Intelligent Monitoring of a Complex, Non-Linear System using Artificial Neural Networks

by Peter Richard Weller

This thesis is submitted for the Degree
of Doctor of Philosophy at the Department
of System Science, City University, London.

Submitted September 1997

Contents

List of Figures

Chapter 7 - Diagnostics

List of Tables

Chapter 7 - Diagnostics

Acknowledgements

It seems almost traditional for works of this nature to commence with an esoteric section comprehended only by a small sub-set of the readers. This thesis is no exception.

I wish to thank my two supervisors: Dr. Ron Summers, at City University, and Dr. Alex Thompson, at the Royal Naval College, for their support and guidance throughout the project. Ron is to be especially thanked for the telephone call that began the whole project.

I must thank Dr. Richard Dybowski, a scientist and a gentleman, for his friendship and acting as a sounding board for many of my ideas. Special mention must also go to Dr. Jacques Lefevre, to whom I am only the Sorcerer's apprentice, for our many discussions and for sowing the seed of studying for a PhD.

Extra special thanks must go to my parents for their love, support and understanding over the last five challenging years. Also in memory of Floyd. The final word must go to my Mother who, given the environment and opportunity to realise her full potential, would, I am sure, have enjoyed doing mathematical things with computers as much as I do.

Thank you all.

* * * * *

I do not know what I may appear to the world, but to myself I seem to have been only like a small boy playing on the seashore and diverting myself in now and then finding a smooth pebble or a prettier shell than ordinary whilst the great ocean of truth lay all undiscovered before me.

Sir Issac Newton

Declaration

I grant powers of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to me. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

This work has been carried out with the support of MOD(PE).

Abstract

This project uses advanced modelling techniques to produce a design for a computer based advisory system for the operator of a critical, complex, non-linear system, typified by a nuclear reactor. When such systems are in fault the operator has to promptly assess the problem and commence remedial action. Additional accurate and rapid information to assist in this task would clearly be of benefit.

The proposed advisory system consists of two main elements. The plant state is determined and then the future condition predicted. These two components are linked by a common data flow. The diagnosed condition is also used as input for the predictive section.

Artificial Neural Networks (ANNs) are used to perform both diagnosis and predictions. An ANN, a simplified model of the brain, can be trained to classify a set of known inputs. It can then classify unknown inputs.

The predictive element is first investigated. The number of conditions that can be predicted by a single ANN is identified as a key factor. Two distinct solutions are considered. The first uses the important features of the fault to determine an empirical relationship for combining transients. The second uses ANNs to model a range of system transients. A simple model is developed and refined to represent an important section of a nuclear reactor. The results show good predicted values for a extensive range of fault scenarios. The second approach is selected for implementation in the advisory system.

The diagnostic element is explored using a set of key transients. A series of ANNs for diagnosing these conditions are developed using a range of strategies. The optimum combination was selected for implementation in the advisory system. The key plant variables which contributed most to the ANN inputs were identified.

An implementation of the advisory system is described. The system should be a single suite of programs with the predictive and diagnostic sections supported by a controller module for organising information. The project concludes that the construction of such a system is possible with the latest technologies.

Chapter

1 Introduction

1.1 Background to Thesis

Recent advances in technology have given rise to a set of complex, critical systems. These are typified by having large, non-linear sets of inter-related variables and, in some situations, require rapid response to system changes. Examples of such systems can be found in power generation, intensive patient monitoring, aircraft flight control and financial markets. Many of these systems are currently controlled by human operators who base their decisions on observations from a large number of displays or screens of instrument readings. In the event of a system change, such as a fault state occurring, the operator has a limited time to react before the situation worsens. The system may be seriously affected, often with catastrophic results. In many cases, the most severe problems demand the fastest reaction times and require the operator to make rapid decisions with what may be potentially serious outcomes.

This situation could be improved if the operator's skill and experience were supplemented by a tool that could monitor the state of the system, advise on future plant condition and rapidly assist the diagnosis of any fault situations. The operator would still retain the overall control of the system, the tool would merely aid the ability to manage the environment safely, quickly and efficiently.

A possible implementation of such an advisory tool could use artificial intelligence (AI) techniques. The various disciplines that contribute to this branch of science have now reached a level of maturity that allows practical implementations to be considered. Some advantages of using AI are the ability to learn, rapid operation and the use of domain experts knowledge. Furthermore, most AI techniques are now available as computer software packages and so remove the need for the development of expensive, specialised equipment.

The work reported in this document investigates applying a combination of two modern modelling techniques; one from Artificial Intelligence, the second from Evolutionary Computation, to the monitoring of a power generation plant, namely a nuclear reactor. The two techniques considered are artificial neural networks (ANNs) and genetic algorithms (GAs). Briefly ANNs provide a crude model of the function of the brain; whilst a GA uses evolutionary techniques to solve problems. The two topics are felt to complement each other as the weaknesses of one technique are compensated for by the second. The two techniques also have some similarities in that practical solutions are obtained from randomly assigned initial conditions, and in some implementations the optimization of the results is an iterative process.

A system of ANNs is proposed and developed which is able to both diagnose the status of the plant and to predict its future condition based upon that diagnosis. Each ANN is optimized to minimize the error in performing their respective tasks. The proposed system will be accurate, quick to operate, easy to update with new information and robust to noisy or missing plant data.

The research on which this document reports was performed during a three year project. The program is part of an ongoing collaboration between the Department of System Science, City University, London and the Department of Nuclear Science and Technology, Royal Naval College, Greenwich.

by a tool that could monitor the state of the system, advise on future plant condition and rapidly assist the diagnosis of any fault situations. The operator would still retain the overall control of the system, the tool would merely aid the ability to manage the environment safely, quickly and efficiently.

A possible implementation of such an advisory tool could use artificial intelligence (AI) techniques. The various disciplines that contribute to this branch of science have now reached a level of maturity that allows practical implementations to be considered. Some advantages of using AI are the ability to learn, rapid operation and the use of domain experts knowledge. Furthermore, most AI techniques are now available as computer software packages and so remove the need for the development of expensive, specialised equipment.

The work reported in this document investigates applying a combination of two modern modelling techniques; one from Artificial Intelligence, the second from Evolutionary Computation, to the monitoring of a power generation plant, namely a nuclear reactor. The two techniques considered are artificial neural networks (ANNs) and genetic algorithms (GAs). Briefly ANNs provide a crude model of the function of the brain; whilst a GA uses evolutionary techniques to solve problems. The two topics are felt to complement each other as the weaknesses of one technique are compensated for by the second. The two techniques also have some similarities in that practical solutions are obtained from randomly assigned initial conditions, and in some implementations the optimization of the results is an iterative process.

A system of ANNs is proposed and developed which is able to both diagnose the status of the plant and to predict its future condition based upon that diagnosis. Each ANN is optimized to minimize the error in performing their respective tasks. The proposed system will be accurate, quick to operate, easy to update with new information and robust to noisy or missing plant data.

The research on which this document reports was performed during a three year project. The program is part of an ongoing collaboration between the Department of System Science, City University, London and the Department of Nuclear Science and Technology, Royal Naval College, Greenwich.

1.2 Aims and Objectives

The aim of this work is to investigate the use of two advanced techniques, ANNs and GAs, in the intelligent monitoring of a complex systems environment, typified by a nuclear reactor.

The following objectives will be addressed to achieve this aim:

* a detailed study of the state of current relevant research;
* an in depth investigation into using neural networks for the prediction of PWR plant variables;
* fundamental work on optimizing number of predictive neural networks required to accurately predict a suitable range of plant transients will be described;
* the appropriateness of ANNs to PWR diagnostic problems will be explored;
* a prototype implementation will be fully described.

1.3 Plan of Report

The remainder of the thesis is as follows. Chapter Two describes the potential of using AI techniques for application to complex systems. This chapter gives a detailed review of relevant research carried out in both nuclear power plants and other complex problem domains. Chapter Three contains the formal description of the problem addressed by this work. The modelling method used is described in Chapter Four. Chapters Five, Six and Seven are concerned with modelling the application.

Chapter Five reports the initial investigations into the predictive nature of ANNs. The Chapter concludes that knowledge about the number and type of transient conditions that can be accurately modelled by a single ANN is an important requirement for the success of the project. This question is addressed in Chapter Six. Two very distinct approaches are considered and developed. The first investigates an empirical relationship between the attributes of the graphical representation of the transients. The second method develops an ANN based simulation of the PWR to predict a wide range of transient conditions. The merits of the two strategies are then discussed and the second method selected for inclusion in the advisory system.

Chapter Seven considers the diagnostic element of the advisor. An iterative approach is adopted in which the number of plant variables and number of time steps of previous information are varied and used as inputs for a range of ANNs. Chapter Eight discusses a prototype implementation of the advisor and highlights some potential problems with possible solutions. Chapter Nine draws some conclusions and indicates future direction..

Appendix A briefly introduces the Pressurised Water Reactor (PWR). Appendix B discusses both ANNs and GAs. The remaining Appendices contain full training details of all ANNs developed and sample computer program listings. These are referred to in the text for the relevant chapters.

Although each chapter includes a brief introduction and summary, it is intended that the chapters should be read in the order that they appear in the report.

Chapter

2 Critical Review

2.1 Introduction

This chapter describes the current research on artificial neural networks (ANNs) and genetic algorithm (GA) applied to complex systems. The review is mainly concentrated on the nuclear power industry, but other complex, critical systems are also considered. Each potential use is described including any specific methodology developed. Finally, a section on hybrid modelling methods in which integration of ANNs and GAs are discussed. Appendix B contains a brief introduction to each topic.

Research into artificial intelligence is becoming increasingly popular in terms of number, scope and geographical location of research groups. The following figure, Fig 2.1, shows the number of research papers on neural networks announced for the years 1969 to 1995. Until 1985 the annual number of papers announced remained relatively low. This could be attributed to the 1969 publication of Minsky and Papert's book `Perceptron' (Minsky, Papert. 1988). This detailed serious shortcomings of the algorithms in use at the time causing a corresponding decline in research funding. This hiatus was reversed by the 1986 publication of the work of Rumelhart and McClelland, the first widely available work containing new algorithms that answered the previous problems (Rumelhart, 1986 and McClelland, 1986). The annual number of papers announced has since rapidly increased from 88 in 1986 to 9119 in 1995.

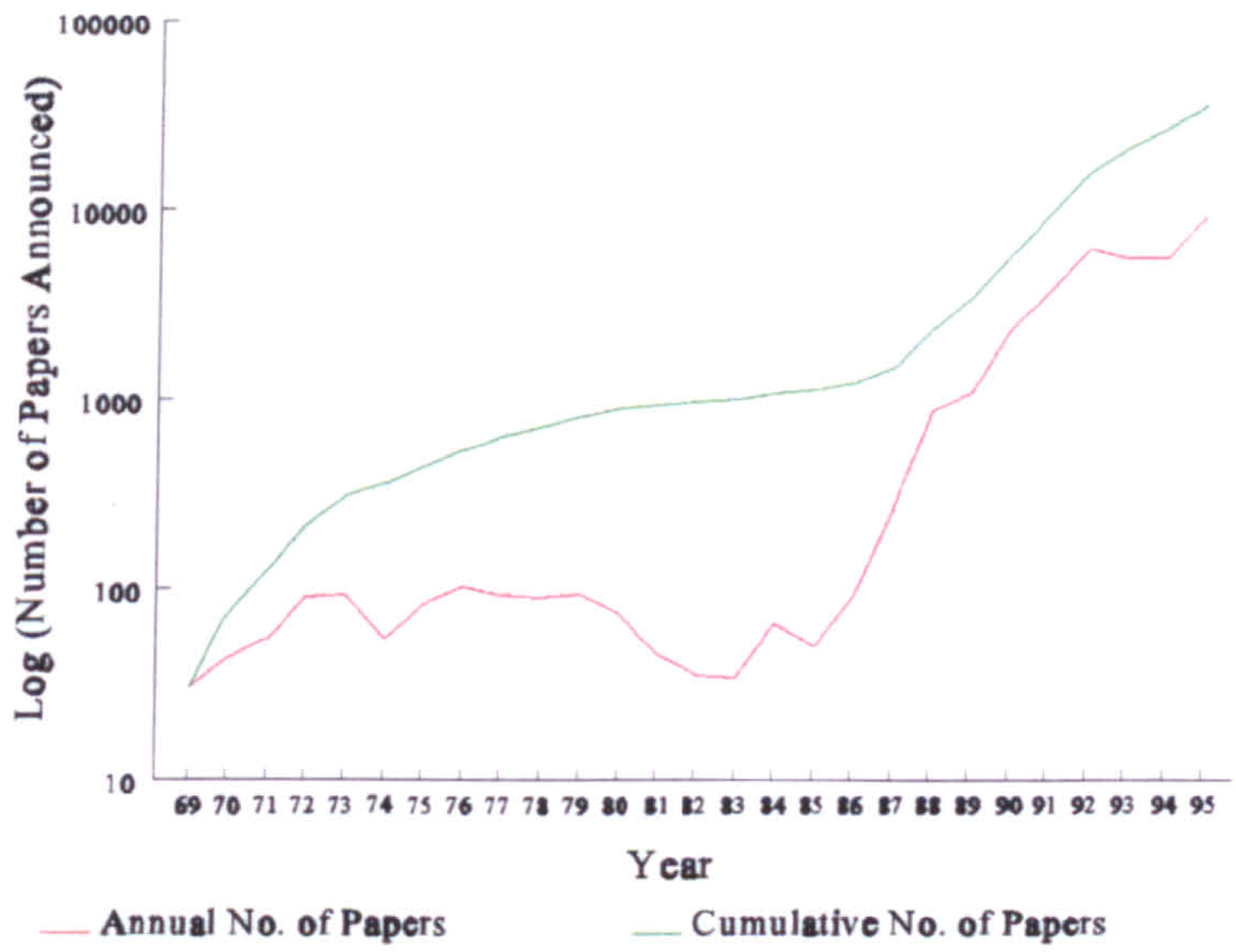


Fig 2.1 Graph of Neural Network Papers Announced

Source: INSPEC Database, Keywords Used: NEURAL(W)NET?

Figure 2.2 shows the corresponding number of papers published for genetic algorithms. There is hardly any published work prior to Holland's landmark book on the subject (Holland, 1975). Since then research interest has steadily increased but not at the same pace as neural network research.

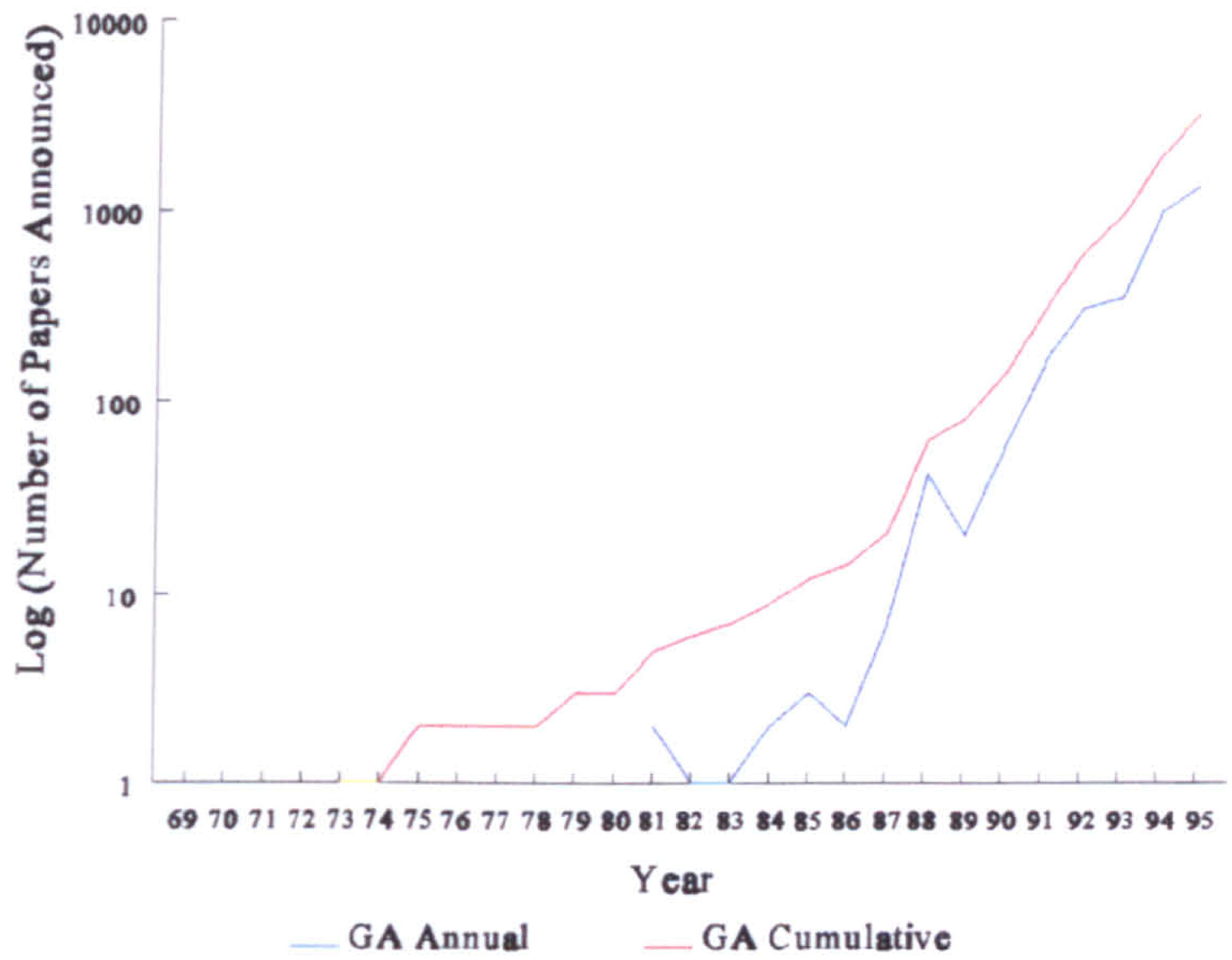


Fig 2.2 Graph of Genetic Algorithm Papers Announced

Source: INSPEC Database, Keywords Used: GENETIC(W)ALGORITHM?

It would be incorrect to attribute the above results solely to the publication of landmark books. Other factors must also have contributed significantly. The greater availability of low price, high performance computers allowed more researchers to explore areas that had previously been the domain of select institutes equipped with mainframes or supercomputers. An increase in the world wide scientific population meant that more researchers were available to explore leading edge subjects. Also the greater dissemination of results, through conferences, journals and electronic methods allowed scientists in different places to exchange ideas and not work in isolation. Finally, the practical need of artificial intelligence solutions was being realized by the world at large. Engineering organisations began to consider incorporating the technology into their products to maintain a competitive lead over their rivals. Consequently they became involved in research either establishing their own research teams or funding academic institutes for a program of research.

The large number of publications, while encouraging for the future, provides a challenge for the researcher wishing to review the subject. Unfortunately, a serious literature search has to concentrate on the most relevant areas of the topic with only little more than a cursory investigation of other applications.

2.2 Review of Neural Network Applications to Nuclear Power Industry

2.2.1 Introduction

As discussed in the introduction to this chapter, research into neural networks and their applications are becoming increasingly popular. Many diverse areas have been investigated to explore the benefits of neural networks, and the nuclear power generation industry is no exception.

The attraction of neural networks to the nuclear industry is understandable. The typical control room of a nuclear reactor is a rather daunting place. The operator has to monitor many hundreds or even thousands of plant parameters. Many of these are in the form of visual analogue signals so the operator becomes familiar with the pattern of settings that relate to a particular plant condition. A pattern change then becomes associated with a fault condition. With many instruments to monitor, the operator could easily overlook a slight change in plant condition which could lead to potential disaster. Most of the reported research, to date, has used the classification ability of neural networks to provide a system which could advise the operator of any problems.

With the exception of a few widely publicised events, there is a shortage of observed information on reactor incidents. Moral, ethical, ecological and financial considerations prevent the destructive testing of nuclear power reactors for training and experimental purposes. Extensive use is therefore made of computer simulations, both to train operators and to simulate fault conditions. Many of the training sets used in the neural networks so far developed have used data from simulators. The information obtained from simulators is free of any noise, as the simulator program functions in isolation of any external interference. It is therefore usual to assess the performance of the developed ANNs with added noise to determine the robustness of the solutions provided.

The published results, to date, are predominantly taken from studies carried out in America; although some recent papers have originated from Korea and Japan. These facts may not give a true guide to the quantity and direction of current research. Some results may be proprietary or even restricted by the nature of the application and so information is not widely available.

The potential of ANNs to the nuclear power industry is described in a key paper by Uhrig

(1991). He introduces several possible applications for ANNs in the nuclear power industry and reports on the progress of research in these areas. It is interesting to note that most subsequent work continues in the directions outlined in this review paper.

2.2.2 Diagnosing Plant Condition

The majority of research to date has considered the diagnosis of nuclear plant condition. Bartlett & Uhrig (1992) have investigated the classification of nuclear power plant accidents. An ANN was trained to identify normal operating and seven fault conditions. A simulator was used to obtain data on the twenty-seven plant variables selected as inputs to the diagnostic network. The output of the ANN was a 3-bit coding identifying the one normal and seven fault conditions. The coding of the outputs was not explained in any detail, although a severity ranking is suspected.

The structure of the ANN was decided by a unique process referred to as dynamic node architecture (Barlett & Basu, 1991). The network learning process began with one node in a single hidden layer. The learning was stochastic and self-optimizing. The network was trained until a performance plateau was reached. A second node was then added to the hidden layer. The learning process continued until another plateau was reached and another node added to the hidden layer. This process of training and adding nodes was repeated until a specified performance level was reached. The least important node was then removed and the training process continued. Eventually the system was found to oscillate around the optimum number of nodes in the hidden layer.

The methodology adopted produced a neural network that quickly responded to the onset of most accident conditions. All fault conditions were diagnosed well in advance of the reactor scram. The network also displayed a graceful degradation with 2% Gaussian noise added to the input set.

This work was continued and reported in a second paper (Basu & Bartlett, 1994). The number of transients considered was increased to twenty-seven, and the number of plant variables raised to ninety-seven. They again used the dynamic node architecture to develop the neural network, but used the standard back propagation algorithm for learning. Again the training data was obtained from a simulator and the three digit coding for the outputs was retained for the presentation of the twenty-seven outputs.

Two neural networks were developed in this study. The first was simply to decide the state of the reactor. If the plant was found not to be operating correctly a second network was used to diagnose the type of fault. Two advantages of this approach, over a larger single network, were offered in the paper. Firstly, each network may only require a subset of the available data set for operation. The number of variables in the training and test sets would then be smaller and this would enable the resulting network to be developed easier. The second advantage is that each network would be developed independently and therefore not rely upon the satisfactory training of the previous network for its own learning.

The two neural networks developed used the same number of inputs, despite the above advantage. However, they consisted of different internal architectures.

2.2.3 Hybrid Systems for Diagnostics

The next set of studies discussed have all adopted a hybrid of artificial intelligence techniques to diagnose fault conditions. The first paper reported used statistical methods and a genetic algorithm to identify a transient. The next two papers each utilised an expert system to enhance the features of neural networks. While the last paper considered a combination of fuzzy logic and neural networks.

Lin, Bartal & Uhrig (1995) developed a k-nearest neighbour system to detect and classify the severity of a transient in a nuclear power plant. A historical database of parameters for a set of known transients was used to identify new transients. This was performed by comparing the variables of the unknown fault condition with the two nearest similar transients in the database. A new metric was developed to determine the similarity and weighting of the neighbours to the new condition and so obtain a diagnosis. The dimension of the vector considered for each transient was optimized by a GA. The GA fitness function was also developed specifically for the problem and was based on the relative error between predicted and observed values.

Using an ANN, Yukiharu Ohga et al. (1993) investigated the identification of the cause of a scram (as described in Appendix A). A time series of five post-scram plant parameters were used as the input. Eight time steps were considered giving a total of forty inputs. The output was a three digit code for the cause of the scram. A simulator was used to generate the data sets. The result of the neural network classification was compared to an expert system in order to confirm the diagnosis. The knowledge base consisted of digital readings from the plant.

Se Woo Cheon et al. (1993) used a different approach. An ANN was developed to provide the knowledge base for a transient diagnostic expert system. The inputs for the network were twenty-four parameter trends or plant states. Each of the fourteen transients included in the paper was represented by a binary valued output. The training and tests sets were produced from a combination of measured plant data and simulation code.

The combination of expert systems and ANNs offers several interesting possibilities. Both techniques have disadvantages; expert systems cannot handle missing or noisy data and require the extensive use of a domain expert, a neural network can offer no explanation of its decision making process. The amalgamation of the two techniques would appear to increase the potential of these techniques for diagnosis in the nuclear industry.

2.2.4 Simulation and Modelling of Plant Condition

As mentioned earlier, simulators and mathematical models are used extensively in the nuclear industry. Parlos et al. (1992 & 1994) have investigated the use of ANNs to model key parameters (steam dome pressure, water level and cold-leg temperature) of a U-tube steam generator. The inputs to the neural network were optimised to those that contributed significantly to outputs. The developed network was used to predict values for these parameters after a defined time step. The results compared favourably to a conventional mathematical models.

A recurrent multi-layer ANN architecture was proposed for modelling nonlinear systems. Instead of feeding the output from a node to the nodes in the next layer as in a feed forward network, the output is also fed back to the node itself and to other nodes in the same layer. The use of recurrent networks is an interesting subject. Rumelhart et al. (1989) report that every recurrent network can be replaced by a feed forward network over a finite period of time.

Two training phases were used on a heuristically developed architecture. The network was first trained on off-line conditions until an acceptable level of performance was reached. A second training set, using on-line conditions, was then used to complete the learning process. This adaptive learning avoids presenting a raw neural network with the large parameter variations that may occur in fault conditions. Instead the network is initially presented with more stable information to recognize and then tuned to the idiosyncrasies of the abnormal conditions. The training set was large, 1870 cases, and consisted of values for a series of plant conditions at a sequence of ten second time steps. The effects of noisy data were implicity included as the empirical model was unable to avoid including the effect of other parameters.

Hyun-Koon Kim et al. (1993) used a neural network to estimate the variation of Departure from Nucleate Boiling point (DNB) of a PWR. The departure from boiling point is a important safety parameter used to assess the core thermal margin. The initial work was to develop a neural network to predict DNB off-line. The training and test sets were randomly selected from a larger, simulator produced data set. Three networks were developed and each consisted of five inputs, one hidden layer of five, five and fifteen nodes respectively and only one output. The five inputs in the final network were selected by a sensitivity analysis of PWR core parameters. Each network was trained on a different sized training set of nineteen, thirty-seven and fifty-six cases respectively. The second network was found to offer the best predictions of DNB which were close to those of the simulator. The paper concluded with consideration of an on-line version of the system.

The use of neural networks for simulation and modelling offers several advantages.

1) Nuclear reactor simulator models are generally of two types. The model can either accurately simulate the reactor process at the expense of operating time; or the model is able to run in real time, but with some loss of accuracy due to assumptions made to speed up calculations. By replacing the whole model or just specific computationly intensive sections, a neural network could speed up the run time for the simulator or allow extra complexity to be added to the model.

2) ANNs can be used to model non-deterministic systems. Firstly, a neural network can derive implicit relationships between variables and could be used to model situations that cannot be calculated deterministically. Secondly, physical and safety restraints may prevent the measurement of variables, for example, the core temperature of a reactor.

2.2.5 Validation of Plant Monitoring Instruments

As mentioned at the beginning of the chapter the control room of a nuclear reactor contains many instruments. The usefulness of the information they portray is dependent on the accuracy of the sensors or measuring equipment attached to the plant. The use of ANNs for signal validation has been investigated by Upadhyaya & Eryunek (1992). They developed an ANN to predict a sensor's signal. Slight changes in sensor behaviour are determined by comparing the anticipated value with that offered by the sensor. The rate of degradation could even reflect the remaining useful life of the device.

In an earlier paper Holbert & Upadhyaya (1990) mapped process states to a hypercube. A similar technique to empirical modelling was used to train the cube on operating states. In operation the system compared the process states against the learned domain and identified any discrepancies.

The later paper replaced the hypercube with a series of independent neural networks. One set of networks were developed for sensor validation during the start up of a nuclear reactor. A second set were produced for complete plant monitoring using multiple signal estimation. All ANNs were small and consisted of only three or four inputs and one output. The structure of each network was calculated using the following equation, from Shannon and Weaver's Information Theory (Shannon and Weaver, 1971).

$$H = I \cdot \log_2 N$$

Where: H = Optimum number of nodes in hidden layer
I = Number of inputs
N = Number of training patterns

The networks were trained with a modified back-propagation algorithm. The new methodology allowed the user more flexibility in deciding the training parameters, including some on line updating. The predictions obtained from the networks were compared to actual plant values and found to offer good agreement, indicating that ANNs could replace empirical or physical models as estimators.

2.2.6 Other Applications

This section briefly introduces other reported nuclear applications of ANNs. The application described is not central to the main ideas considered for this project. However, the underlying theory might be potentially useful. Also, to have omitted them would have given an incomplete picture of the potential of neural networks in the nuclear industry.

2.2.6.1 Database Applications

Gacem et al. (1990) has investigated using the implicit relationships developed by ANNs to extract additional information from a database of nuclear incidents. A self-organising Kohonen network was used to map cause and effect relationships onto a two-dimensional map. The centroids of these mappings was then used to construct a multi-layer network that could identify sequences of patterns and detect unusual patterns. A prototype network, containing a large number of incidents, was developed. The initial results appeared encouraging; however, no subsequent progress has been reported.

Heger et al. (1989) described an application to develop an adaptive interface for a nuclear database using ANNs. The interface became an associative information evolving environment that allowed the user to refine queries to the database. A test database was used to prove the systems robustness and fault tolerance.

2.2.6.2 Fuel Reloading Planning

Another interesting use of ANNs in the nuclear industry is for planning PWR fuel reloading cycles. Fuel assemblies are classified as fresh, once burned and twice burned. During the reloading cycle the twice burned fuel assemblies are replaced by fresh ones. After every reloading process three key parameters; local peaking power, maximum burnup and effective multiplication factor have to be considered both for safety and economic reasons.

Kim Han Gon et al. (1993) used two networks to predict the local peak power factor and the effective multiplication factor of a PWR. The neural networks they developed and justified were unusual as they were composed of a large number of hidden nodes; 500 nodes in one hidden layer for 21 inputs and 18 output nodes. The training and test patterns were randomly selected from a range generated by a commercial deterministic programme.

The system they developed proved to be a very fast predictor of the two parameters, being several hundred times faster than the reference numerical code. It also demonstrated that neural networks can be used to predict core parameters. However, they concluded that the accuracy, although usually within an permissible range, was not sufficient enough. They proposed several possible solutions to this question including improved selection of the training patterns.

Uhrig (1989) and Miller et al. (1992) both considered a similar application for ANNs. The conventional core calculations were emulated with neural networks. Again two networks were developed; one for fuel depletion calculations, the second for the neutron calculation. An expert system was then used to shuffle the fuel in order to improve the core loading. The method was not approved by the Nuclear Regulatory Commission so could not be used for licensing. It did however produce a near optimal solution much faster and cheaper than existing methods.

2.2.6.3 Safety Control

Jouse et al. (1990, 1993) considered the safety aspect of using ANNs in nuclear plant. A self-programming neural network, based on the Barto-Sutton architecture, was developed and trained to solve increasingly difficult tasks. The assignments ranged from stabilizing a reactor core to the multi-variate control of a PWR during start up.

A system model was used to produce the required parameters for the network. The final system was demonstrated to synthesis the control of a four-loop PWR. The network was shown to be robust and able to handle new problem domains with a graceful degradation.

2.2.7 Discussion

As a result of the literature review several observations have become apparent. These are now discussed.

There is a noticeable increase in the amount of ANN research published each year; for example, a recent journal on nuclear engineering contained three papers relevant to this study. The number of institutions reporting research results has also grown. The increase in research activity may give a guide to the potential for neural network applications in nuclear reactors.

There have, to date, been no large scale ANN applications of an entire plant reported in the literature. All investigations have considered small subsets of the reactor, for example one loop of the primary circuit. The ANNs developed have also been small, not only in number of nodes but in terms of data sets. A possible reason may be that most of the existing work is a preliminary investigation to determine the validity of the approach and that real, full plant applications are now being studied.

There is no universally accepted method of deciding the architecture of ANNs. Most papers adopted a heuristic approach and used an educated guess to refine the number of nodes in a hidden layers. Some researchers developed their own algorithm to solve the problem, for example the dynamic node architecture of Basu (1994). Custom solutions may create further problems in determining the accuracy of a solution. A neural network may fail to classify correctly due to limitations with the bespoke algorithm as well as problems in the nature of the application.

Each paper developed a unique validation method for their neural networks. Usually this was a comparison of the predicted output with that obtained from a simulator or plant. This technique leads to a very restricted solution set. An ANN that accurately predicts the performance of one reactor may not enjoy similar success on a slightly different model.

It is felt that the above problems could be improved by the use of a combination of artificial intelligence techniques. Uhrig (1991) briefly discussed the potential of this approach. The strengths of one approach could compensate for the weaknesses of another system. A possible example of this is an expert system providing a more rigid explanation of neural network decision making process. A later section of this chapter considers further possibilities from other fields of artificial intelligence.

2.3 Neural Network Applications to other Non-linear Systems

This section examines applications of neural networks in other fields of research. The review is by no means exhaustive and concentrates on topics and techniques relevant to the proposed direction of this particular research project.

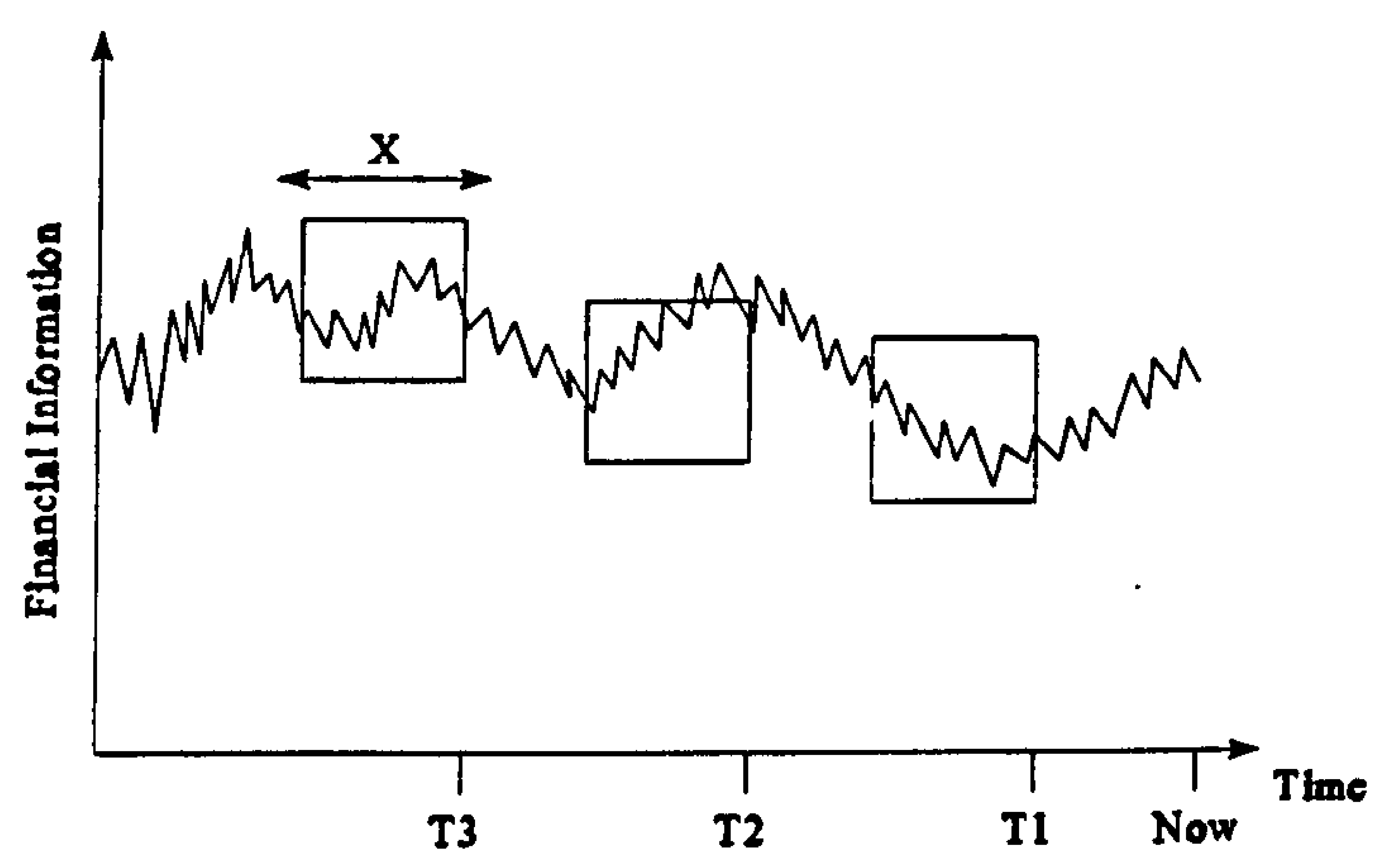
2.3.1 Financial Applications of Neural Networks

Traditionally the financial world has made extensive use of statistical methods, such as exponential smoothing and auto-regression, to predict share prices, exchange rates, economic turning points and sales forecasting. The use of neural networks in these, and similar areas, has been the subject of much investigation in many countries. The majority of the applications require a predictive capability and this is the area concentrated upon in this report.

In general, information on the neural networks developed has been scant. A possible reason for this may well be to maintain any competitive advantage gained from using neural network technology. Any slight improvement gained by ANNs over existing techniques can result in large profits. Consequently the developer of a successful system is unlikely to publish full details of the implementation and give competitors the benefit of his labours.

Refenes (1991) and Refenes et al. (1993) used an ANN to predict the exchange rate between the American Dollar and the German Mark. Two types of forecasting were developed. The first was a multi step prediction to identify general trends. This technique used a feedback approach in which the predicted values were fed back into the neural network as inputs. The second type of forecasting was a single step prediction. The output was the exchange rate one time step into the future. No feedback, using the output as input for longer predictions, was considered as the ANN only predicted one time step ahead at a time.

A technique called 'windowing' was adopted for the forecasting. This method is used to identify relationships in noisy datasets. The diagram below shows the basic concept.



Windowing for Financial Prediction

Fig 2.3 Windowing of Information

The two windows were used to examine the dataset. The first was used as the input to the ANN, the second as the output. The relationship between the two windows is a mapping determined by the training process. For subsequent predictions the windows are both moved along the time axis a defined step size. The choice of window size and step size are critical to the accuracy and convergence of the final network.

The papers also introduced a dynamic learning procedure in which nodes are added to the hidden layer during the training process. Two similar types of dynamic learning were discussed. The difference between the techniques was the method of creating the inputs for the hidden layer. The first approach used cascade correlation, the input nodes and the output from the last hidden node as inputs to each hidden node. The second method was a fully connected network with every previous node being connected to each hidden node. The paper did not report a preference or any network details. The final network was able to successfully predict trends in the exchange rate.

In a later paper, Refenes (1994), the performance of an ANN is discussed along with testing strategies and metrics. The paper comments that neural network literature contains many poorly tested results. One reason for this may be the inherent difficulties of statistical tests for non-linear models.

Gia-Shuh Jang et al. (1993) used a similar approach to predict the buying or selling of shares on the Taiwan Stock Exchange. The developed ANN comprised of two modules; the first used a short term window of the last twenty-four days trading and the second module considered a long term window of the last seventy-two days trading. Feature extraction was used to pre-process the datasets before training. The final output of the neural network was a weighted combination of the results of both the short term and long term predictions.

Again, a new method of training the network was developed. After randomly setting the weightings, an initial training set of share prices from the previous year was presented to the network. A moving window training scheme was then used for sensitivity tuning of the weights to both long and short term trends. The architecture of the network was developed heuristically with the testing of several hidden layer sizes until an optimum was found.

2.3.2 Medical Applications of Neural Networks

Another area where ANN applications have been developed is in the medical domain. The medical use of artificial intelligence has a long history, going back to the 1950s. Expert systems such as MYCIN and some statistical systems incorporating Bayesian-based estimators were very successful. It is no surprise that the potential of neural networks has also been investigated.

Yearworth and Sharpe (1992) report on the current situation of medical ANNs applications in the EC OpenLabs AIM project. The vast majority of reported medical applications of ANNs are for diagnosis of particular conditions. The inputs for these ANNs are usually patient symptoms, medical images or biophysical parameters.

Mulsant (1990) investigated the use of an ANN in the diagnosis of dementia. The input set of his network consisted of a series of conditions that were classified for binary representation. For example, the age of patient had three binary inputs and was categorised as Young (<50) (1 0 0), Middle (50-70) (0 1 0) and Old (>70) (0 0 1). Several parameters consisted of two inputs; for the condition being present and absent. The case of missing or unknown information was permitted by setting both values to 0,5. He also reduced the number of possible diagnosis of dementia to seven main classes. The final network was found heuristically to consist of two hidden layers of ten and seven nodes respectively. An additional feature was a confidence facility that requested further information on complicated cases.

Weller (1993) used a neural network to classify the organisms causing septicaemia. Again a feature selection process was used to reduce the input set to only include the most significant parameters. This selection process used a domain expert to identify the important variables. The initial set of fifty-one parameters was reduced to nine. Two of the final categories, `year of infection' and `age of patient', were real valued inputs. The range of these parameters was modified to use a binary representation. The `year of infection' category was represented as a binary coding of the particular year with respect to when the recording began. For example, the recording of septicaemia bacteria was begun in 1969; so a patient who became ill in 1986 would be given the code 1 0 0 0 1 (17) for the `year of infection'. The `age of the patient' was classified into four intervals. These were 0 to 3 months, 3 months to 4 years, 4 years to 62 years and 62 years to 100 years. A forty-five year old patient would therefore have an age input of 0 0 1 0 and a two year old child's age input would be 0 1 0 0. This method enlarged the neural network to sixteen binary inputs.

The paper also introduced a method of developing the optimum neural network architecture. It proposed using a genetic algorithm to evolve the number and size of the hidden layers. Genetic algorithms and their application in this technique are explained in the next chapter.

The techniques of feature selection and evolving ANN architecture are being further explored in current work. The feature selection has been performed using statistical methods to identify the prominent parameters. This data set is then used to evolve the neural network architecture. This approach has the benefit of reducing an initially large data set to a more manageable size while still retaining the important features.

A potential problem with ANNs is for one outcome or result to dominate and so affect the accuracy of the output. Medical records consist of a set of patient related readings. These values are measured or observed over many days and provide a guide to the eventual outcome or diagnosis. If an ANN were used to aid the diagnostic process the training and test sets would be composed of values from these readings. However, the construction of these data sets requires some careful thought.

As an extreme example consider two patients, both of whom have similar symptoms but different conditions. One patient is in hospital for only one day for the diagnosis to be complete, whereas the second person requires twenty days observation for an analysis. If readings are taken daily the first patient would only have one set of readings compared to the twenty for the second person. If a neural network is now developed to aid the diagnosis of these conditions, the daily readings would be used to generate the training and test sets. Of the twenty-one members of these data sets only one has an output for the first condition whereas twenty are for the second illness. The data sets would be weighted towards the second condition. The trained neural network would be therefore be biased to this ailment and could give incorrect diagnosis.

One way to resolve this dilemma is to only consider one daily reading from the second patient. The reading used could be selected at random from all recorded values. While this method would result in an equal number of data sets for each condition it would leave a lot of redundant relevant information; the nineteen other readings for the second patient. Further work is needed to develop a suitable compromise.

There are many similarities between applying neural networks to medicine and nuclear power applications. They are both concerned with the well being of a complex system; there is a need to diagnose the condition of this system accurately and quickly; lastly, in each case there are a large number of possible parameters that may have unexplored implicit relationships that neural networks can work with. The progress of medical neural network applications has been monitored during this project to identify common problems and solutions.

2.4 Hybrid Systems

2.4.1 Neural Networks and Genetic Algorithms

One of the biggest weaknesses of neural network applications is the development of the final structure of the network. The problem is usually addressed in one of three ways.

The final architecture can be found heuristically, an educated guess is refined until the results obtained are acceptable. This method, while very common, has an element of luck or at least `black magic' as part of the procedure. There is no guarantee that the obtained solution is optimal or even efficient.

A second method is to use a formula to determine the structure of the network. The formulae used have all been developed for a particular application or based on a problem-specific theory and are not necessarily appropriate to the current area of interest. For example, an equation designed for a network to classify general trends would not be efficient in an application requiring specificity.

The third method involves the development of a custom procedure to determine the network form. These algorithms usually function very well in the envisaged application. However, they require resources to develop and could well be problem specific and not transfer well to other problem domains. Furthermore, there is no method of verifying the procedure, a poor experimental result may be due to either the application or the algorithm, or even both.

As an alternative to these commonly found methods, a genetic algorithm could be used as an universal method of developing neural network architectures. This concept has already attracted some attention in the literature.

Weller (1993) coded the number and size of the hidden layers into a genetic string. A maximum number of possible nodes was defined, determined by the number of inputs. The coding then consisted of a binary representation of the size of each hidden layer. A maximum of four hidden layers was allowed, although networks with less than this number were permitted by the coding. The string was translated into a neural network architecture, for example the code 0 0 1 1 1 0 1 0 1 1 0 0 0 1 0 1 became a four layer network with three, ten, twelve and five nodes in the respective hidden layers.

Each network in a population was trained and the fitness of each string calculated. The fitness was a linear combination of RMS error, network complexity and learning time. The traditional genetic algorithm techniques of parent selection, crossover and mutation were then applied to produce the next generation. The following diagram, Figure 4.1, further explains the process.

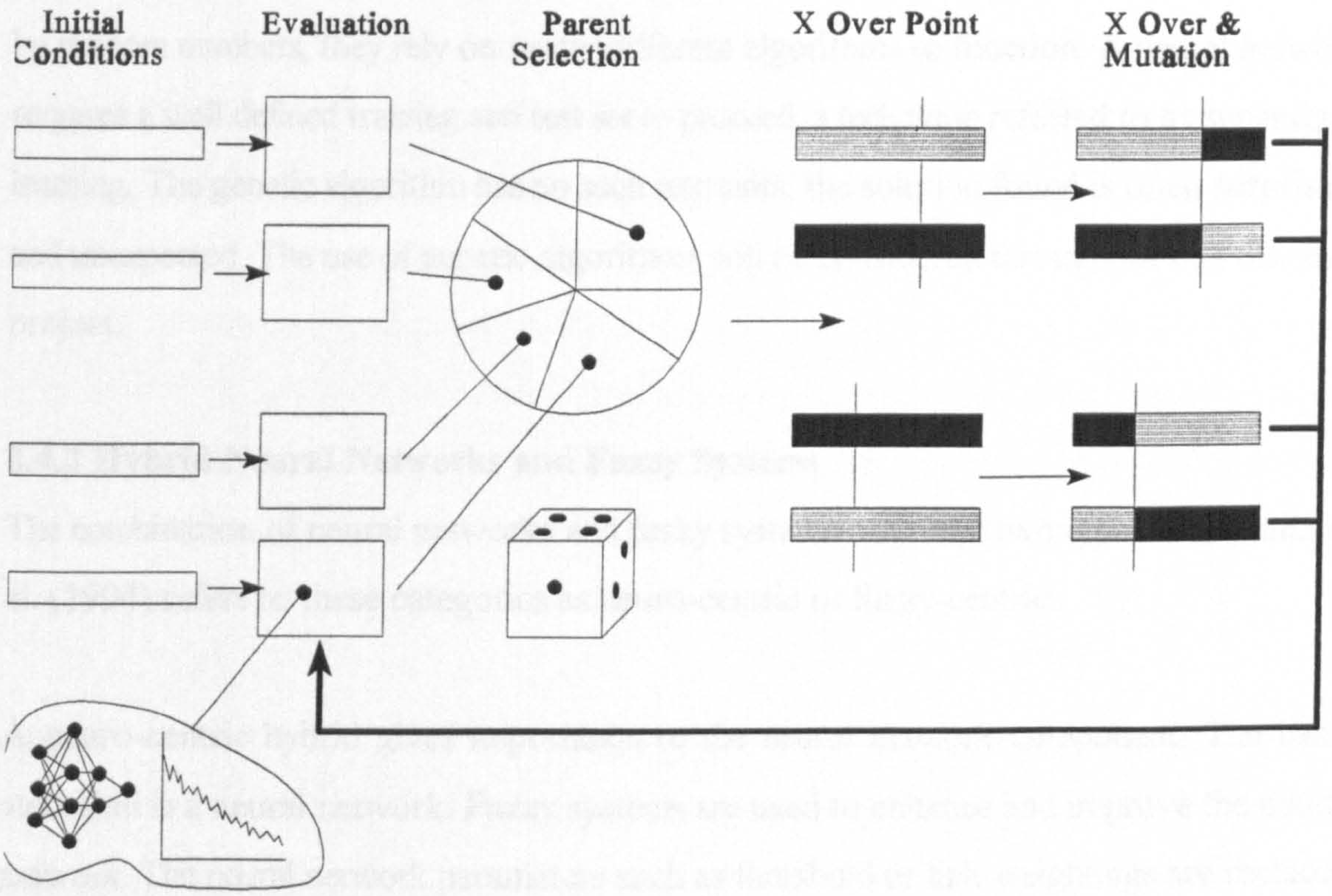


Fig 2.4: Procedure to Select Neural Network Architecture with a Genetic Algorithm

Another popular approach to the task of deciding neural network architecture has also used a genetic algorithm for selecting the interlayer links of a neural network, see Dasgupta et al. (1992), Elias (1992). The number and size of the network was first established. A coding for the existence of a link between nodes was then developed along with a fitness coding. The results reported seem to encouraging if idiosyncratic.

Training a neural network with a genetic algorithm has also been investigated. Potter (1992), Karunanithi et al. (1992) both investigated methods of replacing the back-propagation algorithm with an evolutionary system.

Guo et al. (1992) used a genetic algorithm for feature selection of the input layer for a neural network. This idea is very interesting and, if successful, could replace the current methods of domain experts or statistical analysis to select the input variables. A difficulty to be resolved is one of fitness, a good input set could be discarded because of a bad neural network architecture, similarly a poor data set could lead to the rejection of the optimum architecture.

The genetic algorithm complements the neural network. Although they are both initiated by random numbers, they rely on vastly different algorithms to function. A neural network requires a well defined training and test set to proceed, a technique referred to as supervised learning. The genetic algorithm has no such restraints, the solution found is often surprising and unexpected. The use of genetic algorithms will be considered throughout this research project.

2.4.2 Hybrid Neural Networks and Fuzzy Systems

The combination of neural networks and fuzzy systems can be of two types. Tsoukalas et al. (1994) refers to these categories as neuro-centric or fuzzy-centric.

A neuro-centric hybrid gives importance to the neural network component. The basic algorithm is a neural network. Fuzzy systems are used to enhance and improve the neural network. The neural network parameters such as threshold or link weightings are replaced by fuzzy regions. An example of this approach was outlined earlier, Sang Ki Moon et al. (1994) developed a system to predict the critical heat flux of a nuclear reactor.

A fuzzy-centric hybrid emphasises the fuzzy system part of the hybrid. Fuzzy rules are first used to formulate an algorithm to address a problem. Neural networks are then utilised to change the rules to an optimum state or to determine membership functions. Ikonomopoulos et al. (1994) used this approach for the virtual measuring of a reactors secondary flow control valve position.

2.4.3 Discussion

Some of the neural network problems identified in the previous sections have been addressed using hybrid techniques. The solutions developed are mostly test cases but the results so far are very encouraging. The strengths of one form of artificial intelligence compensates for the weaknesses of a second form. A good example of this is the expert system and neural network combination. It is felt that the use of hybrid techniques will be of increasing importance in future artificial intelligence work.

2.5 Summary

The points raised earlier in the chapter appear to have also been considered by the wider ANN modelling community. In most arenas the applications of neural networks still appears to be in an investigative stage. There are as yet no large scale implementations of neural networks. It is possible that instead of one large all encompassing network, with the corresponding problems of convergence, the trend may be for flexible, linked multiple smaller systems.

Again the development of the architecture of a neural network appears to have attracted a lot of interest. At present no one way has been adopted for general use with many researchers developing their own algorithm or using heuristic methods. It is hard to accept that the development of neural network architectures is application specific; there could well be a universal algorithm for all situations.

The validation of neural networks is a challenging problem. Several researchers have considered this aspect but again the approaches have been very piecemeal. It is possible that once the use of neural networks becomes an accepted tool the means to validate them will assume a higher priority and methods would have to be developed for neural networks to become universally accepted. Conversely the widespread use of ANNs may depend on a successful validation process. In the critical, complex systems being considered by this thesis ANN safety justification is a critical component for their acceptance and use. Important work has been carried out in the area of confidence factors for ANNs, for example Bishop (1996).

The review of current ANN research has shown that the techniques have been applied in two distinct problem areas, namely diagnosis and prediction. The diagnostic applications are typified by the medical domain where, generally, a set of patient related inputs are used to train an ANN to classify the patient condition. The predictive role of ANNs, typified by the financial domain, frequently considers the history of a set of relevant variables and predicts the next set of values in the series. The results of the review show that researchers working in the nuclear industry have considered ANNs for both diagnostic and predictive tasks. The nature of nuclear reactor technology may have encouraged a diverse range of applications to be considered. Alternatively, engineering domains generally may have a greater requirement for rapid development and be more prepared to investigate emerging disciplines.

The application of ANN technology was examined in several domains. Research in the nuclear, financial and medical domains was examined and discussed. In these cases ANNs appear to have been fairly successful. However the applications considered were small. The method of determining the optimum ANN architecture is not standardised. The safety justification of ANNs for complex systems still requires a formal definition.

Chapter

3 Problem Definition

3.1 Introduction

This chapter introduces the specific problem to be addressed in this thesis. The environment of a nuclear reactor control room is examined and key factors that contribute to operator efficiency identified. These elements highlight the need for additional assistance for the operators to control the plant safely during fault conditions. This provides a formal definition of the problem to be addressed. Finally a systems approach is considered and the problem formulated in both systemic and systematic terms.

3.2 Problem Definition

A nuclear reactor typically has a single control room manned by a team of operators under the control of supervisors. The teams work on a shift basis. The control rooms are usually ergonomically designed for ease of operation. The many hundreds, or even thousands, of measured plant variables are displayed on large panels of both analogue and digital meters. Throughout their shift the operators monitor these meters for signs of abnormality. Any unexpected readings are referred to the supervisor for confirmation before any corrective action is taken. The strategy to be followed in any condition is well defined in authorised standard operating procedures. Maintenance and repair methods are also well defined by authorised procedures

Operator reliability is an essential component for the safe operation of nuclear power plants. The operator is a vital link for both identifying the reactor condition and for instigating the corrective measures required to return a faulty plant to a safe situation. Studies have been conducted to investigate the operator's ability to perform their tasks in varying situations (Swain and Guttmann, 1983). Using the Technique for Human Error Rate Prediction (THERP) they conclude that the two most crucial factors affecting operators efficiency are stress and time.

The reported effects of stress on operator success are shown in the following figure, Fig 3.1.

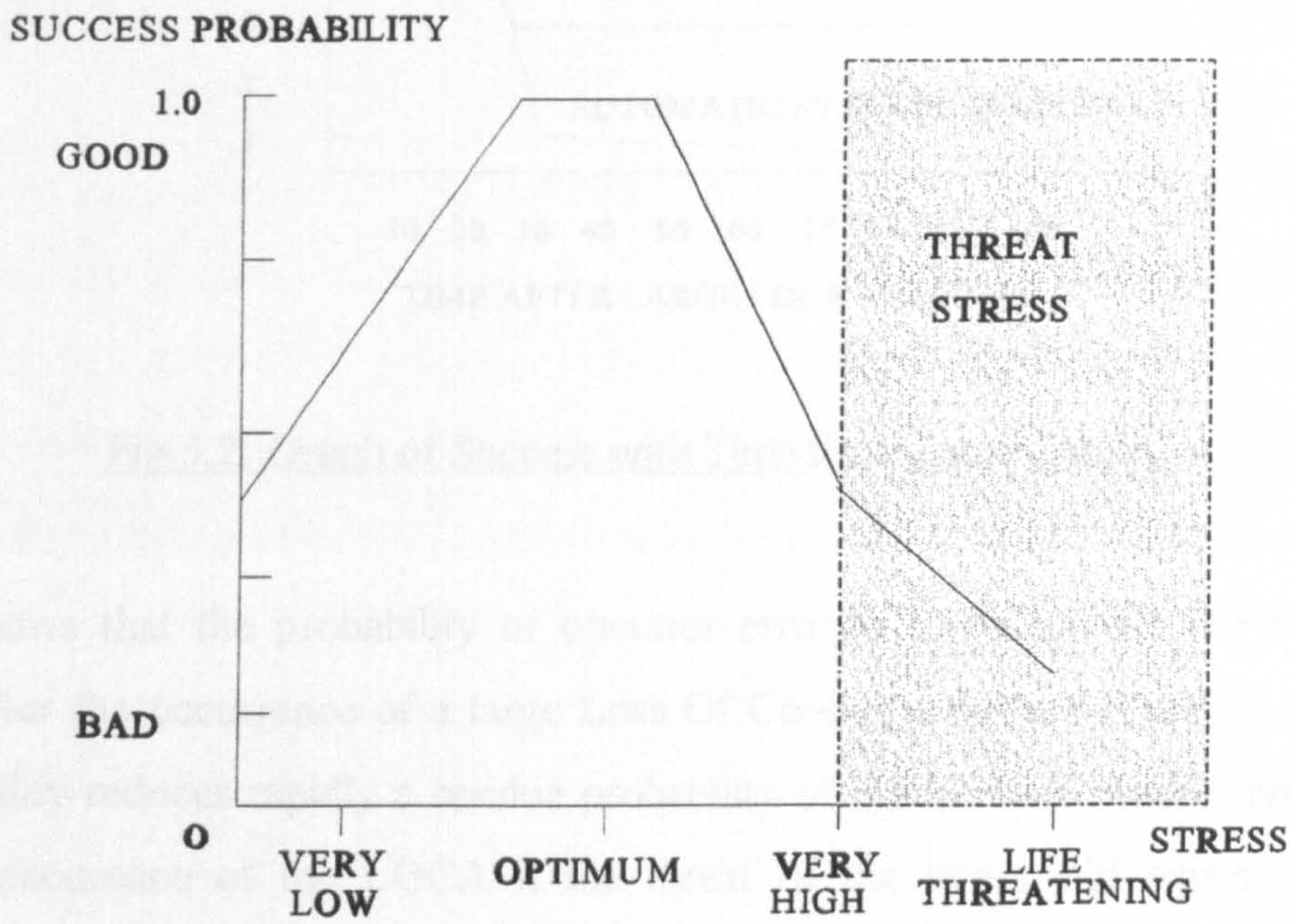


Fig 3.1: Graph of Stress and Success

The optimal operator performance occurs when some stress is present. Periods of low stress induce boredom and brain activity becomes largely imaginative. Operators could "invent" situations to test themselves, potentially with dramatic results. Periods of high stress, such as during a reactor transient, also reduce the efficiency of the operators. The presence of a group of operators further amplify the effects of stress. In these situations it is conceivable that an operator could misread or misinterpret instrument and meter readings. The effects of stress can be partially reduced by robust plant design, standard procedures for managing emergencies and regular operator training in simulated emergency situations. However, stress remains an important factor in operator efficiency and therefore reactor safety.

The results of elapsed time on operator efficiency after a major incident are shown in the following figure, Fig 3.2.

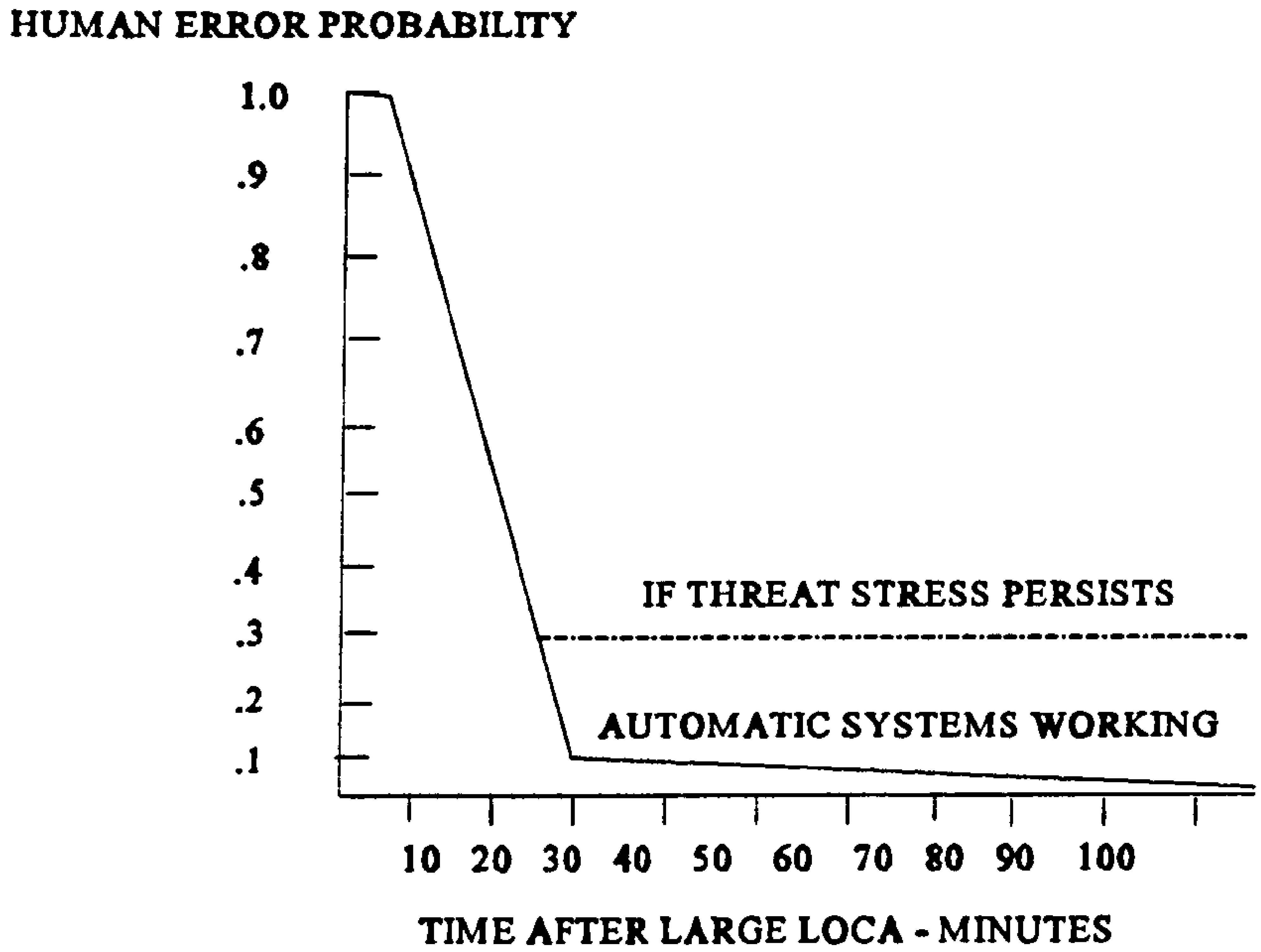


Fig 3.2: Graph of Success with Time for a Large LOCA

Fig 3.2 shows that the probability of operator error is very high for the first thirty minutes after the occurrence of a large Loss Of Cooling Accident (LOCA). Although the probability reduces rapidly a residue probability of 0.3 remains twenty five minutes after the occurence of the LOCA if the threat of the stress still exists. With the automatic safety systems in operation this probability drops to 0.1 after thirty minutes. The high initial probability of human error highlights the problem of a safety system reliant solely on human operators. This problem has been addressed at the nuclear power station at Sizewell B where the operator is denied access to the system for the first thirty minutes following a major fault sequence. A computer-based system controls the recovery process until the operators are permitted into the system.

Considering the above two factors it is clear that the operators, despite their training and experience, are the vulnerable element for the safe operation of a nuclear reactor.

Indeed, the THERP trials concluded that the probability of successful operator diagnosis was too low to justify inclusion of the operator as a contributory factor in the safety justification of civil plant design. Other factors, such as fatigue and illness, have not even been considered. Yet the human operator is still an essential component of the plant. Human intuition and reasoning cannot yet be efficiently assimilated. An important addition to increasing operator efficiency would be a reliable aid which contributed to reducing stress and, at the same time, increased awareness of the state of the reactor so giving the operator time to consider responses to each situation.

The problem addressed by this work is therefore the design of a computer-based advisory system for the operator of a PWR. This system should be able to quickly both diagnose the current condition of the plant and predict the future reactor state based of that result. The human operator must remain an integral part of the control strategy responsible for final decisions relating to the system. The work will consider the primary circuit of a PWR and concentrate on LOCA transients.

A LOCA is a serious potential fault situation in a Pressurised Water Reactor (PWR). At the high operating pressures of a PWR even a small leak can lead to a grave situation. It is therefore important that any potential LOCA is identified as soon as possible after its occurrence, so that a program of corrective and safety measures can be instigated. An early diagnosis of possible faults gives a maintenance team vital time to respond and contain the condition, thus preventing it from becoming more serious. In nuclear reactors the situation can change very quickly. The Chernobyl disaster of 1986 is an example of this; the time between the connection of the fourth cooling pump into the circuit and the explosion was just over twenty minutes (Reid, 1993).

It is not the intention that a computer-based advisory system would replace staff but to support the operators and provide additional information at the times when it is most needed.

While the above problem is considered on its own merits, it is noted that it is an example of controlling a complex, non-linear system in a safe, economical and reliable manner. It is hoped that the theory and techniques developed for this project would be applicable to other equally demanding fields of study.

In addition to the above discussion, a computer-based advisory system would have the following advantages.

1) The system would be permanently on duty and not tire or become bored or distracted during operation. It would not require briefing at every duty change over.

2) The same advisory system would be applicable to all models of the plant for which it was developed. This would provide both transferability and portability of the technology.

3) The system would be very easy to upgrade or replace. A computer-based advisory system could be constructed from commerically available hardware and avoid the vast use of expensive bespoke equipment.

4) The proposed advisory system would not be required to attend extensive training courses nor gain a wealth of experience to operate fully. All the required knowledge and experience would be programmed into the system. The initial development expense may be high but the subsequent operating costs will be low.

5) A record of reactor history could be stored and so provide a built in audit trail.

Although the development of an advisory system is hoped to make an important contribution to reactor system safety by supporting the human link in the reactor control chain, the work has wider reaching attractions. A similar problem exists in other non-linear, critical systems. Examples of this are monitoring of patients in intensive care, controlling air traffic and share dealing on financial markets. In all arenas a key requirement is for the safe operation of critical systems with a quick and accurate reaction time.

3.3 Systems Approach to Problem

A system has been defined as a set of interrelated parts or subsystems, each one of which is in charge of some mission or task. The problem defined in the previous section can also be described in terms of a systems approach to controlling an external environment, the PWR.

The key component of the system is the operator who has ultimate control over the reactor system. Information from the environment, in the form of instrument readings, displays and other observations is passed to the operator who decides on the appropriate action to be taken. In normal circumstances the operator needs to monitor the plant, make any operational changes and ensure safe performance of the reactor. The diagram below, Fig 3.3, shows this system.

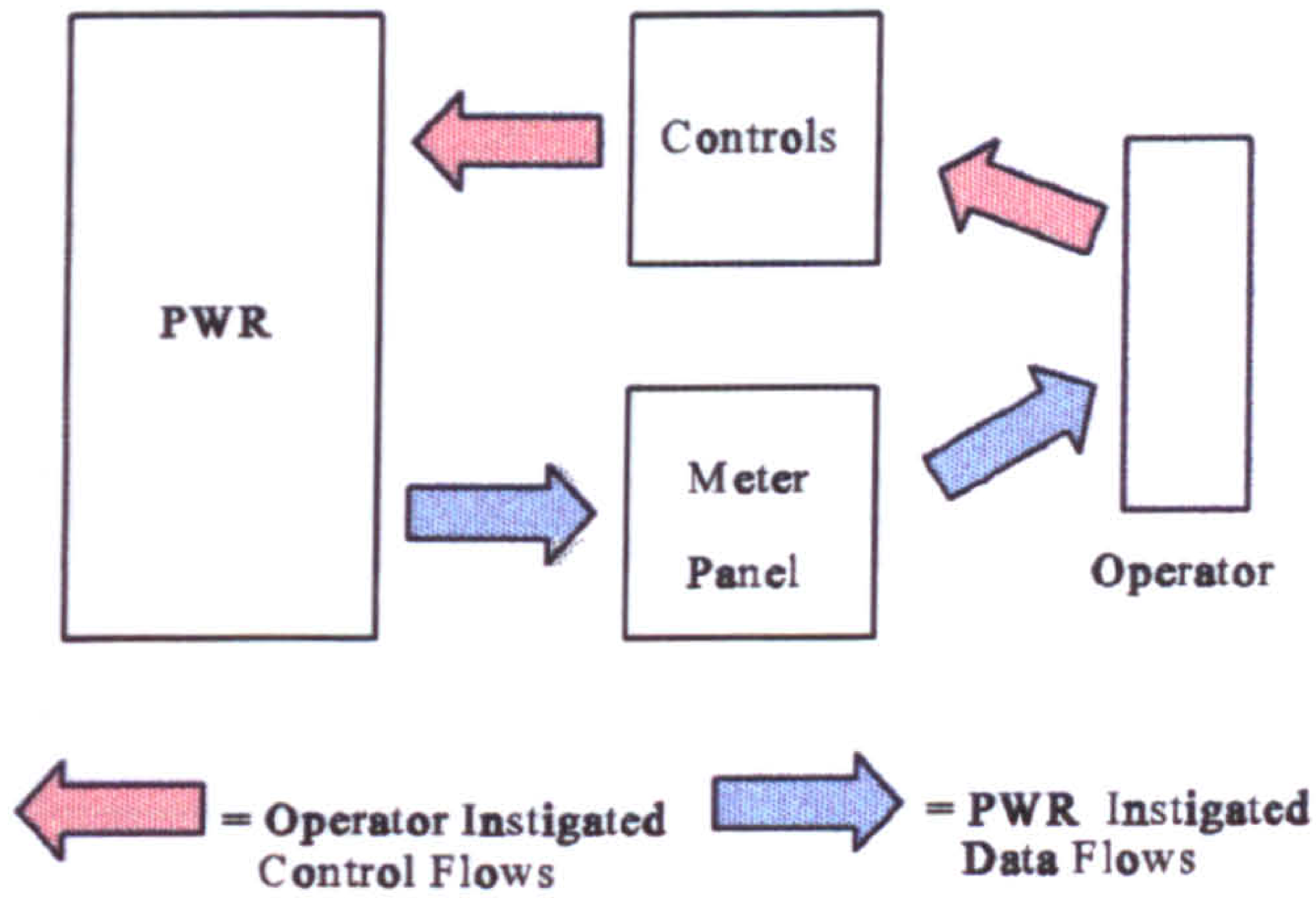


Fig 3.3: Normal Operation of System

In a fault situation the systems environment has changed, as depicted in Fig 3.4. A different set of observations are now received by the operator. The safety sub-system is now required and called into action by the operator. The safety system may include control rod movement, emergency cooler activation and the dispatching of personnel to begin repairs. In the previous example of Sizewell B, this system would also include the thirty minute operator lock out period.

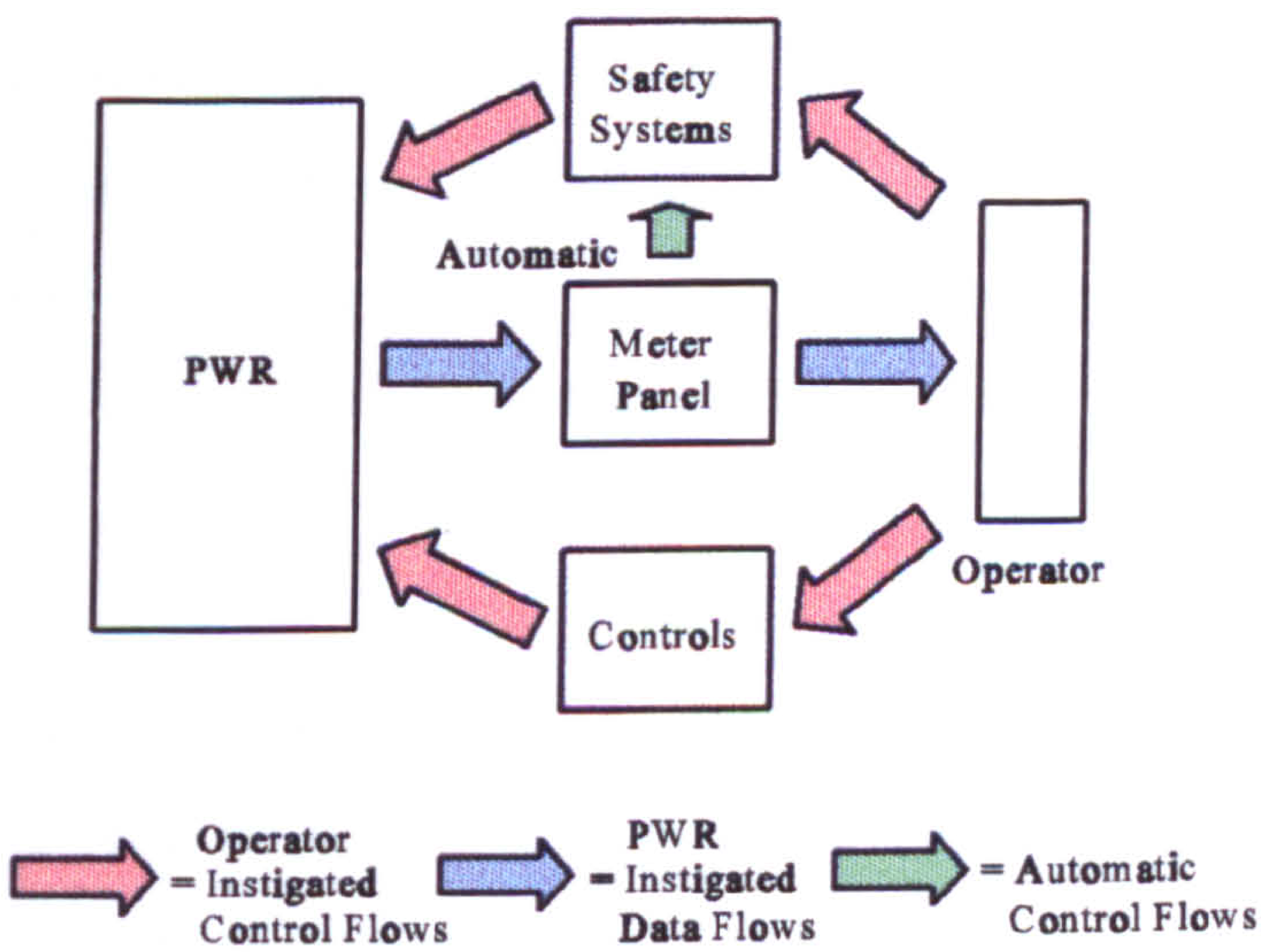


Fig 3.4: System in Fault Condition

The above two scenarios show the key position of the operator. Nearly all responses to plant condition are instigated by the operator. It is important therefore to ensure that the operator's decisions are based on accurate information including the current and possible future state of the PWR. A computer-based advisory system could provide the operator with rapid information. The diagram below, Fig 3.5, shows this option.

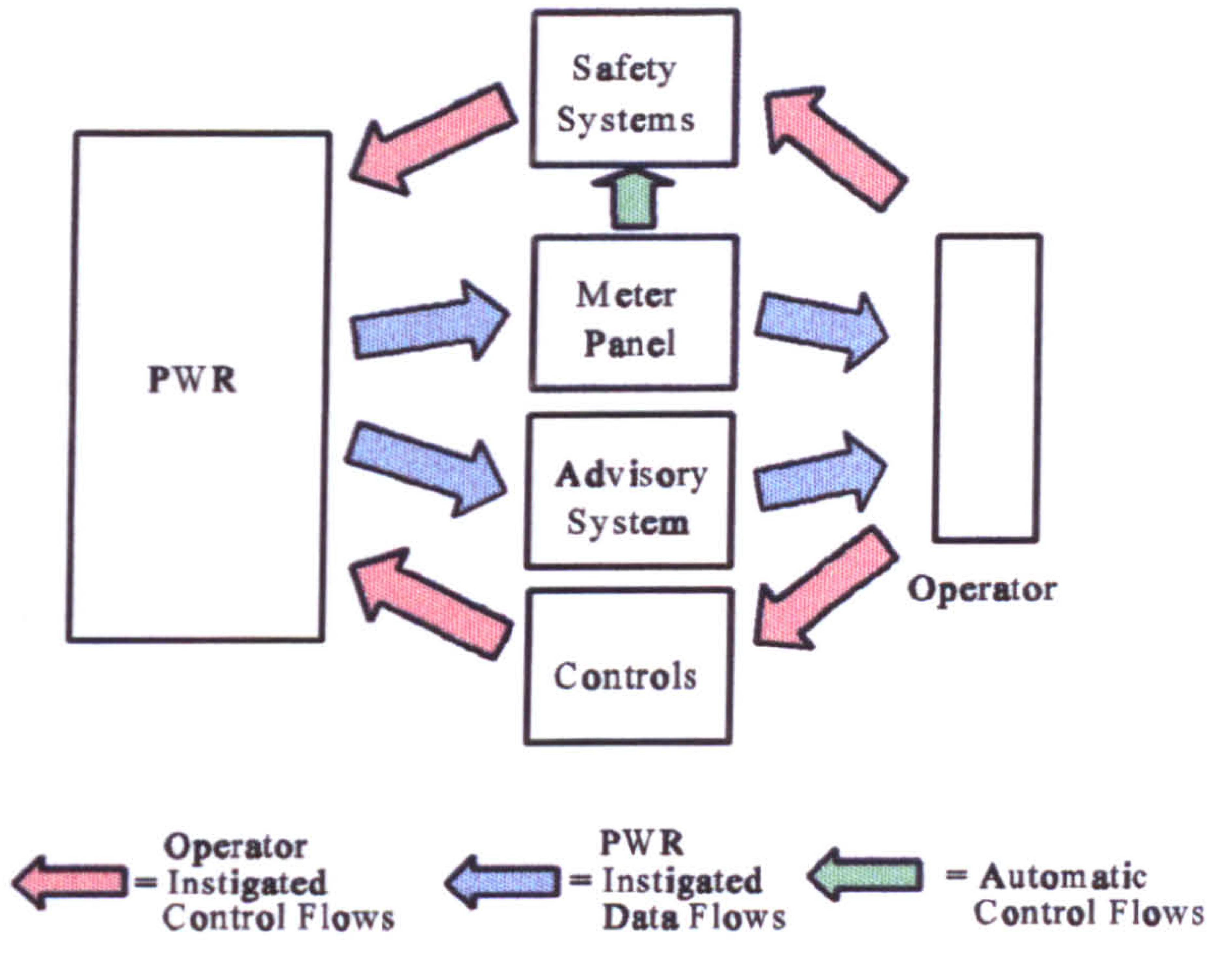


Fig 3.5: System with Operators Advisor

The previous diagrams also show the dependence of the operator on the information obtained from plant monitoring equipment. The control of many complex systems is established upon the value of indirect measurements. For example, both safety and practical reasons prevent direct measurement of PWR core temperature. If any sensor or meter was to malfunction the operator would have an incomplete picture of plant condition. Experience and training would compensate to an extent but this could be a time consuming process. The advisory system could provide further rapid additional support in the case of instrument malfunction.

3.4 Summary

This chapter has discussed the problem to be addressed by the remainder of this thesis. The role of the operator of a nuclear reactor was examined and factors that affect efficiency, such as stress and time, identified. The need for an aid to assist the operator during periods of high stress and quick reaction times, as in the occurrence of a LOCA, was introduced. The aim of the project was then defined as the design of a computer

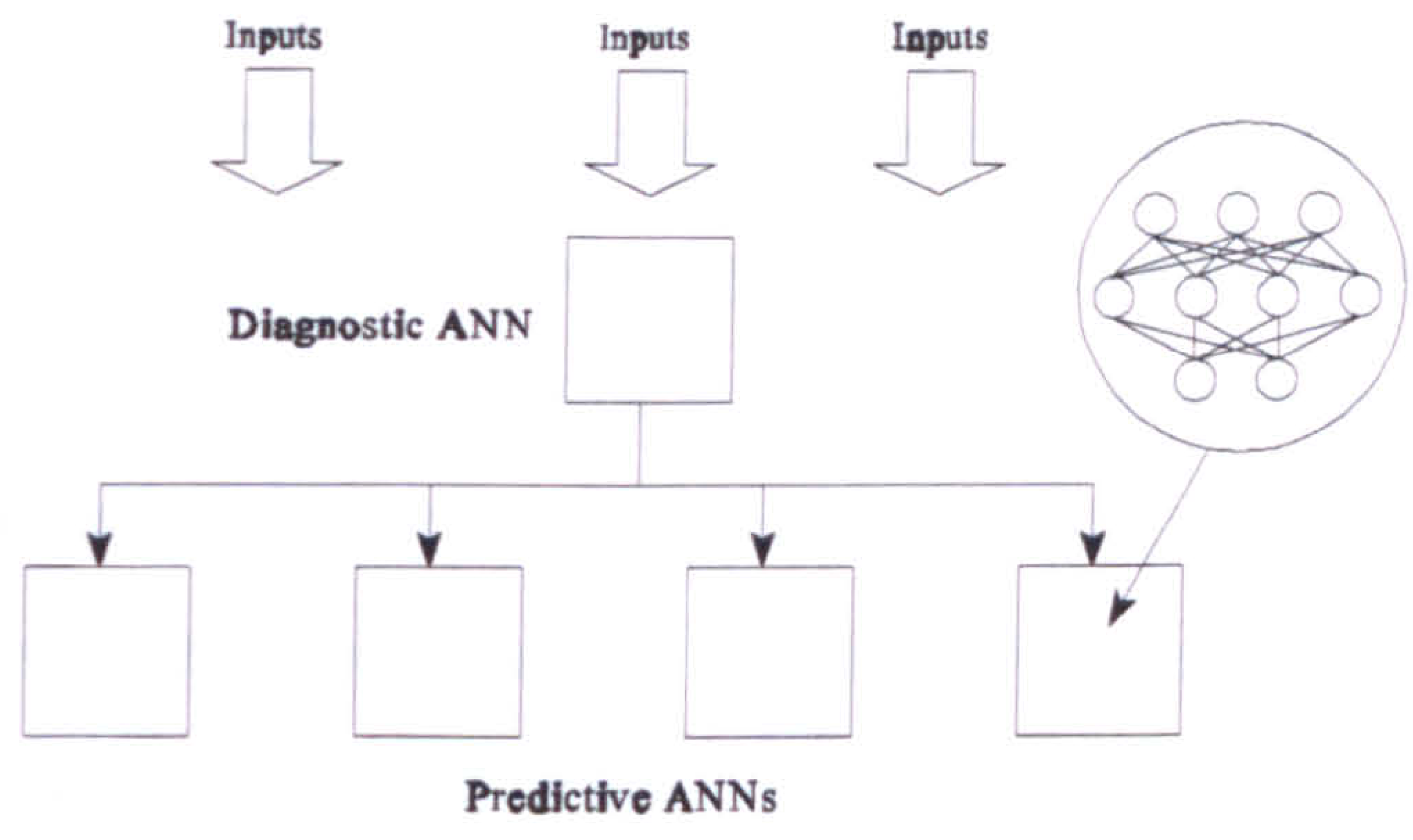


Fig 4.1 Hierarchy of ANNs

The top most level is developed as a diagnostic tool to report the current condition of the reactor. A number of readings for plant variables are fed into the network. This information could include recent history of the reactor. This input set could be a sub-set of the whole range of possible plant variables. The output from this part of the system would be an identification of reactor condition and, if in fault, an identification of transient type and location. Additional information could be made available to the operator and allow him to consider remedial action. The figure below, Fig. 4.2, shows the proposed system. The diagnosis is performed at a pre-defined time interval. However it is possible that this interval could be modified to reflect the seriousness of the reactor condition. If a reactor was diagnosed as being in a serious fault condition the time interval between subsequent diagnosis could be reduced so more frequent observations were made of the plant than in normal operating conditions.

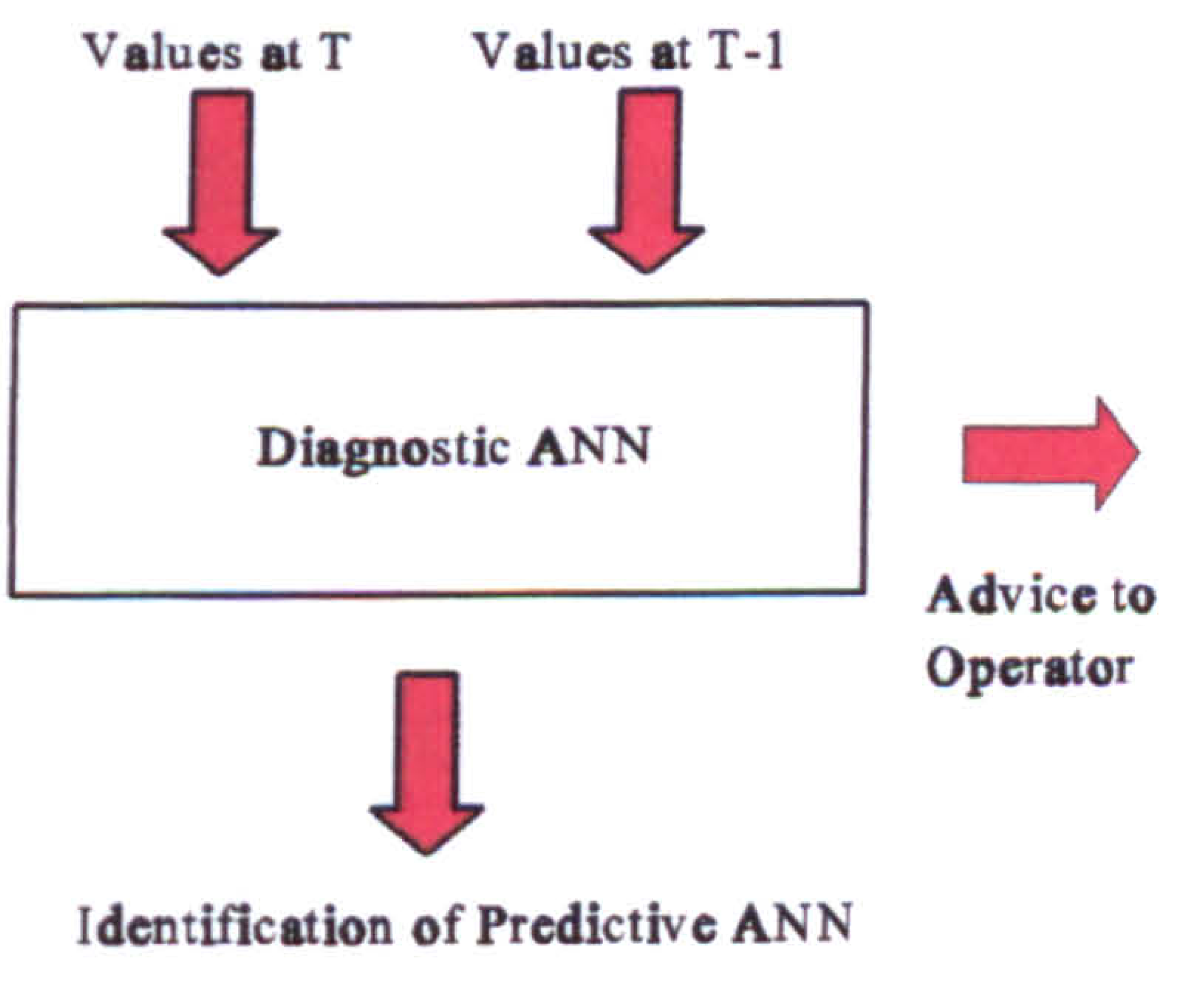


Fig 4.2: Diagnostic Stage of Advisor

principle to predict many steps ahead. Networks trained for more than a single transient are then considered and ANNs are developed to predict many transients. Finally a discussion on predicting PWR condition with an ANN and concludes that the number of transients a ANN can accurately predict is paramount to the successful development of the predictive layer of the operators advisor. This question is addressed, in detail, in the next chapter.

This chapter mentions PWR components and principles, a detailed introduction to the PWR is included in Appendix A.

5.2 One Step Prediction

The simplest form of modelling the future condition of a reactor is to predict a single state of the plant in the near future. A set of reactor variables for a certain time are fed into a suitably trained ANN. The corresponding output is a prediction of these variables for a future time. The ANN is trained to be able to predict a set of such future values. For accurate predictions it may be necessary to included some of the reactor's recent history in the input set, as shown in the figure below, Fig 5.1. This information may consist of past values of the variables under consideration. Alternatively, the recent state of the plant could be represented as a single input for current condition, say a `0' for normal operation and a `1' for a fault condition.

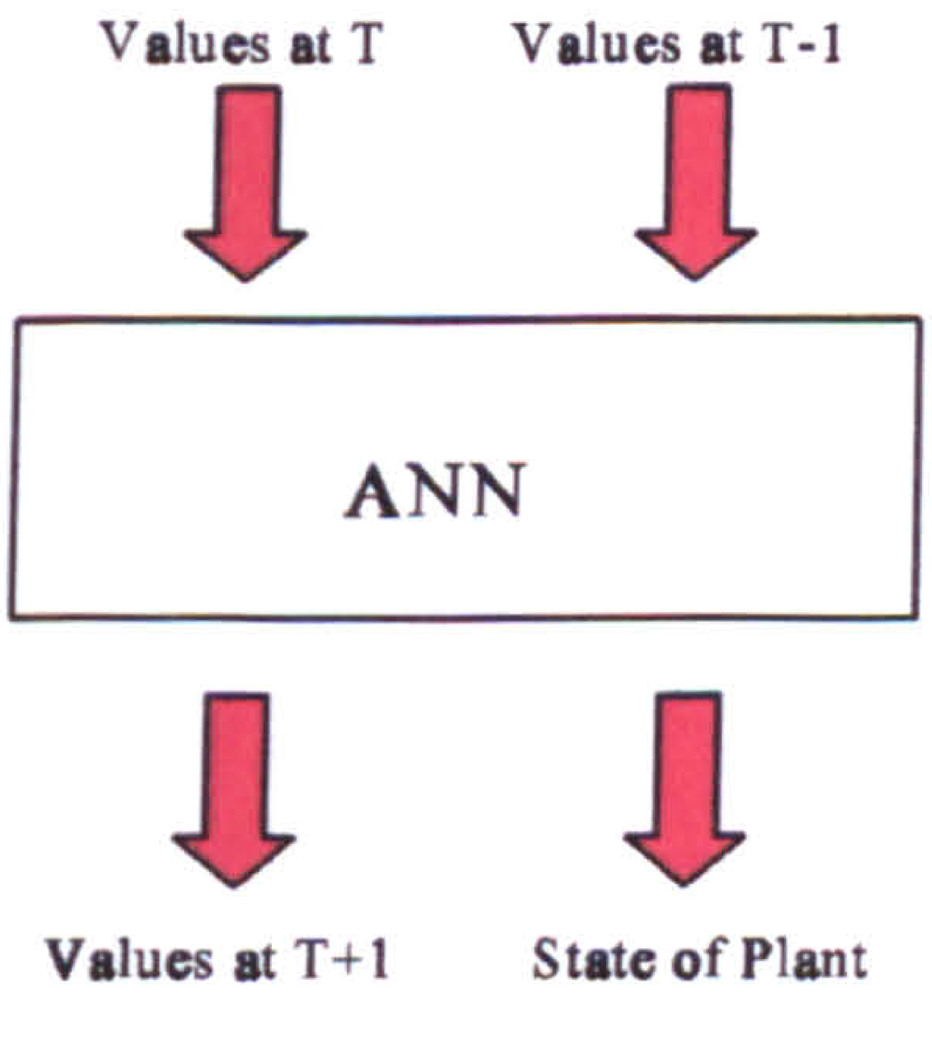


Fig 5.1: Prediction of Future Reactor Variables

ANNs have been successively used to predict the future values of time series from other complex system domains. However, little work has been carried out to assess the applicability of ANNs for prediction in the nuclear industry and PWRs in particular. A series of tests were, therefore, conducted to investigate the ability of an ANN to predict a

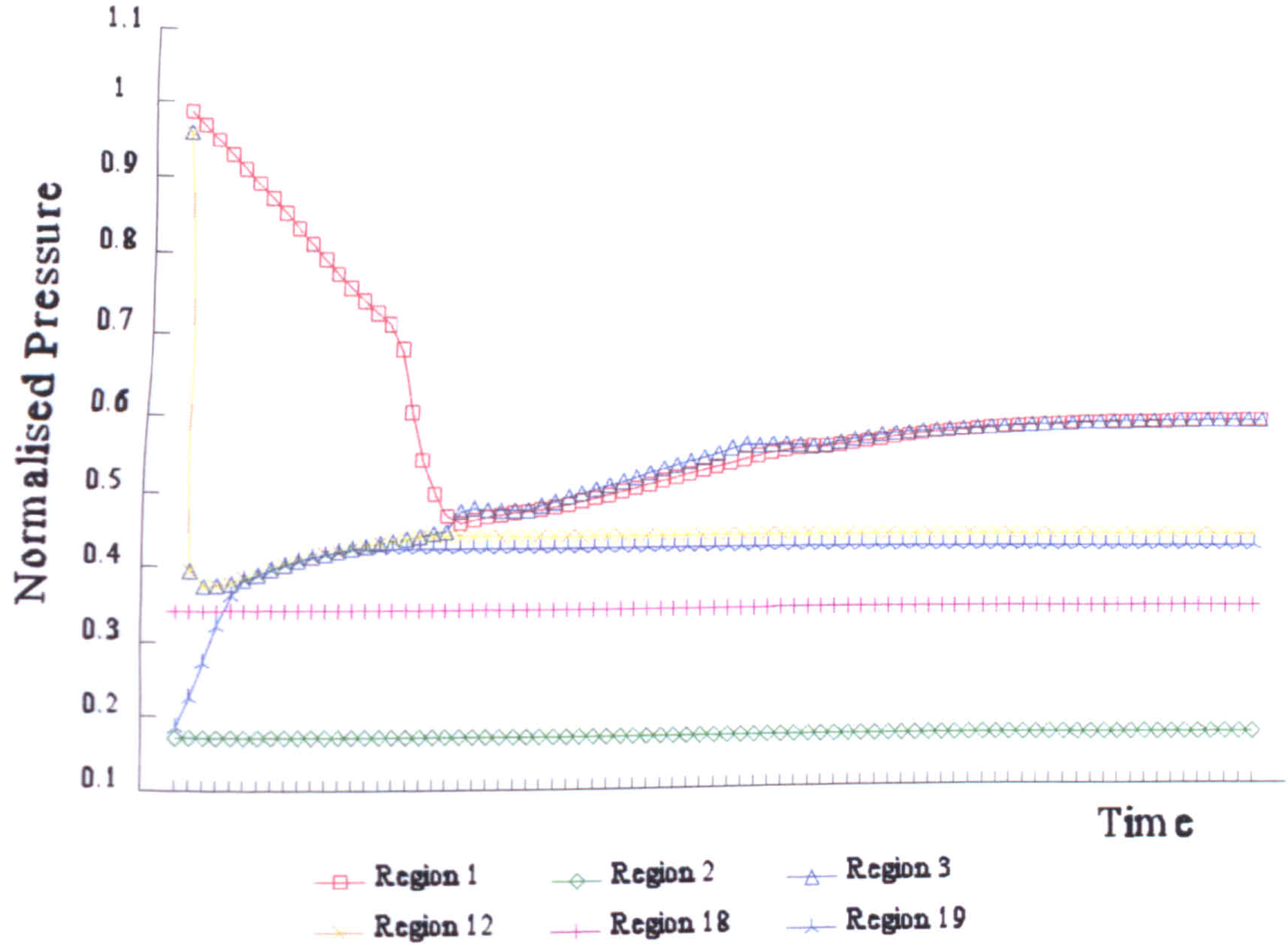


Fig 5.2: Pressure Outputs for Transient 1

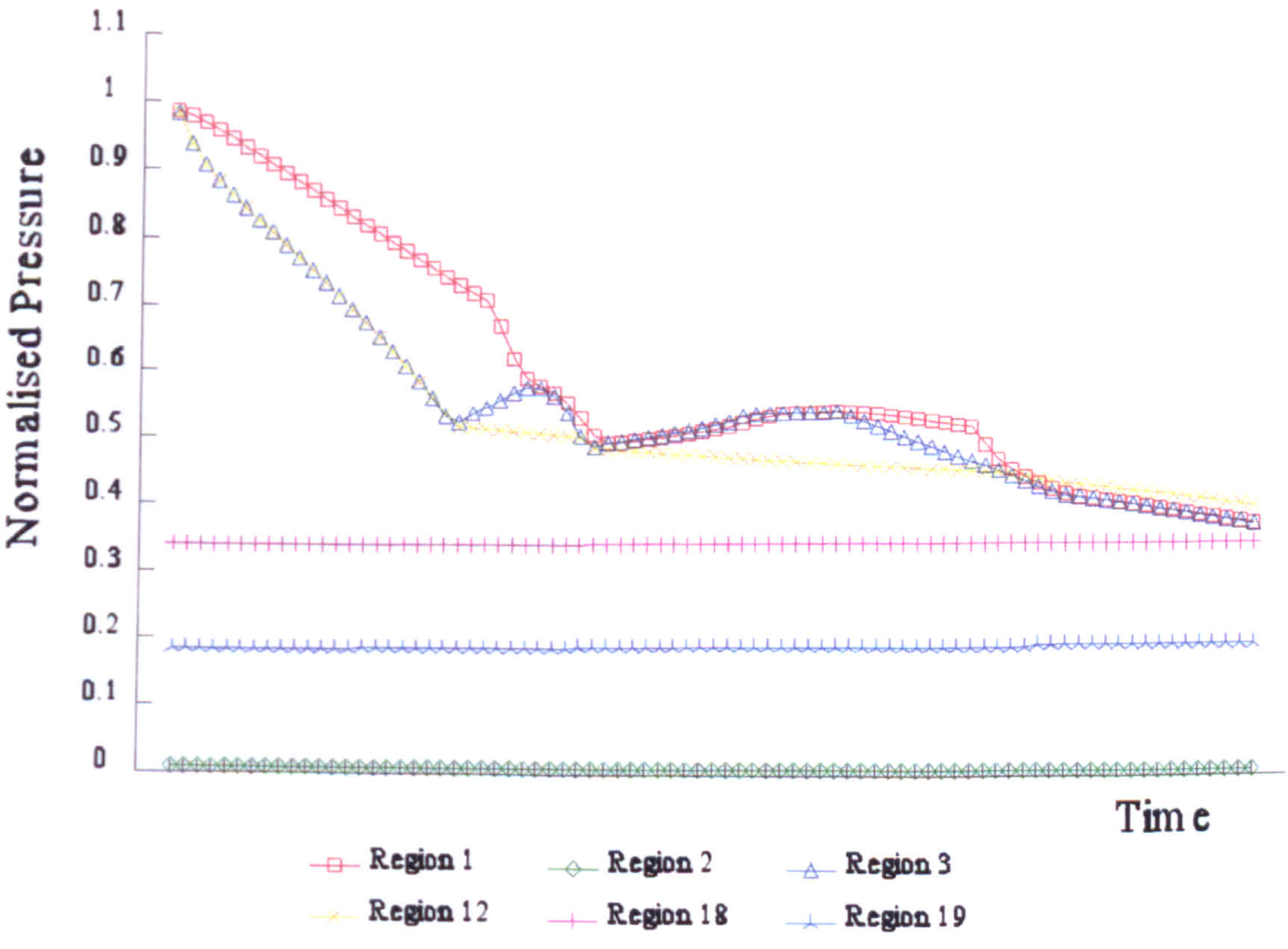


Fig 5.3: Pressure Outputs for Transient 2

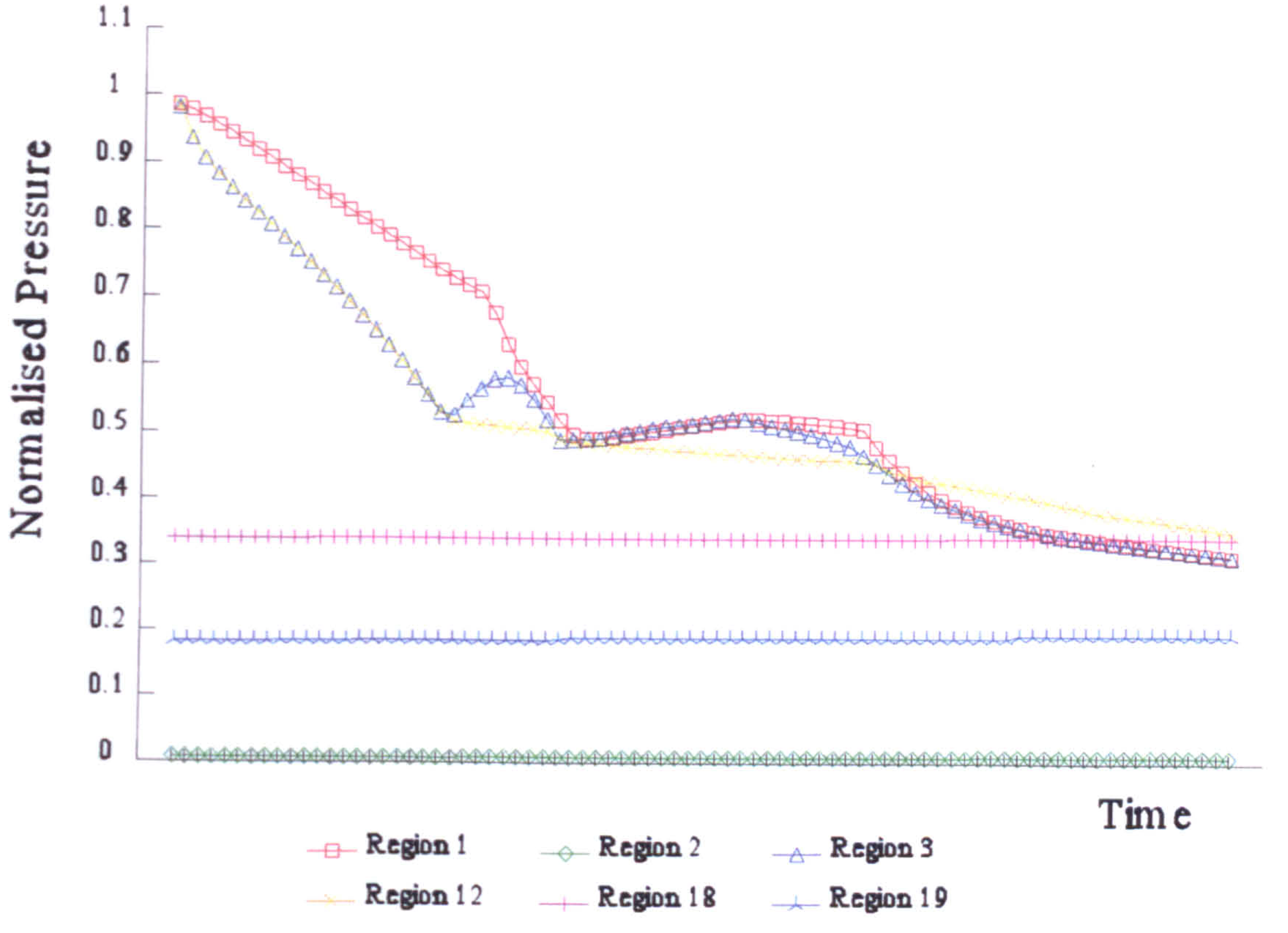


Fig 5.4: Pressure Outputs for Transient 3

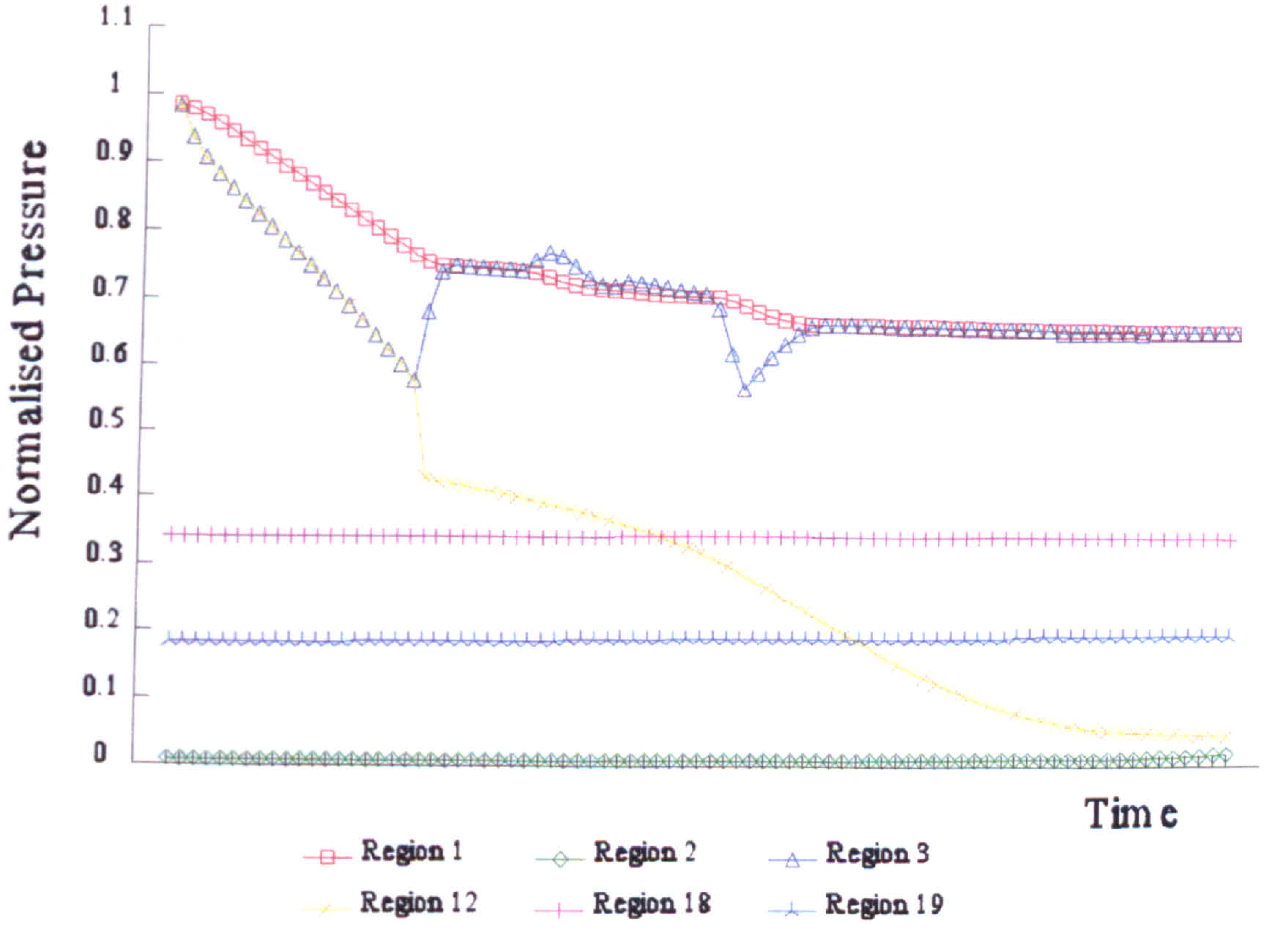


Fig 5.5: Pressure Outputs for Transient 4

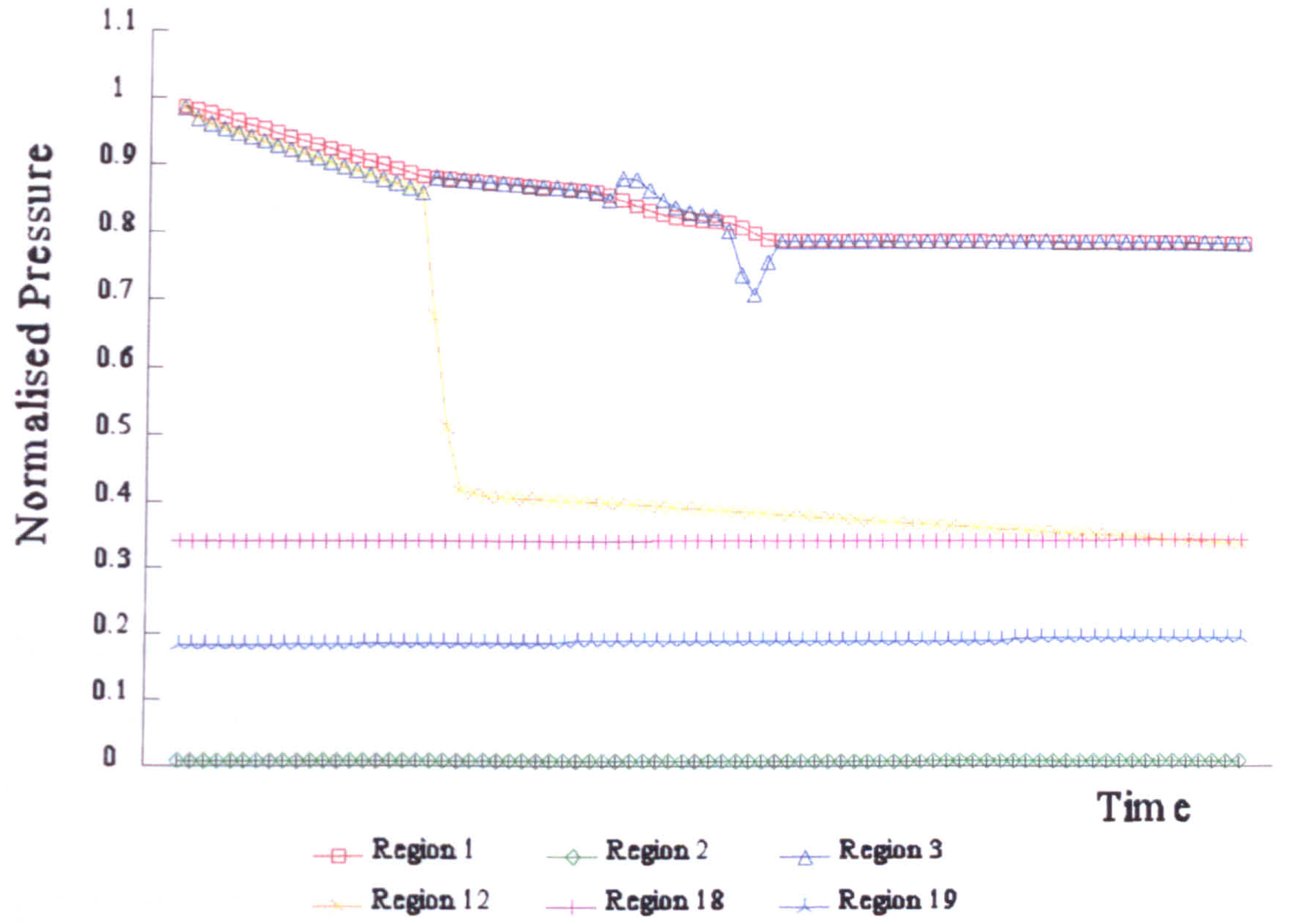


Fig 5.6: Pressure Outputs for Transient 5

A separate ANN was then developed to predict the values of these pressures one time step into the future for each set of inputs. The pressure values calculated by the simulation program, stored in a separate file were used to produce the data for training the ANNs. The initial output file consisted of a time reference and the values of the pressure in each region at that time. This file was modified to remove any reference to time and to reflect the number of steps of past reactor history being used to train the ANN. For example, to train an ANN to predict a set of future pressure values with information from two time steps a training set would require data on the two successive steps of output from the simulator as inputs and the next set of values as output. To illustrate this idea further consider Table 5.1 which shows a representation of four successive sets of nodal pressures in a simulator output set, the letters a to f signify the different PWR regions under consideration and the number identifies the time step, ie c2 is the second pressure value for region c.

The following table contains the results of the best ANNs trained to predict Transient 1 for each combination of input time steps and ANN hidden layers. The full results are given in Appendix C.

Time Steps	Hidden Layers	No. of Nodes	RMS Error
1	1	6	0.0314
	2	10, 8	0.037558
2	1	4	0.0281
	2	6, 4	0.0381
3	1	4	0.0261
	2	4, 4	0.0305
4	1	4	0.0351
	2	4, 4	0.0395

Table 5.3: Best ANN Results for Transient 1

The following figure, Fig 5.7, shows the one step prediction for the best of the above ANNs. The accuracy of the outputs can be determined by comparison with Fig 5.2.

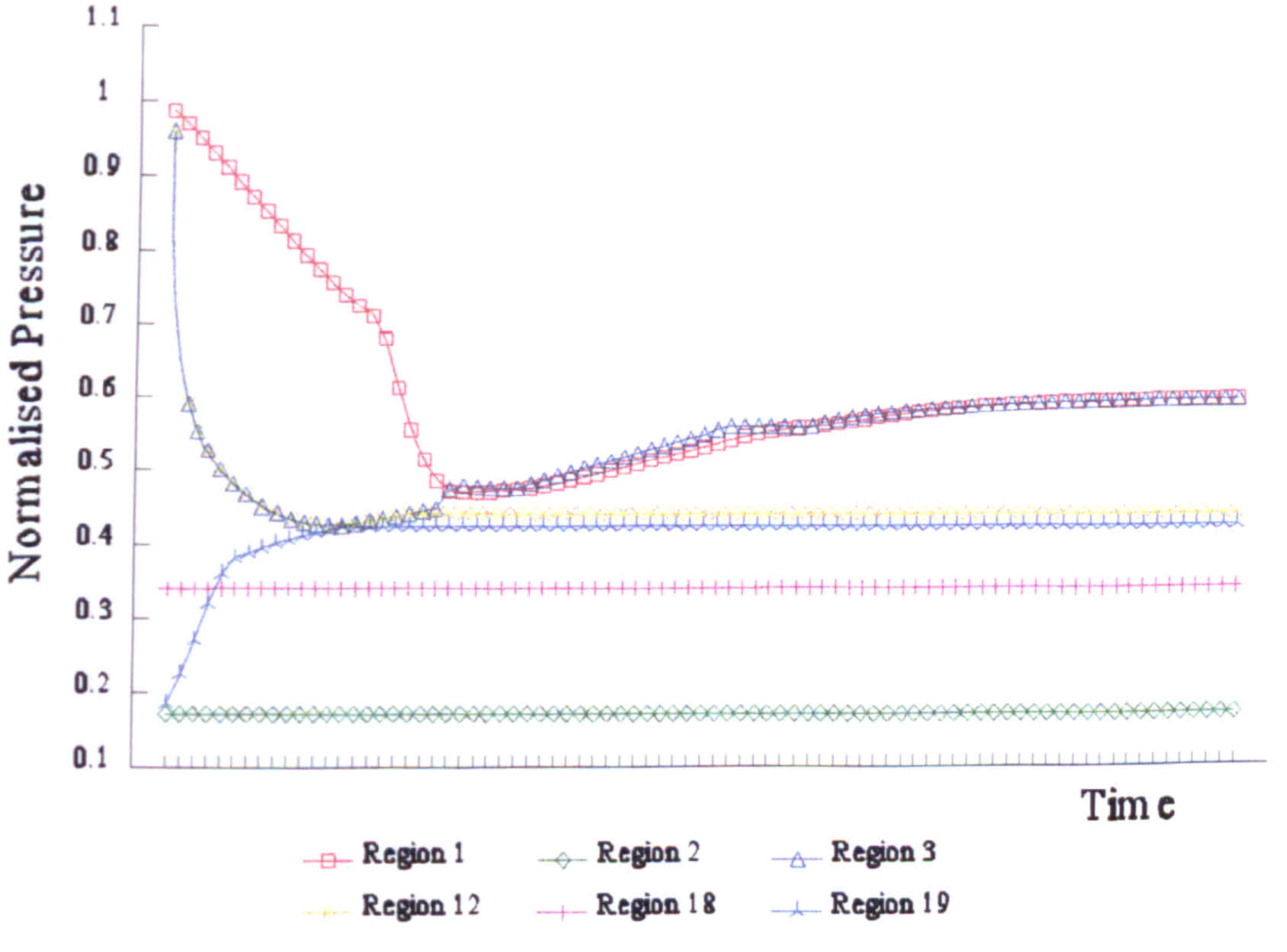


Fig 5.7: Outputs for Best ANN Predicting Transient 1

The following table contains the results of the best ANNs trained to predict Transient 2 for each combination of input time steps and ANN hidden layers. The results are given in full in Appendix D.

Time Steps	Hidden Layers	No. of Nodes	RMS Error
1	1	4	0.0306
	2	10, 8	0.037558
2	1	6	0.0283
	2	6, 4	0.0382
3	1	4	0.0272
	2	6, 4	0.0384
4	1	4	0.0351
	2	2, 6	0.0380

Table 5.4: Best ANN Results for Transient 2

The following figure, Fig 5.8, shows the one step prediction for the best of the above ANNs. The accuracy of the outputs can be determined by comparison with Fig 5.3.

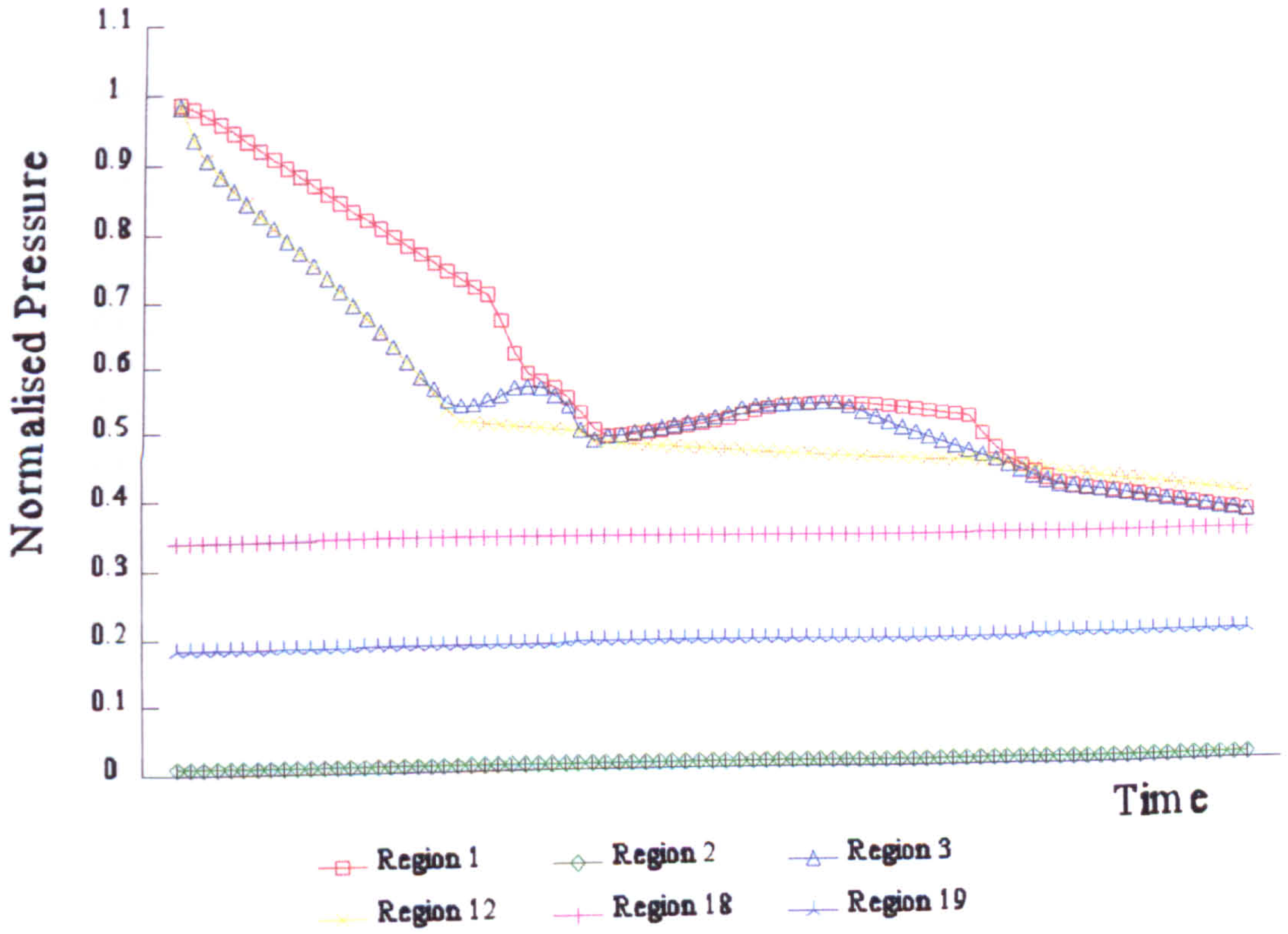


Fig 5.8: Outputs for Best ANN Predicting Transient 2

The following table contains the results of the best ANNs trained to predict Transient 3 for each combination of input time steps and ANN hidden layers. The full results are given in Appendix E.

Time Steps	Hidden Layers	No. of Nodes	RMS Error
1	1	6	0.0325
	2	4, 4	0.0372
2	1	8	0.0328
	2	4, 10	0.0380
3	1	8	0.0324
	2	6, 6	0.0393
4	1	4	0.0351
	2	4, 8	0.0383

Table 5.5: Best ANN Results for Transient 3

The following figure, Fig 5.9, shows the one step prediction for the best of the above ANNs. The accuracy of the outputs can be determined by comparison with Fig 5.4.

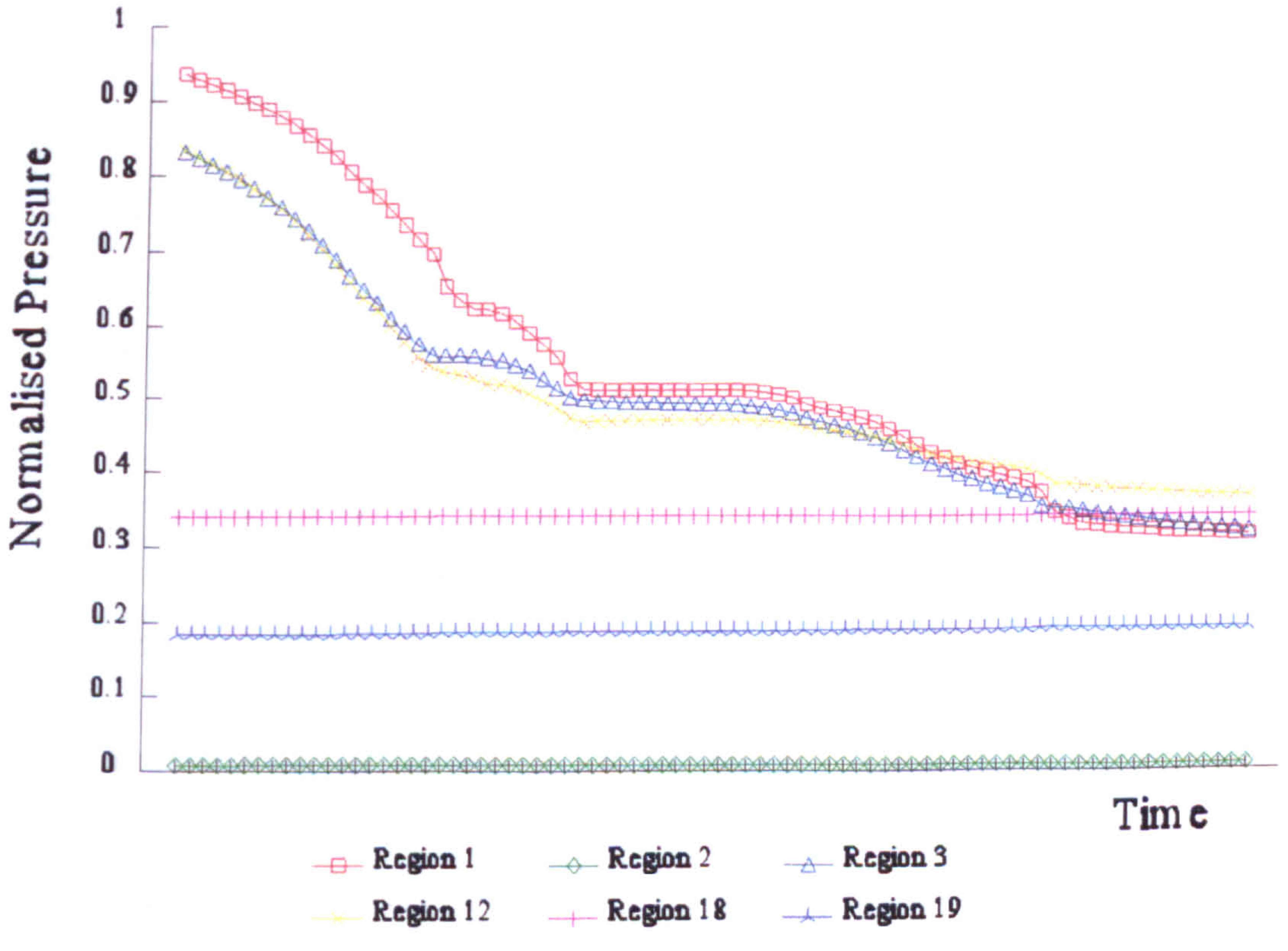


Fig 5.9: Outputs for Best ANN Predicting Transient 3

The following table contains the results of the best ANNs trained to predict Transient 4 for each combination of input time steps and ANN hidden layers. The full results are given in Appendix F.

Time Steps	Hidden Layers	No. of Nodes	RMS Error
1	1	6	0.0321
	2	4, 10	0.0396
2	1	4	0.0295
	2	6, 4	0.0358
3	1	4	0.0259
	2	6, 4	0.0339
4	1	4	0.0359
	2	6, 6	0.0388

Table 5.6: Best ANN Results for Transient 4

The following figure, Fig 5.10, shows the one step prediction for the best of the above ANNs. The accuracy of the outputs can be determined by comparison with Fig 5.5.

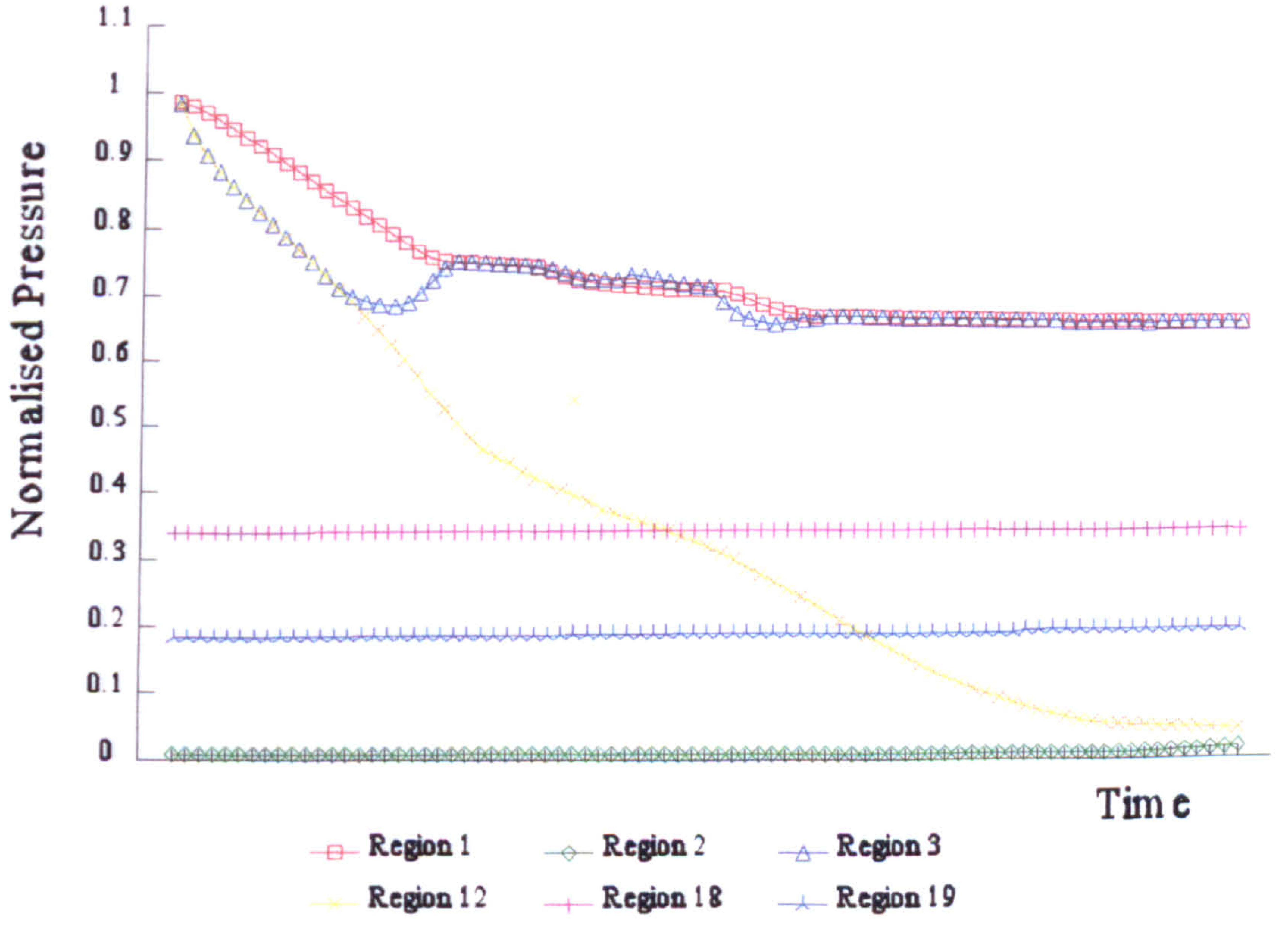


Fig 5.10: Outputs for Best ANN Predicting Transient 4

The following table contains the results of the best ANNs trained to predict Transient 5 for each combination of input time steps and ANN hidden layers. The full results are given in Appendix G.

Time Steps	Hidden Layers	No. of Nodes	RMS Error
1	1	6	0.0355
	2	10, 8	0.0376
2	1	4	0.0303
	2	4, 6	0.0355
3	1	6	0.0291
	2	6, 4	0.0340
4	1	15	0.0358
	2	6, 4	0.0385

Table 5.7: Best ANN Results for Transient 5

The following figure, Fig 5.11, shows the one step prediction for the best of the above ANNs. The accuracy of the outputs can be determined by comparison with Fig 5.6.

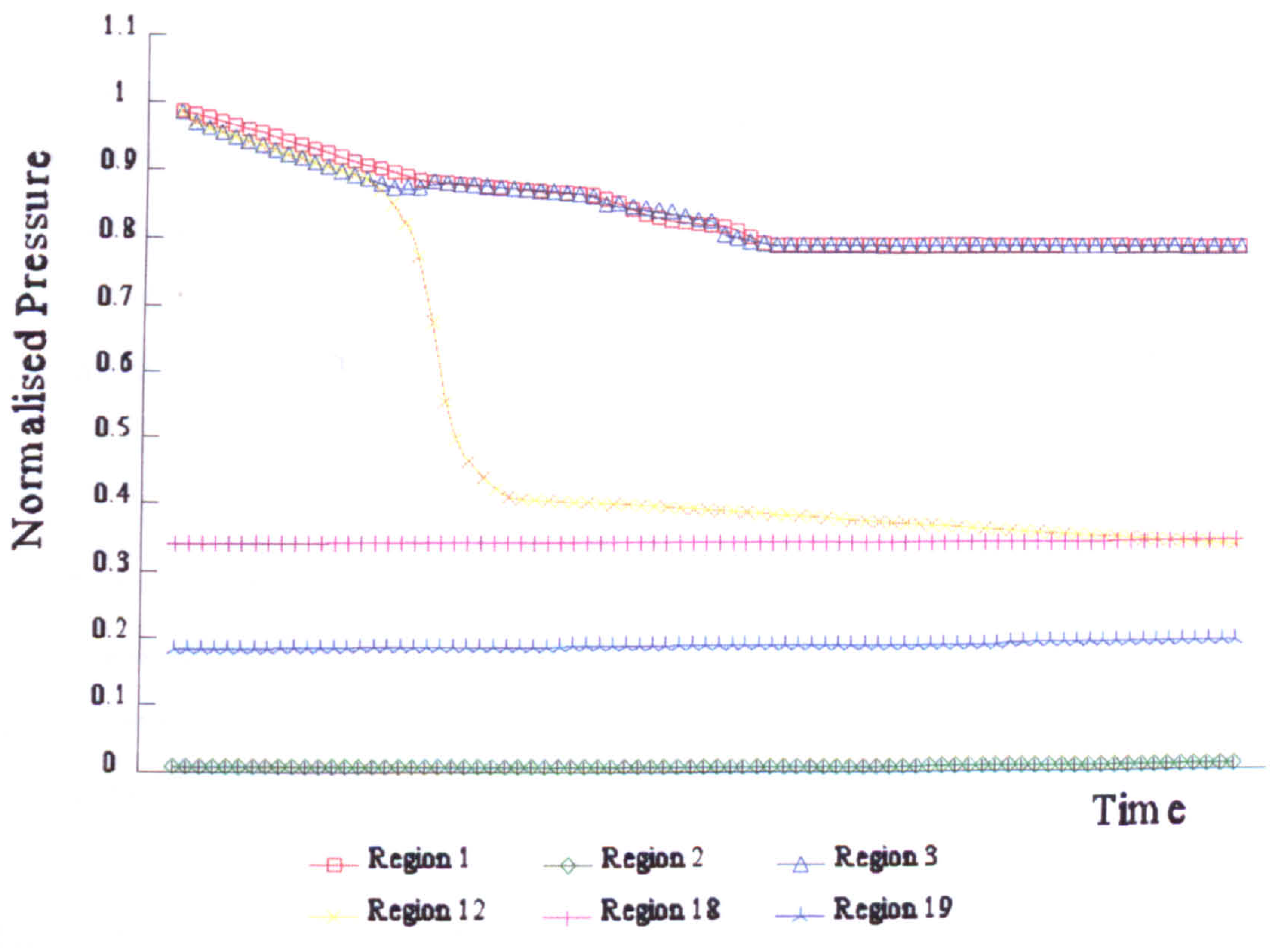


Fig 5.11: Outputs for Best ANN Predicting Transient 5

The results obtained are very consistent. All five transients are fairly well predicted for each combination of time steps of input and ANN architecture. The ANNs are not given any concept of time. The inputs are presented to the network in an arbitrary order, yet the ANN is able to return valid predictions. Perhaps most importantly the outputs from the ANNs follow the general trend of the desired outputs. The rate of change of the ANN outputs matched the simulator outputs very closely. The main discrepancy between the two occurs at turning points, for example from a negative to positive gradient, in these cases the ANNs smooth the output to a spline. An example of this is shown below in Figure 5.12.

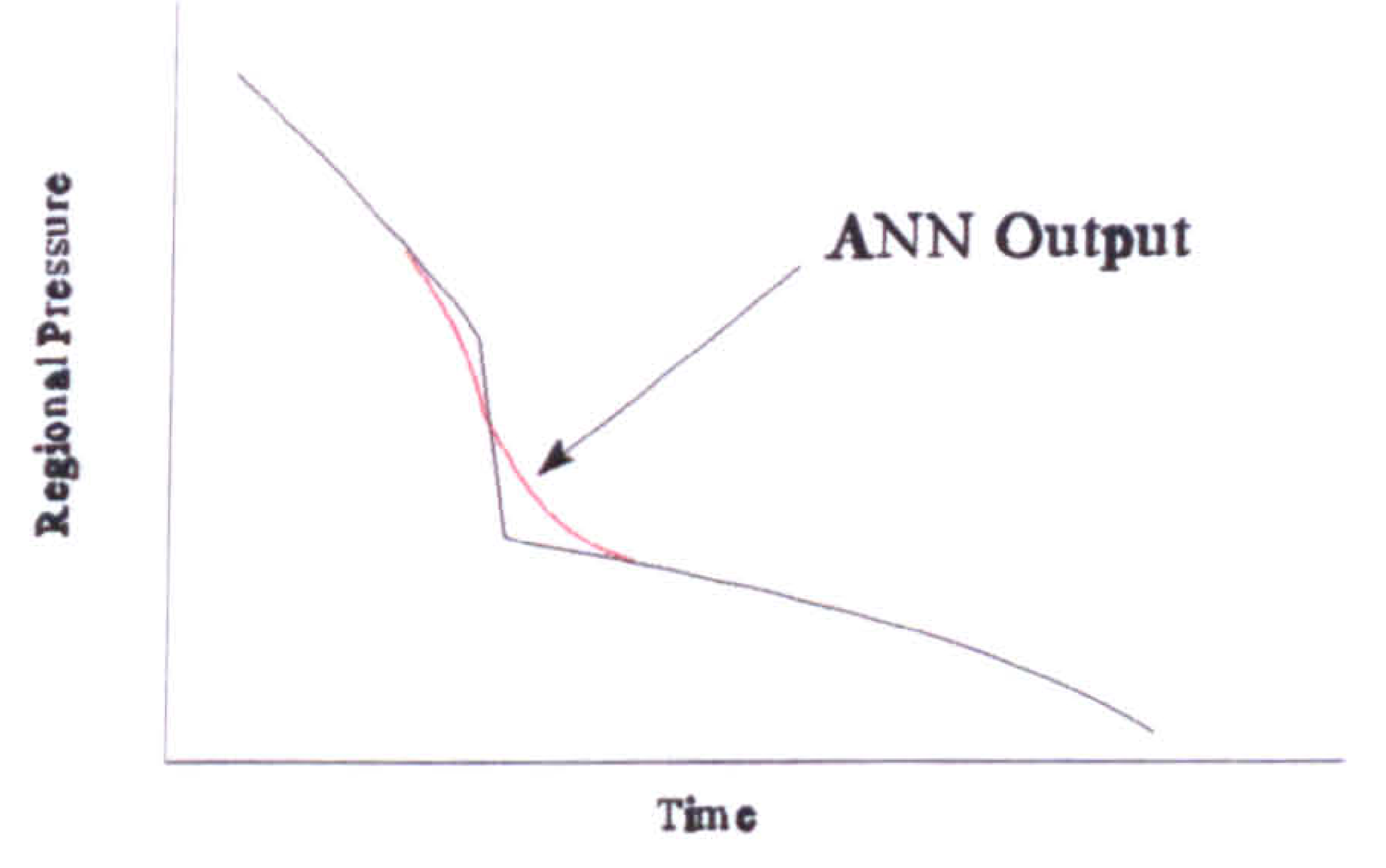


Fig 5.12: Typical ANN Output for PWR Variable Prediction

All the ANNs converged to similar RMS error levels, all less than 0.04. This result shows the robustness of using ANNs to predict PWR variables. The ANNs with a larger internal structure may well have not converged to the same level as the smaller ANNs given the relatively small number of cases in the training set. If the results had not been acceptable a larger training data set could have been produced by two methods. The first would have been to continue the simulation of each transient for a longer period; however, as seen in Figures 5.2 to 5.6 the variables are already at new steady state values and any further similar values would cause the ANN training set to be dominated by the new steady state. The second method of producing more training data would be to record further values during the transient simulation. A time interval of 3 seconds was used for the current tests but this could have been reduced to enable the creation of sufficient information.

An examination of the internal structure of a sample ANN, 3_2x1af.nnd, is shown below in Table 5.8 with Figure 5.13 showing the related links between the ANN nodes. The key connections for each node are identified by the size of their inter-nodal weighting. For clarity these are highlighted in red. An examination of these weights reveals that no particular input or time step dominates the structure. All the inter-nodal weights, with the exception of the constant value outputs, are of a similar value. The ANN is not solely using the most recent values for the prediction. All the time steps appear to have a reasonable influence in the final output. The implicit relationships developed during the training process use all time steps to determine the next value of the pressures. The straight line output variables seem to be the most independent.

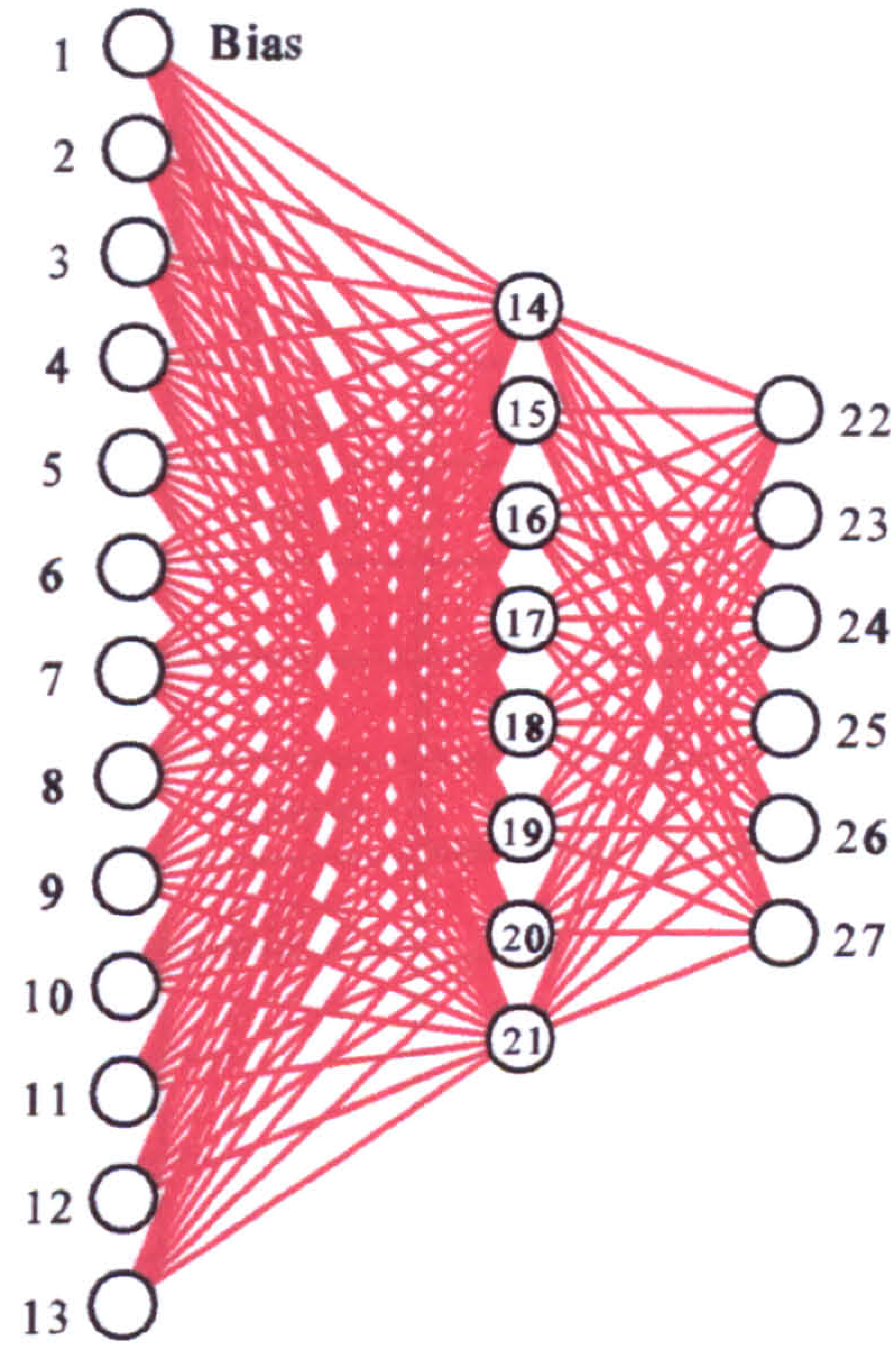


Fig 5.13: Node Structure of ANN

Link	Weight	Link	Weight	Link	Weight	Link	Weight
1, 14	-0.0226	2, 17	-0.1999	3, 20	-0.8079	20, 23	-0.9658
2, 14	-0.3045	3, 17	-0.4938	4, 20	0.0706	21, 23	0.0183
3, 14	**0.5263**	4, 17	0.1165	5, 20	0.0256	1, 24	-0.0125
4, 14	-0.0902	5, 17	-0.1633	6, 20	-0.4506	14, 24	0.0450
5, 14	0.1735	6, 17	-0.0193	7, 20	-0.3619	15, 24	0.4584
6, 14	0.1327	7, 17	-0.1742	8, 20	-0.1084	16, 24	-0.3021
7, 14	0.1870	8, 17	-0.1244	9, 20	-0.7346	17, 24	0.4079
8, 14	-0.2903	9, 17	-0.5980	10, 20	-0.0551	18, 24	-0.7338
9, 14	0.4661	10, 17	0.2044	11, 20	-0.0869	19, 24	0.1096

Link	Weight	Link	Weight	Link	Weight	Link	Weight
10, 14	0.0311	11, 17	0.0293	12, 20	-0.4379	20, 24	0.4081
11, 14	-0.0361	12, 17	0.0750	13, 20	-0.7293	21, 24	-0.6659
12, 14	0.2900	13, 17	-0.4114	1, 21	0.1362	1, 25	0.3263
13, 14	0.4384	1, 18	0.2267	2, 21	-0.1874	14, 25	0.2004
1, 15	-0.1036	2, 18	-0.0714	3, 21	-0.0810	15, 25	0.5196
2, 15	-0.1731	3, 18	-0.0127	4, 21	-0.6280	16, 25	-0.5254
3, 15	-0.2788	4, 18	-0.4911	5, 21	-0.3957	17, 25	-0.0407
4, 15	0.1964	5, 18	-0.4601	6, 21	0.0725	18, 25	-0.6743
5, 15	0.3370	6, 18	-0.1839	7, 21	-0.3306	19, 25	0.3177
6, 15	-0.2732	7, 18	0.1006	8, 21	-0.2145	20, 25	0.1891
7, 15	0.1964	8, 18	-0.0651	9, 21	-0.0698	21, 25	-0.8558
8, 15	0.1473	9, 18	0.0357	10, 21	-0.7058	1, 26	0.2151
9, 15	-0.3240	10, 18	-0.4807	11, 21	-0.7047	14, 26	0.1749
10, 15	0.3574	11, 18	-0.4343	12, 21	0.0036	15, 26	-0.5085
11, 15	0.1740	12, 18	-0.0988	13, 21	0.0307	16, 26	0.3544
12, 15	-0.2531	13, 18	0.2338	1, 22	0.1653	17, 26	-0.0798
13, 15	0.1299	1, 19	0.2716	14, 22	-0.4452	18, 26	0.2769
1, 16	0.1965	2, 19	0.0257	15, 22	0.2928	19, 26	-0.7340
2, 16	-0.0795	3, 19	0.0047	16, 22	-0.5376	20, 26	-0.8271
3, 16	-0.0305	4, 19	0.1486	17, 22	-0.0270	21, 26	0.5596
4, 16	-0.2421	5, 19	0.2696	18, 22	-0.2398	1, 27	0.1525
5, 16	-0.4907	6, 19	-0.3136	19, 22	0.9020	14, 27	0.5169
6, 16	0.0230	7, 19	-0.1476	20, 22	0.1945	15, 27	0.0280
7, 16	0.0441	8, 19	0.3078	21, 22	-0.6283	16, 27	0.3370
8, 16	-0.2659	9, 19	0.0422	1, 23	0.1944	17, 27	-0.3628
9, 16	0.1326	10, 19	0.0325	14, 23	0.7980	18, 27	0.5085
10, 16	-0.5474	11, 19	0.3644	15, 23	-0.3080	19, 27	-0.6361
11, 16	-0.2496	12, 19	-0.6369	16, 23	0.1536	20, 27	-0.9227
12, 16	0.1779	13, 19	-0.3529	17, 23	-0.6310	21, 27	0.2237
13, 16	0.0893	1, 20	0.6729	18, 23	0.1454		
1, 17	-0.0861	2, 20	-0.1861	19, 23	0.0348		

Table 5.8: Internodal Weights for Transient 3 Prediction ANN

based advisory system for the plant operator. The system would diagnose the current plant condition and then predict the future condition based on that diagnosis. Finally the problem was examined in terms of a system approach.

Chapter

4 Modelling Method

4.1 Introduction

This chapter discusses the proposed solution to the problem introduced in the previous chapter. The rational for employing the chosen techniques are described. An outline of the modelling methods used are introduced. Finally a hybrid model of the two techniques is introduced. Appendix B of this report includes a basic introduction to the Artificial Neural Networks (ANNs) and Genetic Algorithms (GAs).

4.2 Modelling Methods

The computer based advisory system proposed in the previous chapter could be implemented in several ways. An automatic system has several advantages.

1) The proposed method would enable a common advisory system to be installed in every plant. This would enable cost effective large scale implementation. Furthermore operators familiar with the system would be able to change plant without retraining.

2) A computer based advisory system would be always available for use and would not become tired or lose concentration.

3) A computer based advisory system could include an automatic audit trail for checking both advisor performance and PWR records.

Such a system could use statistical methods, such as Bayesian Belief Networks, or Expert systems. These two approaches each have a disadvantage. A Bayesian belief network does not allow feed-back loops, a calculus that can cope with feed-back has not yet been developed for causal networks. Expert systems have difficulty with missing or noisy data. Even fuzzy expert systems are not yet fully tested in this respect. A further possibility is to develop the advisor with ANNs.

ANNs have several advantages over other possible tools for developing an operators advisor.

1) An ANN is relatively easier to update with new conditions and examples. An existing ANN can be re-trained with a modified training set.

2) An ANN solution can be developed in a relatively short development time. The time consuming element of process is the acquisition and processing of the training data.

3) ANN technology is cheaper to both research and implement as high specification computers and available ANN software are now cost effective. The wide availability of suitable tools removes the need of expensive bespoke equipment.

4) ANNs can try to approximate a best attempt solution for unknown fault conditions.

5) The development of an ANN does not require extensive use of a domain expert

However, ANNs do have some shortcomings.

1) The safety justification of such a system is problematic although this problem is being addressed by other researchers (Bishop, 1996).

2) An explanation of decision process used by an ANN is not easily available. The popular conception of ANNs is of a "Black Box" technology

3) The available training data may not result in the ANN converging to an acceptable level

ANNs will be the predominant tool used for developing this advisor. In addition to the above advantages of ANN technology there are several reasons for making this selection. It must be stressed that the human operator retains final decision regarding PWR control. The current research in AI has not yet been successful with modelling intuition, innovation, serendipity or even consciousness.

In a nuclear reactor, or any complex system, there is a large set of possible fault scenarios. Not all of this set can be realistically modelled or considered in a practical advisory system. The knowledge base would be far too large to use and it would not be certain that all possible combinations had been considered anyway. The use of a neural network is a good solution to this dilemma. During training the ANN develops implicit relationships between the variables which are used when considering previously unseen situations.

Once trained an ANN has instantaneous operation and so gives quick results.

As already stated an ANN based advisory system can be implemented on a computer based system using a commercial software package thus avoiding the need for expensive bespoke equipment. If, at a later date however, cost effective production quantities are required then the developed advisor system could be migrated to a commercially available neural network integrated circuit.

4.3 Hierarchical Approach

The computer based advisory system, described in the previous chapter, has two very different requirements. Whilst each requirement is a pattern recognition task, they have different forms of output. The diagnosis of current reactor condition is a classification problem where the input is assigned to one of a discrete number of classes. The prediction of future states based on the result of the diagnosis is an iterative problem where the outputs are values of continuous variables. A single ANN arrangement would be unable to fulfill this task as the two requirements are very different and would make contrasting demands on the network. The diagnostic component is designed to give a one off guide to the state of the reactor, while the prediction component has to give a sequence of outputs for the future condition of the reactor.

A method of resolving this dilemma and also increasing the functionality and practicality of the advisor would be to use a hierarchy of ANNs (Basu & Bartlett, 1994). The figure below, Fig 4.1, shows a simple hierarchy of two layers. This arrangement is the minimum configuration as in practice there may well be several more levels. The blocks in the figure could represent arrangements of ANNs. However the underlying reasoning still remains valid.

The lower levels of the system are used to predict the future state of the reactor in the diagnosed condition. In normal circumstances the prediction will consist of steady state conditions. It is envisaged that the prediction system will consist of a number of prediction branches. Each of these systems could be developed for a particular range of transient conditions. The particular predictive system to be used is identified from the output of the diagnostic system. Each prediction structure so selected uses additional, relevant readings of plant variables as the inputs. These systems could also use the short term history of the reactor's condition to increase prediction accuracy. The output from the system is a prediction of the reactor variables at a defined time step into the future, again this function could be windowed to reflect the reactor condition. This reading may itself be used as an input to the predictive system, using a feedback principle. By this method a real-time, or faster, prediction for the full period of the fault condition may be obtained. The upper level of the advisory system will continue to monitor the reactor and, with additional information, refine the diagnosis.

When the transient has been detected and addressed, the advisory system can also monitor the recovery of the reactor. The diagnostic element will still report the transient but can advise the operator of a reducing seriousness. The prediction system will, in the meanwhile, hopefully predict the full recovery of the reactor and be able to provide an indication of recovery time.

The hierarchical system, described above, has some additional advantages over a large, single ANN based system. The networks in the proposed system will be much easier to update and modify than a single network. A revised prediction ANN can be trained, using additional inputs if required, and inserted into the system to replace its predecessor when trained and tested and approved. The remainder of the system is unchanged and functions as before.

The hierarchical nature allows the system to be extended to extra prediction levels permitting greater detail to be included. Correspondingly, the diagnostic system can also be enlarged to provide more information to the operator, such as location or size of fault. Outputs to reactor transients unknown to the advisory system may still be produced from this enhanced system. The accuracy of the diagnosis from these new inputs may not be as good as that obtained from transients used for training the ANNs. However, the implicit relationships developed in the ANN during training should enable the ANN to produce some guidance to the operator for unfamiliar transient scenarios.

Once the particular ANNs are trained, tested and implemented the system would perform very quickly in operation. The diagnostic system will give a result as soon as the required number of time steps used for input have elapsed. The speed of the prediction system returning a guide to future reactor condition will depend upon the frequency of estimates and the length of the prediction. However once the initial plant data is available the predictions should be available very quickly. The system will perform considerably faster than the real time operation of the PWR.

The outputs from both the diagnostic and predictive elements of the advisor could be saved to provide a record of reactor history. This information could be used for audit purposes and to monitor the operational accuracy of the advisor.

4.4 Hybrid Modelling

A nuclear reactor is a complex, data rich environment where many readings of plant variables are recorded. The nature of these readings includes power, temperatures and pressures. The operator monitors this myriad of information to assess the plant condition. If all possible data were to be used to construct an ANN it would, of necessity, be very large with many inputs and hidden nodes in the architecture. This size of network would be undesirable and has several disadvantages such as possible over-fitting of data compared to smaller networks (Masters, 1993, p176).

Firstly, a large ANN would be less likely to converge to the same level of accuracy as that of a smaller network. During training an ANN builds implicit relationships between the variables. Some of these connections may detract from the efficiency and accuracy of the performance of the network. Although some ANN training algorithms, typically back-propagation or a node pruning algorithm, will reduce the importance of such relationships their effect will not be totally removed. A smaller network consisting of only the most significant variables would not suffer from the same dilemma. Furthermore the ANN learning complexity is a function of the number of variables in a problem. For an increase in the number of variables in a problem to n, the associated complexity increases faster than a polynomial of order n (Basu & Bartlett, 1994).

Secondly, were the large network capable of being trained to an acceptable level of convergence, the time taken to do so would be considerable. The large number of inter-nodal weights in such a network need time to be iteratively optimised. A large network would also require a great number of examples in both the training and test datasets. One method used is to develop a training set with at least twice the number of inter-nodal weights in the network (Masters, 1993, p177). A bigger ANN therefore requires larger training and test sets. The presentation of such datasets would also increase the training time required to converge the network. Finally in an environment such as a nuclear reactor the acquisition of data itself may be problematical and unavoidably result in small training sets. Although the use of simulators can alleviate this situation the amount of available data can still be a problem.

A possible disadvantage of using the optimum number of relevant inputs is the susceptibility of the ANN to noisy, corrupt or missing data. The larger ANN with a number of possible superfluous inputs and implicit relationships will probably be more tolerant of input data problems. The smaller network is tuned to perform with the selected input set and has very little spare processing ability to handle bad or noisy input data.

The above discussion implies that a small network dedicated to a particular task is preferable to a large all-encompassing ANN. A method of achieving this would be to train the ANN only on the most significant PWR variables. A GA could be used to evolve the best combination of plant readings for a particular condition.

4.5 Summary

This chapter has described the hierarchical neural network approach to be adopted for the remainder of this report. The top-most levels are to be used for diagnosing the condition of the reactor, while the lower levels predict the state of the PWR based upon that diagnosis. The chapter has also introduced an approach, using genetic algorithms, to minimising the size of the neural networks used in the hierarchy. The next three chapters explore the components of the hierarchy, diagnosis and prediction, in greater depth.

Chapter

5 Initial Prediction Investigations

5.1 Introduction

This chapter examines the prediction element of the advisor in greater depth. The concept of the prediction layer of the proposed adviser has previously been described in Chapter Four. For a diagnosed reactor condition the artificial neural network (ANN) based prediction system could predict the behaviour of the reactor during the transient and inform the operator on the management of the transient. The inputs to the system would be values of recordable plant variables and the outputs will be values of key variables at a defined time step into the future. These values may be fed back into the prediction system to ascertain the behaviour of the plant for a longer period.

The remainder of this chapter is as follows, the work being described in the chronological order that it was performed. The initial investigations into the feasibility of using ANN technology for prediction in the Pressurised Water Reactor (PWR) environment progresses into using ANNs to predict future values of selected plant variables, mainly regional pressure, using the recent history of the variables as input. One step prediction is described first in which a single view of the PWR condition is determined. This is followed by using a feedback future value for a reactor during a transient. A series of ANNs were developed for a set of different PWR transients. These ANNs were then tested and compared to evaluate the potential of the ANN approach for the predictive layer of the advisor.

The following five different transients were selected to explore the validity of one step prediction using ANNs.

1) Large break from Primary to Secondary circuits
2) Medium/Large Non-Isolated break to Pressure Vessel Outlet
3) Medium/Small Non-Isolated break to Pressure Vessel Inlet
4) Large break to Hot Leg
5) Small break to Hot Leg

These transients were selected to provide a wide range of possible fault scenarios and also to compare similar conditions, for example the size of leak in transients 4 and 5 . Each transient was separately modelled on a PWR Simulator. The first four minutes of the transient was modelled in each case. The calculated nodal pressures were recorded for every 3 seconds of the leak duration giving 81 sets of values per transient. The pressures for six key regions of each transient is shown in Figures 5.2 to 5.6 below. These selected variables provided a combination of both non-linear and linear terms to explore an ANN's ability to predict dissimilar forms of output.

Time	Simulator Output
1	a1 b1 c1 d1 e1 f1
2	a2 b2 c2 d2 e2 f2
3	a3 b3 c3 d3 e3 f3
4	a4 b4 c4 d4 e4 f4

Table 5.1: General Example of Simulator Output

Continuing the previous example of using two time steps for prediction, one vector set in the ANN training data would include the values from times 1 and 2 as inputs and the values from time 3 as the corresponding output. A second vector set would consist of times 2 and 3 for inputs and time 4 as the output. set. Table 5.2 shows this method applied to the data from Table 5.1 for three time steps of input.

Number of time steps in ANN	Input of training set	Output of training set
One	a1 b1 c1 d1 e1 f1	a2 b2 c2 d2 e2 f2
	a2 b2 c2 d2 e2 f2	a3 b3 c3 d3 e3 f3
	a3 b3 c3 d3 e3 f3	a4 b4 c4 d4 e4 f4
Two	a1 b1 c1 d1 e1 f1 a2 b2 c2 d2 e2 f2	a3 b3 c3 d3 e3 f3
	a2 b2 c2 d2 e2 f2 a3 b3 c3 d3 e3 f3	a4 b4 c4 d4 e4 f4
Three	a1 b1 c1 d1 e1 f1 a2 b2 c2 d2 e2 f2 a3 b3 c3 d3 e3 f3	a4 b4 c4 d4 e4 f4

Table 5.2: Generic ANN Training/Test Sets

The above table shows that the total number of inputs to a predictive ANN depends on the number of time steps of reactor history considered. In the current tests a one time step predictive ANN consists of six input and six output nodes. Correspondingly a two time step network contains twelve input and six output nodes; a three time step ANN has eighteen inputs and six output nodes.

The total number of members in the training and test sets (N) are also determined by the number of time steps used (T). As the above example demonstrates for the given size of data examples there are three members in a one time step training set, two members in a two time step set but only one member in the three time step set. In general,

$$N = P - T$$

where P = Number of records in output file

Four different time steps were investigated in these tests. The smallest was a single time step while the largest used four time steps for prediction. The larger number of time steps was felt to be the maximum that could be used while still retaining a quick response from the advisor. To avoid any bias in the tests, the same number of members were used in all training and test sets, irrespective of the number of time steps considered. This was 78, the number of four time step, the lowest, cases. The information was then randomly divided between training and test sets in the ratio of approx 2:1.

A range of ANNs were trained for each size of input time step. The backpropagation algorithm was used for optimising the inter-nodal weights. The networks varied in number and size of hidden layers. A series of tests were performed to determine the best transfer function, in terms of RMS error, for this application. The Tanh function was found to offer the best performance and was used for all ANNs developed in the remainder of this chapter. All networks were trained for 120,000 cycles and then for a further 40,000 cycles with testing every 100 cycles, the best network being saved. The training process was repeated with different starting conditions to alleviate the possible effect of the training becoming caught in a local minima of the solution space. A 10% gaussian noise was included in the training stage to improve the robustness of the ANNs. The results obtained from the best ANNs for each input time step are given below. The full results for all ANNs developed are given in Appendices C to G. With each table of results is a graph, Figures 5.7 to 5.11 respectively, of the output from the best ANN for each transient. By comparing these with the corresponding simulator output, shown in Figures 5.2 to 5.6, the accuracy of the predictions can be seen. Points representing the same parameter on the graph are shown joined; however there is no direct connection between them, the graph merely shows the outputs returned when the a set of previous values were input into the ANN.

From the results achieved in this section an ANN appears capable of accurate one step prediction for a range of possible transient scenarios in a PWR. ANNs would therefore seem to be an ideal tool to employ for the prediction component of the advisory system. Although each ANN has been developed for the prediction of a single transient further investigation into more sophisticated approaches is justified by these results.

5.3 The optimum number of time steps for prediction

As seen in the previous section the immediate history of the reactor is an important component to consider when predicting future reactor condition. This information can give a meaningful guide to the changes that are occurring inside the reactor and, based upon these changes, a sensible prediction of the future changes can be made.

Throughout the project extensive use is made of PWR computer simulations for data generation. The simulation program performs calculations at pre-defined time steps, typically 0.1 seconds, and these information has been used to produce ANN training data.. A prediction system will produce an output at a defined time step into the future. In both cases the number of time steps used for the prediction is an important element for the accuracy and speed of the proposed adviser. Consider the following example. Fig 5.14, shown below, show the range of a typical reactor variable for a single arbitrary transient. The nature of the transient causes the selected variable to perform in a non-linear manner. Several values, such as value X in Fig. 5.14, occur more than once during the fault, although the gradient at each occurrence is different. A system to predict the reactor's performance under this transient would require a sufficient knowledge of the immediate previous history of the reactor.

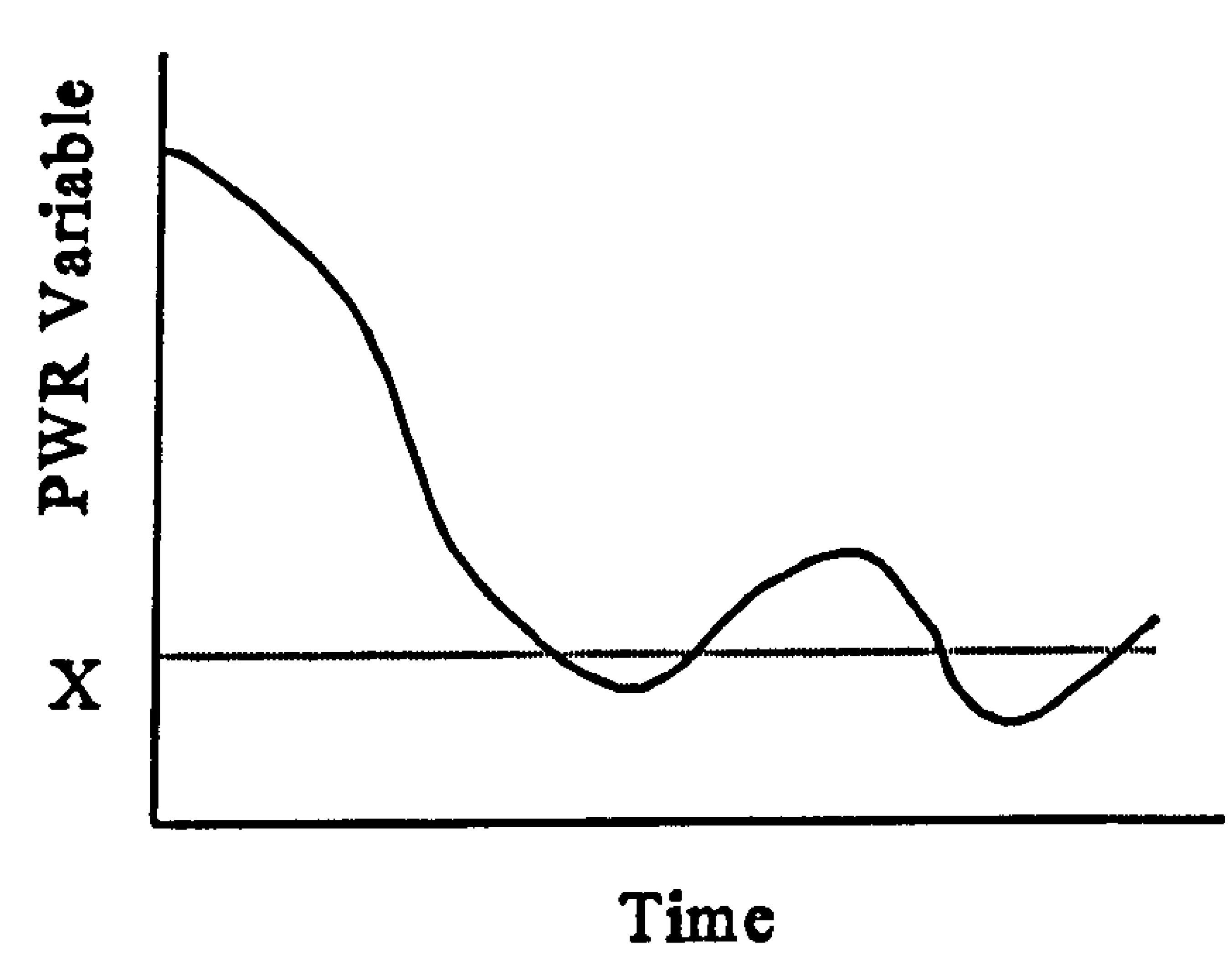


Fig 5.14: Sample PWR Variable during a Transient

Figure 5.15 shows such a system with the value of a single previous time step used as an input to the prediction system. Using a single time step to determine future condition does not permit the prediction system to clearly identify the section of the transient currently under consideration and so accurately predict the future values for the variable. Obviously the whole prediction system would examine more than one plant variable at a time but the implicit relationships established between the variables will still not guarantee the prediction of the correct sequence of values for the variable.

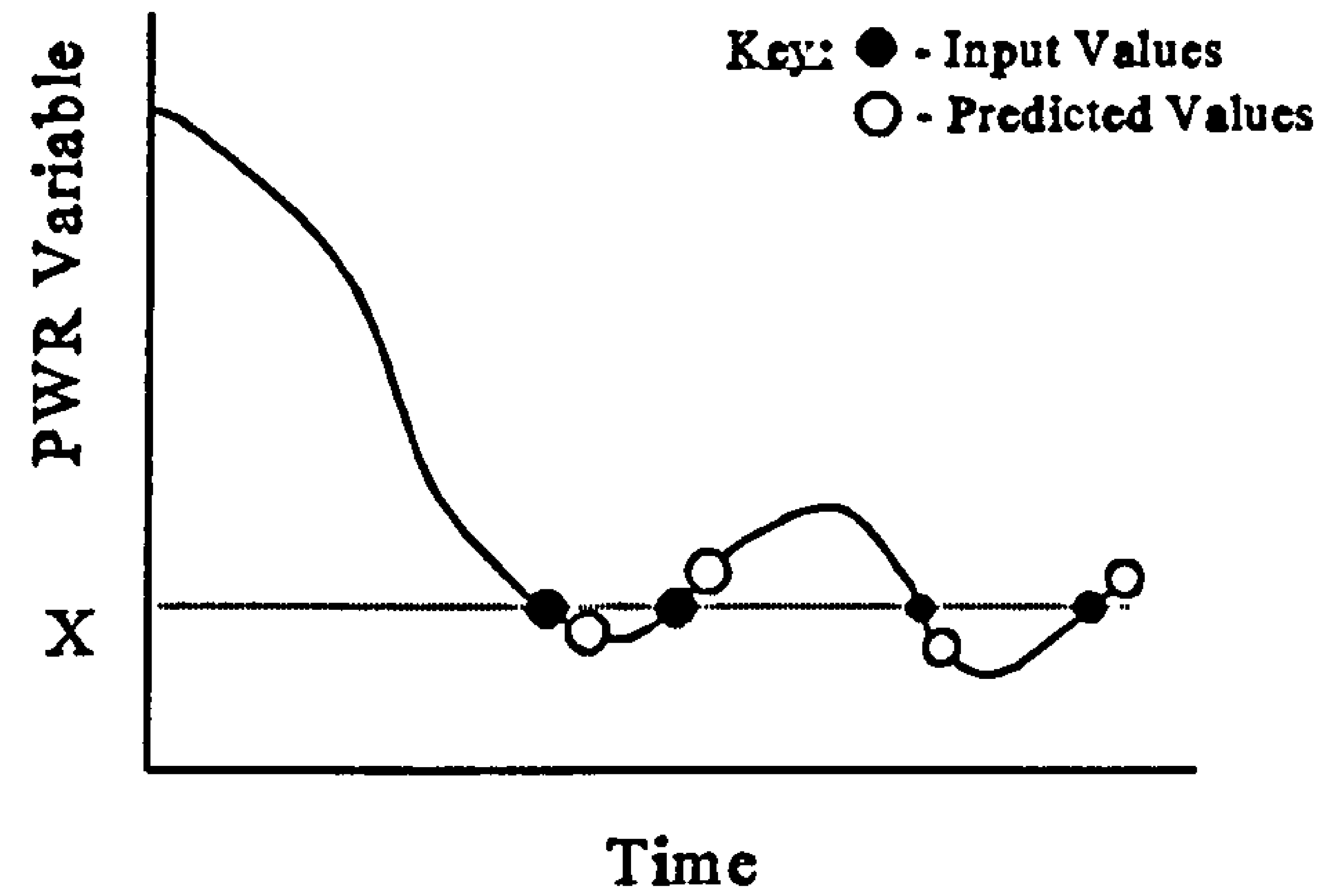


Fig 5.15: One Time Prediction for PWR Variable

Figure 5.16 depicts the same system but with a much larger number of time steps for the input to the prediction system. This size of the time step is such that most of the transient has occurred before the prediction system has acquired enough information to allow prediction of the remainder of the transient. This extreme is also undesirable as the operator gains no additional information from the advisor. The period when the prediction would be an asset to the operator is occupied by collecting data.

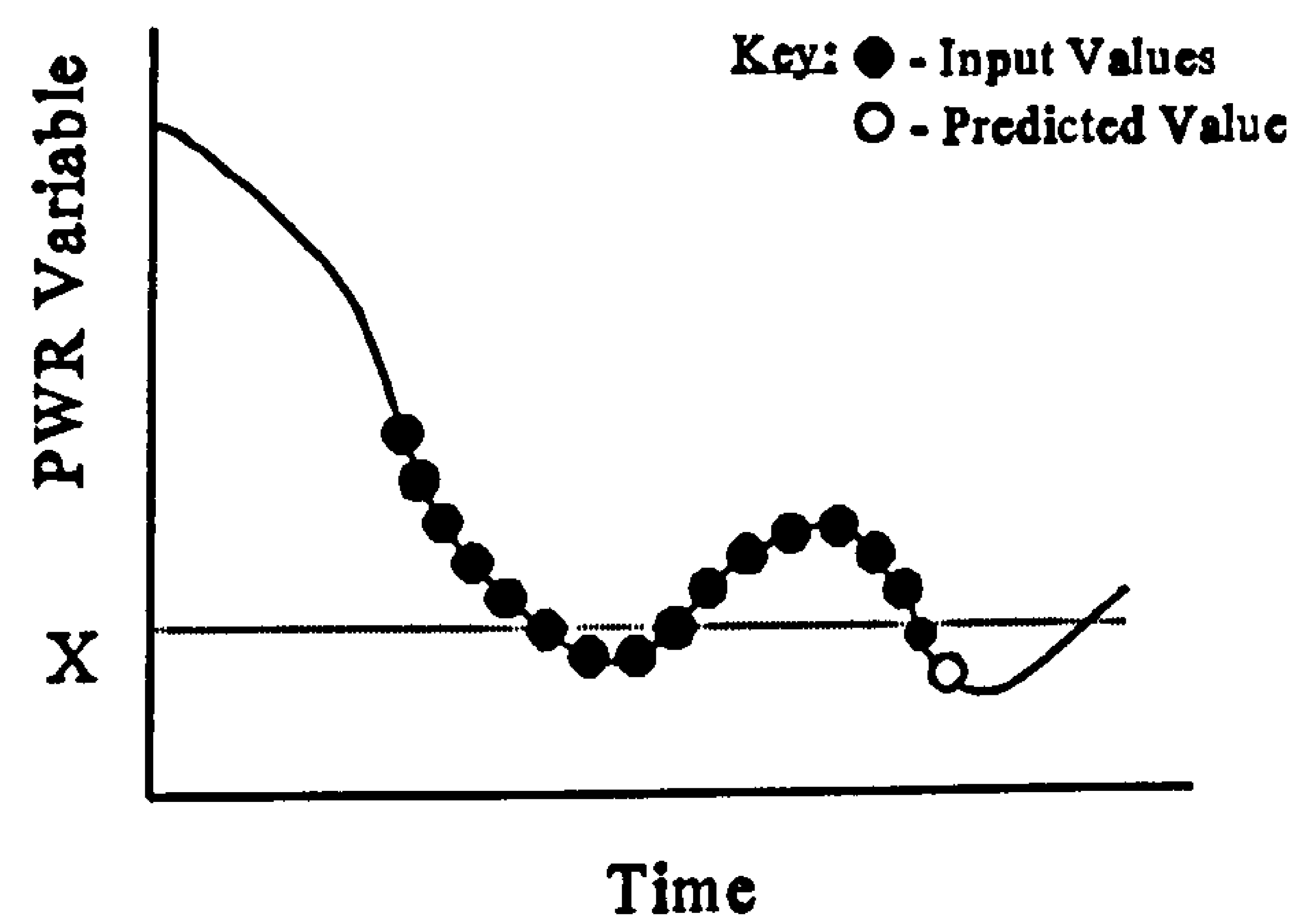


Fig 5.16: Multi-Step Prediction for PWR Variable

Clearly an optimum situation must exist between these two extremes. The ideal number of time steps for the advisory system must be short enough to enable a quick response to a transient but it should be of sufficient length for accurate prediction. This situation is depicted below, in Figure 5.17. The time step used for prediction is sufficiently long for the ANN to determine the gradient of the variable but still short enough for the predictor to be able to respond quickly to fault situations.

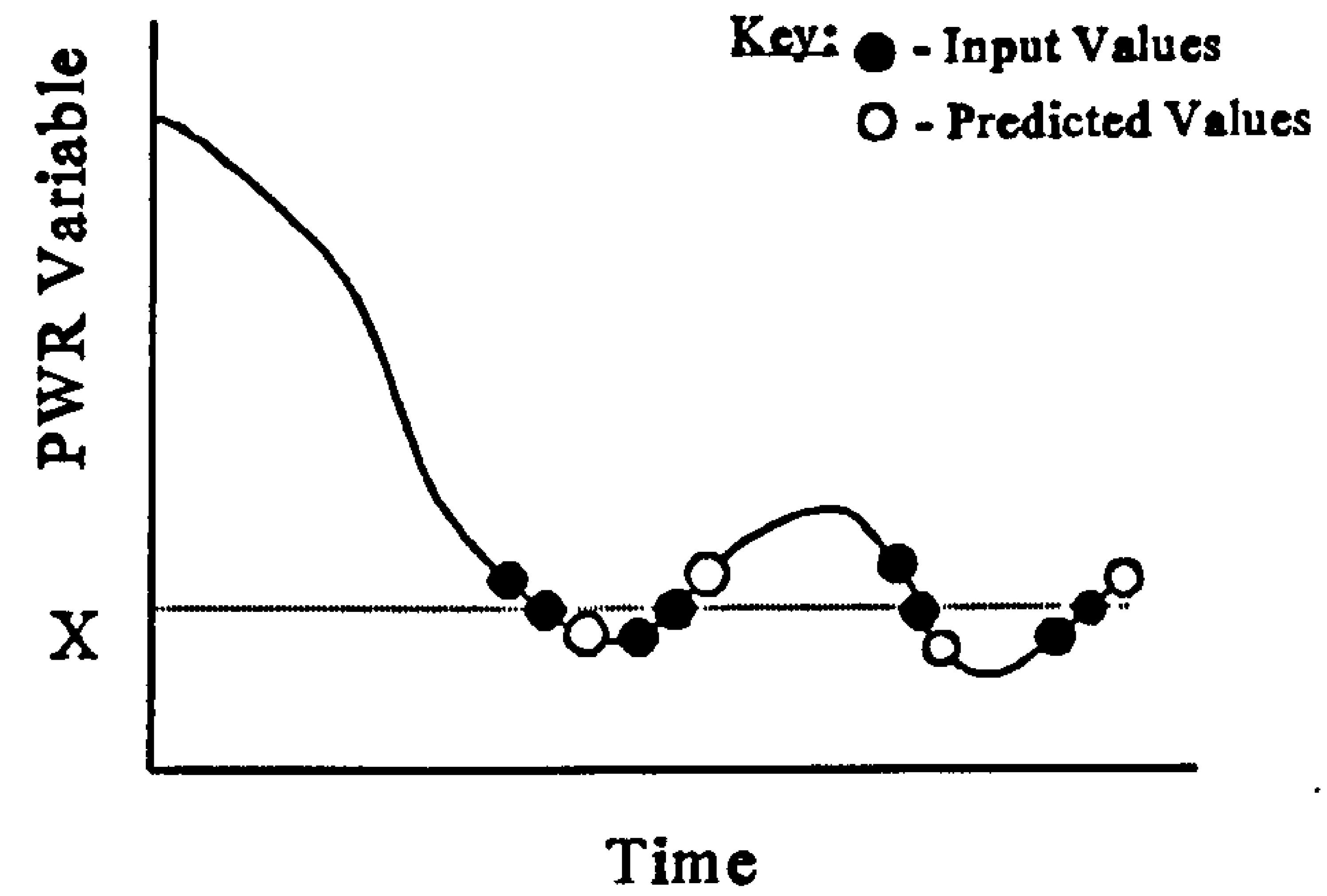


Fig 5.17: Optimum Time Step Prediction for PWR Variable

The prediction of future reactor variables can also be explained in terms of mapping the previous values to a future value. In Figure 5.15 above, using a single time step does not give a one to one mapping for the prediction. Figure 5.16 shows a one to one mapping but at the expense of response time. The current investigation can also be defined as optimising number of time steps required to produce a one to one mapping, if such a mapping should exist.

The results from the previous section show that for the PWR variables elected the number of time steps of information used for the training data does not have a large impact on the predictions from the ANNs. The accuracy of the predictions for each transient is not greatly effected by the number of time steps used. The choice of the optimum number of time steps to use for the remainder of the work on prediction is therefore unimportant in terms of accuracy of ANN output. In terms of the speed of response of the system a smaller number of time steps is preferable so a single time step would seem to be optimal. However, to allow for possible noise in the system two time steps will be used for the ANN inputs for the remainder of this work.

5.4 Continuous Prediction using Feedback of predicted values

The ability to accurately predict one time step into the future is a simple but useful feature for the advisor. An ANN trained on a suitable data set would be able to predict the state of the reactor for longer time steps, of the order of tens of minutes or hours after the diagnosis of the transient. This form of ANN could be used to provide the operator with a quick guide to the state of the plant during a particular transient. A tool could then be developed that would permit the operator to quickly investigate the outcome to a set of possible, what if, corrective actions.

If a set of one step predictive ANNs, each trained to predict different periods, were available the reactor condition could then be determined for any interim period. As an example, consider a set of five ANNs. Each ANN is developed to predict the PWR variable values for a fixed future time. Let these time steps be 1, 2, 4, 8 and 16 seconds. The ANN trained to predict plant variables 4 seconds ahead would therefore be trained to predict the PWR variables at time T given an input set composed of values at T-5 and T-4 using the two time step input developed in the previous section. Similarly the ANN predicting 16 secs ahead would use inputs from T-17 and T-16 to predict the values at T. Using combinations of these networks it is possible to predict reactor variables and therefore determine the plant condition for any period between one and thirty-one seconds. As shown below in Figure 5.18a, to predict the reactor condition twelve seconds ahead the current values for the variables would be input to the four seconds ANN the output from would then be fed into the eight seconds ANN the output of which would provide the required prediction. Similarly, in Figure 5.18b a prediction of variables twenty-seven seconds into the future is composed of the one second ANN followed by the two, eight and sixteen seconds ANNs outputs.

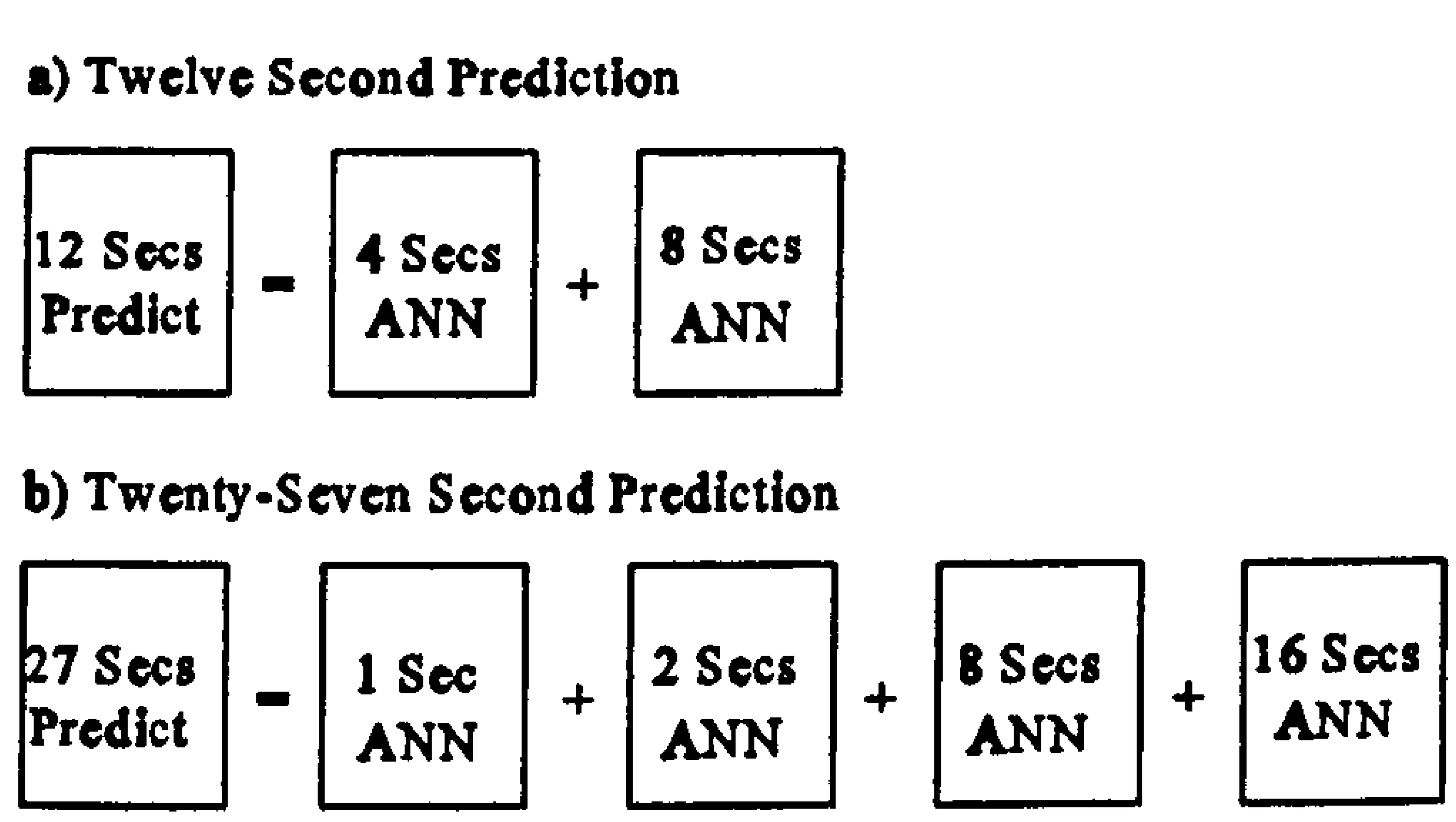


Fig 5.18: Prediction of PWR Variables using Single ANNs

In each of the above examples sufficient operations of the ANNs are required to produce the required number of ANN input sets. The order of the ANNs is not believed to be important. However the accuracy of each ANN is critical as error would be compounded over the number of presentations of the data to the systems.

Alternatively, a smaller number of ANNs could be developed and then used more than once to determine future variables values. Suppose for example that ANNs were developed to predict reactor variables one, five and ten time steps ahead. The first of the above cases could then be modelled by one iteration of the ten time step ANN followed by two iterations of the one time step ANN, as shown in Figure 5.19a. Similarly the second case could be represented by two iterations of the ten time step ANN, followed by one iteration of the five time step ANN and two iterations of the single time step ANN, Figure 5.19b.

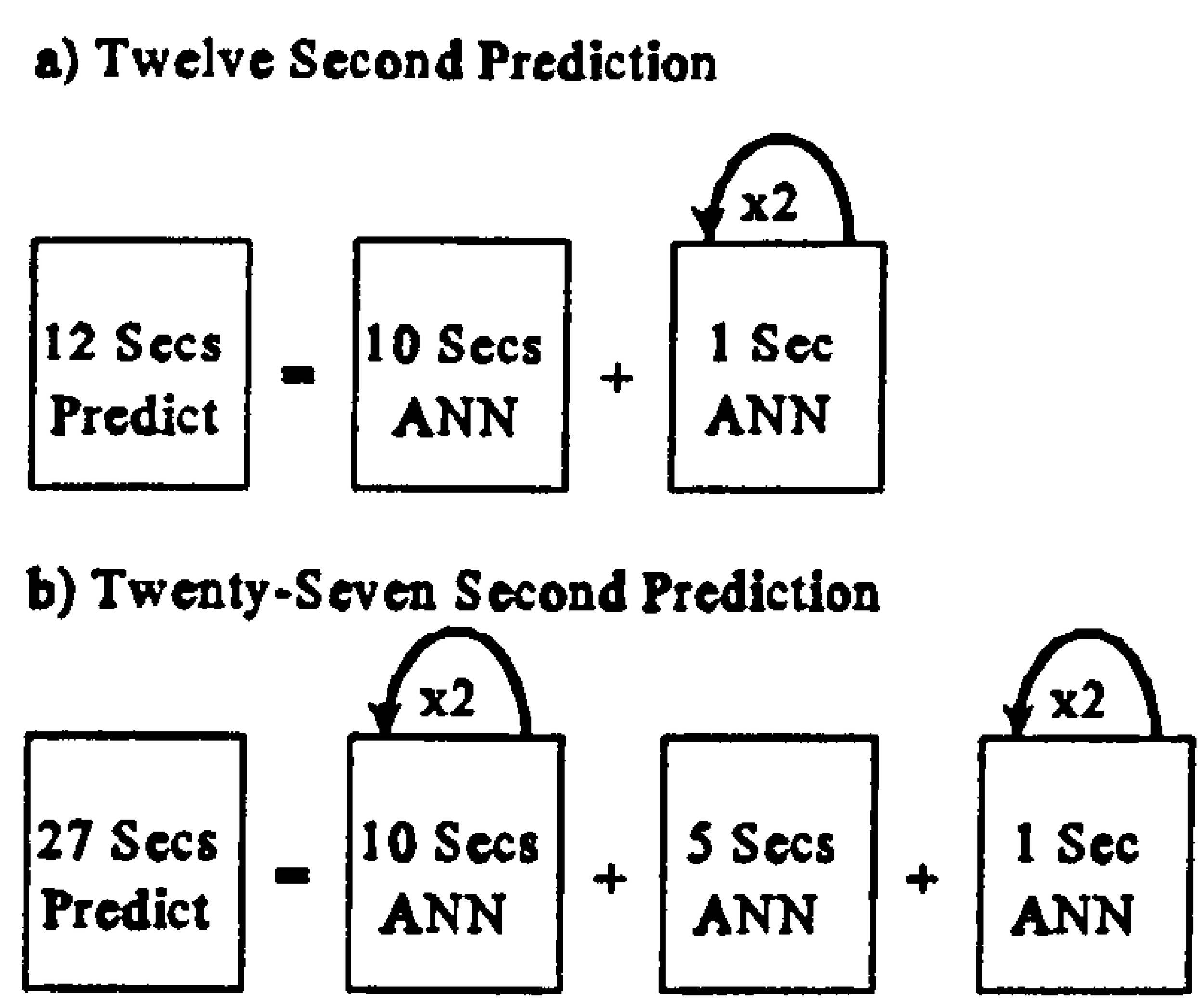


Fig 5.19: Prediction of PWR Variables using Repeated ANNs

Either of the above methods could be expanded to consider predictions for larger time steps. The only requirement for such a system is the construction of a minimum spanning set of acceptably trained ANNs. A potential disadvantage of these systems is the error accumulation of the multiple ANNs. The ANNs need to suffer graceful degradation for noise on the inputs, they must be robust for small variations on the input values.

The robustness of the ANNs can be explained using the following two curves as examples. They each represent a cross section through the input/output curve of two different trained ANNs. Each curve depicts the effect an error in the input value has on the corresponding output value. Both ANNs have converged to a minimum value in their respective solution space. However, the gradients in the area surrounding the minimum values is very different for the two examples. The first network, shown in Figure 5.20a, has a much lower gradient increase in the neighbourhood surrounding the minimum value than has the second example, Figure 5.20b. The error curves are represented as simple, symmetrical functions but in practice this need not be the situation.

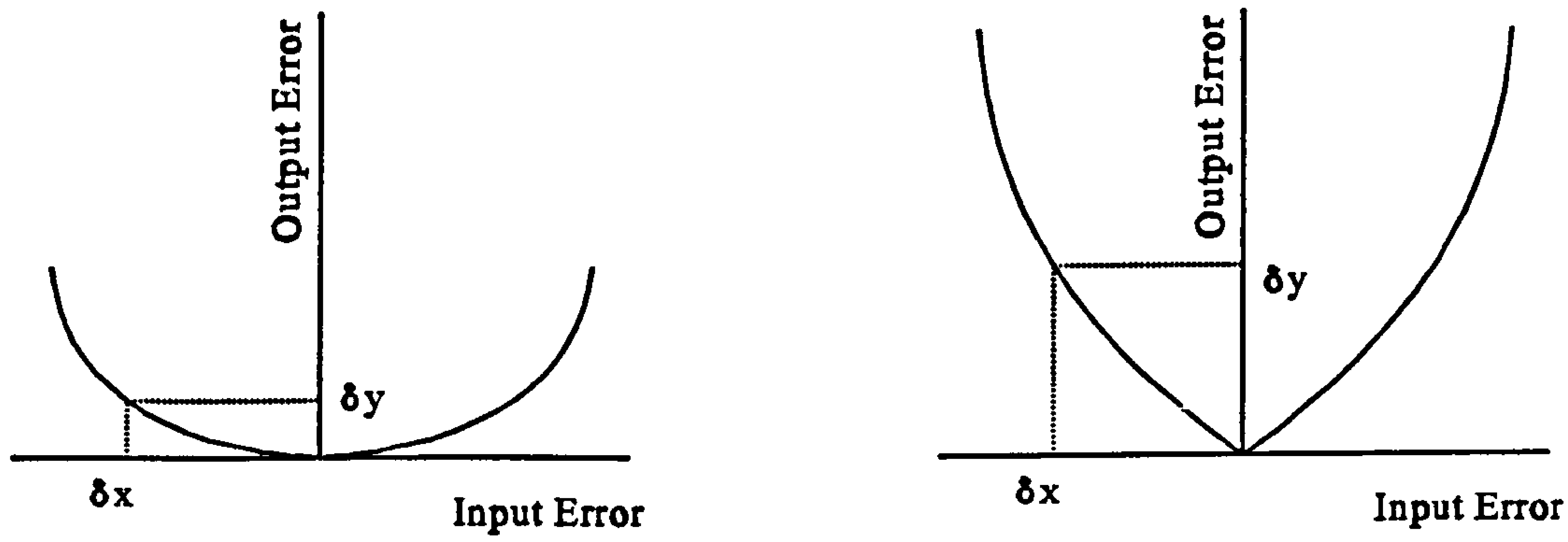


Fig 5.20a & b: Cross Sections of ANN outputs

If an input with the same error (δx) is now presented to both ANNs the resulting error in the outputs (δy) can be compared. The nature of the slope of each curve results in quite different outputs. The gradient of the first ANN's error curve, Figure 5.20a, is small so the ANN is quite tolerant of an error in input value. The second ANN's error curve, Figure 5.20b, has a much steeper gradient and so the same error in the input value produces a far larger error in the output.

The situation in the second case can be due to over-fitting. The ANN may have an excessively large number of nodes in the hidden layers. This excess of processing capability enables the ANN to learn insignificant features of the training set. These aspects are specific to the particular case being learnt not typical of the general population of cases. The network develops implicit relationships for both the unique and general features but with no method of distinguishing between the two categories. Such an ANN may well produce excellent results when evaluated with the training data but disappointing results when presented with inputs from a more general case. The problem can be resolved by using an appropriate number of hidden nodes, enough to enable the ANN to solve the problem but not too many for the general features to be lost amongst trivial and specific idiosyncrasies.

The initial idea of a feedback ANN is reasonably straightforward. An obvious extension of this approach is to use the same ANN to predict all future values. The value of PWR variables can be determined for any future period by using the same ANN for the required number of operations. This method only requires the development of a single ANN so is quicker to implement than the multi-ANN prediction systems previously discussed. The problem of error build up, introduced above, becomes increasingly acute in the feedback situation so the ANN developed is required to be very accurate.

A set of investigations were performed to explore the applicability of using feedback for the ANN prediction of variables during a PWR transient. The best two time step ANN developed for each scenario in the previous section was embedded in a computer program and then used to predict the full transient. The first two sets of variables as the initial inputs and thereafter each predicted set of values used for the most recent input set. Appendix H contains a sample program listing of the best predictive ANN for Transient 3. The results of the feedback program for this transient are given below.

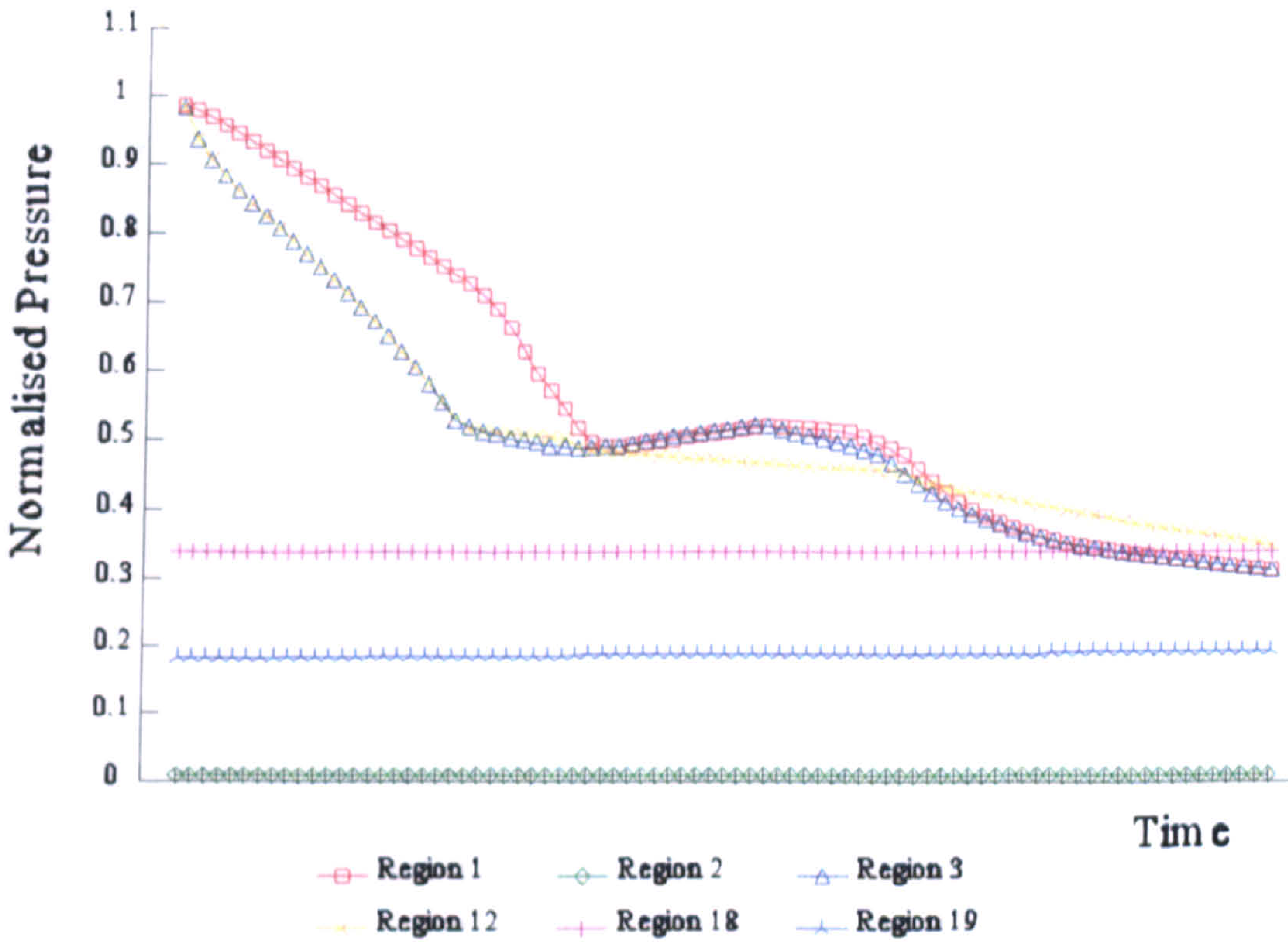


Fig 5.21: Results of Feedback ANN for Transient 3

The results show that the transients are generally well predicted using the feedback programs

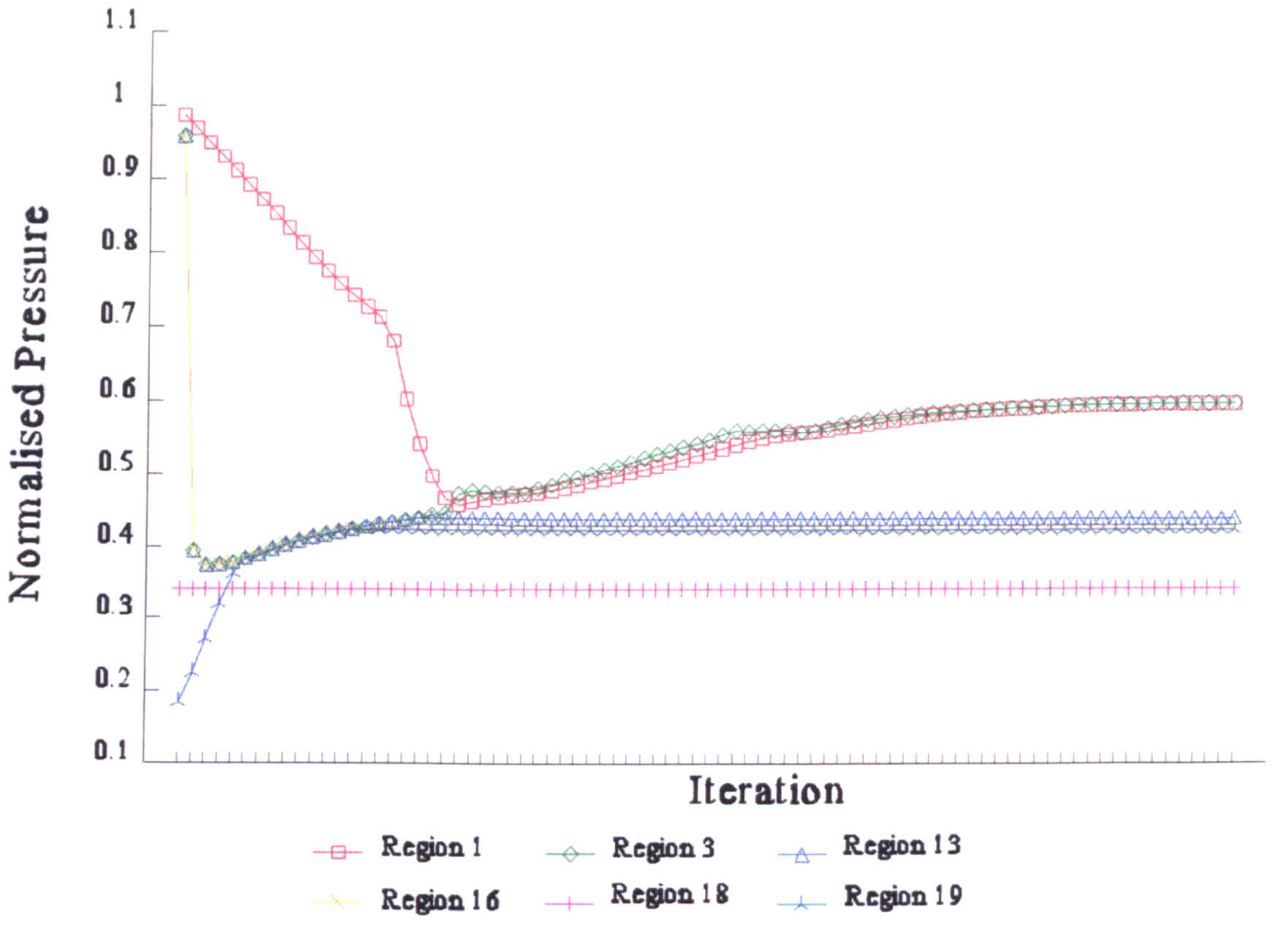


Fig 5.22: Pressure Outputs for Transient A

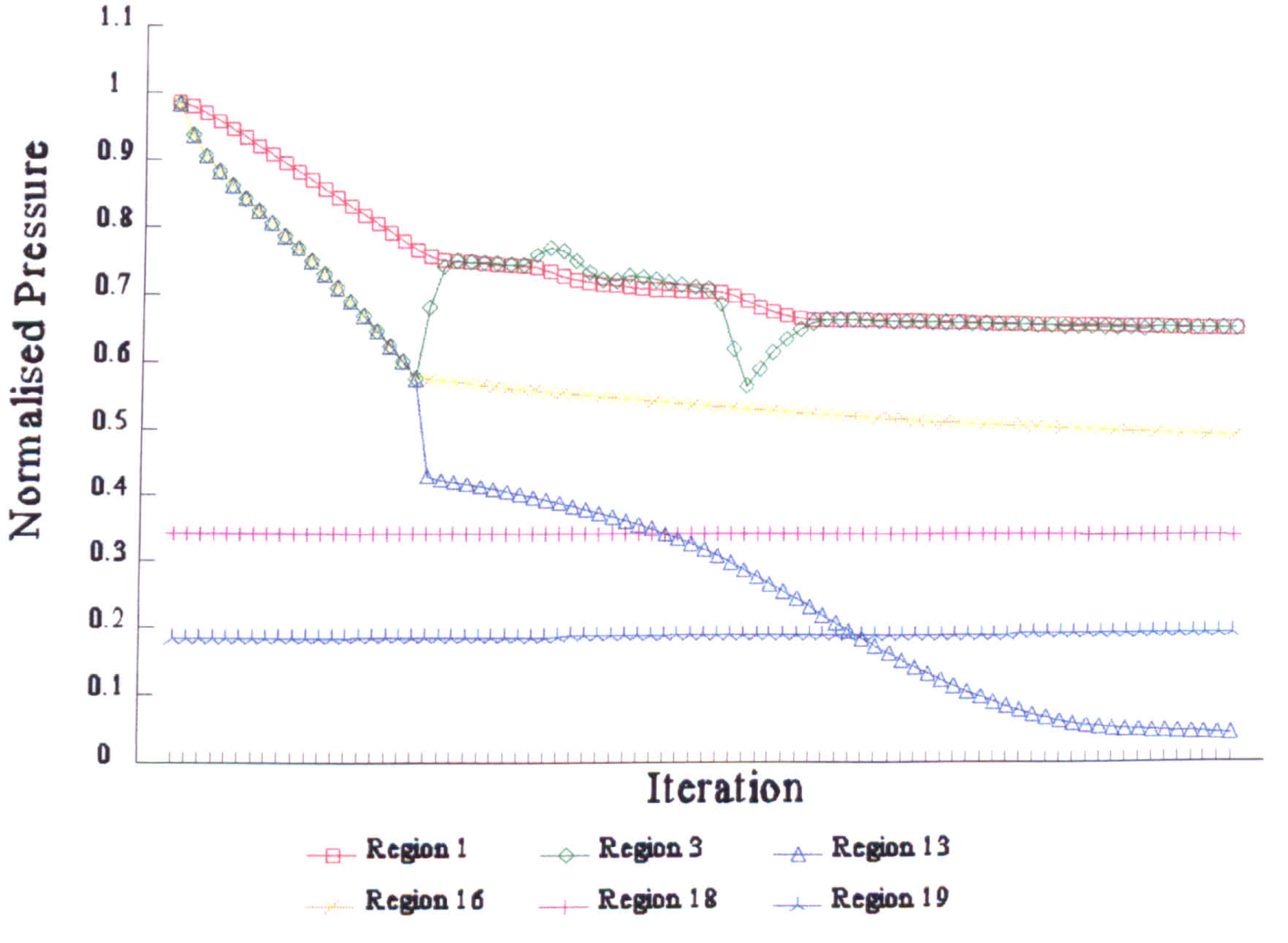


Fig 5.23: Pressure Outputs for Transient B

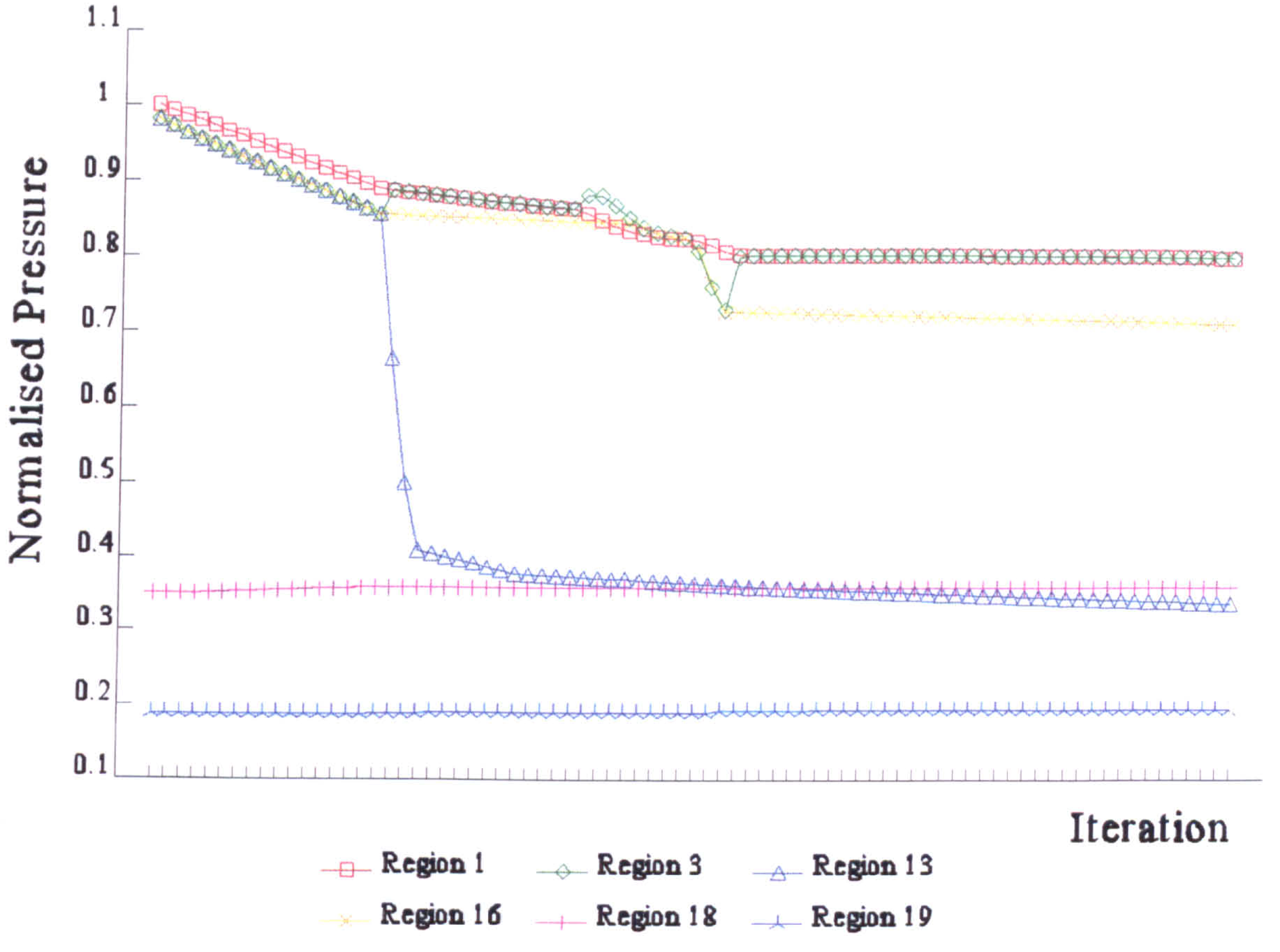


Fig 5.24: Pressure Outputs for Transient C

The training and test sets for each transient were constructed using the two time steps result from the previous work. In addition extra inputs were included to simulate the result from the diagnostic level of the advisors hierarchy. Initially this was a single input, set to a value "1" for transient A, "0" for B and "-1" for C. However, it was decided that, in spite of varying this order, this approach assigned a form of ranking to the transients which did not exist (Masters, 1993, p278). The training and test sets were each modified to include an extra three binary inputs, one for each transient. This approach was felt to simulate the result of the diagnostic level more accurately while allowing for two, or more, transients to occur at the same time. The final details of the ANNs were 17 inputs, 7 outputs.

Four sets of training and tests data were produced, one for each pair of combinations from A, B and C, and a set with all three transient data. The data sets were constructed by concatenating the required individual sets which each contained 80 vectors. Again the ratio of training to test sets was approximately in the ratio of 2:1. A range of neural networks were trained for each combination of transients. The best RMS errors obtained for each association are given in the following table. The full results are given in Appendix H.

Transient Combination	ANN Structure	Best RMS Error
A and B	20	0.0243
A and C	10	0.0228
B and C	10	0.0528
A, B and C	20	0.0229

Table 5.9: Best Results for Training Combined ANNs

The following six graphs, Figures 5.25 to 5.30, show the output from each neural network when the information for each training transient is input. Although the points are connected for clarity, no concept of time has been included in the development of the networks. The outputs are merely the predicted value given two successive values for the plant variables.

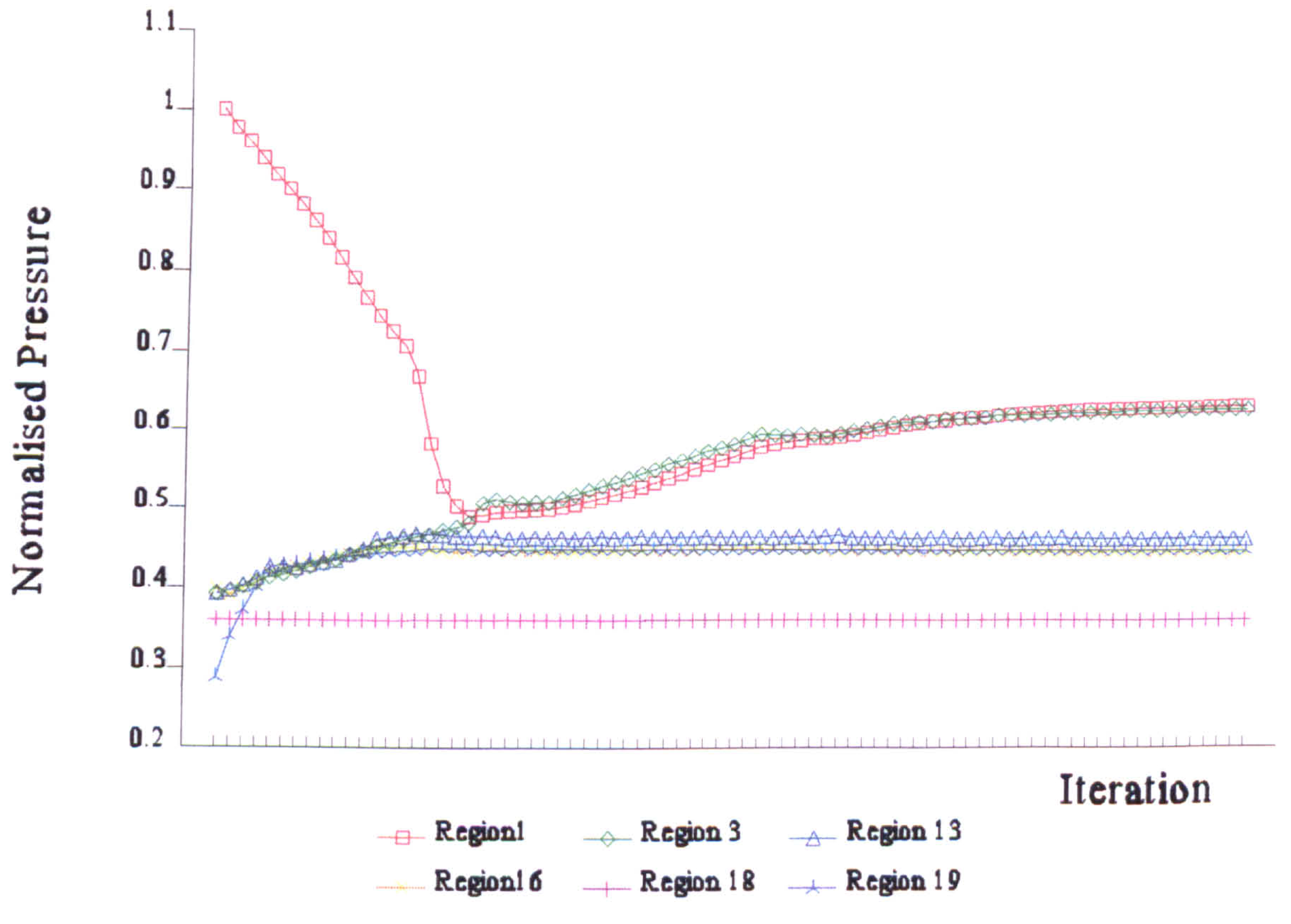


Fig 5.25: Transient A Predicted from A & B ANN

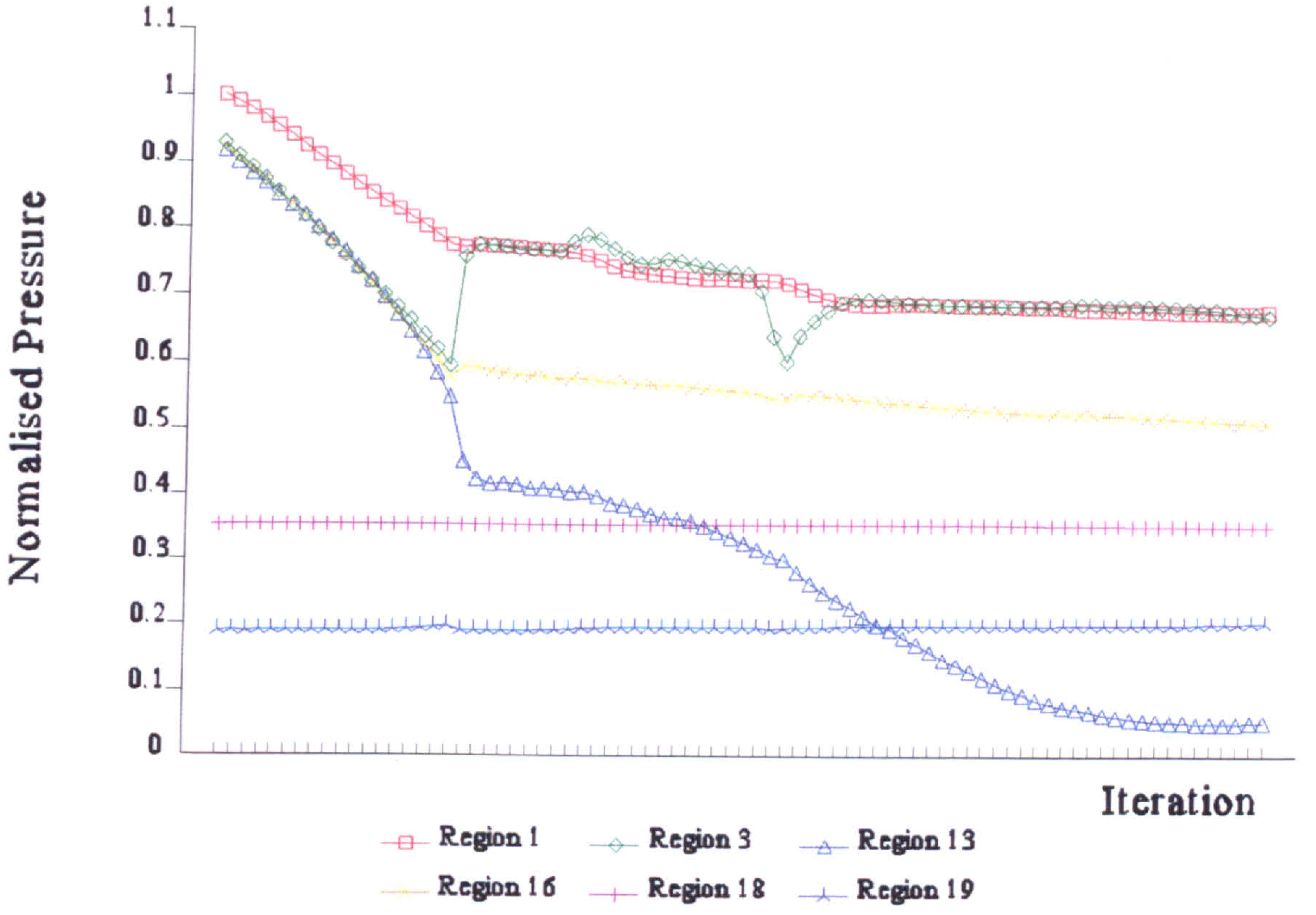


Fig 5.26: Transient B Predicted from A & B ANN

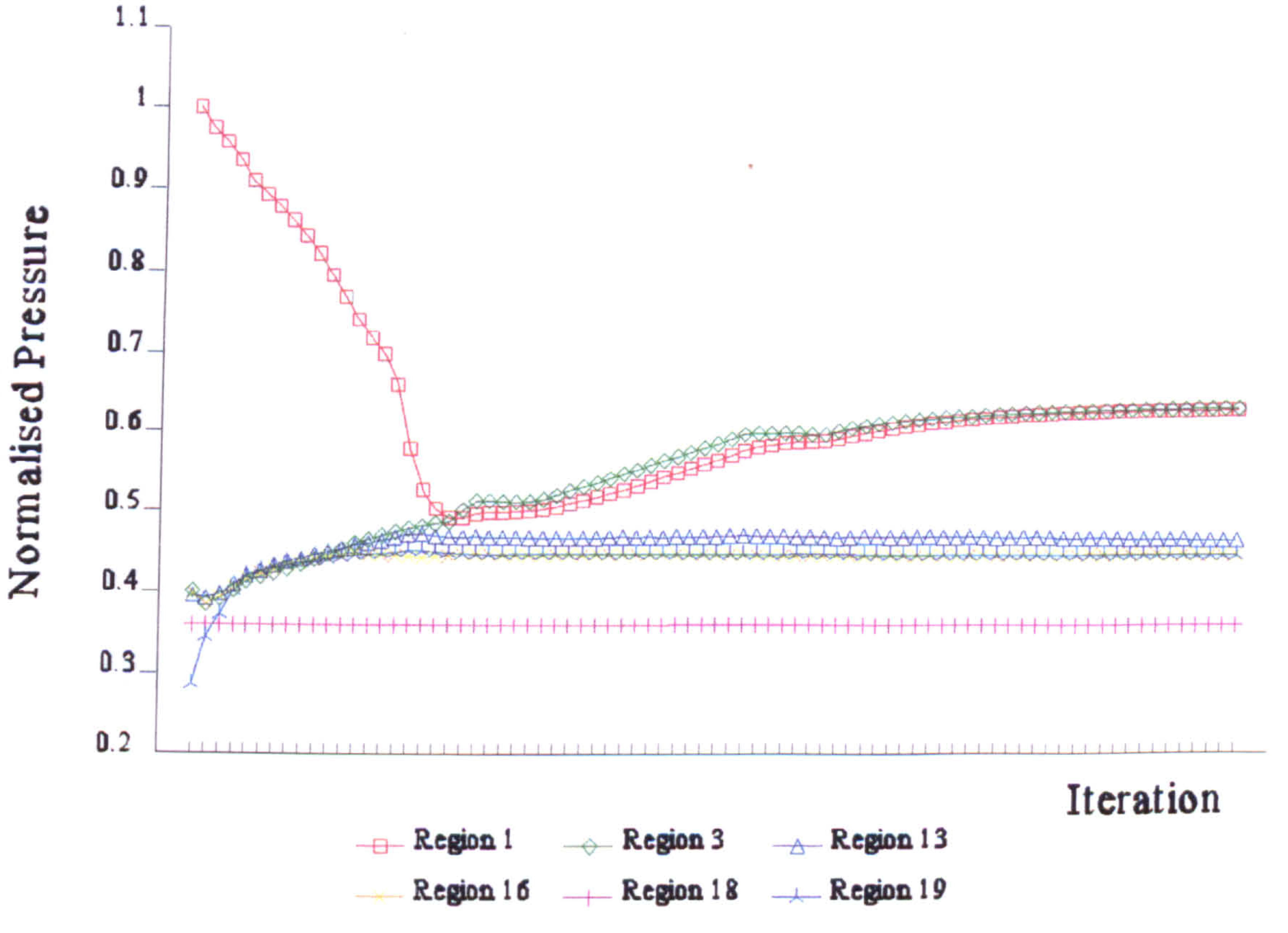


Fig 5.27: Transient A Predicted from A & C ANN

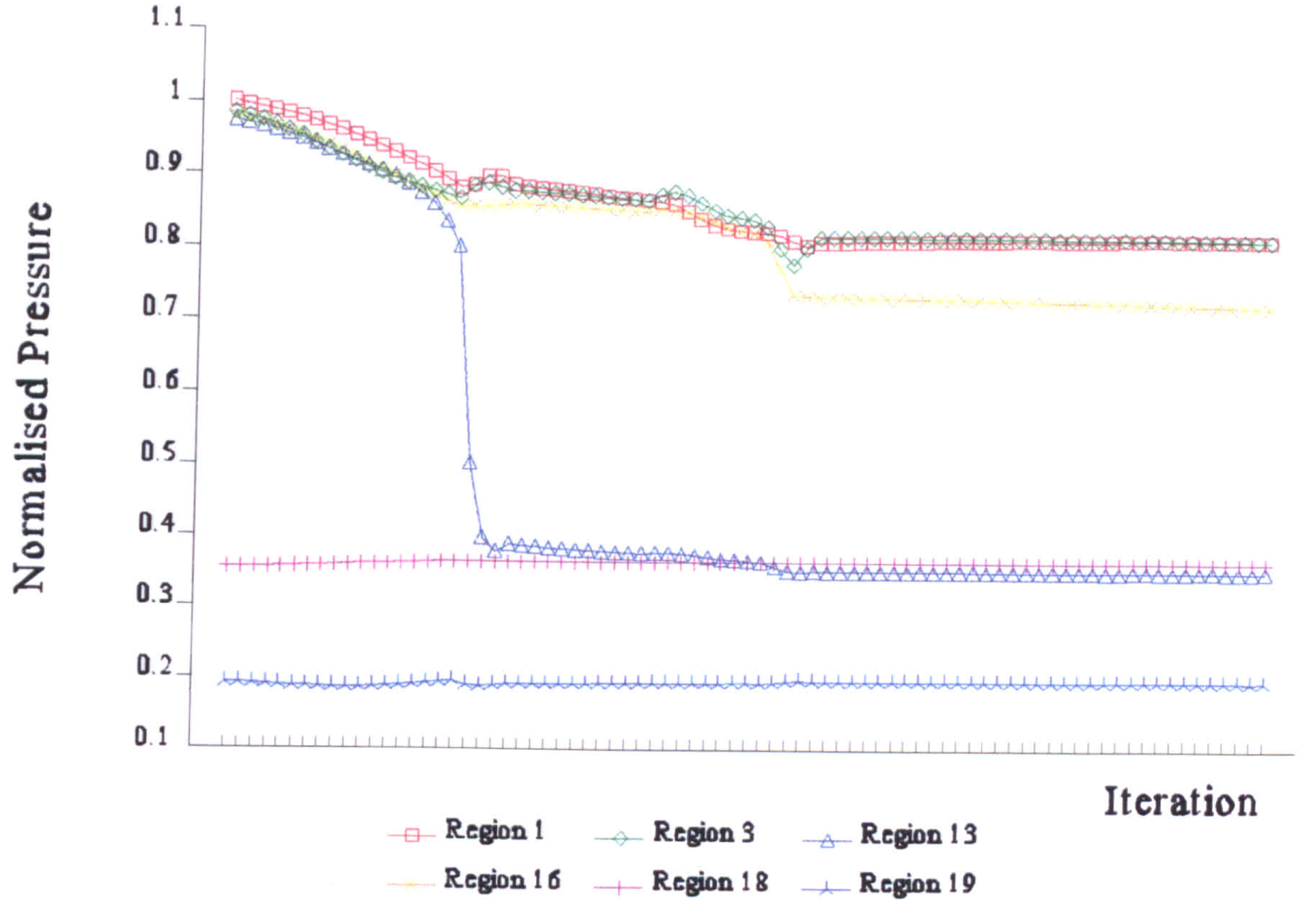


Fig 5.28: Transient C Predicted from A & C ANN

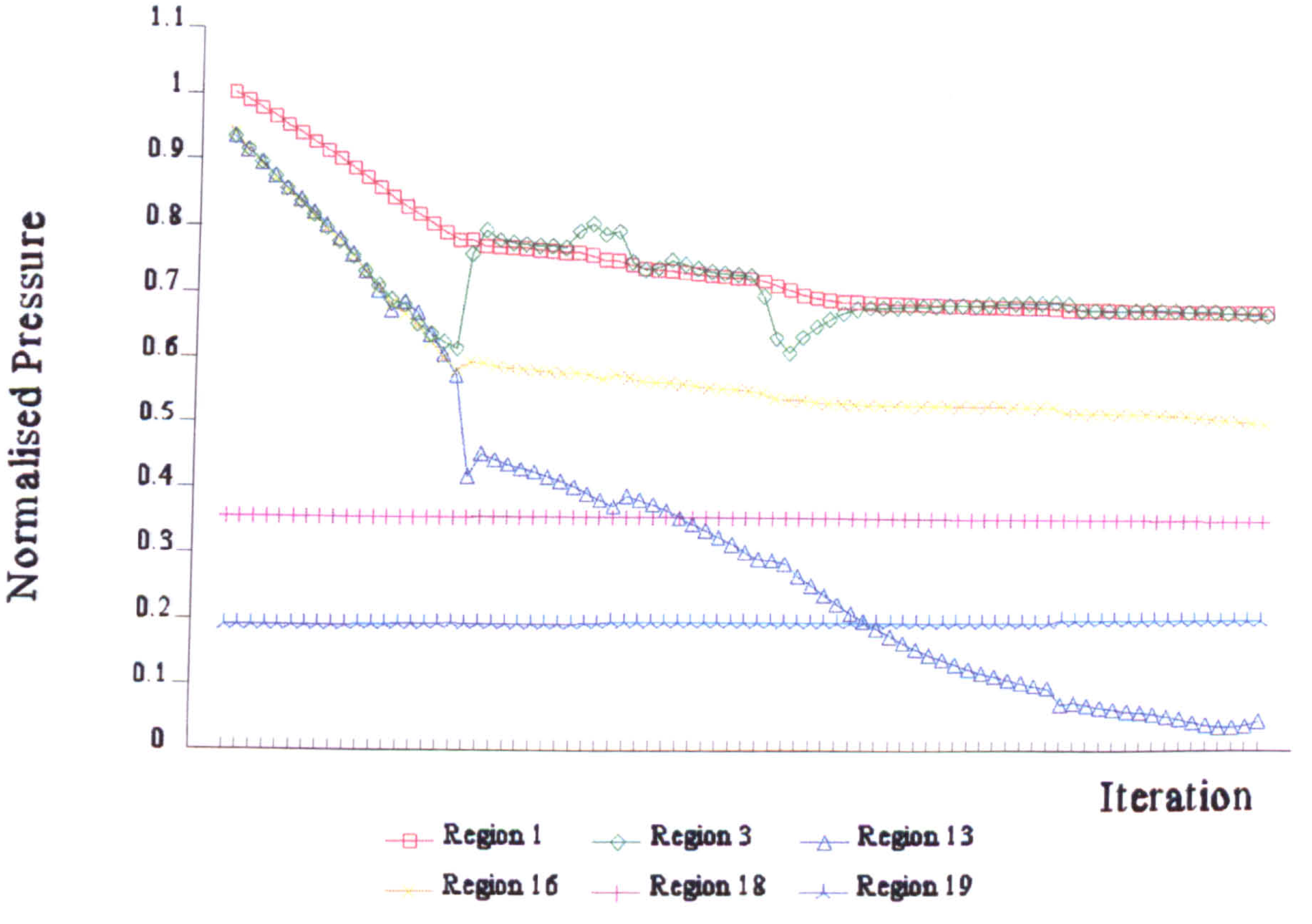


Fig 5.29: Transient B Predicted from B & C ANN

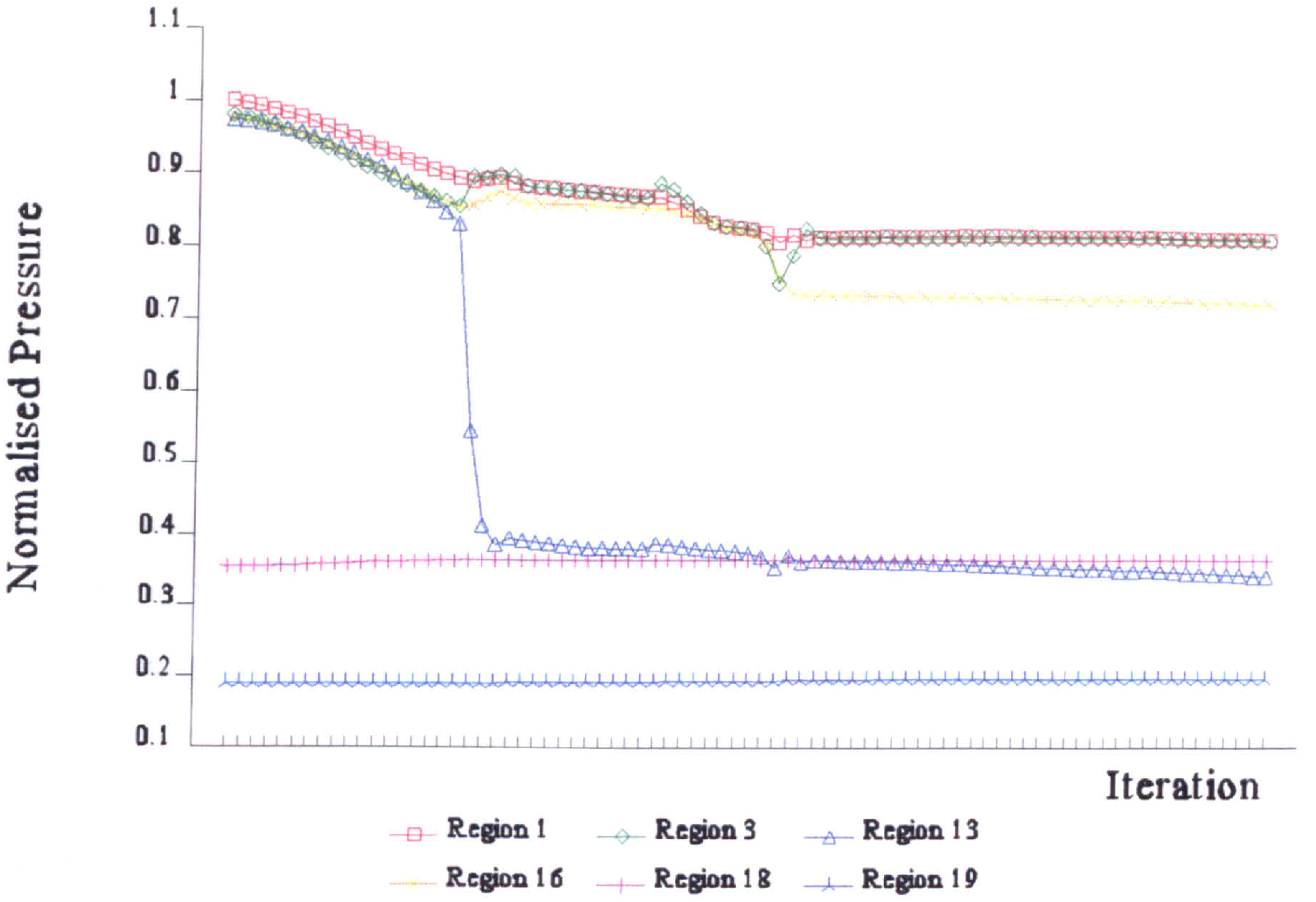


Fig 5.30: Transient C Predicted from B & C ANN

The results show that all combinations of transients are well predicted by ANNs using a single step operation, ie. no feedback. The network developed for predicting dissimilar transients, A with C and A with B, have a lower RMS error value than the ANN developed for predicting similar transients, B with C. This difference may be explained by the similarity in the shape of the pressures curves for the transients. The graphs for transients B and C, Figs 5.23 and 5.24, show a similar pattern for the pressures in nodes 1, 3 and 16. The graphs have very similar curves for the latter stages of the respective transients. The shape of the plots for pressures 1 and 3 show comparable gradients and temporal features in both graphs. The pressures for transient A, Fig 5.22, behave in a very different fashion throughout the duration of the transient, notably the pressure in node 19.

While each neural network is being trained it is attempting to provide a mapping between the inputs and outputs of the training and test sets. Similar transients exasperate the situation as the mapping is more complicated due to the data sets containing alike information.

The results from the ANNs trained to predict all three transients are even more remarkable. The best ANN has an RMS error lower than that of the two transient ANNs. While this difference is not large for the dis-similar transient ANNs (RMS error of 0.0229 compared to RMS errors of 0.0243 and 0.0228) it is still a surprising result. The following three graphs, Figures 5.31 to 5.33, show the prediction of each transient variables using this ANN. A possible reason for the results may be that the training data set is larger for this investigation than the previous cases. The training sets were formed by concatenating the individual transient data for each required transient. The latter training set therefore comprised of three sets of information compared to two for the earlier cases. This larger training set may have enabled the prediction solution space to be better mapped during training compared to the smaller training sets. Another possible explanation is that all inputs for the latter ANN consisted of non-zero numbers. As discussed earlier the transient being predicted is identified by the inclusion of three extra binary inputs which were set to 1 for the transient being predicted.

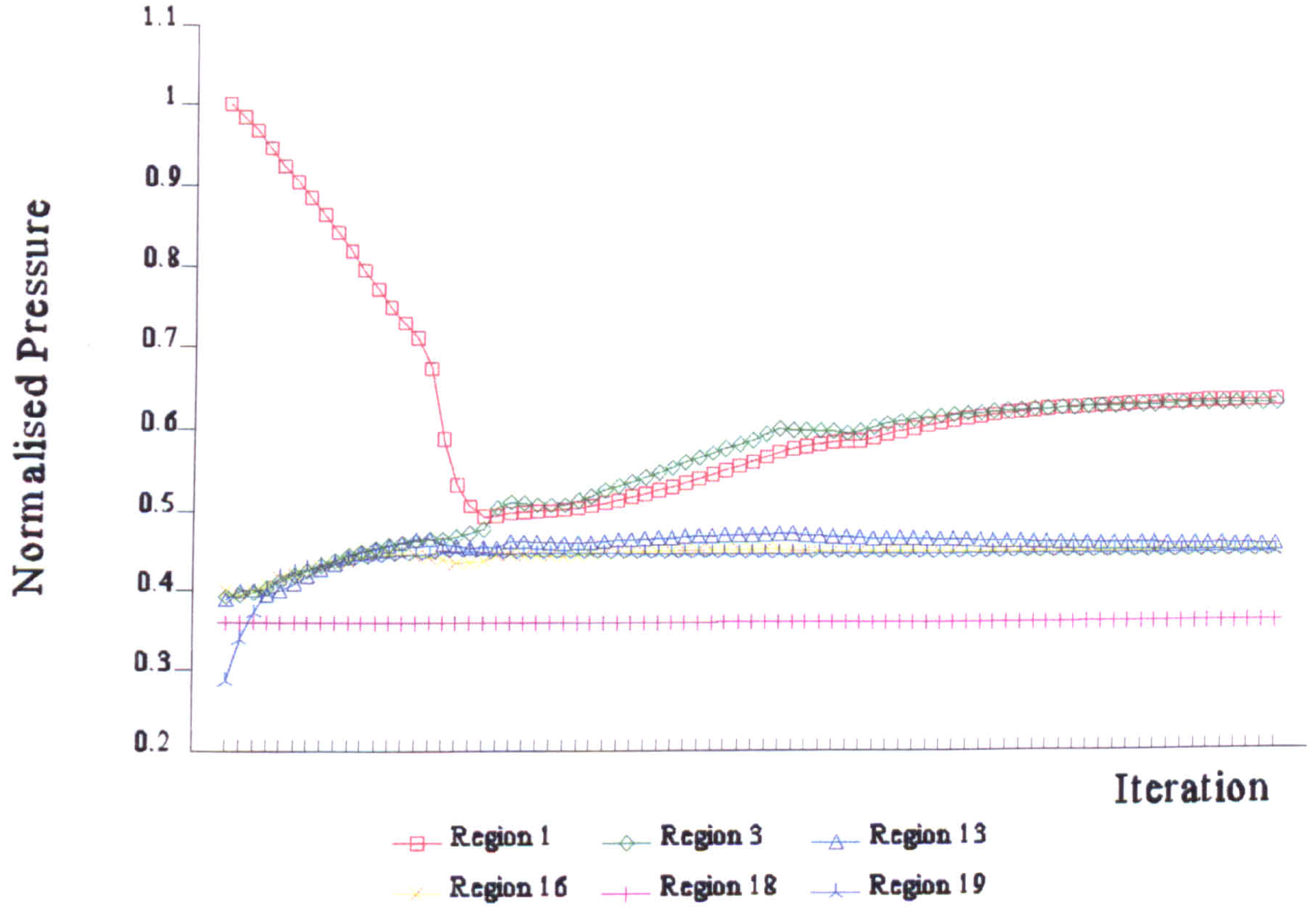


Fig 5.31: Transient A Predicted from A, B & C ANN

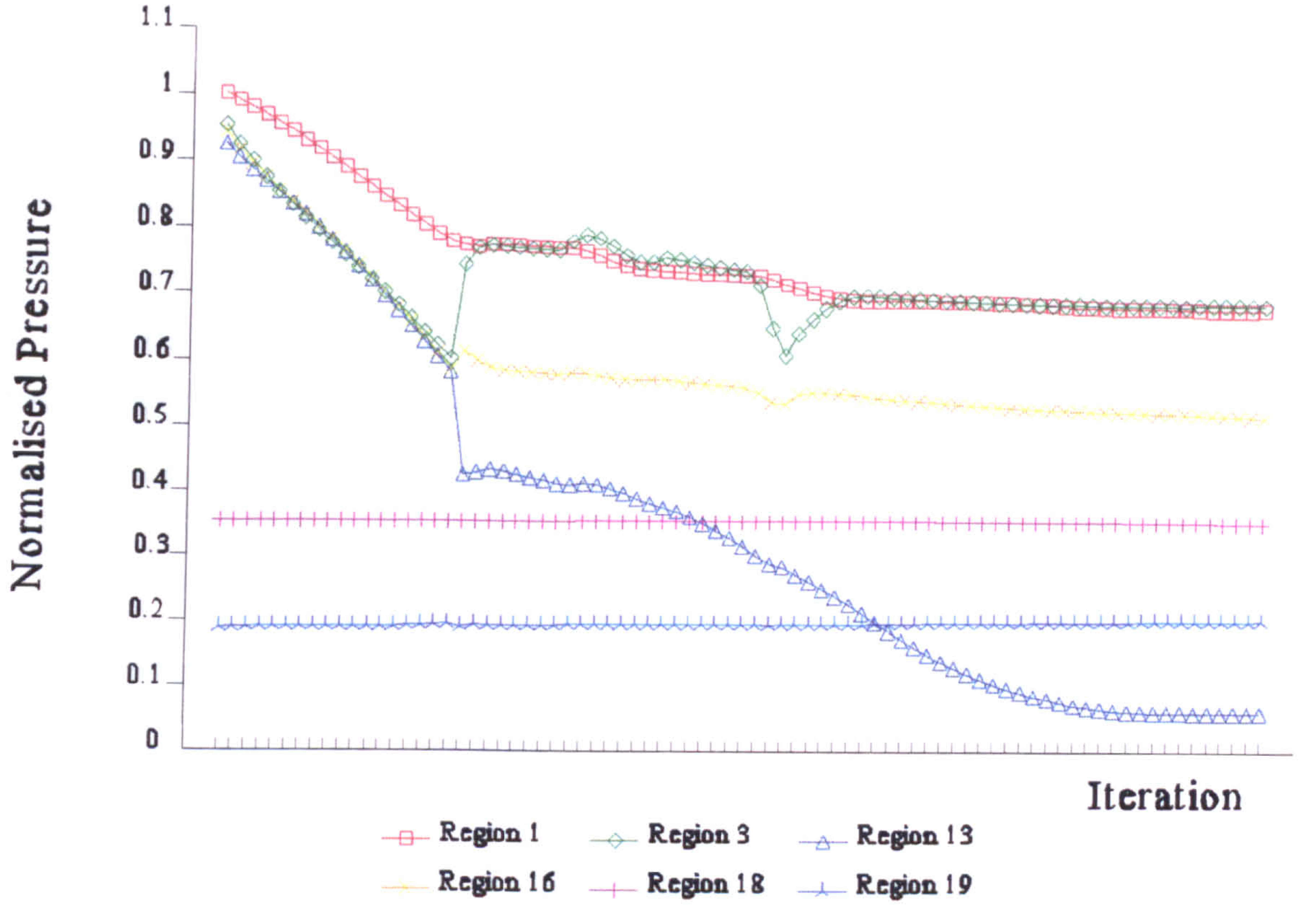


Fig 5.32: Transient B Predicted from A, B & C ANN

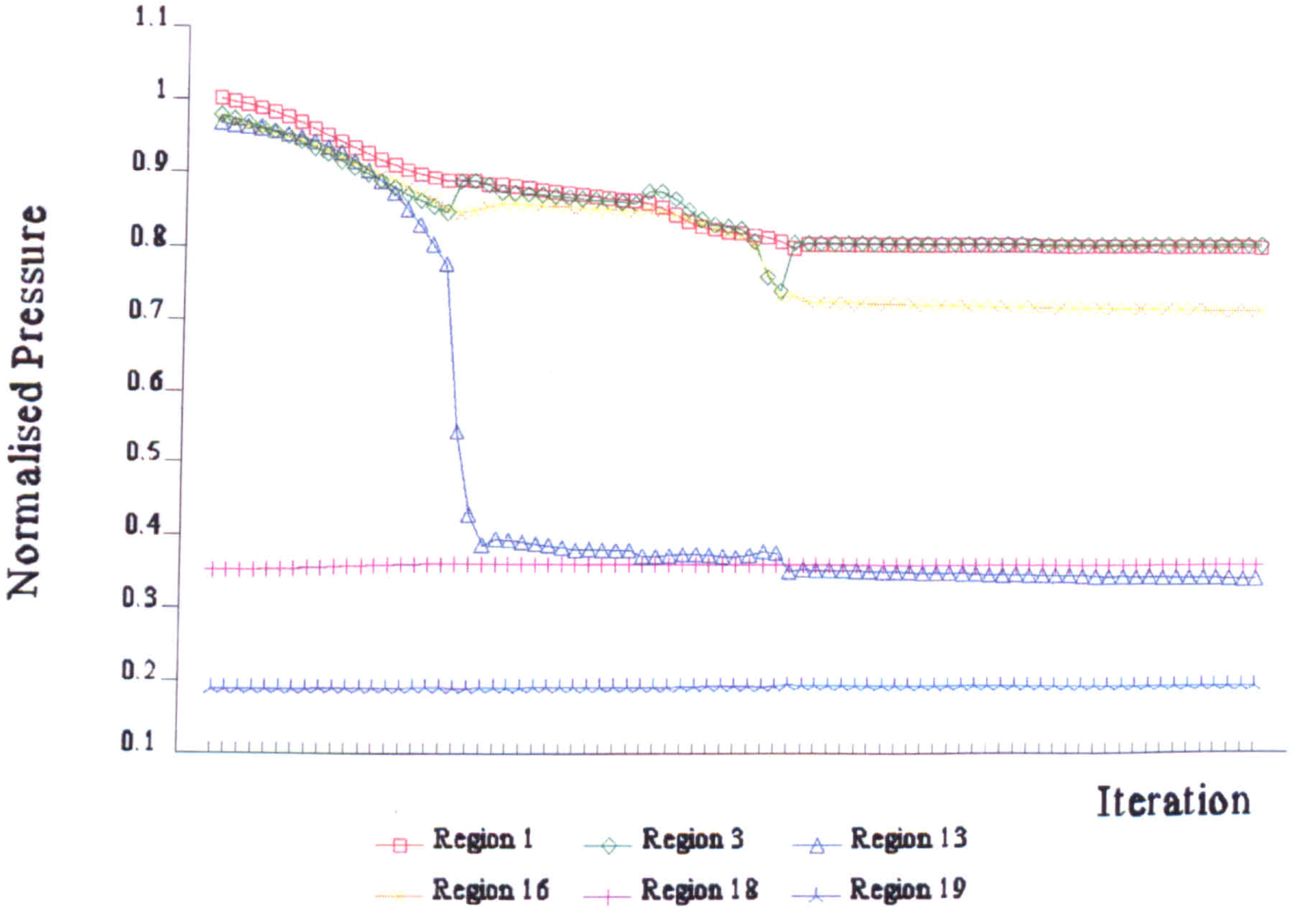


Fig 5.33: Transient C Predicted from A, B & C ANN

Transient	Nodes in Hidden Layer	RMS Error
1	6	0.0286
2	5	0.0093
3	5	0.0130
4	5	0.0191
5	6	0.0286

Table 5.11: Results of Individual Transient ANNs

A program was produced for the best ANN of each transient. The programs featured a feed back of the pressures in nodes 1 and 3. Graphs of the outputs of each program are given below.

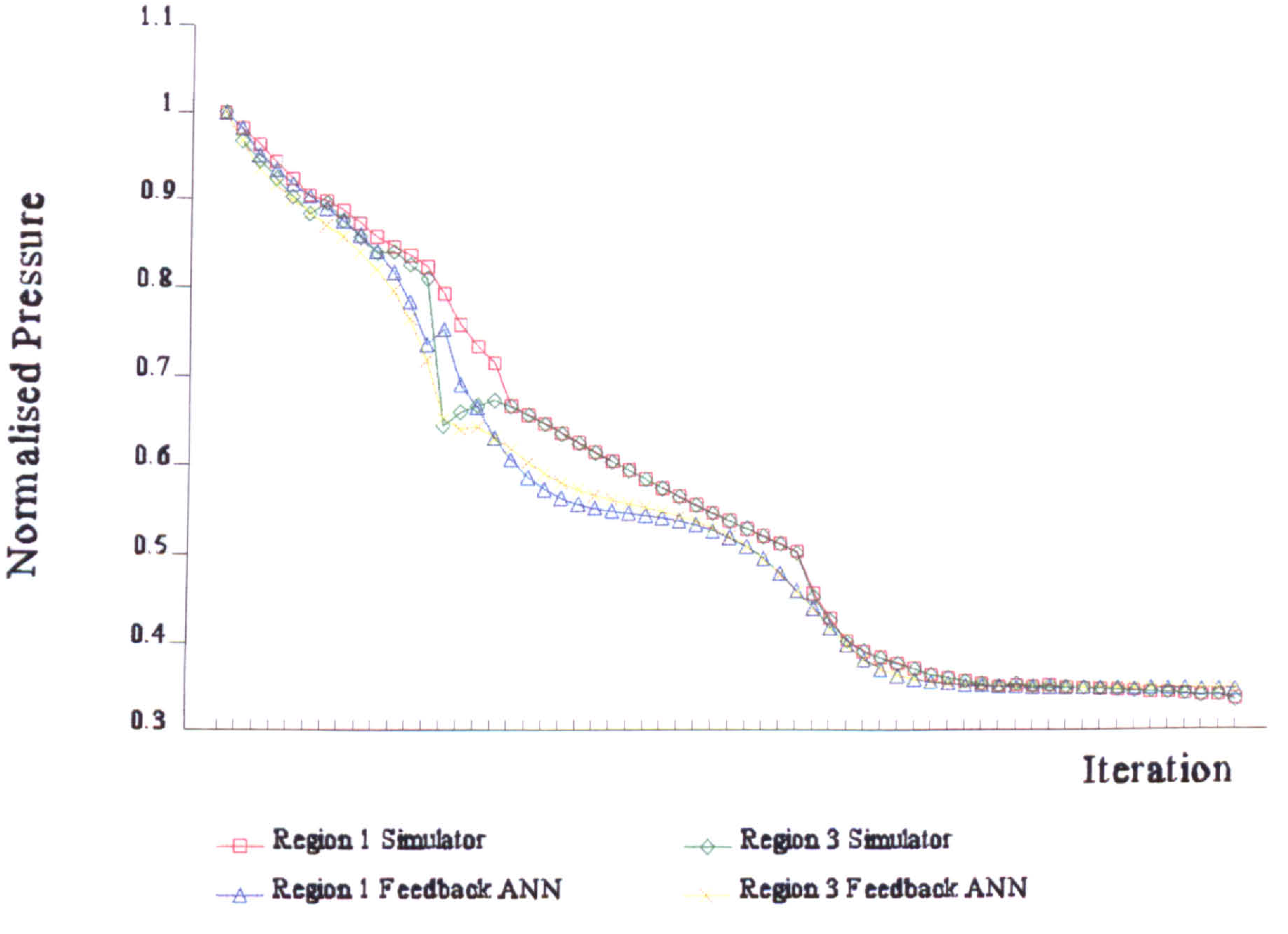


Fig 5.34: Feedback ANN and simulator Results for Transient 1

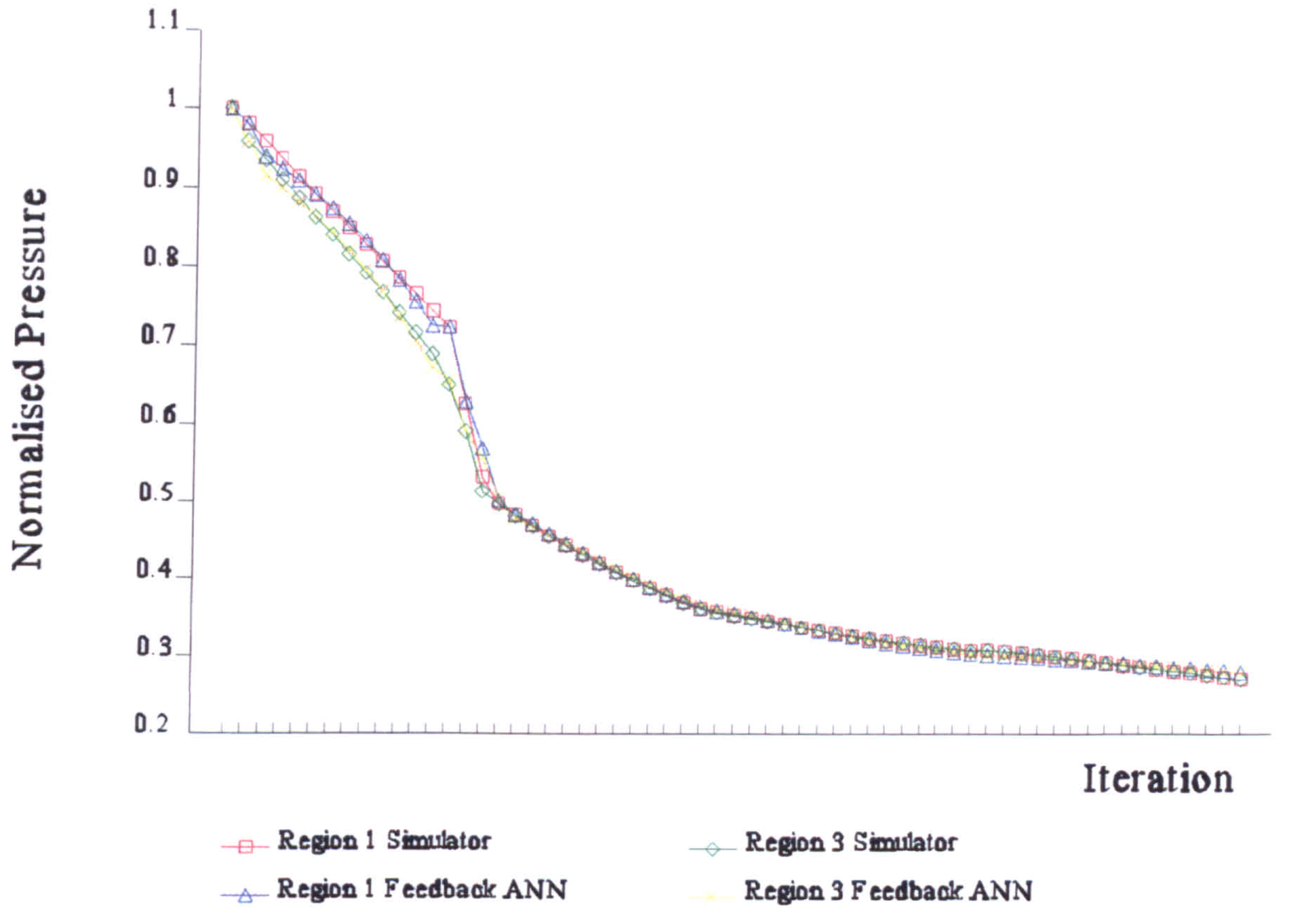


Fig 5.35: Feedback ANN and Simulator Results for Transient 2

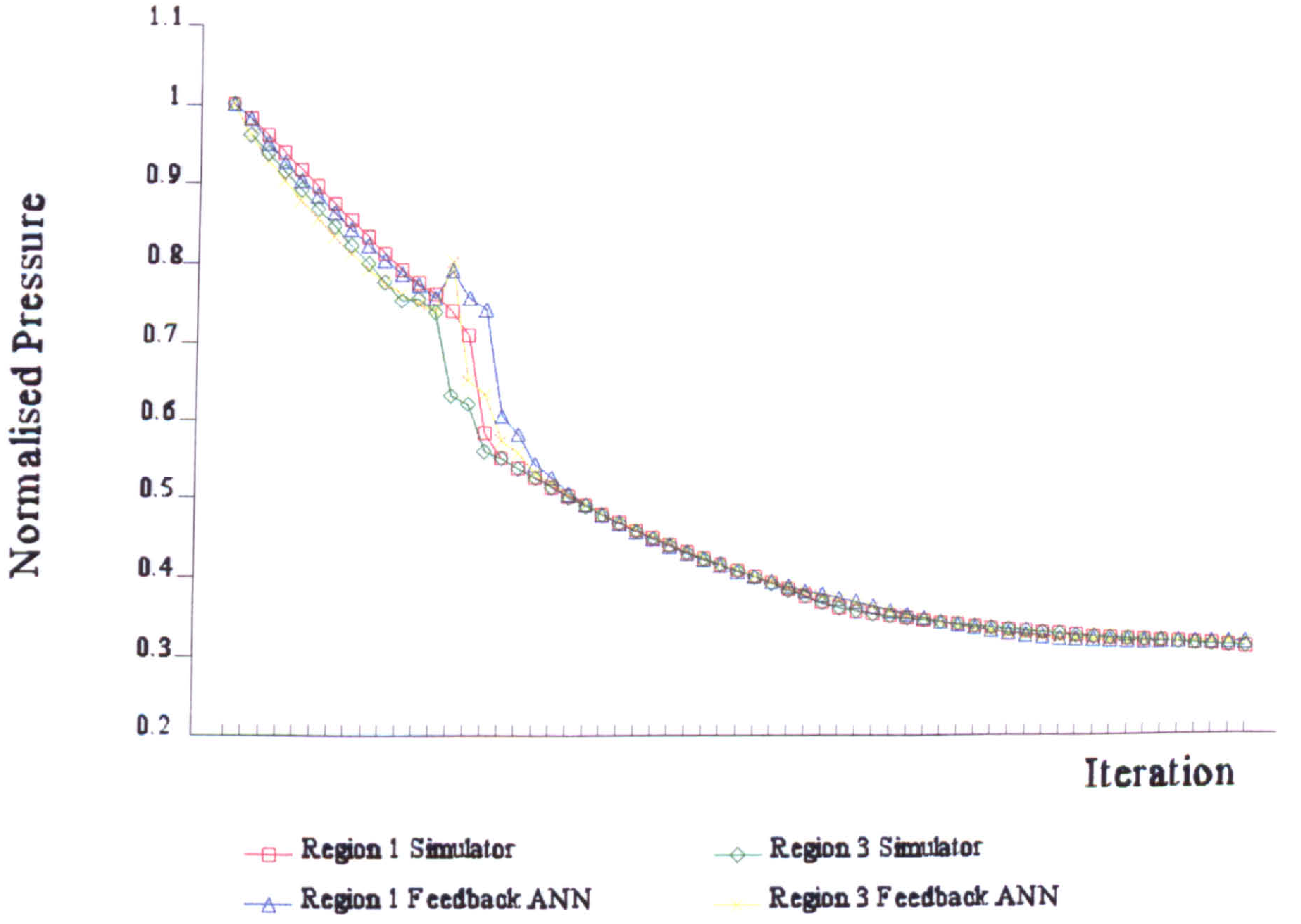


Fig 5.36: Feedback ANN and Simulator Results for Transient 3

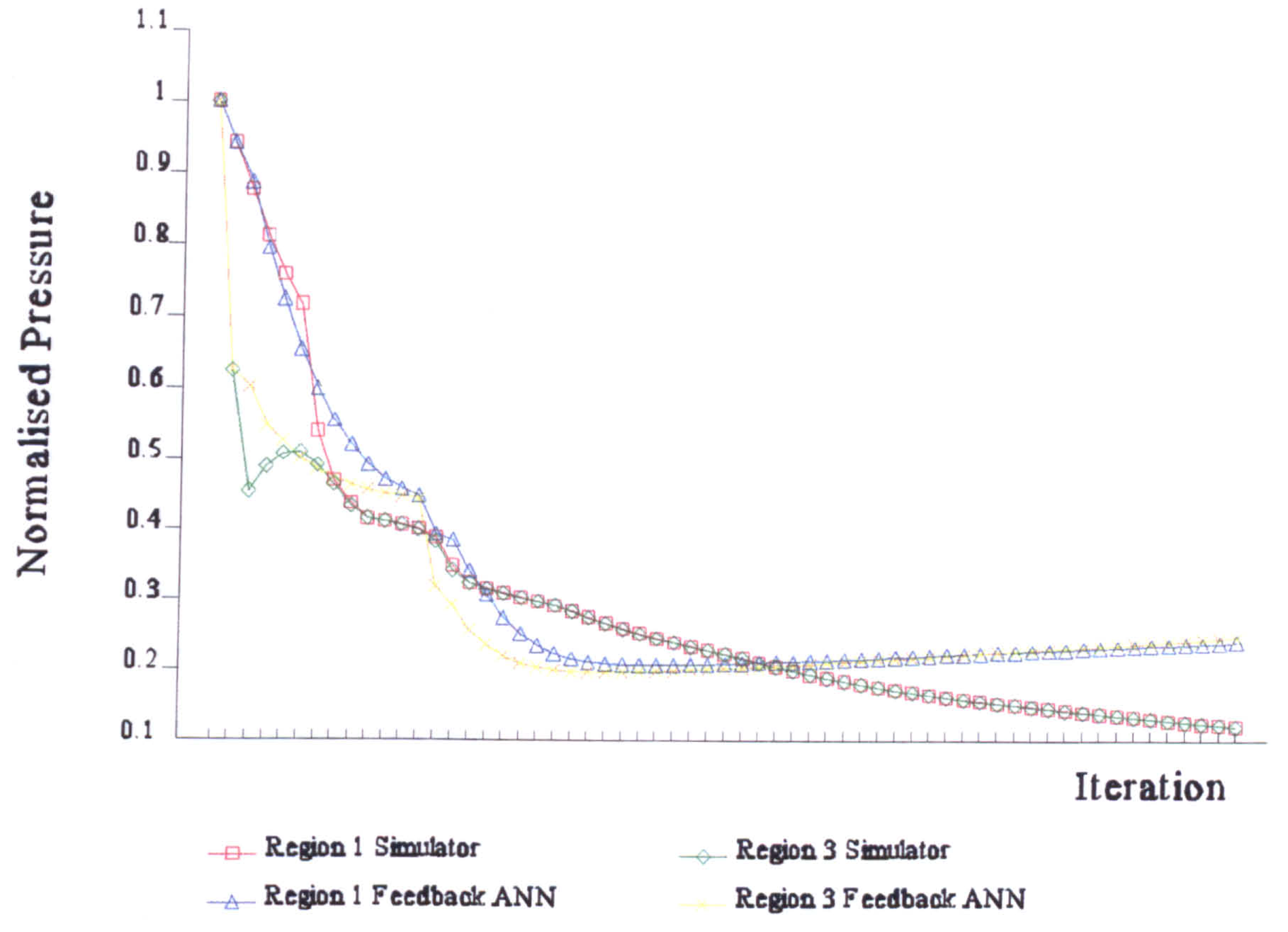


Fig 5.37: Feedback ANN and Simulator Results for Transient 4

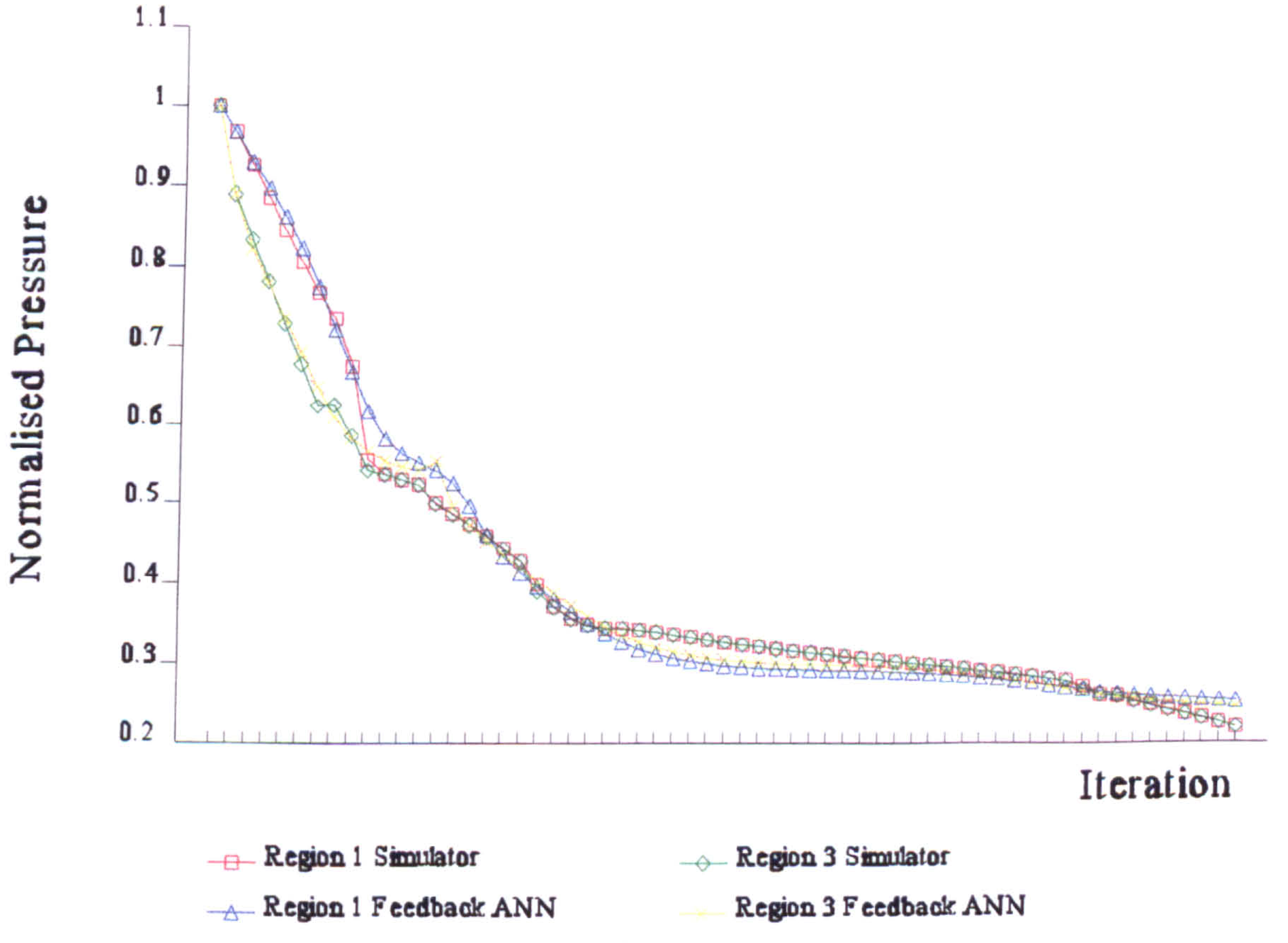


Fig 5.38: Feedback ANN and Simulator Results for Transient 5

was used to predict each of the five transients. The results of the predictions are given below, in Figures 5.39 to 5.43. The original simulation output is included for comparison.

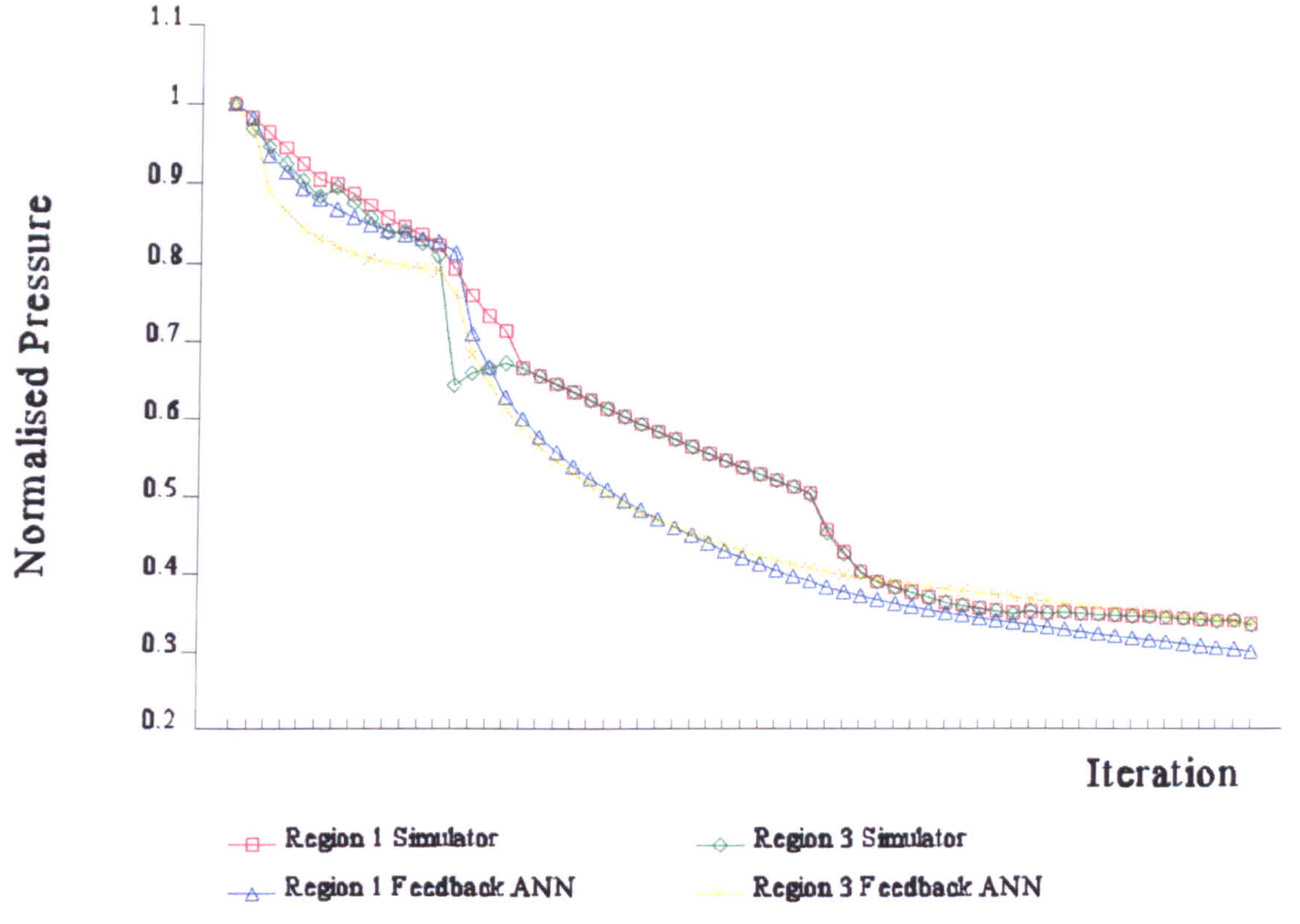


Fig 5.39: 5 Transient Feedback ANN and Simulator Results for Transient 1

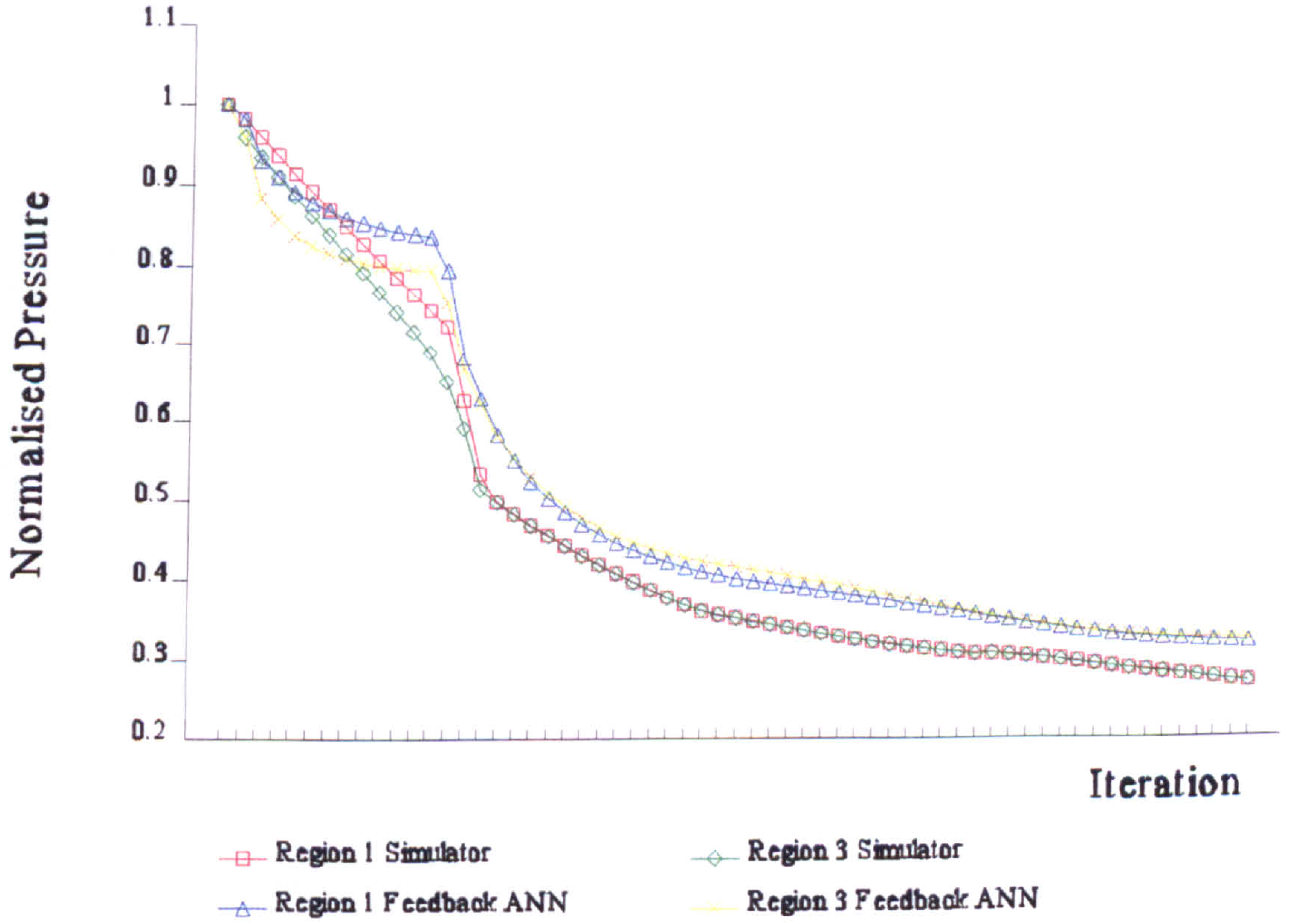


Fig 5.40: 5 Transient Feedback ANN and Simulator Results for Transient 2

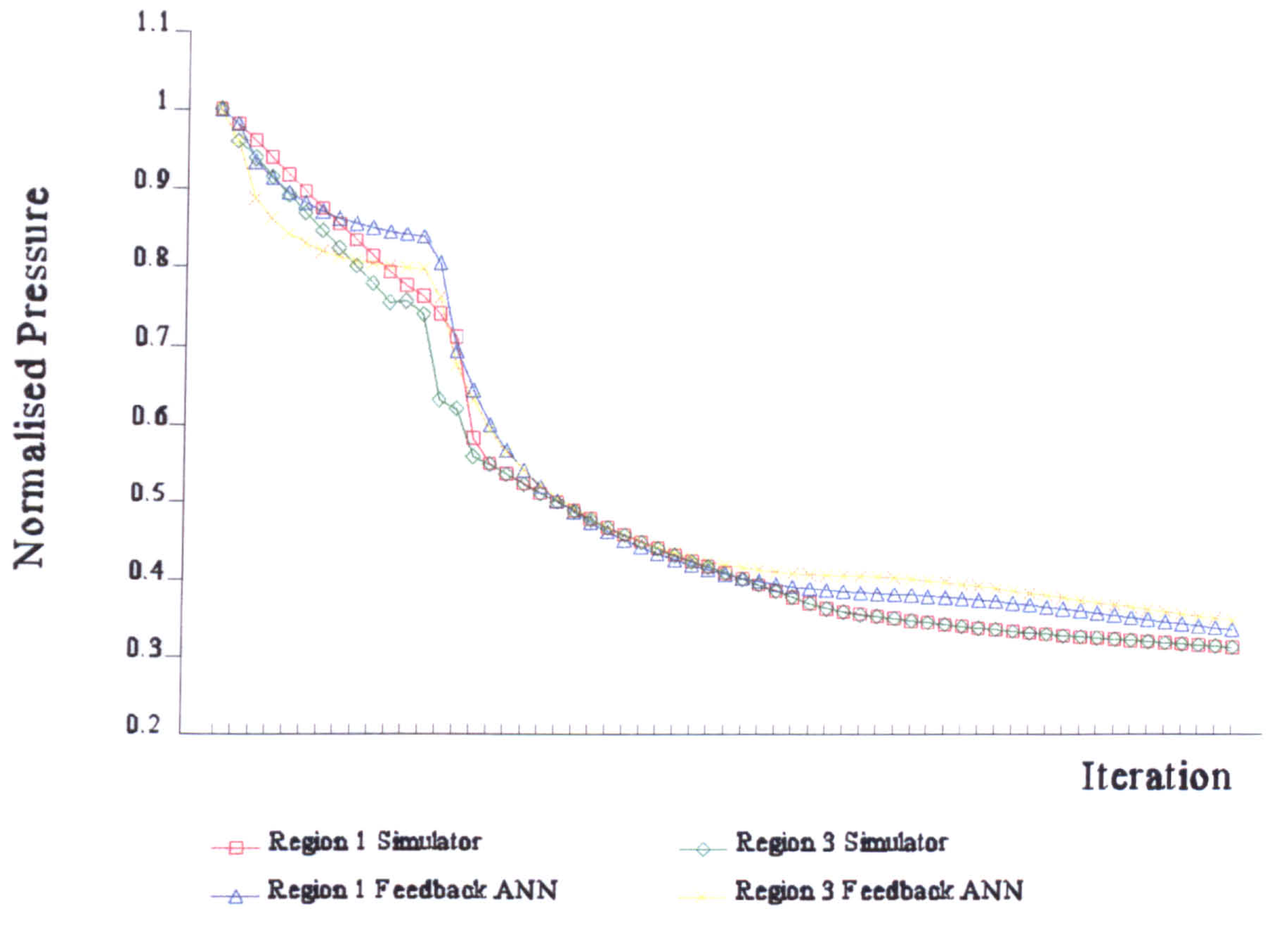


Fig 5.41: 5 Transient Feedback ANN and Simulator Results for Transient 3

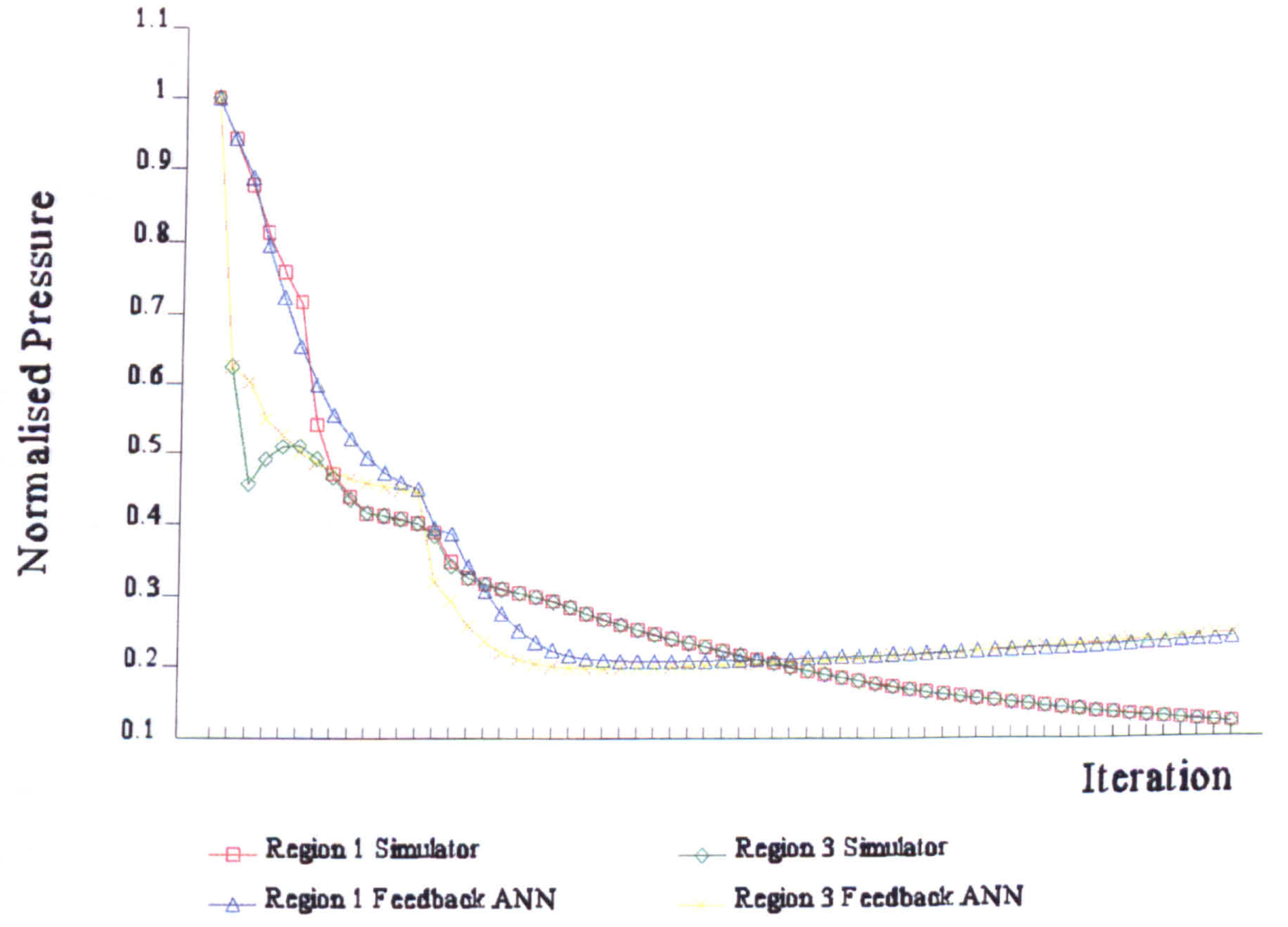


Fig 5.42: 5 Transient Feedback ANN and Simulator Results for Transient 4

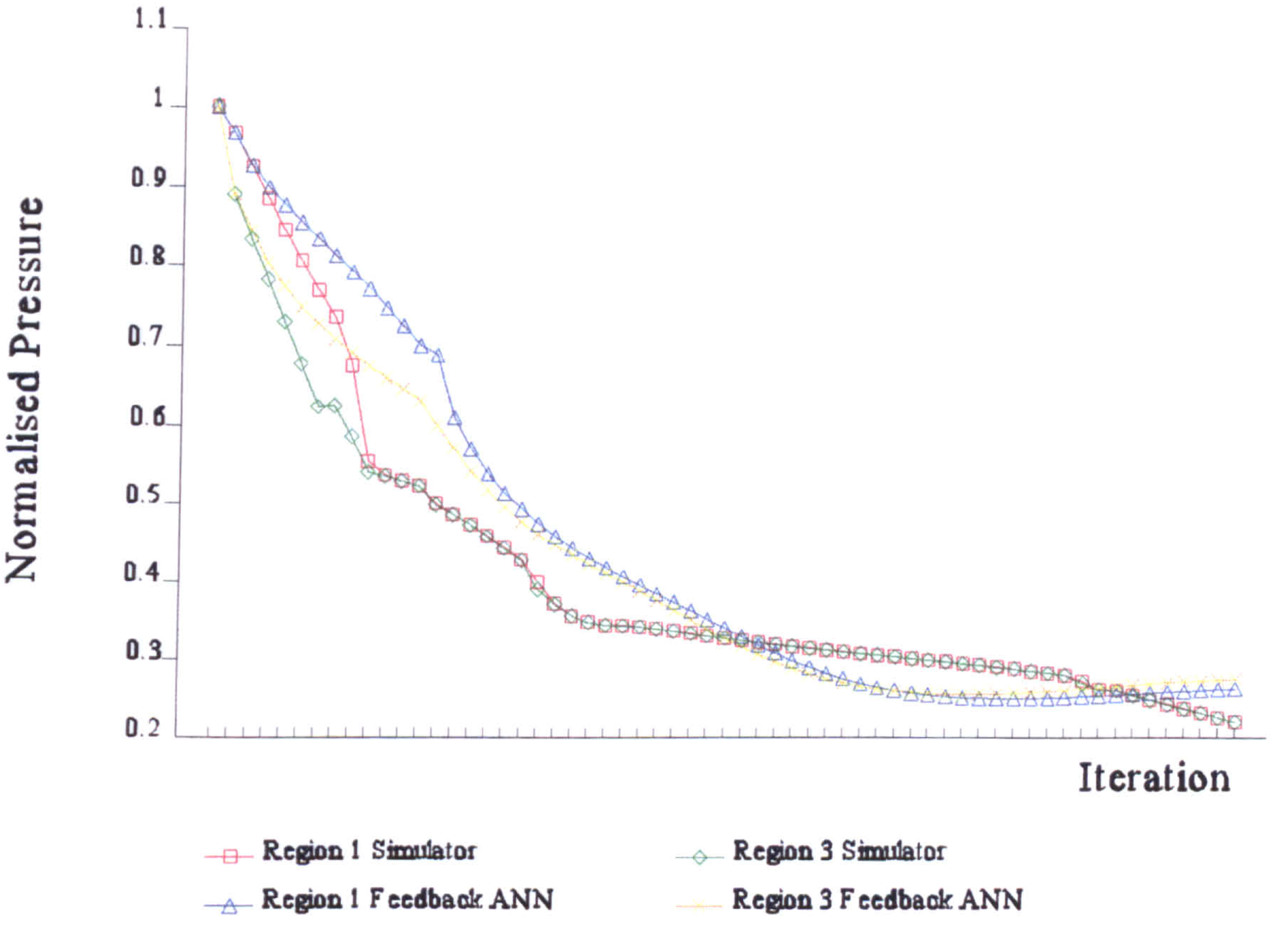


Fig 5.43: 5 Transient Feedback ANN and Simulator for Transient 5

All five transients are well predicted in view of the feedback nature of the tests. Additionally the general trends of each transient are followed favourably. This last test was a significant advance in using ANNs for the prediction of PWR variables. The success of this investigation logically leads to the consideration of the number of transients that a single ANN can accurately predict. This question is a key milestone to the development of the advisory system. If a satisfactory solution can be determined then the prediction element of the system becomes a realistic possibility. Conversely the lack of suitable result may severely limit the practical use of the advisory system.

5.7 Discussion

The results from this chapter indicate that ANN technology is an acceptable tool for the development of the prediction component for the proposed PWR operators advisor. The suitability of ANNs for PWR predictive tasks has been extensively explored and generally the ANNs developed have produced accurate predictions in all the investigations. Furthermore ANN techniques have been shown to be quite robust for PWR predictive applications. The performance of the ANNs have not been adversely affected by the

5.5 Predicting both similar and different transients

The above investigations have all considered only one transient per neural network. This approach would not be practical for the prediction level of the operator advisor. There are several reasons for this. Firstly, the number of neural networks required to implement a full PWR advisor with a full range of possible transients would be very large and unwieldy. Secondly, it would be very unlikely that all possible scenarios could be fully encapsulated in a discreet number of individual networks. Lastly, the maintenance and upgrading of such a system would be challenging.

A far better solution would be to develop ANNs that can each accurately predict a set of transients. The transients included in each neural network would be selected to ensure acceptable results. This investigation explores the basic criteria of the transient selection for inclusion in a single ANN.

Three of the PWR transients used in Section 5.2 of this chapter were considered in the investigation. They are summarised below, together with the corresponding numbers from Section 5.2:

Transient A: Large leak from primary to secondary (Transient 1).
Transient B: Large break in the hot leg (Transient 4).
Transient C: Small break in the hot leg (Transient 5).

Two of the conditions, B and C, were similar with only a different leak size. Transient A is very different to the other two. The graphs of the pressures in the regions on the PWR are shown below in Figures 5.22, 5.23 and 5.24. As before a simulator program was used to produce the data. The regions of the PWR considered are also the same as the previous test.

5.6 Forms of input for predictive ANN

The work so far reported has only considered one type of input to a predictive ANN. The value of pressure in key nodes of the PWR simulator has been used both as inputs and outputs of ANNS. There are other forms of input that should also be considered as these are important components of the full PWR system. These inputs are external to the plant and are either controlled or involuntary. Controlled inputs are either specified by the operator or by automatic systems. The operator defined inputs include throttle position, valve states, power levels and rod positions. The automatic system inputs are typified by the emergency cooler safety system. The main source of involuntary inputs are leaks in the primary circuit.

All these types of input can be represented in one of four ways to an ANN.

1) Defined at the beginning and then fixed throughout a transient period
2) Defined at the beginning but changed during the transient period
3) Defined at every step of the transient period
4) Defined initially but thereafter output fed back as input

An example of the first input type could be an output, such as leak site, from the diagnostic level of the advisor. The second form of input is typified by the valve and pump settings of the PWR. The settings of these components could be changed by the operator following the defined corrective sequence. The third input type could model the behaviour of the leak during the transient. The leak size could be defined at every time step on a "look up" table which is referred to by the ANN at every step of the transient or this value could be determined by the diagnostic level of the advisor. Alternatively, the values could be calculated at each time step using mathematical expressions. The last form of input is used to model the PWR variables such as pressure and temperature. These are used to determine the reactors condition during the various combinations of transient and operator defined situations.

To investigate the effect of the different forms of input on ANNs a series of tests were devised. A range of five transients were identified and a selection of input types chosen to cover all possible forms as described above. These transients were different to those already used during this chapter. The different forms of inputs were represented with suitable reactor variables, as shown in Figure 5.34, below. The heating rate of the reactor was defined at the start of each test and remained unchanged throughout the transient. The time dependent input was represented by the emergency cooler valve. An initial valve setting was defined, along with a time step number for changing the state. For this initial work this valve was binary but the approach could be modified to cover multi-stage settings. The transient leak flow rate is used to represent the third form of input. This value is defined at every time step using a look up table. This approach will give additional control over the system as the leak can be realistically modelled for different situations. The pressures in nodes 1 and 3 are used to model the PWR condition. An initial value will be defined for each variable and all subsequent values will be determined by the neural network and fed back into the system. The details of the transients used for this investigation are given below, in the Table 5.10.

Transient	1	2	3	4	5
Heat Rate	4	1	2	2	2
Leak Flow Rate (kg/sec)	10 - 3.357	10 - 5.208	10 - 4.674	50 - 6.912	20 - 5.345
EC Valve Open (secs)	120	120	120	120	120

Table 5.10: Details of Transients

The simulator program was run for 600 seconds for each scenario with information being recorded every 10 seconds. The emergency cooler (EC) valve was opened after 120 seconds. The neural network training and test sets were produced from the final data set using two time steps and the same method as before. A range of ANNs were trained for each transient. Each network had ten inputs (the five inputs for time T and T-1), and two outputs (the pressure in nodes 1 and 3 at time T+1). All the ANNs developed had one hidden layer and used the back-propagation algorithm for training. The results for the best networks are given below. The full results for all ANNs developed for these tests are given in Appendix I.

The results show that generally the trends of the transients are well predicted by each ANN with feed back. The opening of the emergency cooler valve, at 120 seconds, seems to cause the most inaccuracies. However, in each case the network recovers and predicts the transient accurately. The recovery may be due, in part, to the presence of the look up table for the leak flow rate. This variable has no error build up from the feed back circuit and is correct at each time step of the transient. The opening of the emergency cooler valve does appear to introduce some instability into the networks. The time of the valve opening has not yet been changed and is the same as that on which it was trained.

The above neural networks have only been tested on the training and test sets used to develop each ANN. This situation is rather artificial as any low error network would be expected to produce a good response to the information with which it was trained. With the exception of the heating rate, the first three transients are very similar. The robustness of these networks could be investigated by considering the heating and leak flow rates from transients other than the training set. The results obtained above were considered encouraging enough to warrant further investigation in this area. The above ANNs were all trained for one transient. This arrangement was considered simplified and unrealistic as the predictive ANNs in the advisor would, if possible, be developed for a number of fault situations. A series of networks were developed to predict the pressure values for all five transients.

The training and test sets used to develop the individual ANNs were concatenated to produce a large data set for training the combined ANN. Using the backpropagation algorithm a series of ANNs were developed with this dataset. Each was trained for 100,000 cycles and then a further 20,000 cycles with testing every 100 cycles, the structure with the lowest RMS error being saved. The results for the best ANN are given below, in Table 5.12, full results are given in Appendix J.

Transient	Nodes in Hidden Layer	RMS Error
tesallh.nnd	10-6-2	0.0448

Table 5.12: Best Result of Five Transient ANNs

This ANN was embedded in a computer program. A feedback feature enabled the entire transient to be predicted given the settings for the initial two time steps. This program different requirements introduced by the tests conducted throughout the chapter, for example feedback of outputs, forms of input and number of time steps of input data.

The last set of tests culminated in the development of an ANN that could successfully predict five single PWR transients. A knowledge of the number and form of transients that can be predicted by an ANN is crucial to the future implementation of an ANN based advisor. The number of ANNs required to predict a realistic number of transients may be very large, indeed given the non-linear nature of the PWR the problem may even be NP complete. Fortunately the results from this chapter indicate that this is not the situation and that a set of ANNs trained on carefully selected sets of transient information could provide a suitable prediction layer in the advisor. Criteria for selecting suitable combinations of ANNs that could be used to develop these ANNs are not defined. However the initial indications would be that dis-similar transients may be a more successful combination for ANN training than alike PWR faults. Clearly more work is required to determine both the nature of transients that can be combined on a ANN and hence the number of ANNs that the predictive layer of the advisor would contain. This crucial work is the subject of the next chapter.

5.8 Summary

The work reported in this chapter has examined the applicability of ANN technology for the prediction of PWR variables for a reactor in several fault conditions. Basic implementations have been explored. The first of these was the simple prediction of a set of PWR variables one time step into the future. The success of this work led to a discussion on the optimum number of time steps to use for inputs into a predictive ANN. The feedback of predicted values back into as inputs was then investigated and again an ANN was found to be a reasonable tool for this task. The form of input to an ANN was explored and four types of input identified. A set of ANNs were developed to determine the effect of these input types on ANN prediction ability. A single ANN was finally developed that successfully predicted future variables for five transients. The number of transients that an ANN can successfully predict was identified as an important requirement for a practical implementation of this component of the operators advisor. This question is considered in greater detail in the following chapter.

Chapter

6 Predicting PWR Transients

6.1 Introduction

This chapter builds on the work of predicting future reactor condition begun in the previous chapter. A key question, posed, at the end of the last section, concerned the number of transients an artificial neural network (ANN) that can be successfully trained to predict. This chapter is devoted to examining and addressing this issue. Some initial theory on predicting a number of transients with a single ANN is followed by the introduction and exploration of two distinct, possible solutions to the question. The first involves developing a set of empirical relationships for the mathematical elements of a transient curve. These relationships could then be used to calculate the parameters for each member of a set of fault conditions to determine which can be successfully combined to predict the transients for a given accuracy. This approach is shown to have severe shortcomings due to the inter-dependencies between the parameters. The second method, Section 6.4, develops an ANN based model of the Pressurised Water Reactor (PWR) capable of predicting a large number of transients. A basic building block is developed and then used to construct a fast, accurate simulator of reactor transients. All the findings are reported in the chapter, however detailed neural network development results are contained in Appendices I to R.

6.2 Theoretical Foundations

This section introduces and discusses some theory relating to ANN prediction of transients. The ideas described are used as a foundation for the later sections in this chapter. The work discusses using ANNs for predicting more than one transient. For the moment each transient being predicted is assumed to only occur singly and in isolation from other fault conditions. Scenarios of two different transients occurring together are unlikely due to the design of the PWR, however one transient may cause secondary problems, for example a steam leak affecting the instrumentation in the reactor compartment. For the current discussion all transients will be considered to occur alone although the ANN may well be trained to predict a number of these single transients.

Consider the three transients, shown in the following diagram, Figure 6.1. For this discussion, and for the remainder of the chapter, the relationships between PWR variable and time used for illustrating the concepts do not represent actual PWR variables but are typical of the form found for several measurable plant factors such as node temperature or pressure.

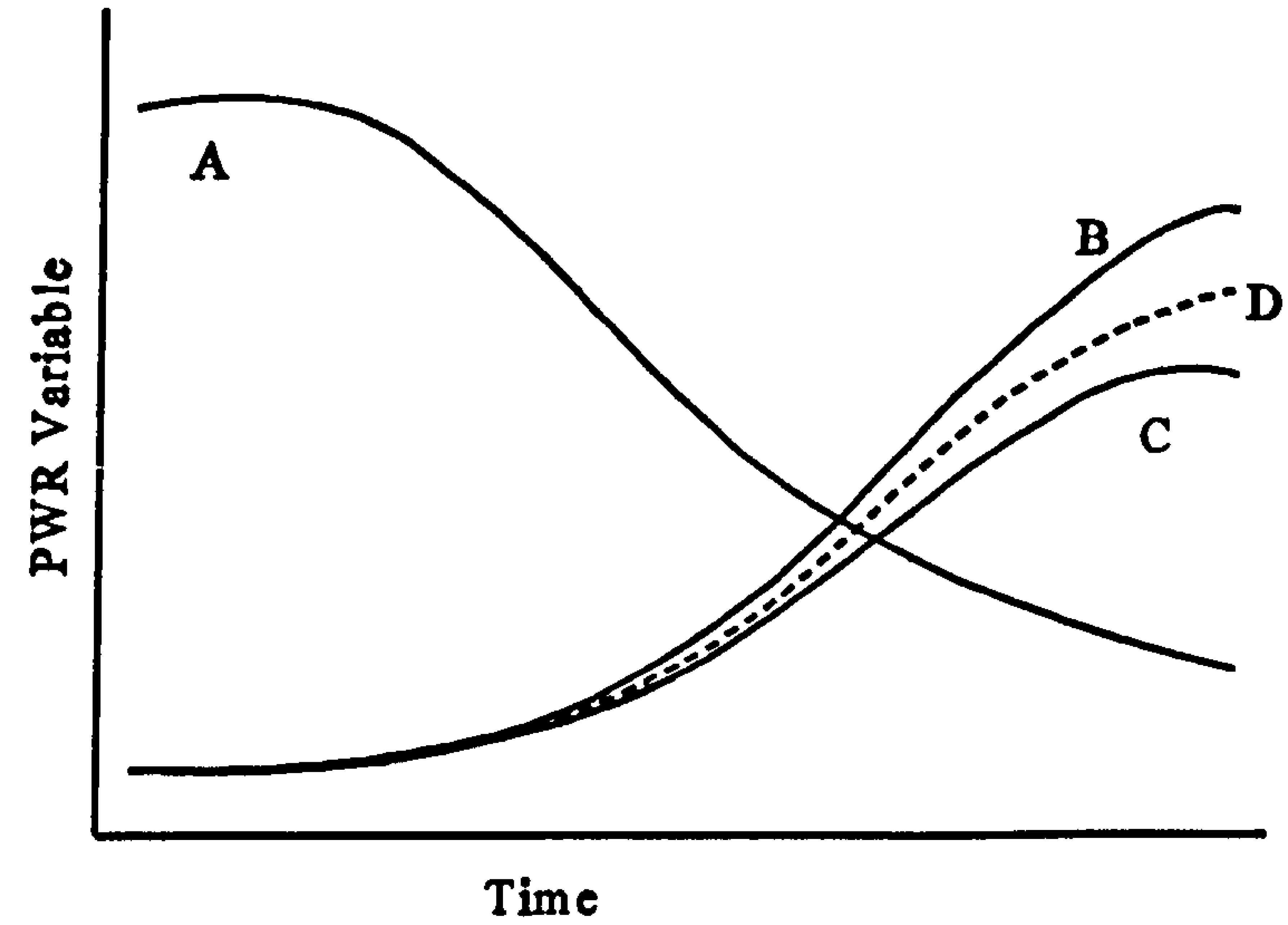


Fig 6.1: Three Transient Curves

The results from the previous chapter, Section 5.4, imply that ANNs trained with data for pairs of transients will be able to predict Curve A and one of the other curves fairly accurately, but may have difficulty predicting Curves B and C. Whilst the two latter curves are very similar for the early stages of each transient, they diverge with time. Unless there are additional inputs to identify particular scenarios, such as the binary inputs used in the investigation from the previous chapter (Section 5.4), an ANN would have difficulty predicting either curve. In practice, with the absence of additional information, the network would tend to produce an output which is mid-way between the two options, as shown by Curve D.

The above three curves are incorrectly depicted as continuous lines. There is no relationship between successive points on the curve, a set of inputs produces a single corresponding output. There is no record for the position of the point in the transient's history. The ANN is not given a sense of time. Each point on the output graph should be shown as a set of one step predictions, as shown in the figure below, Figure 6.2. For the remainder of this section the curves under discussion will be depicted as a set of disjointed points.

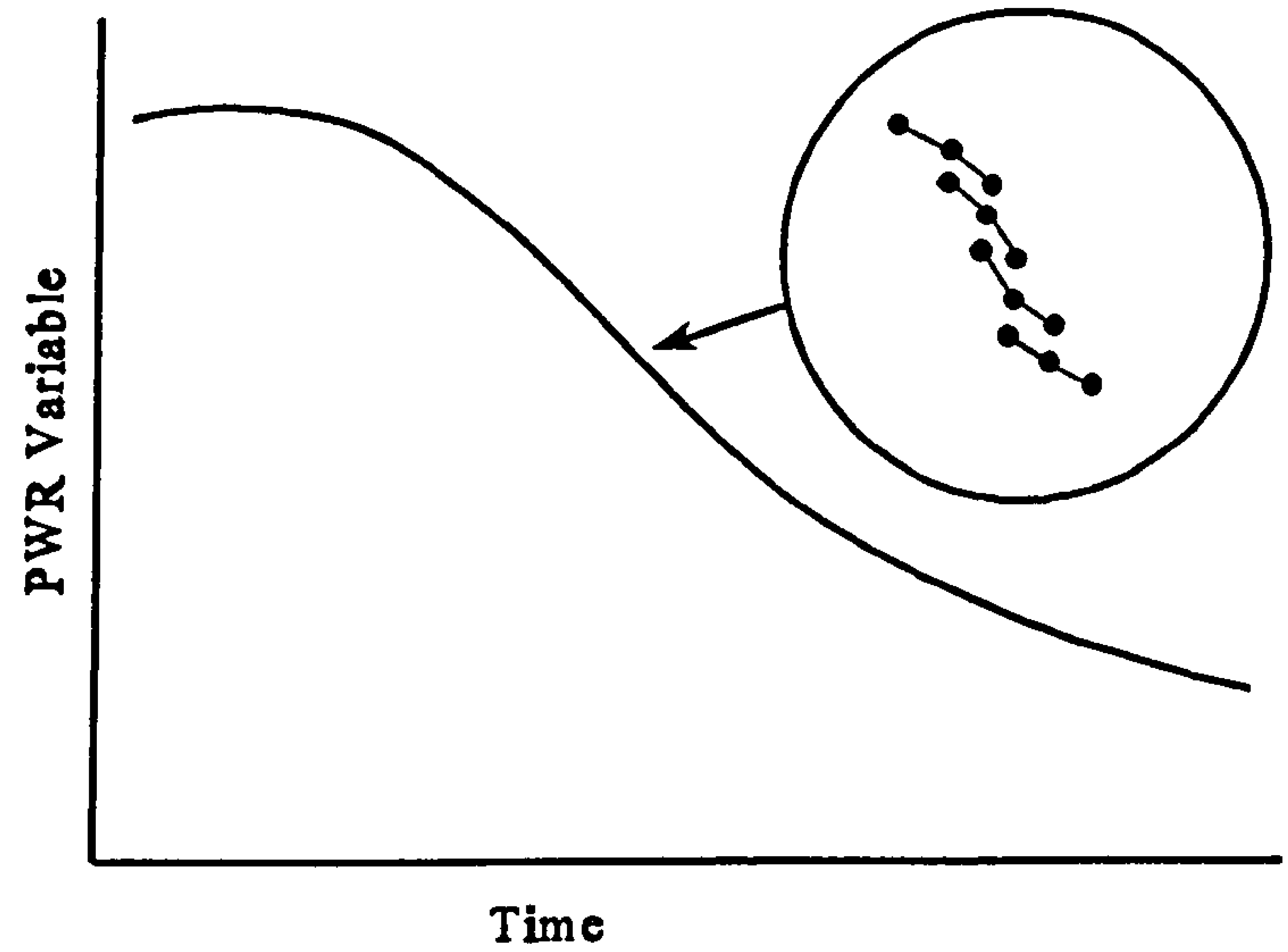


Fig 6.2: True Depiction of Transient Plot

A closer view of the intersection of the curves A and B is shown below, in Figure 6.3.

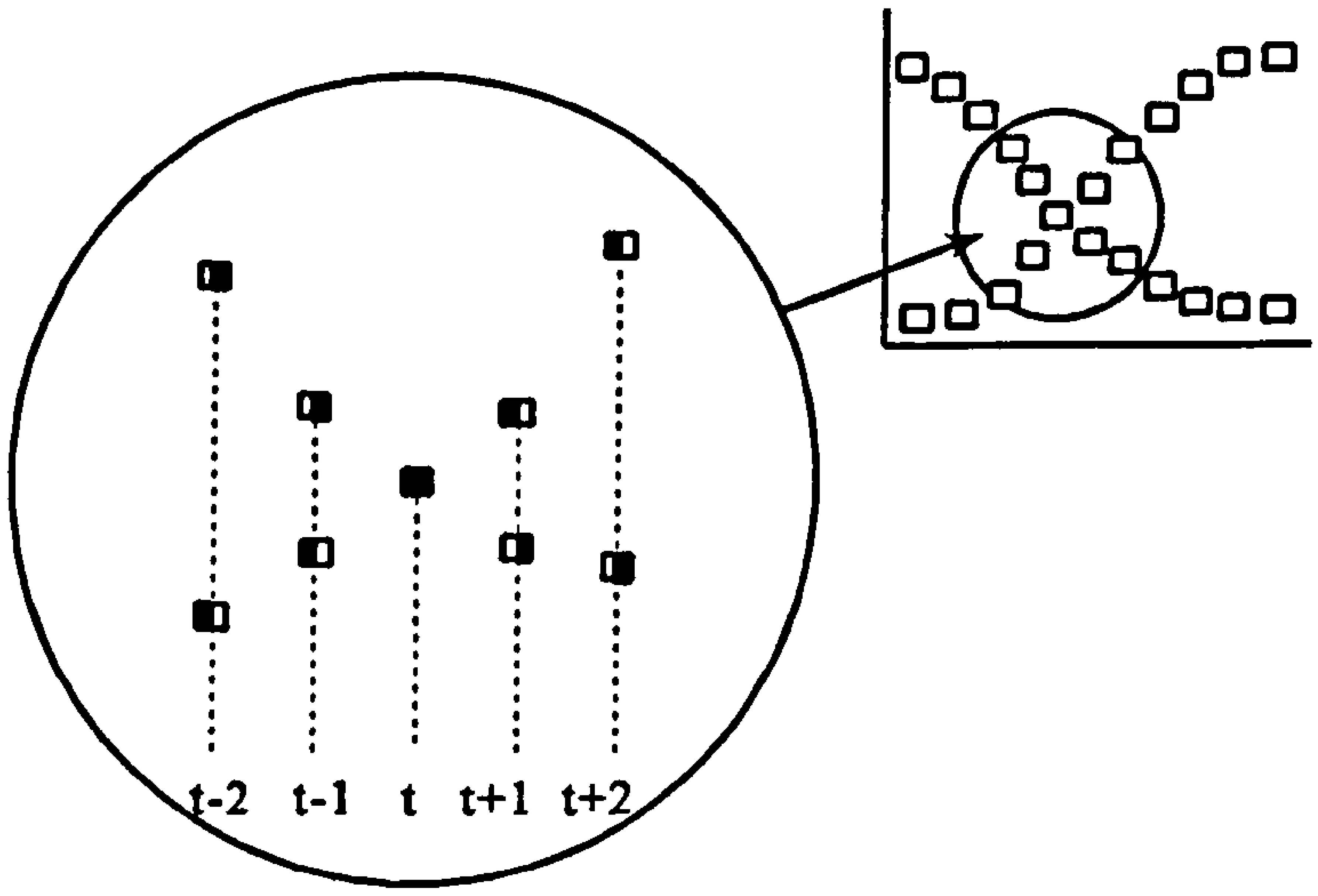


Fig 6.3: One Overlap Intersection

The two curves share a common point, at time t. An ANN, with a minimum of two time steps for the input set, should be able to distinguish between the curves as either t-1 and t or t and t+1 are elements of the inputs and each has only a single common point, time t. However, if instead of a single intersection point the curves shared two common points, as shown in the following figure, Fig 6.4, a different situation arises.

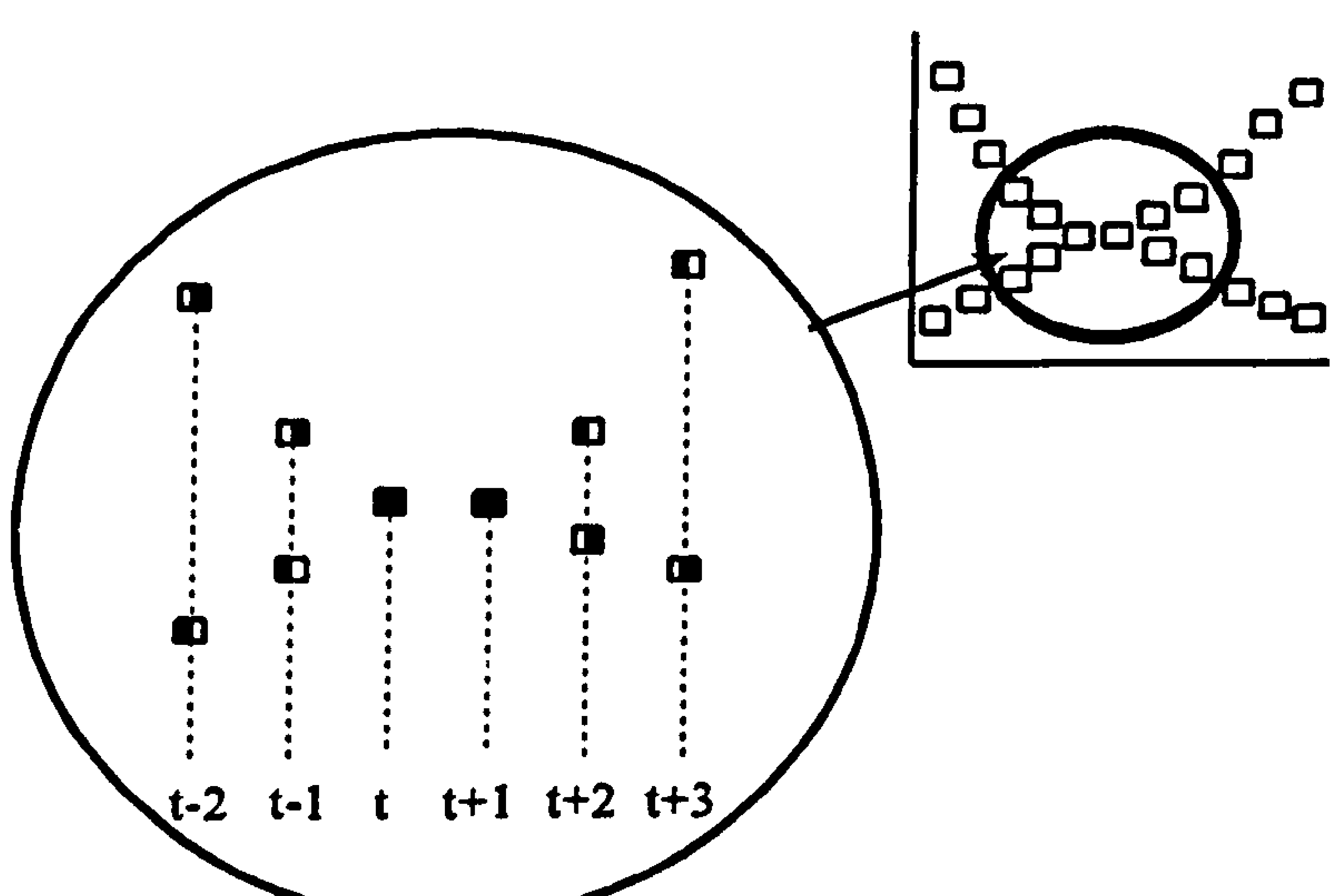


Fig 6.4: Two Overlap Intersection

If two time steps are used for predicting the next value of the variable a difficulty arises for time t+2 as the inputs for both transients are identical. Without additional information the ANN has no guide as to which transient is being predicted. The ANN has no long term history of previous inputs only the current input set on which to determine the output. If this information is ambiguous the network cannot accurately predict the next value of the reactor variables.

This example used two input time steps but the argument is valid for all values. A larger time step allows a greater number of points to be considered for the input set but the situation could still arise where the number of common intersection points is equal to the number of time steps. This situation may be less likely to occur for larger time steps but still remains a possibility. The above examples emphasise the requirement for additional information to be available to the network. The extra inputs could be the results from the diagnostic level of the adviser, identifying the transient under consideration.

The above discussion has only focused on coincident points. A similar argument is valid for points that are close together. It has been assumed above that if two points were not coincident then an ANN could successively distinguish between them. However, this cannot be the case in practice as each ANN has a small error in the output values. While this error is minimised during the training process it still exists. It is possible that, with similar curves, the actual output is closer to that of the second fault than to the transient under consideration. This problem becomes compounded and more critical for ANNs using feedback to predict an entire transient as the error could increase during the transient run.

For a defined ANN error there could be a minimum distance between various possible transient curves to enable accurate prediction. Below this distance spurious input signals, from feedback, noise and faulty sensors, could cause the ANN to incorrectly predict the second transient values even with additional input signals.

The above discussion has considered an ANN with a single output. In practice a predictive ANN would output values for several relevant plant variables. These would have implicit relationships established during the training process and may well assist during the prediction process. The example of Figure 5.5 illustrates this point as the change of gradient in one variable, Node 3 pressure, induces the corresponding changes in the slope of the other variables, Node 12 pressure, and so helps in the prediction.

This section has introduced some basic ideas for predicting many singularly occurring transients with an ANN. There are several potential problems of predicting the fault under consideration. These are mostly concerned with enabling the ANN to determine the correct sequence of reactor variable values when the transient is in close proximity to a second possible curve. These conflicts can be resolved by supplying the ANN with sufficient information to distinguish each particular transient implicitly. In the proposed operator's advisor system this detail could be provided by the output from the diagnosis level of the hierarchical system, which identifies the transient occurring.

6.3 Empirical Approach

The previous section introduced several characteristics of transient curves, the notion of intersections and similar points were discussed. These features are not unique and other geometric characteristics can be identified. If an appropriate relationship between them could be established it may be used to determine whether an ANN could be successfully trained to predict a given set of PWR transients. The features for each member of a given set of transients would be used This section expands on this initial idea. Initial investigations result in the identification of a set of key geometric features of transient curves. A numerical relationship between these elements and their relationships is then explored. If successful the equations so developed would then be used to calculate which combinations of transients could be successfully predicted, by a single ANN, for a maximum defined error.

6.3.1 Classification of Points of a Transient Curve

One of the simplest forms of empirical classification for transient curves would be to allocate a numerical value to the relationship between corresponding pairs of points for the two transients being considered for inclusion in a predictive ANN. To illustrate this idea consider the following figure of two typical transient curves for a undefined plant variable.

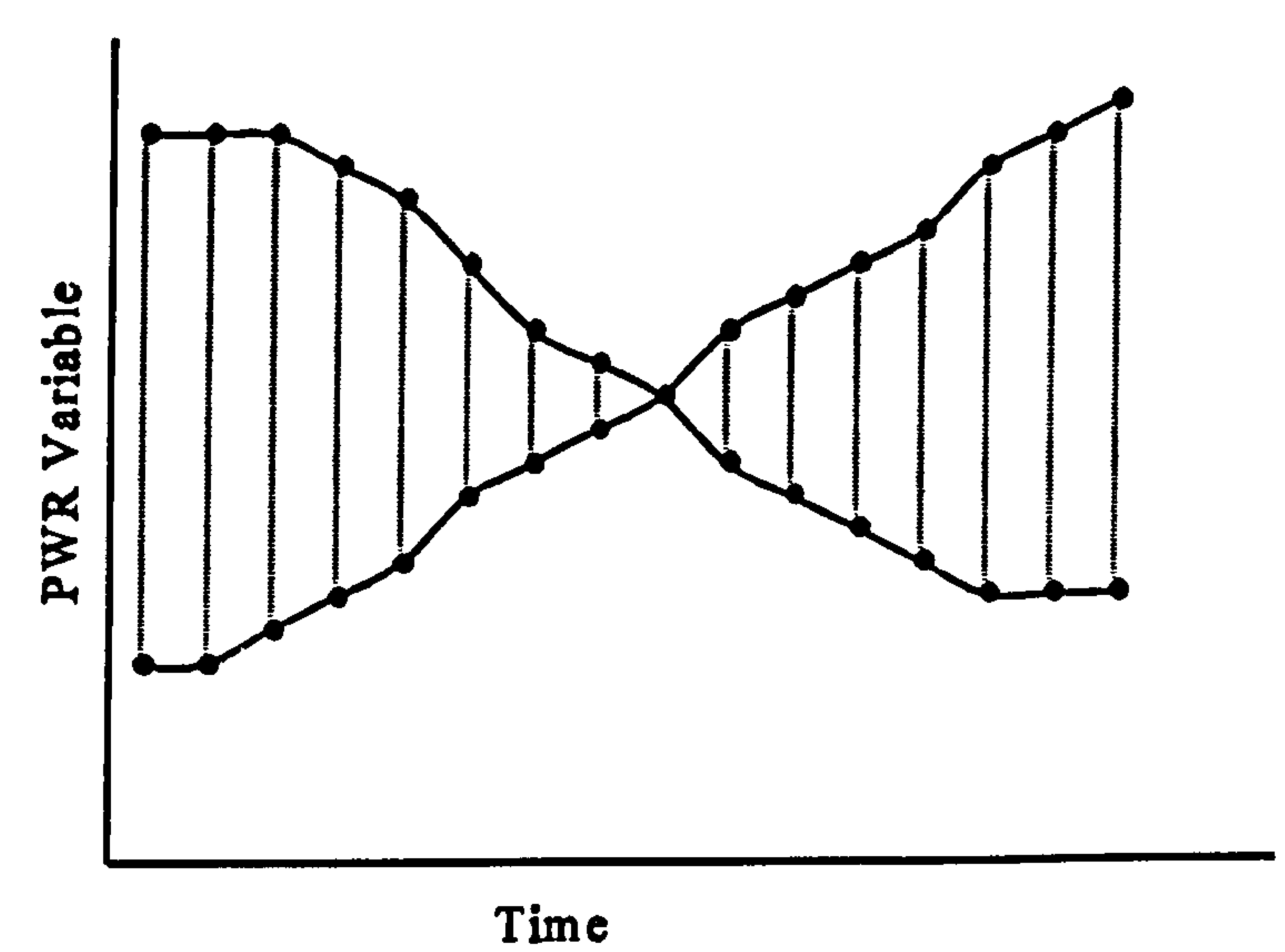


Fig 6.5: Distances between points on Transient curves

If the two curves are considered in vertical sections according to the distances between the pairs of points. Some sections of the curves, where the pairs of points are quite separate, would be relatively easy to distinguish between and so predict. However, other sections of the curves are less well spaced or even coincident and this could lead to problems in the transient predictions. The proximity of points and the length of each section of like points could be a guide to the suitability of the two transients for inclusion on the same predictive ANN. If the length and form of each section was assigned a numerical value, these values could then be summed over the entire length of the transient to produce a numerical rating for the given combination of transients. As an interim step the type of sections could be classified according to their length and the proximity of the points included in the section, a possible classification is given in the following table.

Code	Classification
1a	1 point coincident
2a	2 points coincident
pa	p points coincident
1b	1 point within 0 → 5%
2b	2 points within 0 → 5%
qb	q points within 0 → 5%
1c	1 point within 5 → 10%
rc	r points within 5 → 10%
1d	1 point within 10 → 15%
sd	s points within 10 → 15%
1	1 point greater than 15%
te	t points greater than 15%

Table 6.1: Classification of Transient Points

It is assumed that an ANN can successfully distinguish between curves with points greater than 15% difference. Using this classification the relationship between the curves in Figure 6.5 could be described as 5e + 3d + 2c + 1b + 2a + 3b + 2c + 1d + 5e. The numerical value for each section should also include reference to some basic parameters of the ANN such as number of time steps used for the input set and the number of reactor variables predicted by the network.

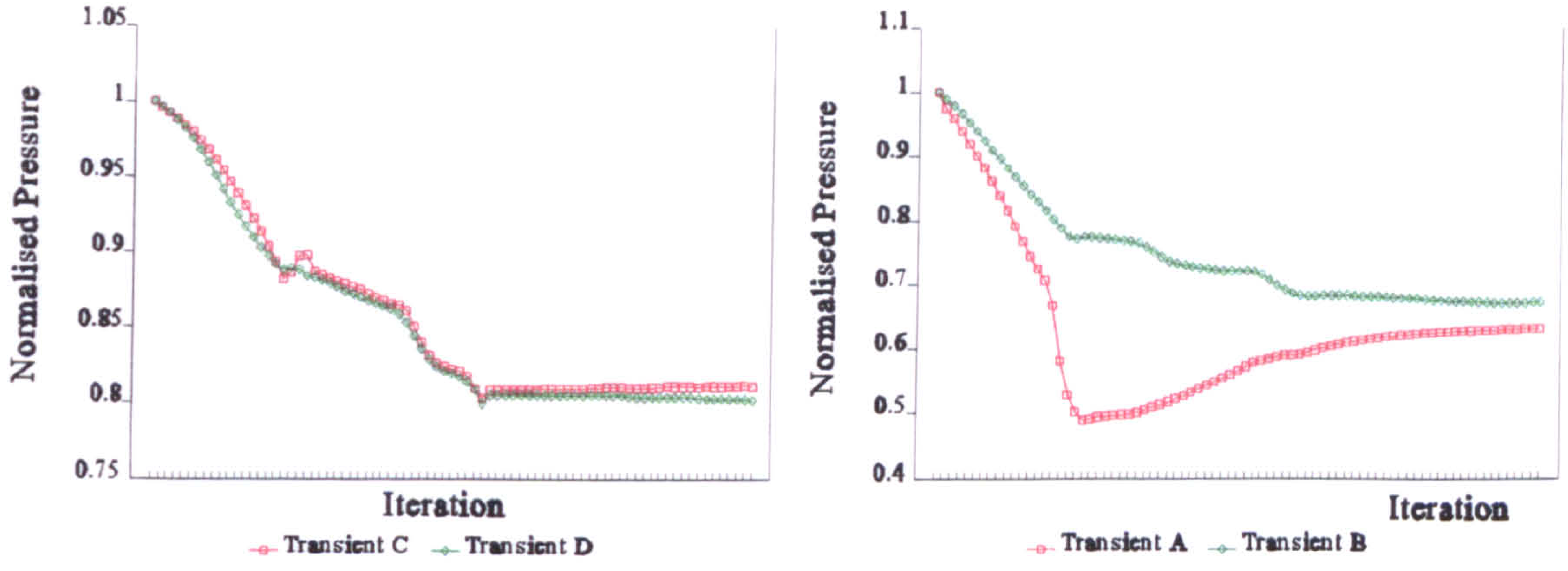
A numerical value could then be defined for the distance between corresponding points for a pair of transient curves. These numbers could then be summed and compared to a previously established threshold value to determine the suitability of training a single ANN to predict the given transients to a defined accuracy.

The determination of the threshold value is problematic. One method of producing such values may be to create a set of 'dummy' transients and then develop ANN prediction relationships for them. The method has the weakness that the result obtained may not be the optimal result for the selected combination occur for two reasons. Firstly, the ANN training may become trapped in a local RMS error minima, a more likely occurrence with complicated transients, so the prediction of the test 'dummy' transients introduces a measure of inaccuracy. Secondly the transient curves are not all equally predicted by an ANN, as seen in the examples in Chapter Five. Any threshold value would require consideration of both an ANNs ability to singly predict each condition under consideration together with predicting the given combination of transients. The problem is further compounded when considering three or more transients.

If a realistic value for the threshold function could be determined for each combination of transients applying the results to PWR transients could be difficult as the inherent error in ANN training may not give the same results as those estimated from the 'dummy' transients. The wide range of possible inaccuracies highlight the inappropriateness of this approach to predicting transients with a single ANN.

6.3.2 Area between Transient Curves

A second possible method of determining the appropriateness of an ANN to predict the above transients could be to consider the relative areas between the curves. The magnitude of the area may be a guide to the success of accurate prediction. To illustrate this idea consider the following two pairs of transient curves as depicted in Figures 6.6a and b. The shading represents the areas between the graphs.

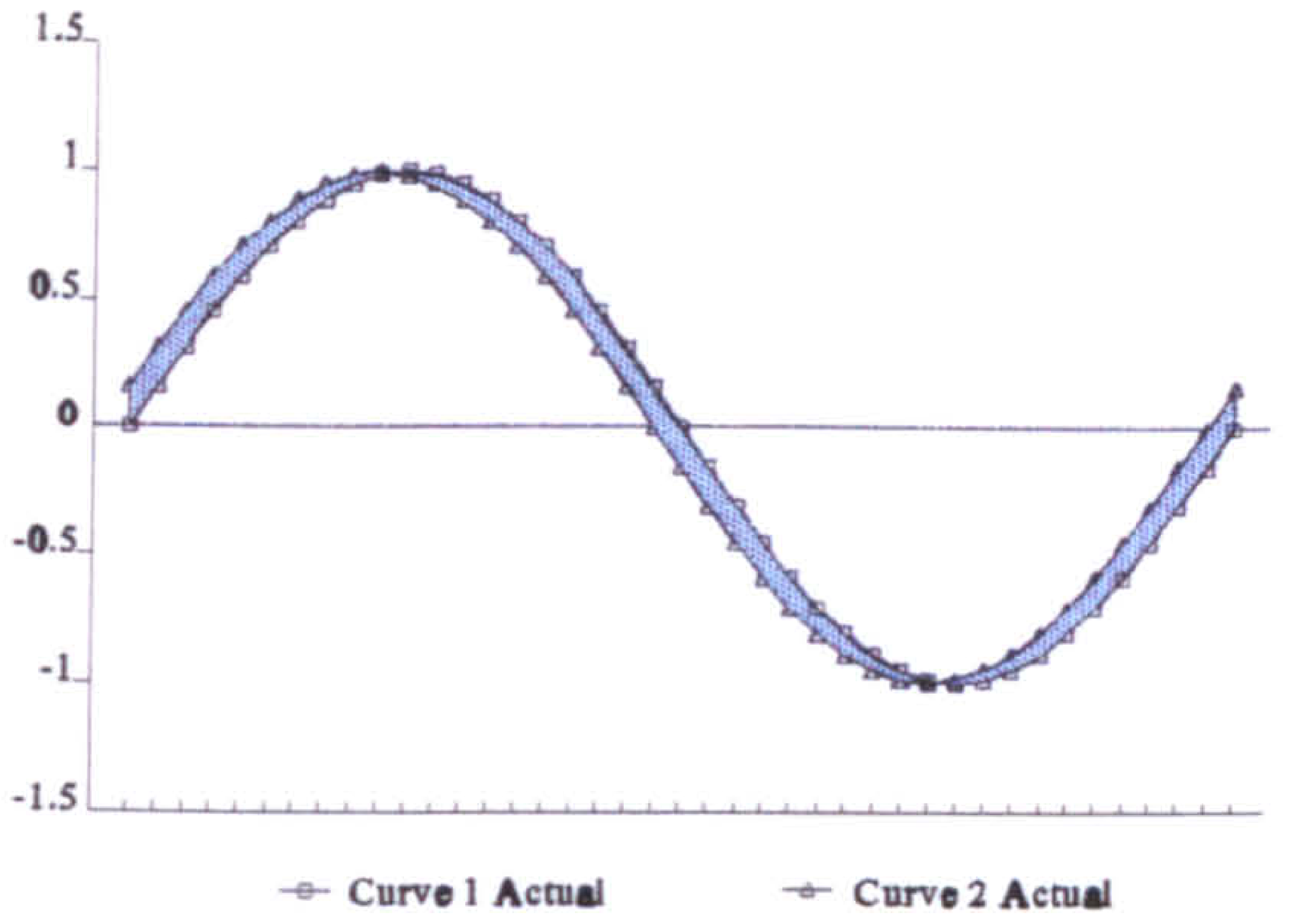


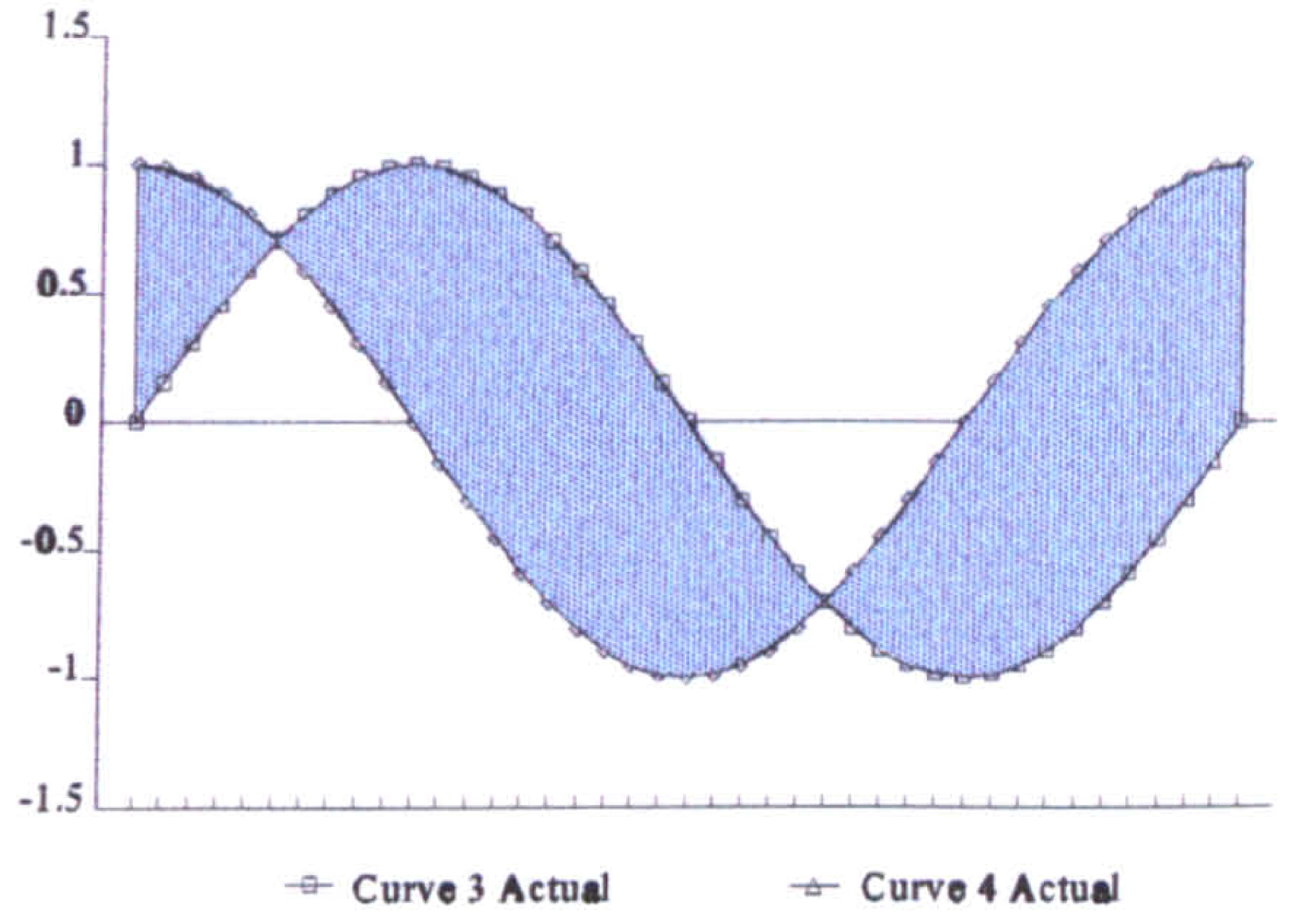
Figs 6.6a & b: PWR Transients

The curves in Figure 6.6a are very similar and so have a small area between them. However the transients represented in Figure 6.6b are quite different so the area between the curves is significantly larger. The results of the previous chapter would imply that the second pair of curves would be more accurately predicted, by a suitable ANN, than the first pair. This result may be related to the size of the area between the curves. A small area could imply a lower accuracy of the ANN to discriminate between the curves and so model the transients. A larger area could signify a better accuracy for ANN discrimination and prediction of the curves.

The mathematical function for each transient is unknown so the total areas would have to be determined by a summation of the areas of a number of small vertical strips. If the curves were normalised this idea could be further developed to equate the total area with a value of the accuracy of prediction. Again only one plant variable has been considered. The number and interdependency of variables used to develop the ANNs would have an important effect on the final result due to the inter-relationships that exist between them. For example, a change in a node temperature could produce a corresponding change in pressure or flow rate.

To explore this idea further the following two situations were considered, Figs 6.7 a and b. Each curve represents the stylised behaviour of a PWR variable during a transient along with the output from the best trained ANN. Pairs of transients are being considered for possible inclusion on the same predictive ANN. Each pair of transients have two intersection points, above and below the horizontal axis. To ease the calculations the variables are all initially represented as sinusoidal curves.





Figs 6.7 a and b: Curves for Area between Transient Investigations

Area between first pair of curves = 0.626 (3 dec. pl.)

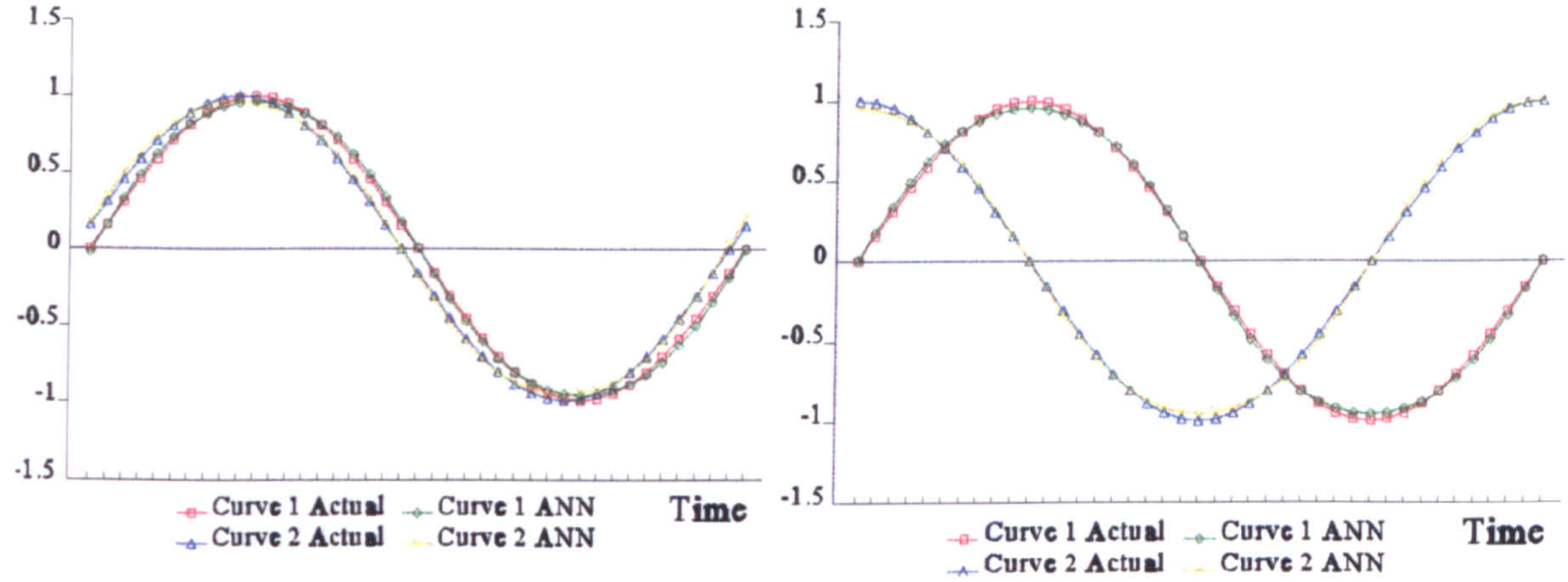
Area between second pair of curves = 5.657 (3 dec. pl.)

The above discussion would suggest that the second pair of curves would be easier to distinguish between, and so accurately predict the transients, compared to the first pair. ANNs were developed for each situation. The output from the diagnostic level of the advisor was simulated by the inclusion of two binary inputs into the training data. In each case one output was set at `1' to signify occurrence and `0' to represent absence. The resultant full data set consisted of 80 cases with 3 inputs and a single output. The third input represented the time from the occurrence of the transient. This set was then randomly divided into ANN training and test sets in the approx ratio of 2:1. The results for the best of the ANNs developed are as follows:

Transient Pair	No. of Nodes	RMS Error
A	6	0.0201
B	6	0.0207

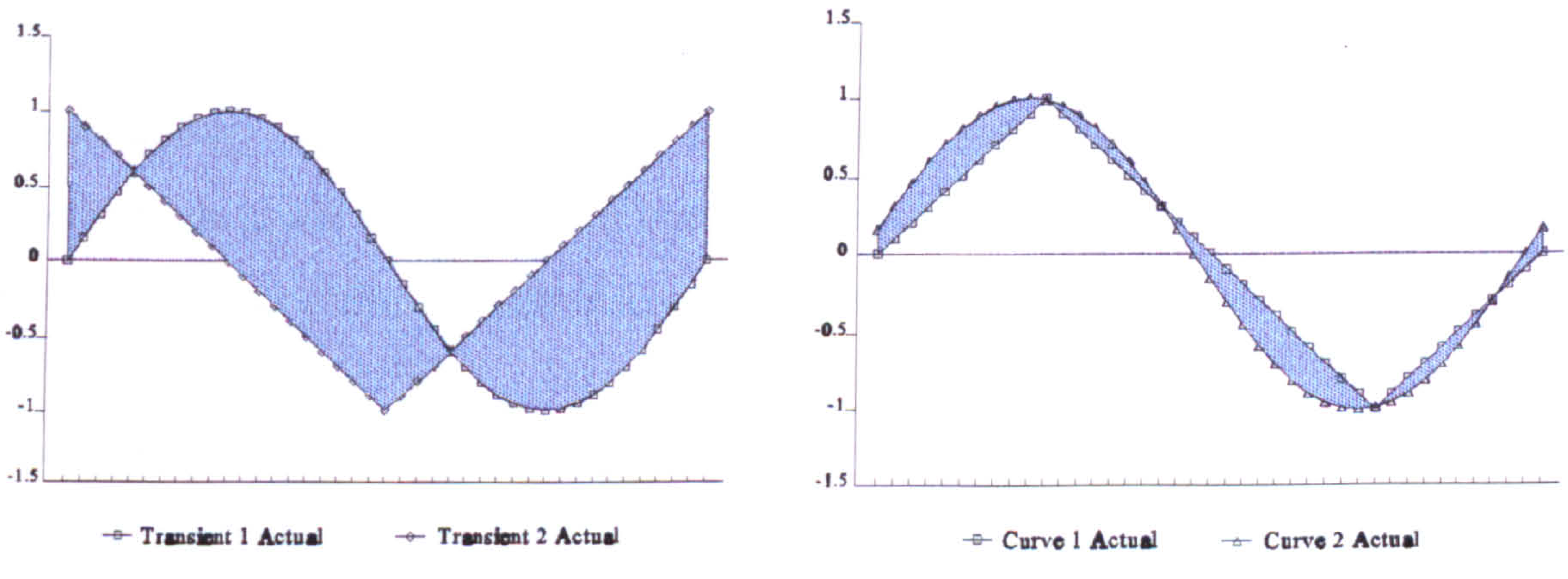
Table 6.2: Results of Area between Curves ANNs

The outputs from these two ANNs were plotted to visualize the accuracy of the predictions. The resultant plots are shown below in Figures 6.8 a and b.



Figs 6.8 a and b: ANN Predictions of Curves for Area between Transients

The best RMS errors are very similar this tends to suggest that the area between the curves is not a useful measure for ANN prediction accuracy. The results may be artificial as the same form of curve was used for each transient. To investigate this further the tests were repeated with a combination of linear and sinusoidal functions to represent the transients. Although this representation is not an accurate depiction of PWR transients it permits an elementary investigation of the underlying ideas. The same procedure as above was adopted for these new tests. The following two figures, Figs 6.9 a and b, show the test curves.



Figs 6.9 a and b: Curves for Area between Transient Investigations

Area between first pair of curves = 4.785 (3 dec. pl.)

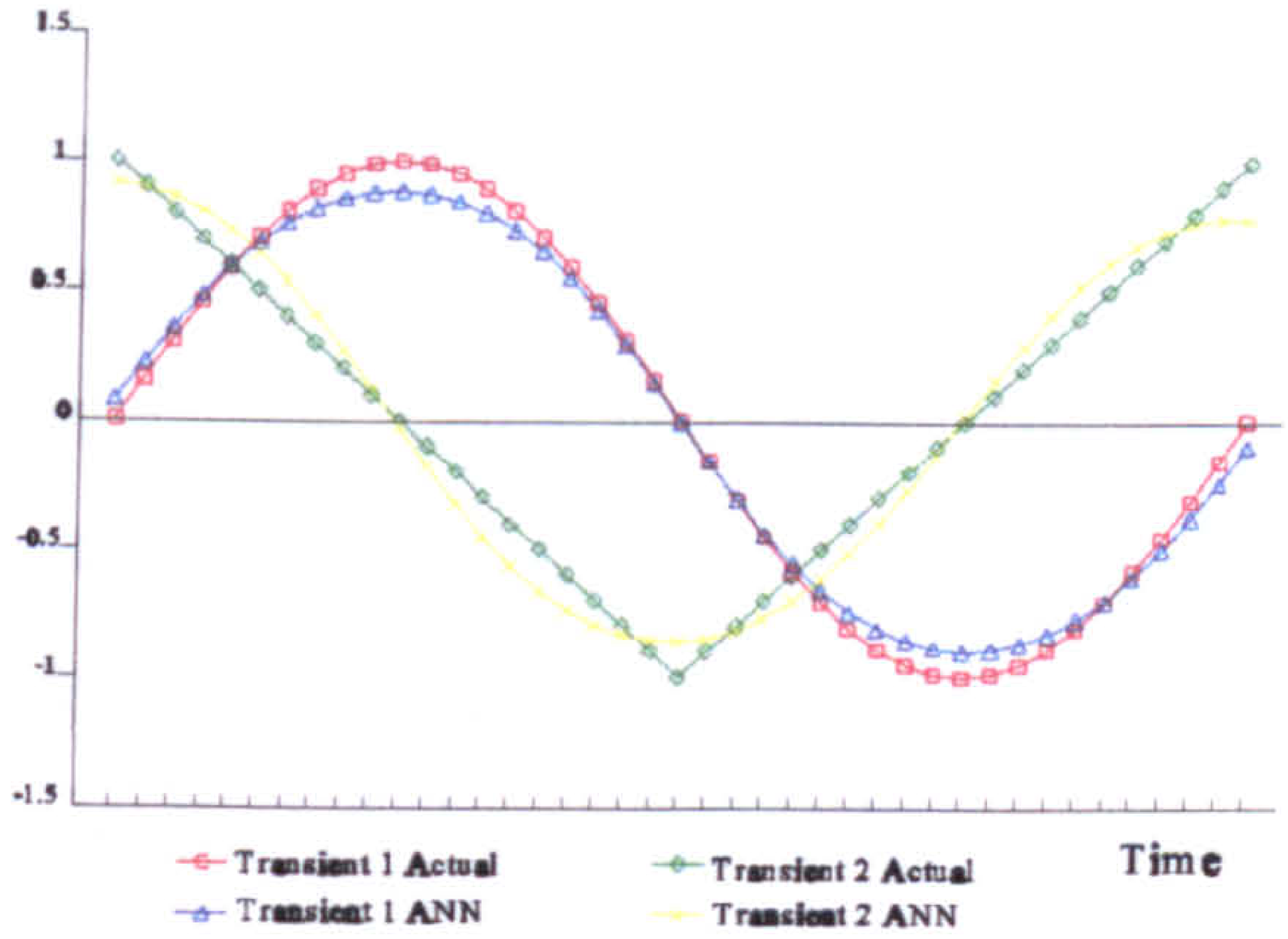
Area between second pair of curves = 0.858 (3 dec pl.)

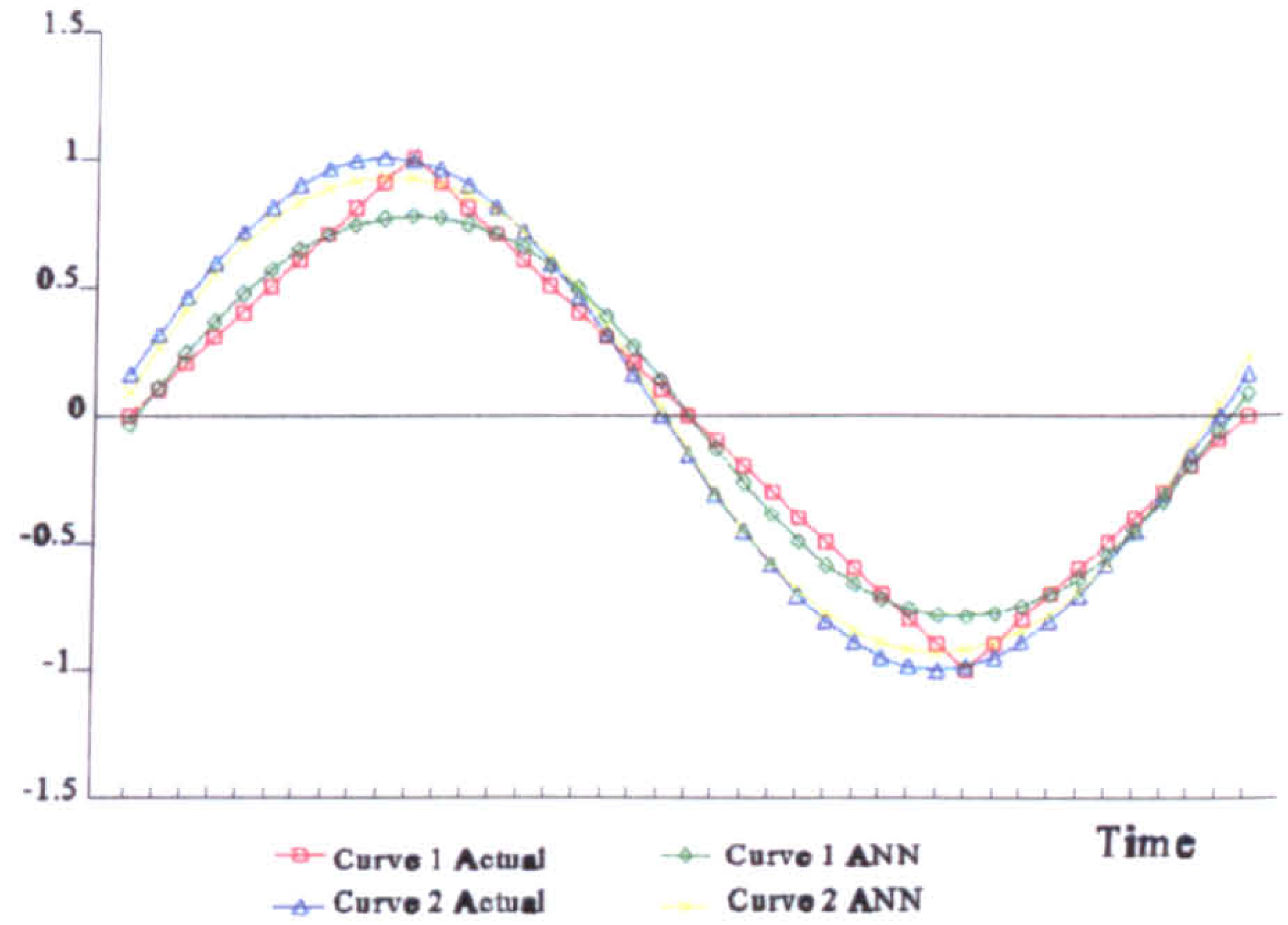
An identical approach to the previous tests was adopted the results for the best of the ANNs developed are as follows:

Transient Pair	No. of Nodes	RMS Error
A	6	0.0804
B	6	0.0792

Table 6.3: Results of Area between Curves ANNs

Again the outputs from these networks was plotted and the results shown below, in figures 6.10 a and b.





Figs 6.10 a and b: ANN Predictions of Curves for Area between Transients

The results confirm that the area between the curves is not a useful measure of ANN predictability. The later examples have similar areas and the same number of intersections as the first two cases but have quite different errors for prediction. There must be other contributing factors to account for this dilemma.

The principal difference between the two sets of examples is the composition of the curves. The first pair consisted solely of sinusoidal representation for the transients, while the second pair were composed of both linear and sinusoidal elements. The results in Figures 6.10 a and b show that the linear components were less accurately predicted than the sinusoidal curves. The turning points for each linear transient were not accurately predicted, the ANN produced a spline to the two converging elements. Several node threshold options were considered in the development of the ANNs, however the best networks in each case used a sine function. This transfer function was probably preferred because of the composition of the curves used in the ANN, a selection of non-sinusoidal curves may have favoured a different threshold function such

These best results are plotted in the following graph, Fig 6.12.

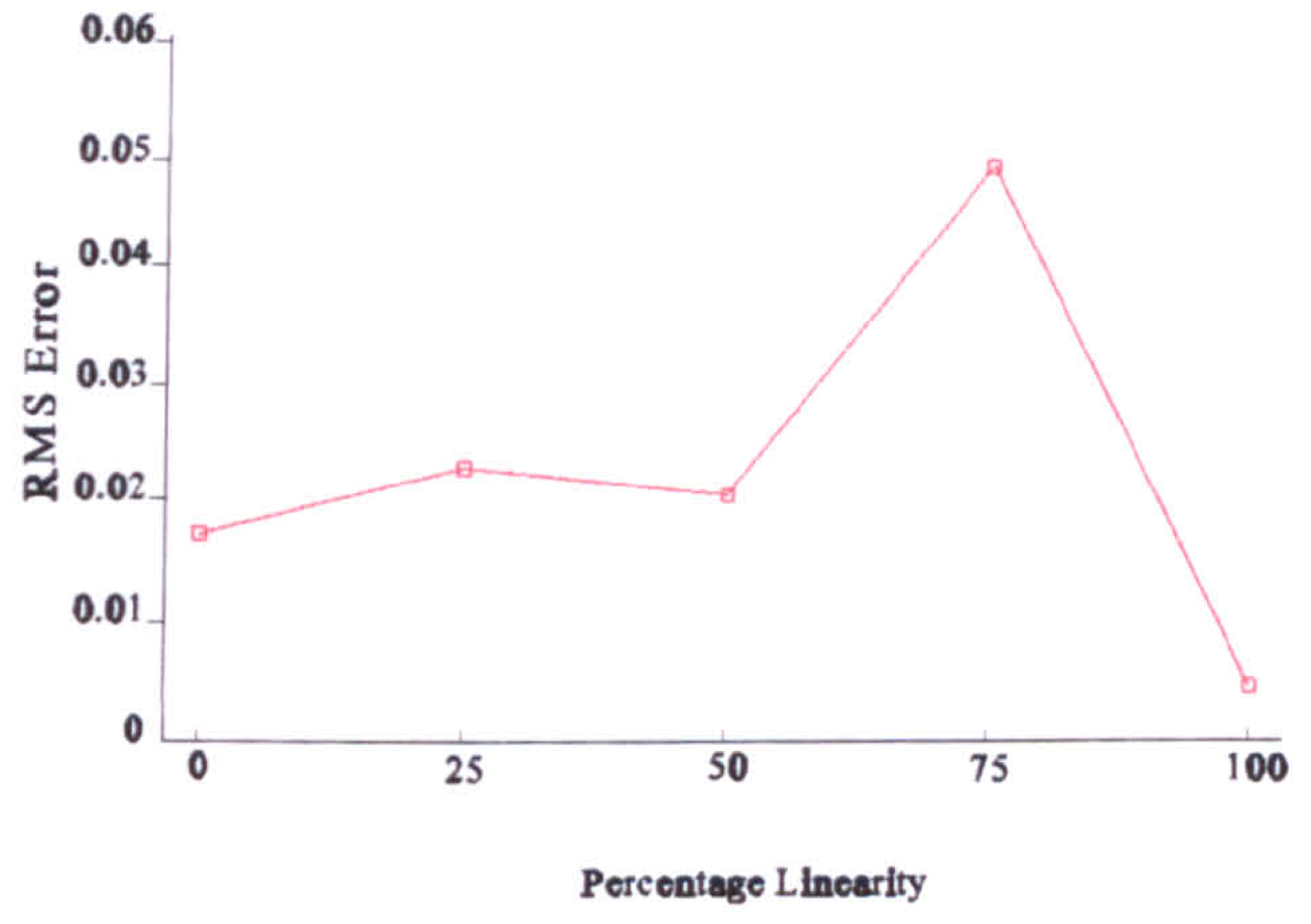


Fig 6.12: Linearity Test Results

The position of the linearity in the curve had an affect on the accuracy of the ANN. The data sets with the linear section at either extreme of the curve were not as well predicted as curves with a linear central section. The following diagrams, Figures 6.13 a and b, show the ANN output for two typical situations. The first figure depicts the prediction of a curve with a 25% linear section at the start, Curve B in Figure 6.11. The second figure consists of an identical sized linear section but in the middle of the curve, Curve C in the Figure 6.11. The linear extreme of the curve is not well predicted compared to the middle linear section.

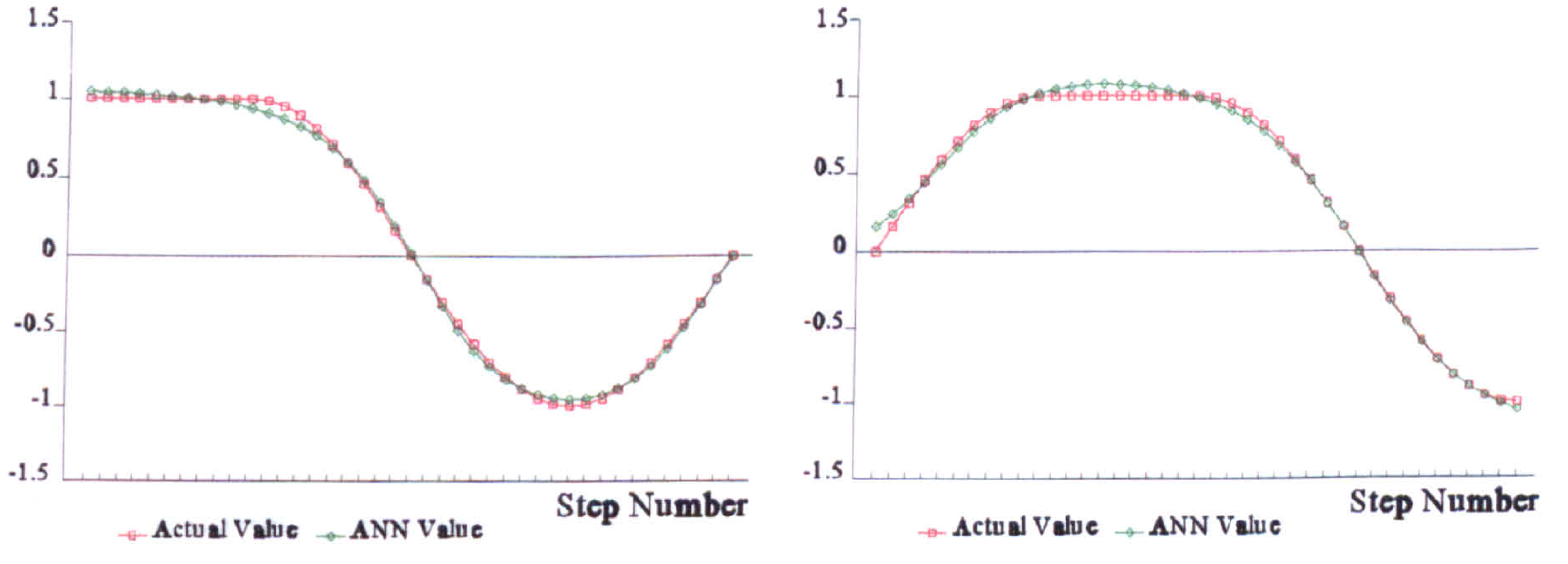


Fig 6.13 a & b: Sample Outputs from Linearity ANNs @ 25% Linearity

The best RMS error results were used to determine a mathematical relationship between percentage of linearity and neural network RMS error. Linear and a quadratic formula were developed and these are given below together with a graph of the functions.

$$y = 3.31235\times 10^{-4}x + 1.37767\times10^{-2}$$
$$y = 2.97144\times10^{-6}x^2 + 6.21327\times10^{-5}x + 1.66799\times10^{-2}$$

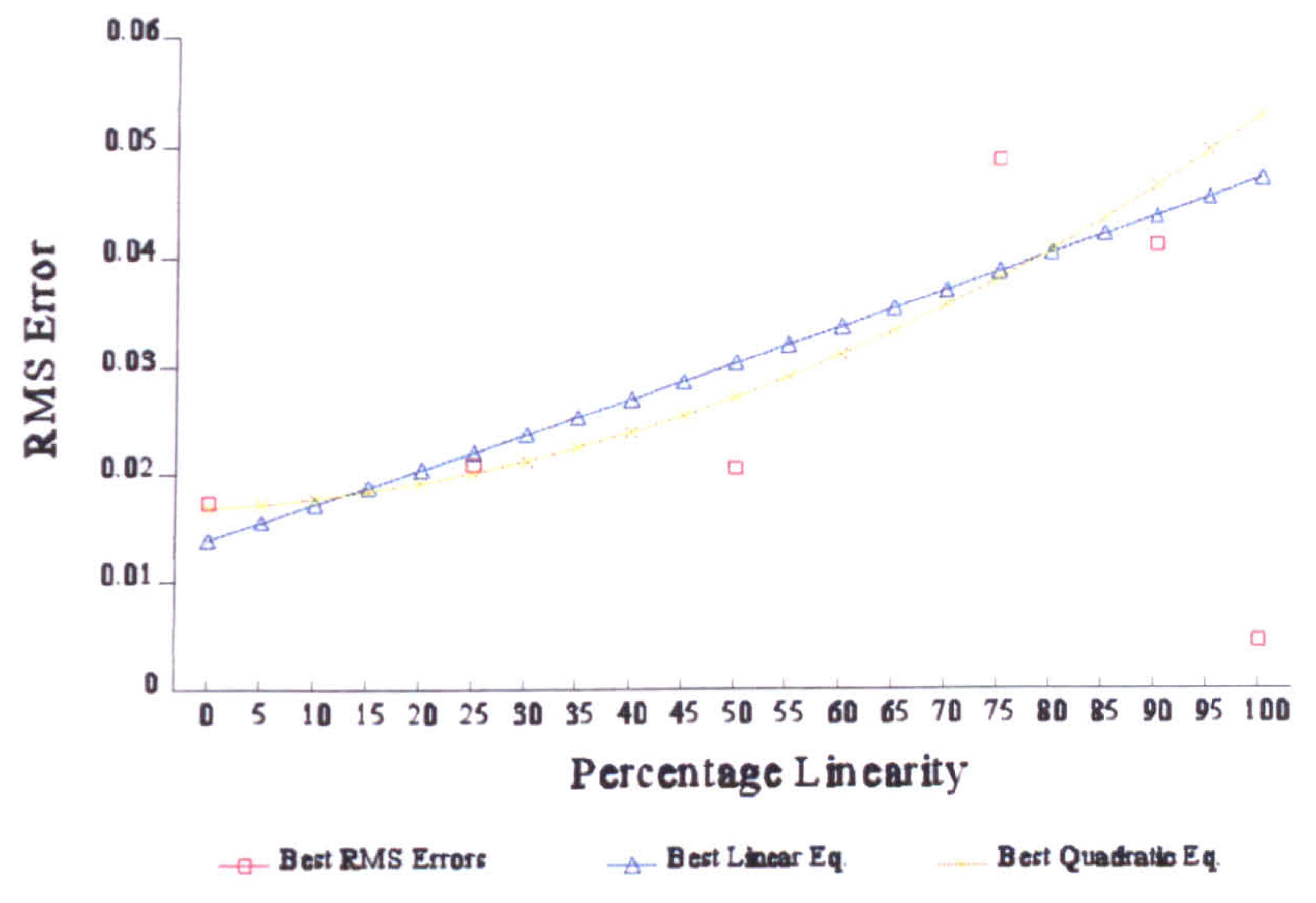


Fig 6.14: Graph of Equations of Linearity

The results show several interesting features. The straight line, 100% linearity, was the best predicted curve. This result is unexpected as the threshold function used was sinusoidal and would not be expected to model a pure linear function. To collaborate this result a series of ANNs were developed for 90% linearity.

% Linearity	Best RMS Error
90	0.0410

Table 6.5: Best Results for Linearity Tests

This result was added to the above graph. The result from 100% linearity was not considered in this graph. The revised figure is shown below in figure 6.15.

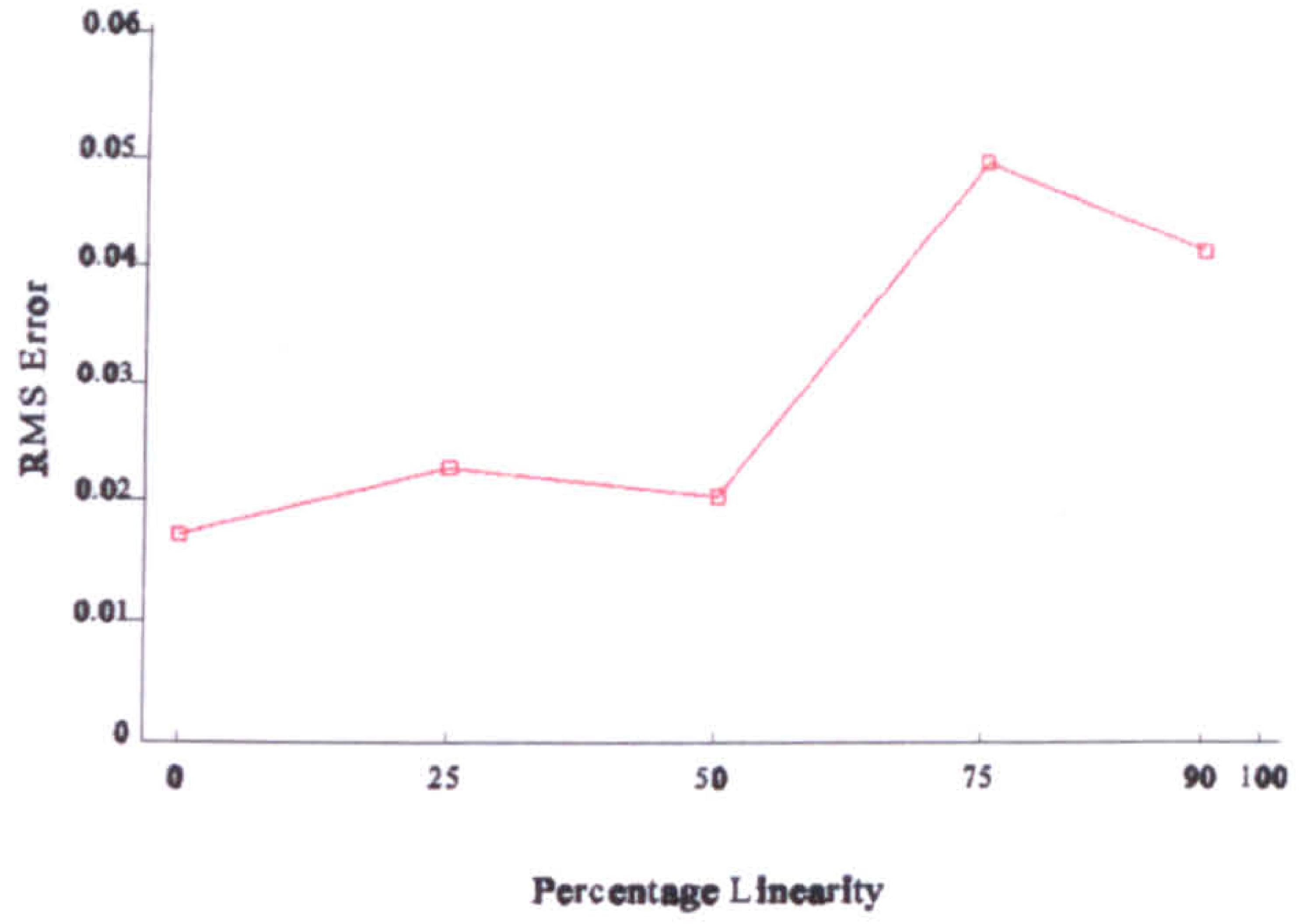


Fig 6.15: Refined Linearity Test Results

The revised equations for the relationship between linearity and ANN error are given below.

$$\mathit{LINEAR}\ \ y = 0.000322177\,x + 0.0138539$$
$$\mathit{QUADRATIC}\ \ y = 3.07169\times10^{-6}x^{2} + 0.0000439959x + 0.016855$$

The quadratic relationship will be used to determine the estimated ANN error for the linearity of a given PWR transient curve.

The relationships are developed on the best ANNs produced to date, however better networks exist in the solution space but were not found in training. Repeating the training for a number of initial conditions gives a guide to the percentage of linearity relationship but the developed formula is not necessarily optimum. Furthermore the linear sections are shown as horizontal but this too may have an effect on the prediction ability of the developed ANNs. It may be that angled linear parts may be predicted with a different accuracy to horizontal elements. The linear and sinusoidal sections were joined tangentially to avoid additional problems of angle of meeting but by considering non-horizontal lines would introduce the angle of the line as a component so the results would not be independent and reflect the effect of linearity.

A relationship has been produced for the linearity of a transient curve on ANN prediction accuracy. However, such an expression requires quantifying to establish the validity of the approach as the isolation of individual elements appears problematic. These observations will be further considered in the next section, which examines the effect of the angle of direction on ANN prediction.

6.3.3.2 Angle of direction as a Elementary Feature

The angle of direction is the second curve characteristic identified as influencing a predictive ANN. The early, exploratory work on the suitability of ANNs for prediction, reported in chapter five, identified several scenarios where the PWR transient curves included acute angles. These situations were usually the result of a large leak or the closing of a system valve. An example of such a condition is depicted in the following diagram, Figure 6.19. The reactor parameter, pressure in Node 3, shows a very steep drop followed by an equally rapid recovery. The output from an ANN, trained to predict this transient, is also shown in the figure. The ANN is not able to accurately predict the transient at this point, the resultant curve being a crude spline to the transient.

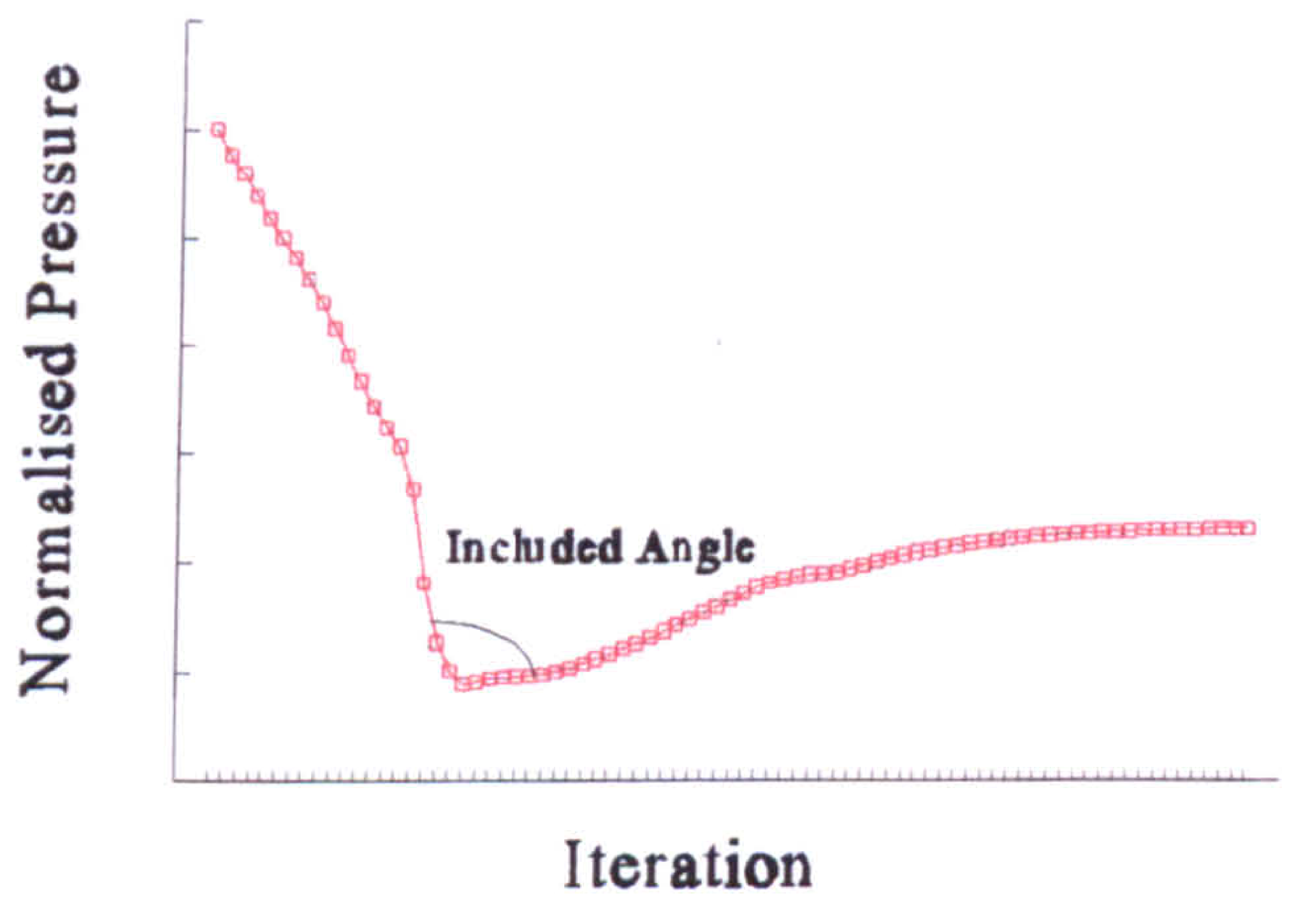


Fig 6.16: Transient with included angle

The included angle appears to be an important factor for predictive ANNs. To investigate this characteristic further a series of tests were created. The aim of these tests was to explore the effect of different included angles on predictive ANNs and, if possible, develop a relationship between included angle and neural network RMS error. The initial opinion was that a transient with a small number of included angles or composed of the better modelled included angles would be more accurately predicted by an ANN, with all other parameters being equivalent.

Data for the included angles was obtained by using the linear construction shown below in Figure 6.17. The arrangement crosses the time axis at three positions; 0, 0.5 and 1. The apexes, at 0.25 and 0.75, are scaled to equal ±1. The included angle is created by varying the 0.25 and 0.75 positions vertically, an apex closer to the time line resulting in a larger included angle.

Included Angle	Best RMS Error
10	0.064678
30	0.065376
50	0.065115
70	0.065009
90	0.063986
110	0.064143
130	0.065722
150	0.064241
170	0.065388

Table 6.6: Included Angle Test Results

The following diagram, Figure 6.18, shows the output from the best ANN in the above table. The diagram highlights both the tendency of ANNs to develop splines to included angles and the difficulty of an ANN to produce a linear output.

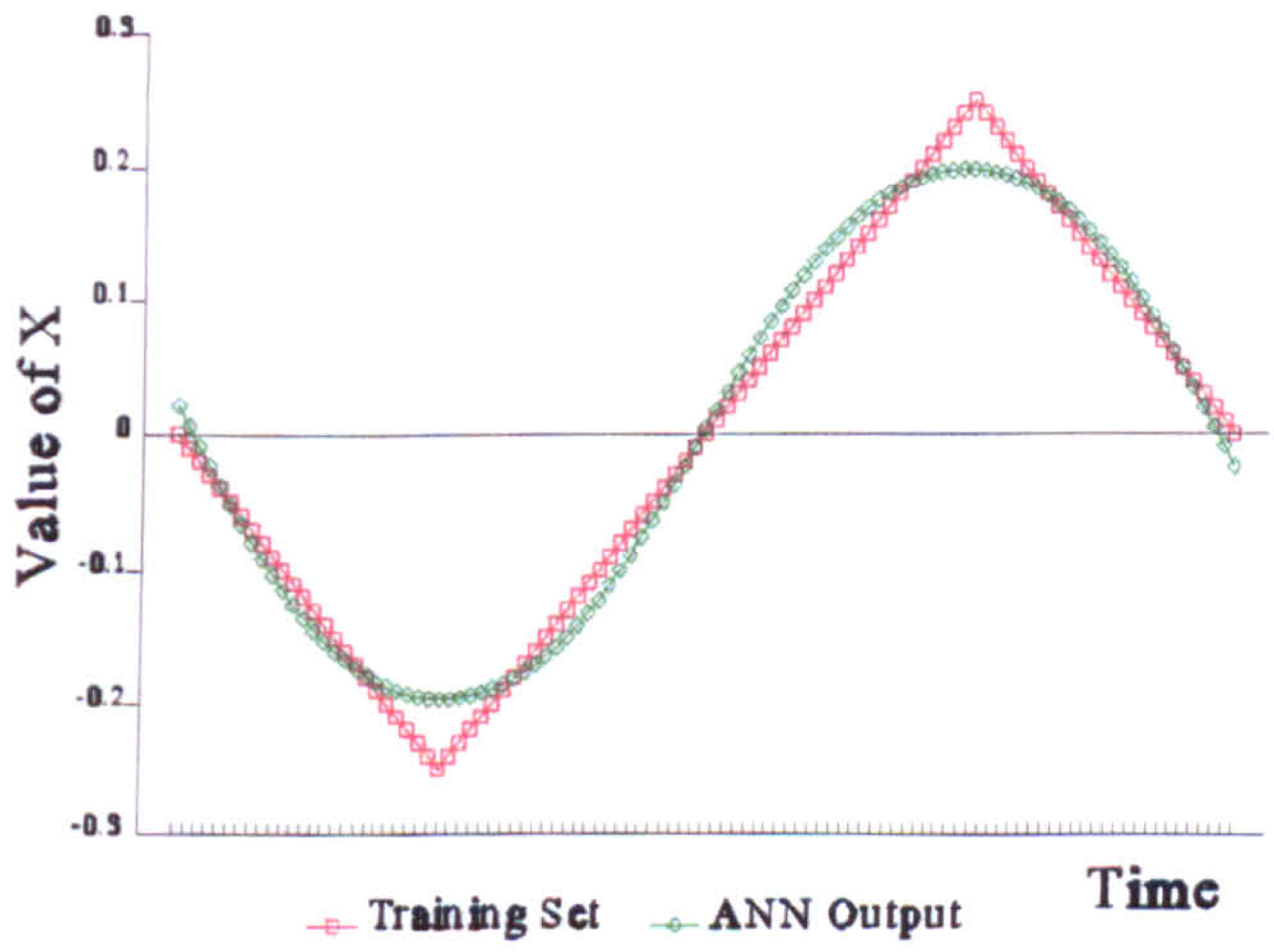


Fig 6.18: Output from Best ANN

The above values were used to determine the following mathematical relationship between included angle and ANN RMS error.

$$y = 6.5102\times10^{-2} - 4.3275\times10^{-6}x$$

Where: y = RMS error of neural network, x = Included angle

6.3.3.3 Discussion

The above sets of tests provide some interesting insights into the inner mechanisms of ANNs but the results obtained do not provide a conclusive argument for the validity of the approach. The main difficulty, which became increasingly apparent during the investigations, was in isolating each of the elementary features. Each test to investigate the effects of one feature always contained elements of the other features. Correspondingly the results obtained were not solely in terms of the characteristic under scrutiny. Consider the following two examples.

The first set of tests considered graphs composed of varying lengths of linear section and curves. The curve sections were all sinusoidal as an ANN can successfully model sine functions, especially with a sine threshold function. However, the joining of the sections was an area of potential confusion. Whilst the different parts always intersected tangentially at a turning point on the sine curve, there is still an inherent included angle as shown in the following diagram, Figure 6.19a. This angle was constant throughout the tests but still had an influence on the results. At the least the results obtained could possibly be considered only as a subset of all linearity tests with a large range of included angles, as depicted in Figure 6.19b.

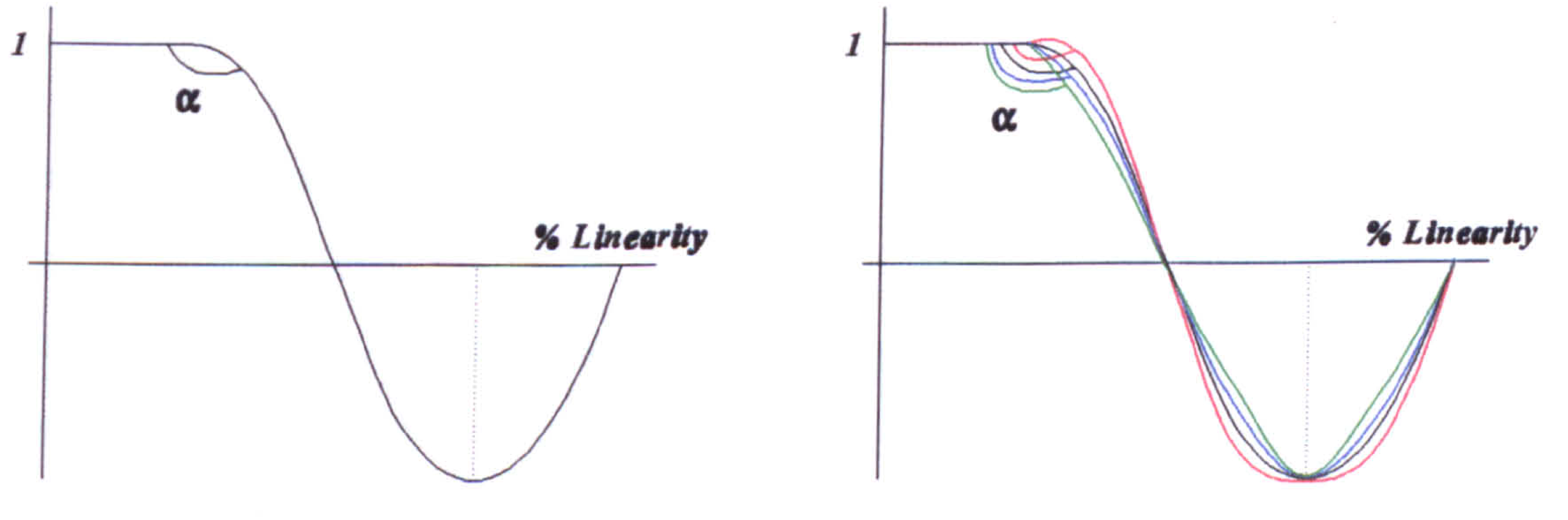


Fig 6.19 a & b: Diagrams of Included Angle in Linearity Tests

The second set of tests, investigating the included angle, were also dependent on more than one elementary feature. The shapes used for the tests were composed of straight lines meeting at the angle under test. The linear components of such arrangements are the elementary features investigated in the first tests of this section. The included angle tests could equally well have been performed with sinusoidal links between the angles, as shown as sigmoid.

These results also confirm the unsuitability of the classification approach to transient grouping discussed in section 6.3.1. The above pairs of curves would all have received different codings yet the results of ANN training show that the effect of proximity of curves is negligible.

6.3.3 Elementary Features of a Transient Curve

While the above results show that area between curves is not a contributing component for ascertaining combinations of transients for ANN prediction, characteristics such as linearity of the transient curves are important factors. Several other elementary aspects of curve construction, such as intersection information and gradient details, can also be identified as possible factors. A possible method of deciding an optimum combination of PWR transients may be to examine these characteristics and to develop a suitable relationship between them. If all the key variables for each transient under consideration could be expressed in terms of these basic properties perhaps by a formula, suitable combinations of transients for a pre-defined accuracy could be determined.

That is for the satisfactory combination of Transients 1 to n in one predictive ANN

$$\Sigma_{k=1}^{n}(V_k) \leq x$$

Where V_k = f(*Elements* of Transient K)

and x = Pre-defined accuracy of ANN

The work in the previous section *considered two combinations of curves. ANNs were* trained to predict two sinusoidal curves and a combination of linear and sinusoidal curves. The two sinusoids were more accurately predicted than the mixed combination. Furthermore, the linear element was found to be less accurately predicted than the sinusoid. The amount of linearity present in a transient may be a possible guide to ANN prediction accuracy.

A second feature that seems to have an affect on ANN prediction accuracy is a change of angle. The predictions in the previous chapter showed that drastic changes in a transient gradient cannot be accurately predicted by an ANN. The resultant spline often fails to predict the value of the PWR variable at the turning position of the curve.

Consequently a transient that contains a number of these features may not be satisfactorily grouped with other transient scenarios for accurate prediction. It is possible that such a fault condition could be successfully predicted by an ANN dedicated solely to the prediction of this transient.

Two further features that seem to have an effect on ANN prediction accuracy concern the nature of any intersections between the curves under consideration. The number of intersections is a important feature to consider as an ANN trained to predict different transients would perform better for curves that do not meet compared to curves that have frequent intersections. The length of the intersection is also a key consideration. An ANN would have great difficulty predicting two curves with common values for a large duration of the transient.

The final set of transient features to be considered for their impact on ANN prediction are the following.

1) Linearity
2) Angle of curves
3) Number of intersections
4) Length of intersections

Each of these properties will now be examined in greater depth in order to establish any possible relationships.

6.3.3.1 Linearity of Curve as an Elementary Feature

A series of tests were devised to explore the effects of linearity of transient on a predictive ANN. A set of test curves were created with a varying linear element. A group of ANNs were developed for each curve. The results from the best ANN in each case were then used to develop an expression for a possible relationship between linearity and prediction error.

The size of the linear section of the test curves ranged, in steps of 25%, from a pure curve, 0% linearity, to a straight line, 100% linearity. The non-linear sections were represented by a sinusoidal waves joined tangentially to the linear sections. A sinusoidal wave was adopted for this task because it can usually be successfully modelled by an ANN. The possible effects of position of the linear section in the curve were considered by repeating the tests for different combinations of linear and non-linear sections. The following diagrams, Figure 6.11, show the full set of curves used in this investigation.

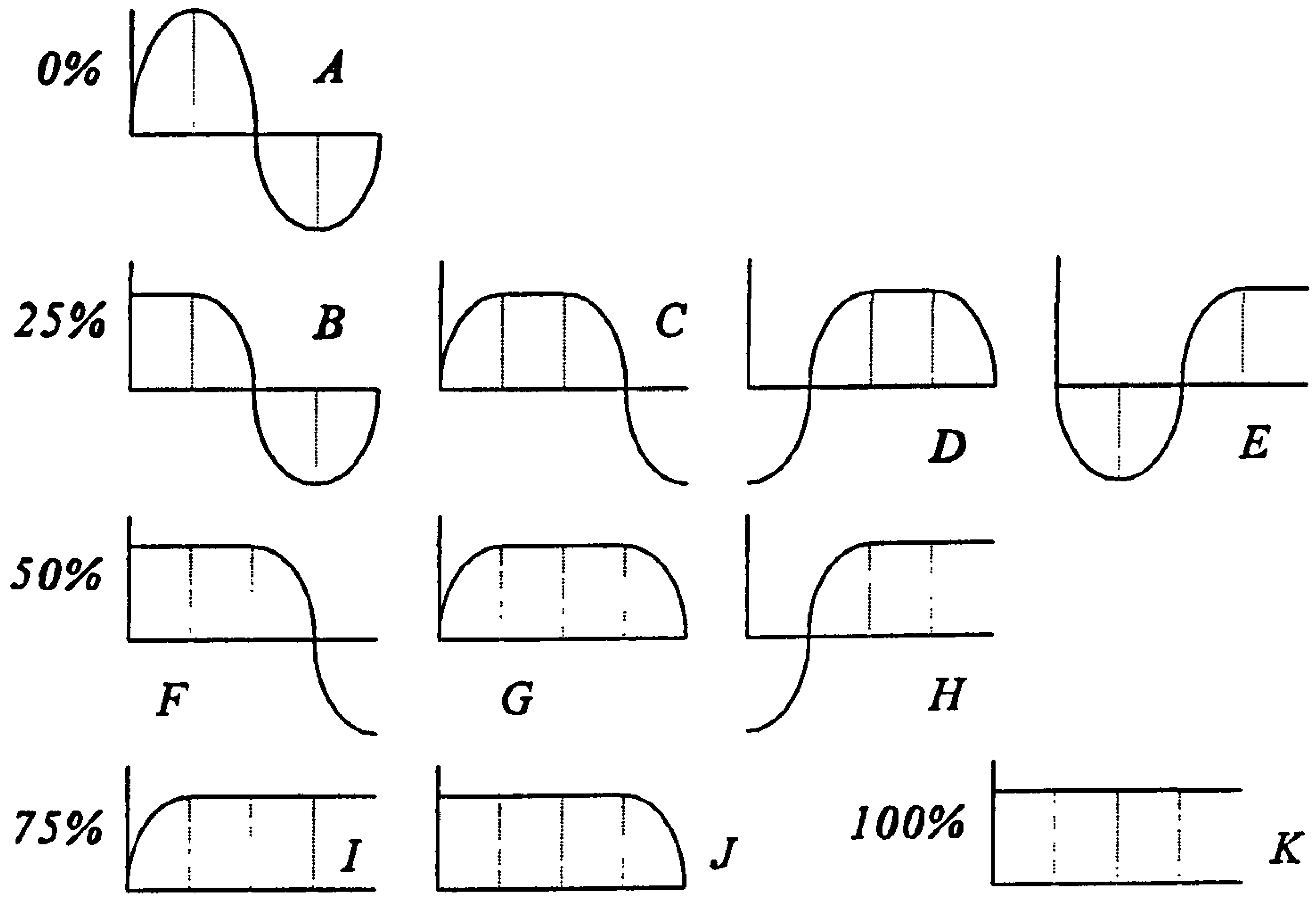


Fig 6.11: Set of Curves used for Linearity Tests

Each curve was represented by a set of forty-one coordinate pairs. These normalised values were divided into ANN training and test sets, in the ratio of approx. 2:1. To keep the models simple so that the effects of linearity could be observed the ANNs consisted of a single input and output nodes. A series of ANNs were developed for each curve with a selection of threshold functions and nodes in the single hidden layer. The best results obtained are given in the following table. The full details of the resulting networks are presented in Appendix J.

% Linearity	Best RMS Error
0	0.017273
25	0.022907
50	0.020590
75	0.048667
100	0.004451

Table 6.4: Best Results for Linearity Tests

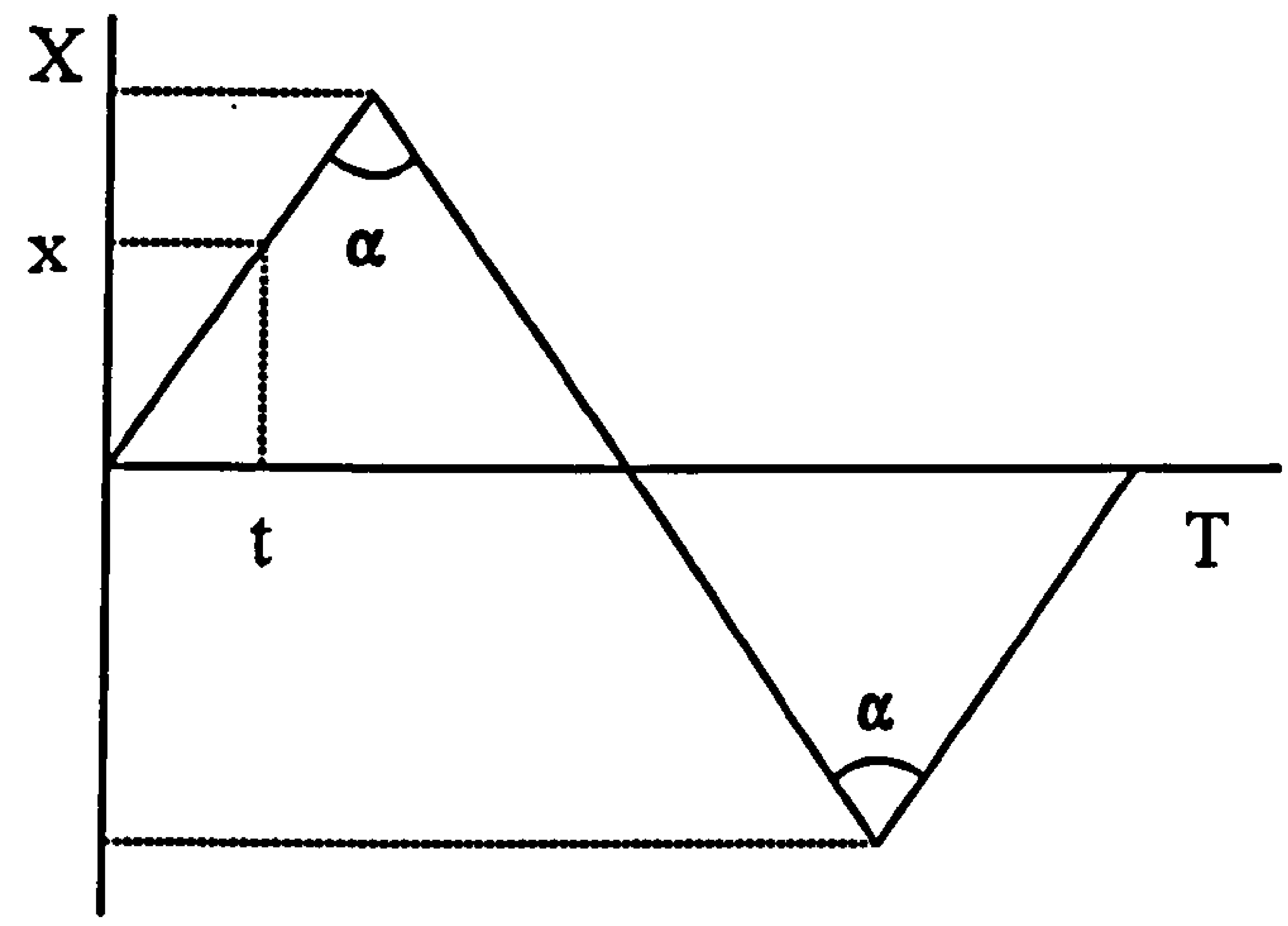


Fig 6.17: Diagram of Included Angle

The training and test sets were produced using one of the following four equations.

$$0 \le t \le 0.25 \quad x = -\left(\frac{t}{Tan\,\alpha}\right)$$

$$0.25 < t \le 0.5 \quad x = -\left(\frac{0.5 - t}{Tan\,\alpha}\right)$$

$$0.5 < t \le 0.75 \quad x = \frac{t - 0.5}{Tan\,\alpha}$$

$$0.75 < t \le 1.0 \quad x = \frac{1.0 - t}{Tan\,\alpha}$$

A computer program was written to automate the production of the training data. The time step was set at 0.01 giving 101 data points which were divided into training and test sets in the approx ratio of 2:1 respectively. The size of the included angle was varied between 10 to 170 degrees, in 20 degree steps. A range of ANNs were developed for each size of included angle. The backpropagation algorithm was used and ANNs were trained for 80,000 cycles with testing for every 100 cycles of the last 20,000, the best ANN being saved. The best RMS error results are given below the full set of results are given in Appendix K.

in Figure 6.20. However, the results obtained in this case could not be directly comparable with linear link results as the included angle is not independent of the links.

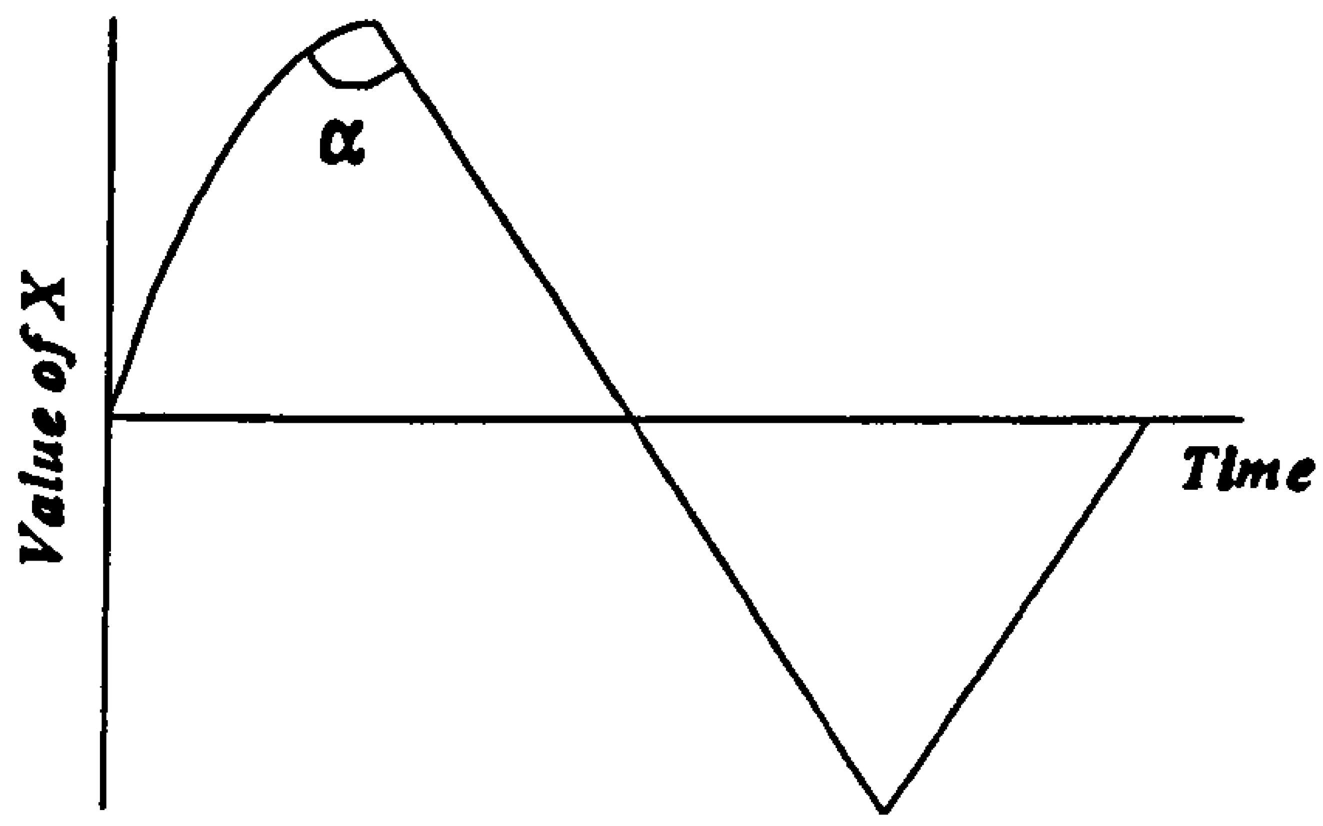


Fig 6.20: Diagram of included angle with linear and sine links

The next series of tests would consider the effect of intersections on the RMS error of a predictive ANN. It was intended to develop a set of ANNs with a number of intersections. The above arguments are exacerbated in this case as both the nature of the curves and the angle of the intersections both have a bearing on the ANN predictability. Similarly the investigation for length of intersection is dependent of several parameters, namely nature of incident curves, angle of intersection and position of intersection in curve.

In each case the variable under examination is not independent of the remaining variables. A relationship would therefore be impossible to determine purely in terms of a single variable. Furthermore the tests carried out have proved to be members to only a small subset of possible combinations of linearity or included angles. The full sets would require far greater testing and development, assuming they could all be identified. Finally the ANNs developed may not be the optimal solution to each case, a different starting point may produce a better final ANN. Using these results to construct mathematical relationships is therefore of questionable validity.

The above problems were together considered sufficient major to curtail further investigations on this approach. The number of transients an ANN can accurately predict is still a crucial question for the development of the advisory system. A new method of addressing this question will need to be investigated.

6.4 Direct Equivalent Network Models

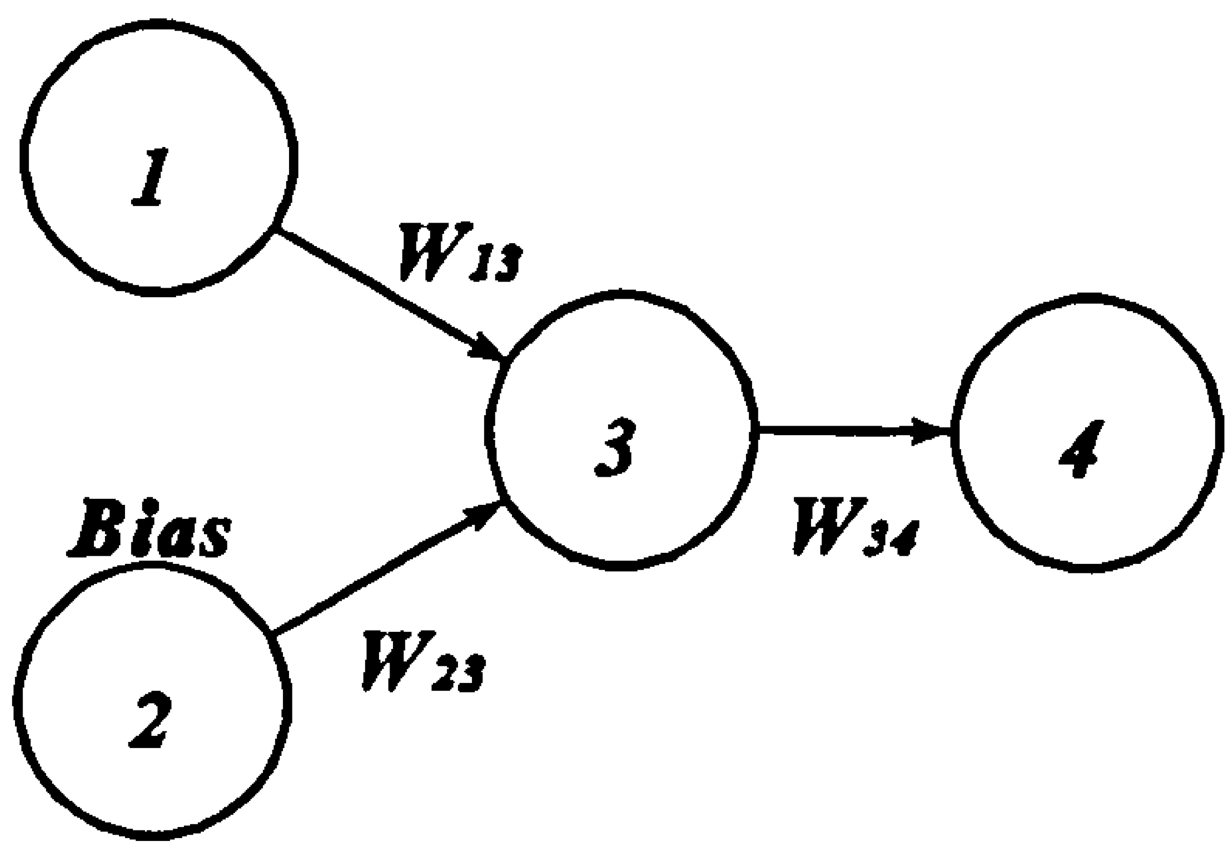
This section discusses a second method of addressing the question of the number of transients that can be successively modelled by an ANN. The approach adopted is very different to that of the previous section in that the proposed network can model a large number of transients compared to the discrete number considered previously. This second method is introduced by examining the method an ANN models simple curves and PWR transients. This work is then enlarged with the development of a basic ANN unit which directly models the important non-linearities of the equation for energy conservation in the PWR primary system. This simple ANN based model is refined, tested and compared with a simulator modelling and predicting the reactor variables. A model of the entire PWR primary circuit is then constructed from a number of these simple, direct equivalent units. This model is further refined and compared with a full computer simulation model of the PWR.

6.4.1 Initial Investigations

This section of work explores the process used by an ANN to model a PWR transient. Very simple examples will initially be considered and by the gradual introduction of complexity the simpler PWR transients will be considered. The transients will be predicted by considering time as the input, however the method is equally valid for other plant variables.

All the models in this section were produced on a spreadsheet program with the cells representing the various components of the ANN. The input nodes were described by a single cell containing the value of that input. The links between nodes were modelled by single cells holding the value of the weighting. The nodes in the hidden layer were represented by a set of cells. The first cell of the set contained the summation of the product of node inputs and weights between nodes. The second cell held the node output, the threshold function acting on the total input. This method permitted value changes to be rapidly performed and results quickly calculated.

The simplest PWR situation to represent is the constant, linear, steady state condition. This situation can be modelled by an ANN consisting of two inputs, a single hidden node and one output, as depicted in the following diagram, Figure 6.21. One of the inputs, Node 2, is the Bias for the ANN while the first node is a reactor related variable. For these initial investigations time is used for this input. The figures given for the weightings between nodes are only simple values for presenting the ideas.



Link	Weight
W_{13}	0
W_{23}	x
W_{34}	1

Fig 6.21: Steady State ANN Structure

A simple constant, linear output, y, is independent of time so the weighting of the time input, w_{13}, is zero. The output y therefore depends solely upon the weighting of the bias, w_{23}, and the transfer function, f, of the output node. This can be represented in the following equations and graph. The output can take both positive and negative values, again determined by the weighting w_{23}. The weighting between the hidden node and the output, w_{34}, performs as a scaling factor for the output from the hidden node. For this ANN the value of w_{34} is fixed at one, no scaling.

$$y = W_{ao}.f(t.W_{1a} + W_{2a})$$
$$W_{1a} = 0, \quad W_{ao} = 1$$
$$\therefore y = f(W_{2a})$$

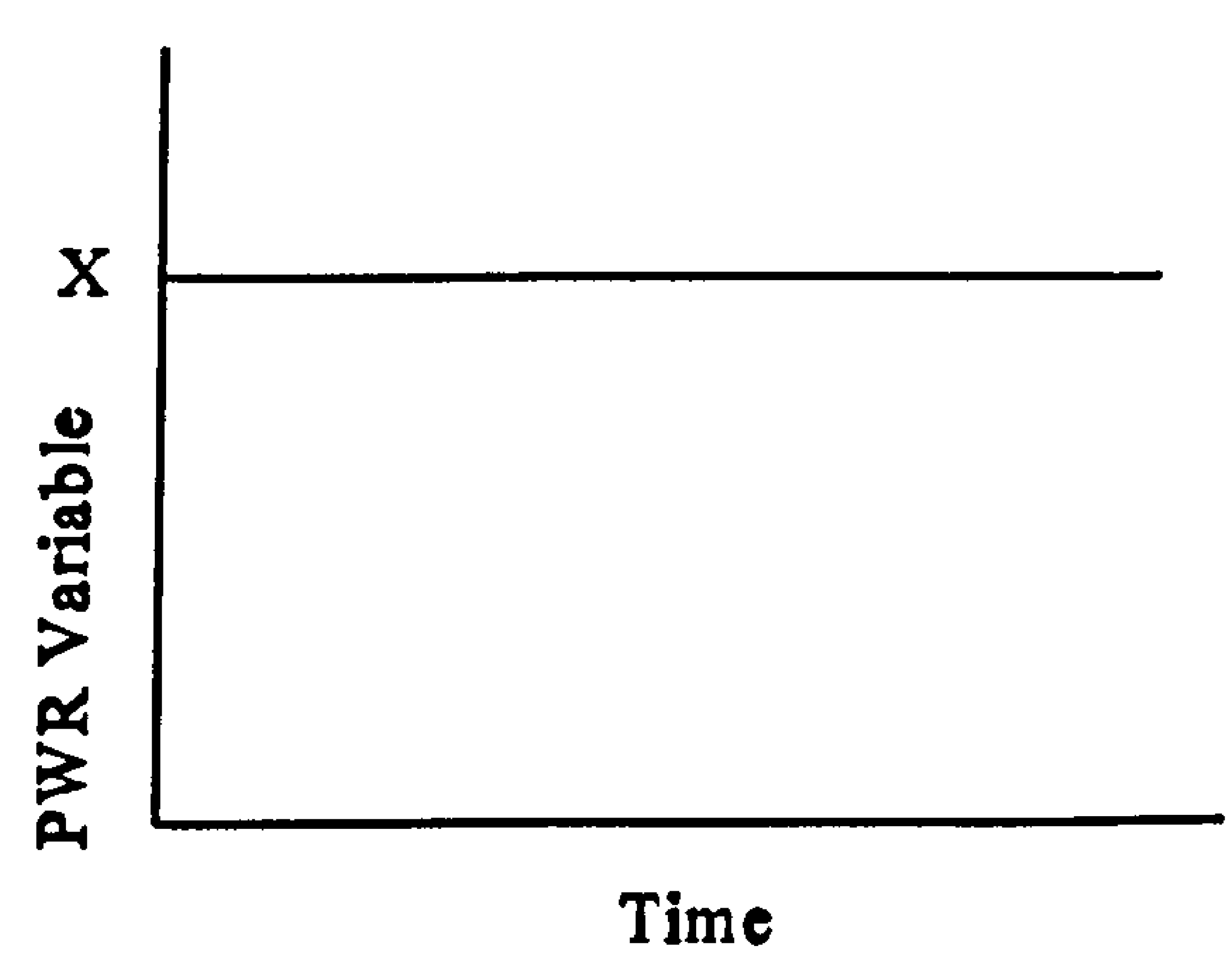


Fig 6.22: Graph of Linear Output

The ANN output can be given a gradient by the introduction of a non-zero value to W_{13}.

The slope of the output is determined by the value of W_{34}, again the value can take both positive and negative forms. The following graph, Figure 6.23, shows the effect of varying the value of W_{1o} for a range of values.

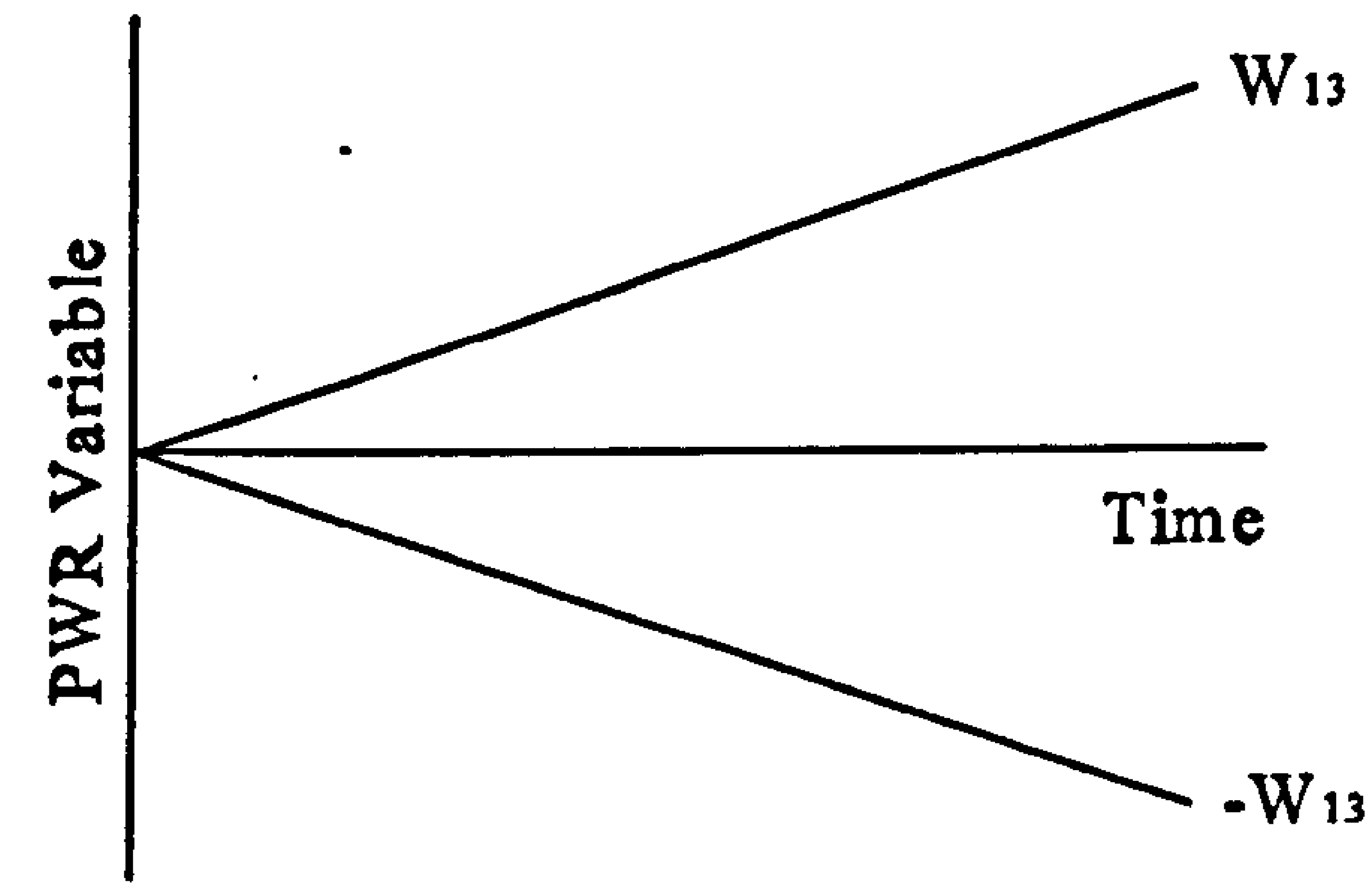


Fig 6.23: Graph of Gradient Output

The transfer function of the hidden node also has an effect on the final curve. A linear transfer function produces a straight line, while using a sigmoidal function results in a curved output. A comparison of the two transfer functions, for the same weight of W_{13}, are shown below in Figure 6.24.

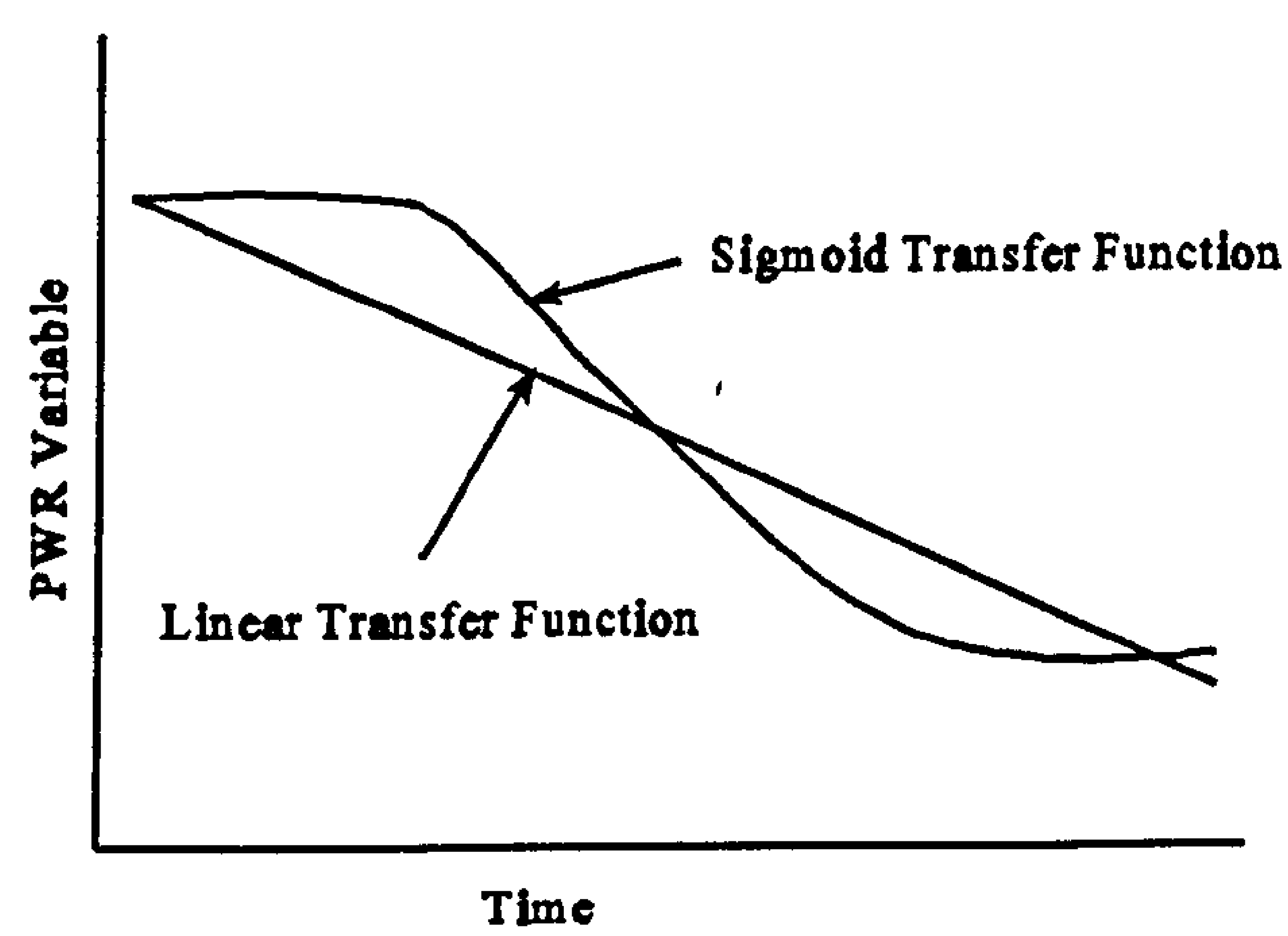
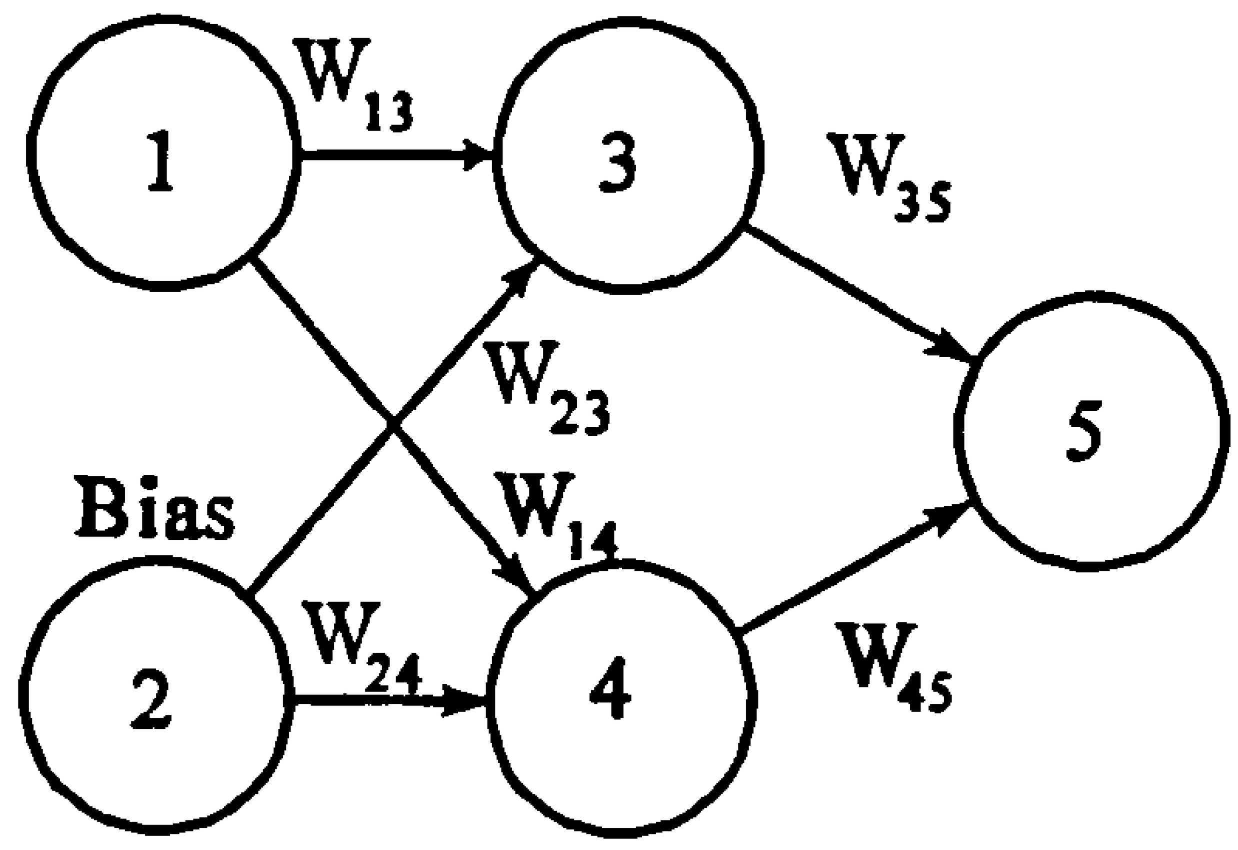


Fig 6.24: Comparison of Transfer Functions

The idea can be further refined by the addition of a second node to the hidden layer. This extra node produces further damping in the output. Node A is the linear element of the output and Node B produces the damped component. The size of the damping is controlled by the weights between Node B and the output, a greater weighting producing a more dramatic damping. A sigmoid threshold function was used for the nodes in this model. The following diagram, Figure 6.25, shows the arrangement of the nodes and the graph in Figure 6.26 shows some results for a range of W_{45}, the weighting between Node B and the output node.



Link	Weight
W_{13}	0.001
W_{14}	-0.1
W_{23}	0.9
W_{24}	-0.1
W_{35}	0.9
W_{45}	0.1

Fig 6.25: Restricted Flow ANN Structure

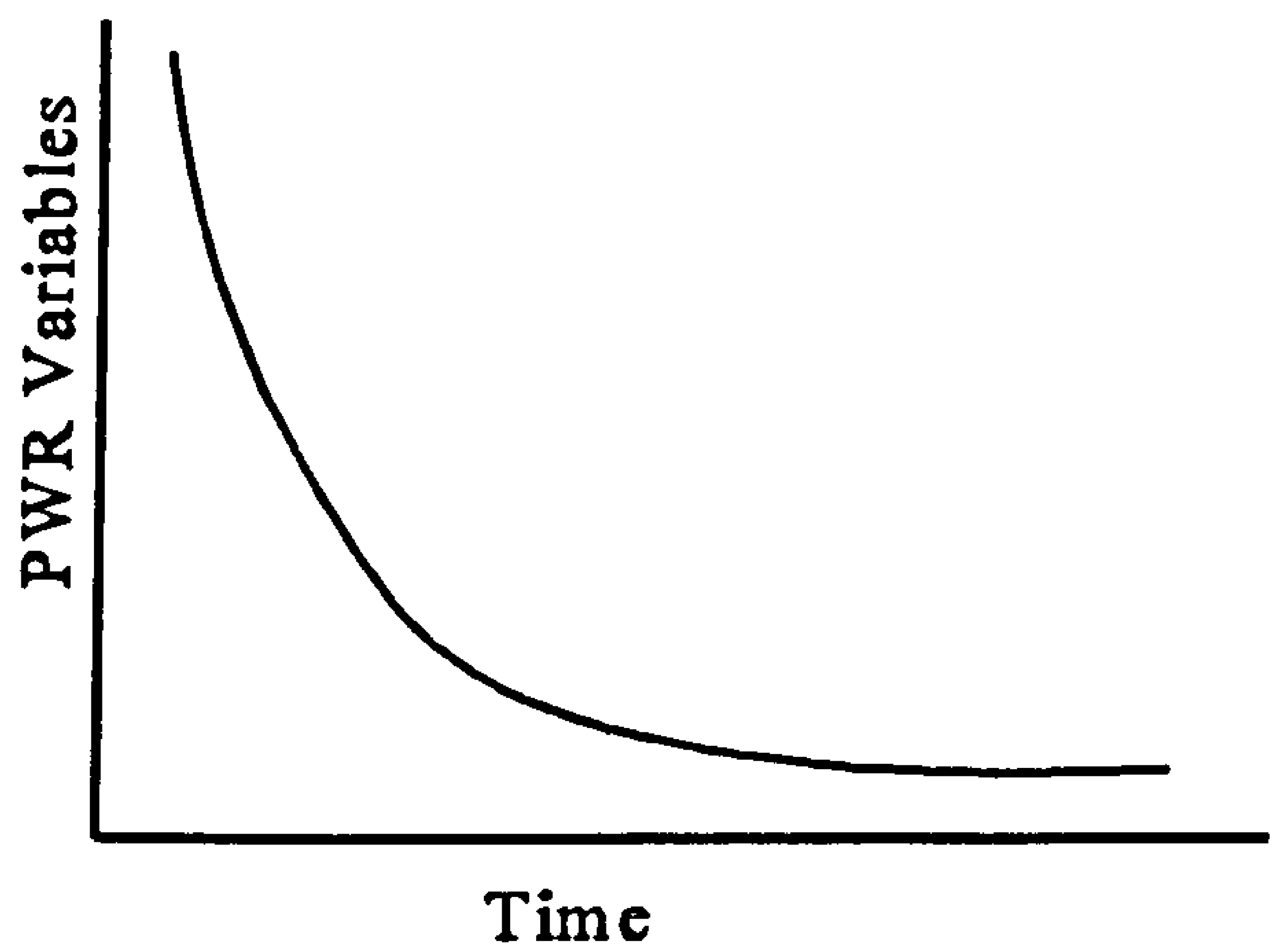
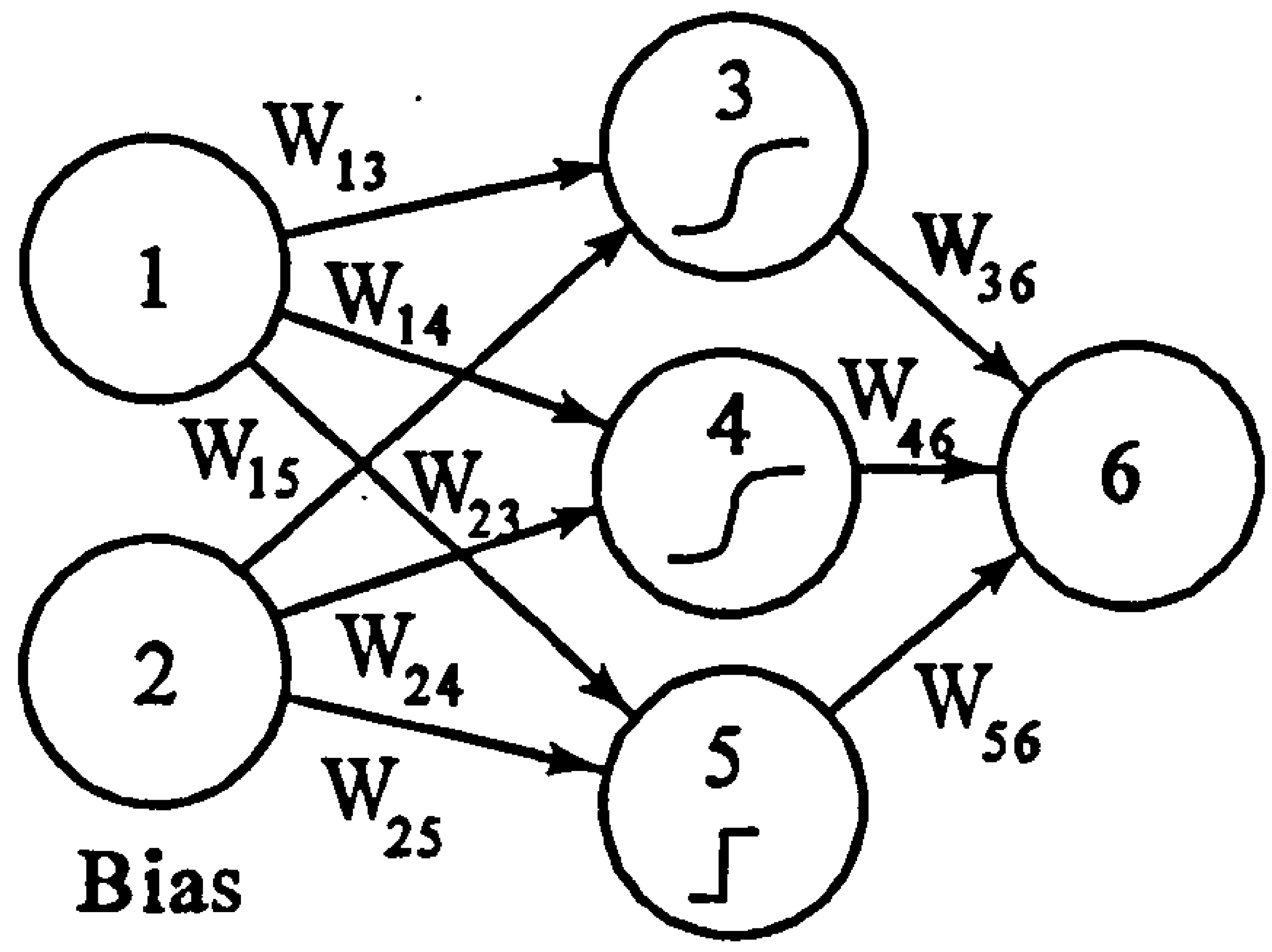


Fig 6.26: Sample results from Restricted flow ANN

These ANNs can be further refined to model the rapid emptying of a chamber, caused by the opening of the emergency cooler valve for example. This feature is produced by the addition of a third node to the hidden layer. A time for the valve to be opened is pre-defined, at t say, before the ANN is constructed. The weighting from node 1, the time node, is the reciprocal of this value, ie 1/t. This new hidden node has a hard threshold function with the following outputs

$$Output = \begin{cases} 0 & if\ input \leq 0 \\ 1 & if\ input > 0 \end{cases}$$

The output from this node is zero until time t, after which the node output becomes 1 and the ANN has an additional damping element on the output, determined by the weight w_{36}. This ANN is depicted in the following diagram, Figure 6.27.



Link	Weight
W_{13}	0.001
W_{14}	-0.1
W_{15}	1/t
W_{23}	0.9
W_{24}	-0.1
W_{25}	-1
W_{36}	0.9
W_{46}	0.1
W_{56}	-0.02

Fig 6.27: Valve Model ANN Structure

In operation this ANN performs identically to the previous network until time t when the valve opens, the output from node C becomes positive, the chamber being modelled rapidly empties and the transient is then modelled for a lower value. The following graph, Figure 6.28, shows the results of a range of valve opening times.

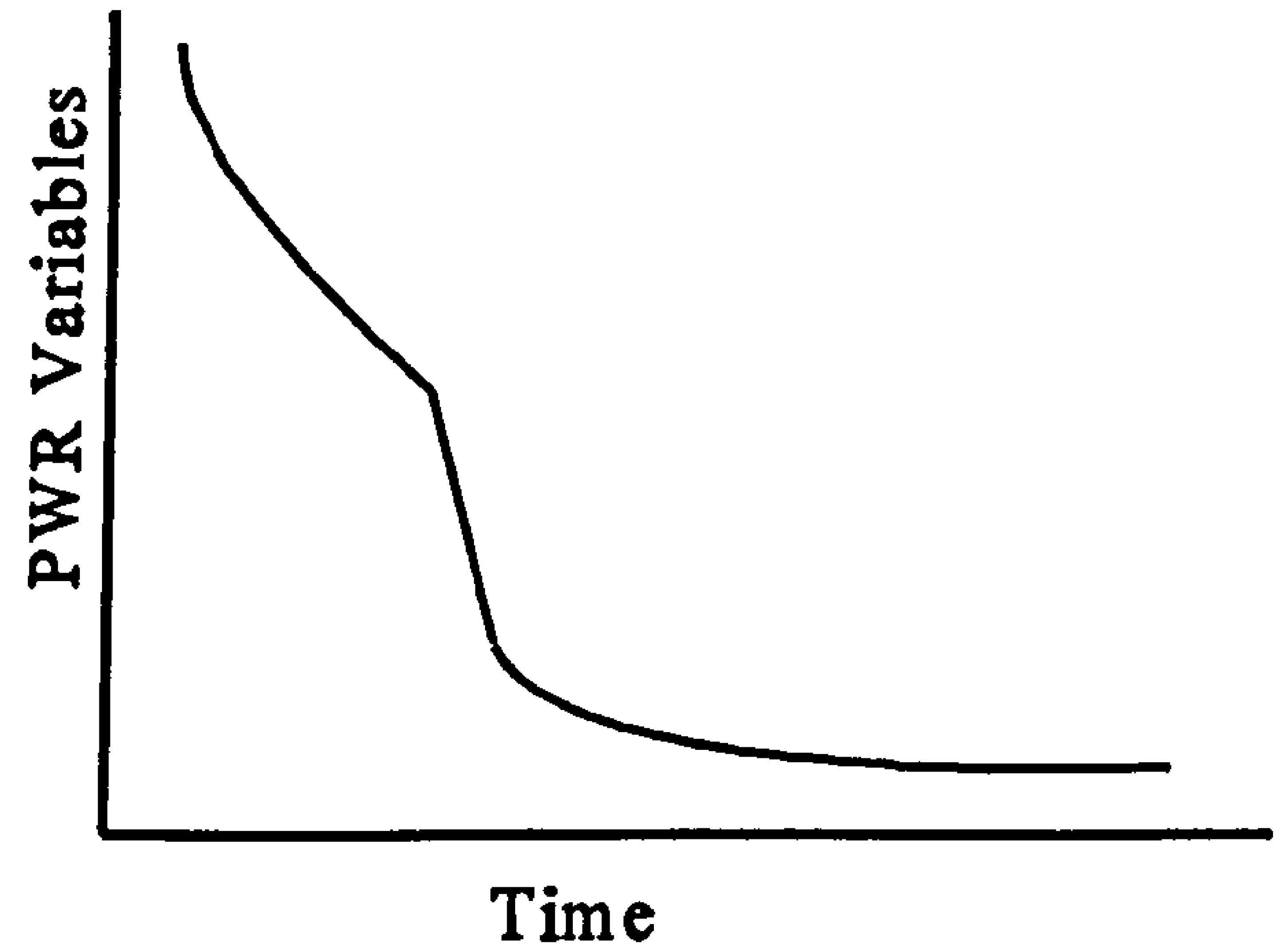
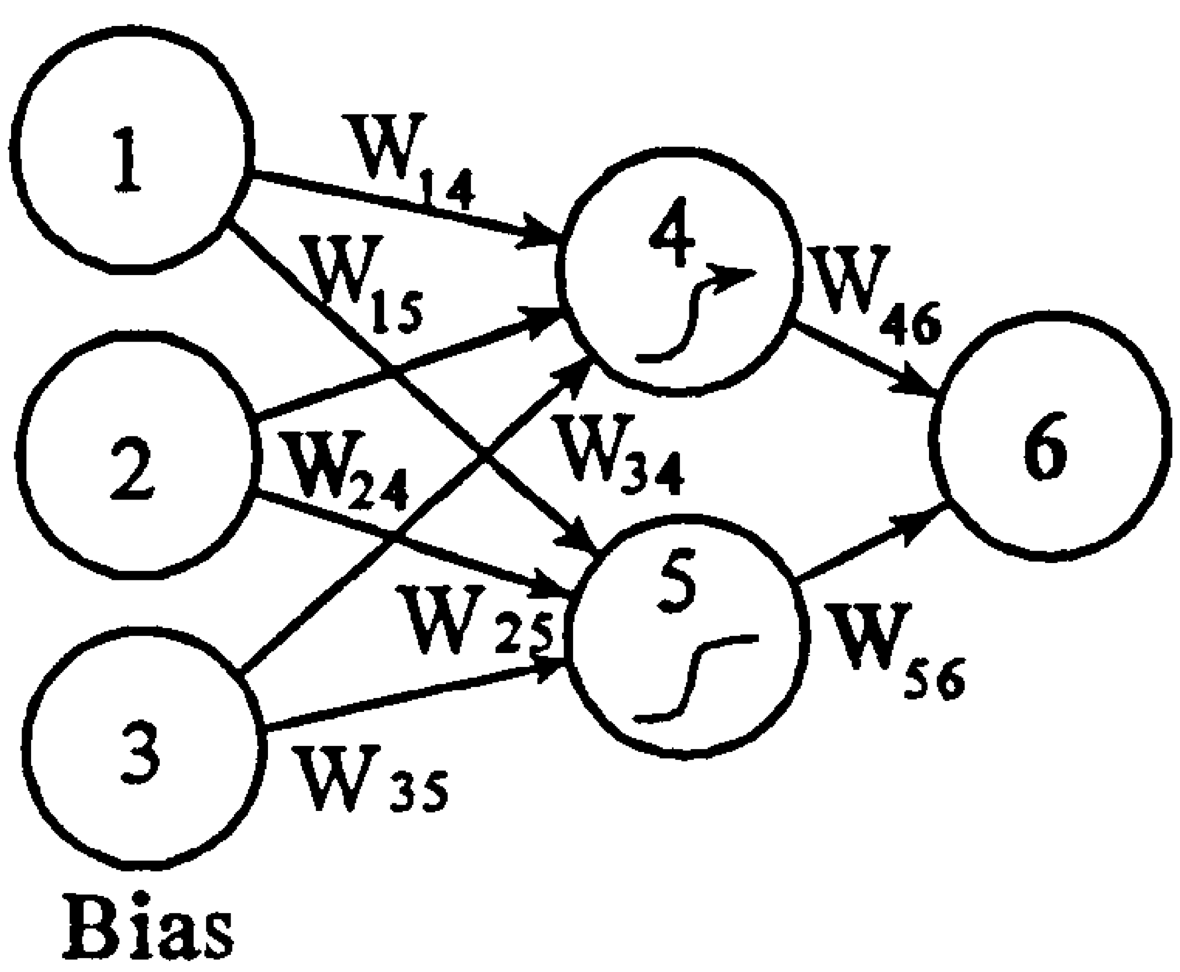


Fig 6.28: Graph of valve opening times

The final variation of this system is to consider various sizes of the leak. This feature is modelled by the addition of a third ANN input to represent leak size. The new node is an extra damped system with an input that reflects the size of the leak with an input of zero for no link. The size of this input affects the degree of additional damping introduced into the network and so creating a range of outputs from the ANN. The developed ANN is applicable for a range of scenarios as compared to the previous models which were all transient specific. The final network is represented in Figure 6.29, below.



Link	Weight
W_{14}	0.001
W_{15}	-0.1
W_{24}	0.0001
W_{25}	-0.9
W_{34}	0.9
W_{35}	-0.1
W_{46}	0.9
W_{56}	0.1

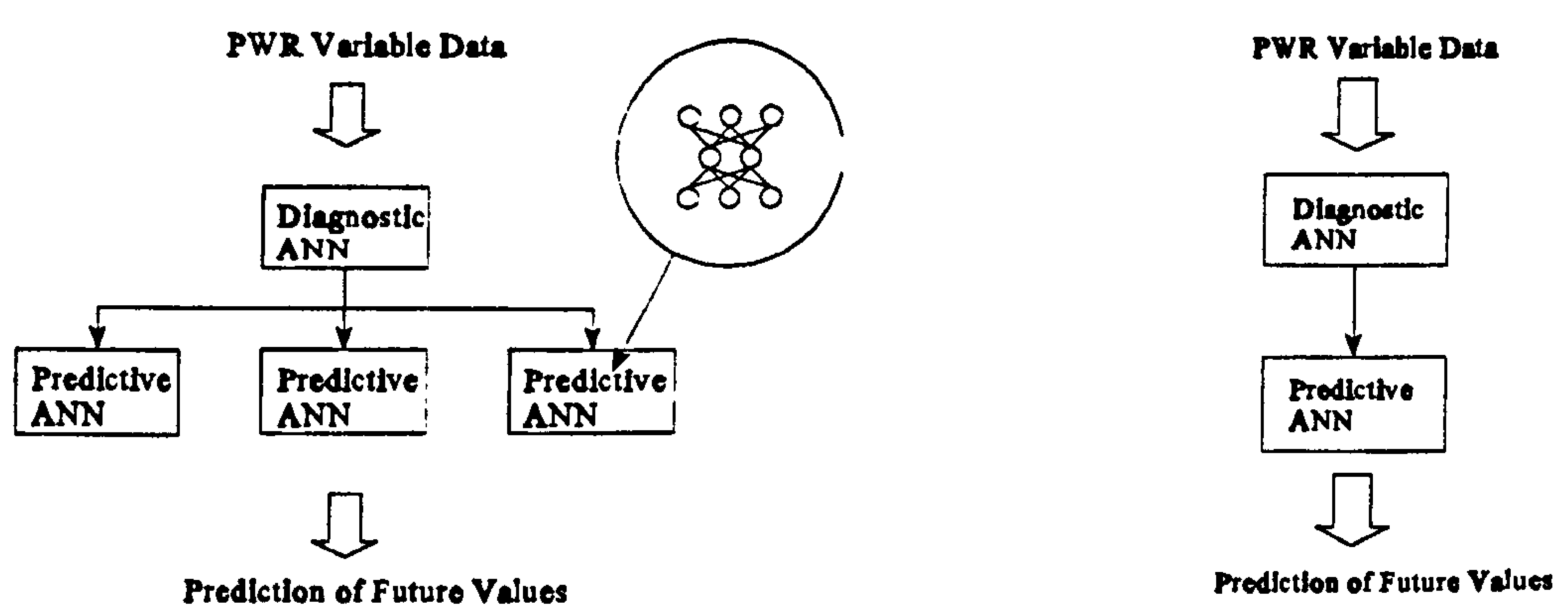
Fig 6.29: Leak Size Model ANN Structure

Even with the rudimentary ANNs so far developed, the basic decay curves of some of the PWR variables have been successively modelled for simple transients conditions. The networks developed above have been for a very limited range of circumstances. Variables such as the damping rates and valve opening times are all pre-defined thus limiting more complex and realistic applications. The addition of extra nodes, to both the input and hidden layers, introduces more complexity into the ANN system. The investigations reported so far in this section have shown that basic PWR transient characteristics can be modelled by ANNs. The networks developed have all been hand designed, with no data sets or training cycles. If the above ideas were to be developed further then a training process could increase the accuracy of the predictions by modifying the inter-nodal weights, as the initial, report weightings were all defined by hand. The severe disadvantage of these networks is their limited range of applicability. The next section considers an ANN model that is able to model a greater range of transient scenarios.

6.4.2 One Compartment Model

The work reported in this section investigates a different approach to the question of the number of transients an ANN can model and predict. Section 6.3 explored methods of determining the number of transients that could be modelled by a single ANN. An alternative approach would be to develop an ANN that could predict a large number of transient scenarios. A method of achieving this aim would be to develop an ANN based simulation model of the PWR primary circuit. A PWR model developed with ANNs would possess several advantages over an equation based simulator. Firstly, the operation of a slow empirical code could be enhanced by ANNs replacing some, or all, of the computationally intensive elements of the code. In the envisaged PWR operators advisor the prediction component would be required to operate well in excess of real time to enable a rapid, usable view of future reactor state. Secondly, the ANN simulator would be able to model plant variables not directly measurable, core temperature for example, and so enhance the model.

This approach to modelling and predicting PWR transients simplifies the original advisors system layout. The previously envisaged scheme included a number of ANNs in the prediction layer, each designed for an optimum set of transients scenarios. The new system retains the diagnostic layer but the prediction element now simply consists of an ANN based simulator. A second ANN simulator model could also be included to model different reactor variables, node temperature and pressure for example. A comparison of the two hierarchical systems are shown in the following figures. Figure 6.30a depicts the original system layout with a number of optimised predictive ANNs in the lower level of the hierarchy. Figure 6.30b shows the new system with a reduced number of ANN simulators in the predictive layer.



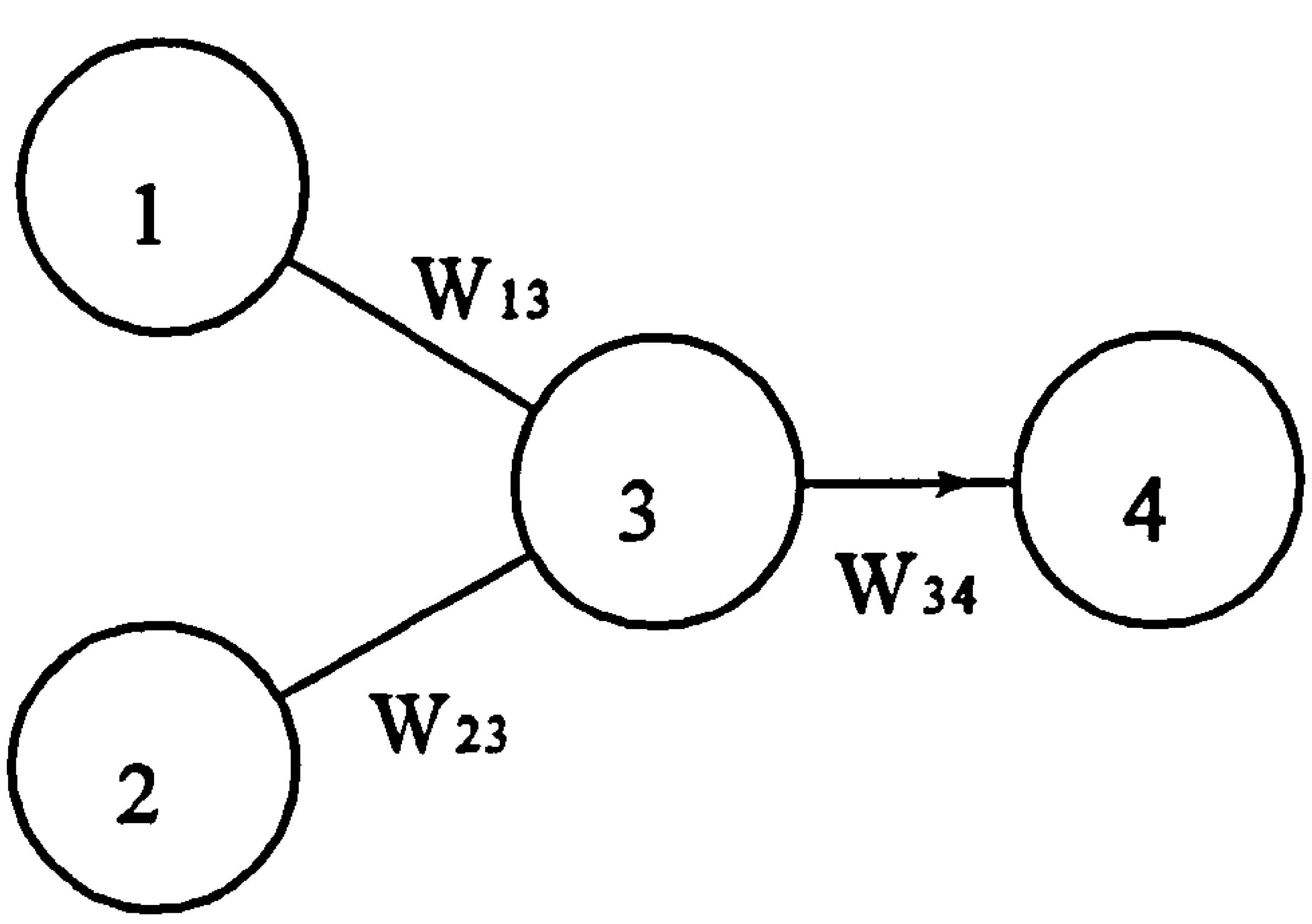
Figs 6.30a & b: Hierarchical Systems

The remainder of this section is as follows. Some initial work on representing mathematical functions with ANNs is first described. An ANN based model of a simplified reactor primary circuit is then developed using these functions. This simple model allows a wide range of scenarios to be explored and tested. The idea is then refined to include standard ANN threshold functions in the model. Lastly the technique is expanded into a model of the primary circuit. This model is refined, tested and compared to a computer simulation program of the system.

A key element to the success of an ANN based simulator is the ability of an ANN to depict basic mathematical functions. Although the final ANN will probably be a complicated mapping of many dimensions the elementary functions will still be represented. An insight into the internal structure of such ANNs could provide benefits when the more complicated models are under consideration.

6.4.2.1 Addition and Subtraction of two numbers with an ANN

The addition of two numbers is simply modelled by an ANN with two inputs, one for each number, and a single output for the summation. The weights between the nodes dictates the multiples of each input in the final total. For simple addition these weights are all given a value of one. The threshold function for the output node can be used to include further refinements, such as multiples, but for standard addition a linear threshold is used. This arrangement is shown below, in Figure 6.31.



Where

W_{xy} = Weight between Nodes X & Y

X_a = Input A

I_x = Input to Node X

O_x = Output to Node X

Fig 6.31: ANN Structure for Addition

Consider Node 3

The input to Node 3, $I_3 = X_1 . W_{13} + X_2 . W_{23}$

The output form Node 3, $O_3 = f_3 (X_1 . W_{13} + X_2 . W_{23})$

If the threshold function f_3 is linear function ie $f_3(x) = x$

$$O_3 = X_1 . W_{13} + X_2 . W_{23}$$

The input to Node 4, $I_4 = W_{34}(X_1W_{13} + X_2W_{23})$

The output from Node 4, $O_4 = f_4(W_{34}(X_1W_{13} + X_2W_{23}))$

If Node 4 has a linear threshold

$$O_4 = W_{34}(X_1W_{13} + X_2W_{23})$$

The required output from Node 4 is in terms of $X_1 + X_2$. If $W_{13} = W_{23} = W_{23} = 0$

then
$$O_4 = X_1 + X_2$$

<u>Addition of two numbers is possible with an ANN</u>

The values of the weights can be changed to produce more complicated functions. For example, if $W_{13} = 3$, $W_{23} = 1$, $W_{23} = 2$,

then
$$O_4 = 6X_1 + 2X_2$$

Finally if $W_{23} < 1$ ie a negative value

then
$$O_4 = X_1 - X_2$$

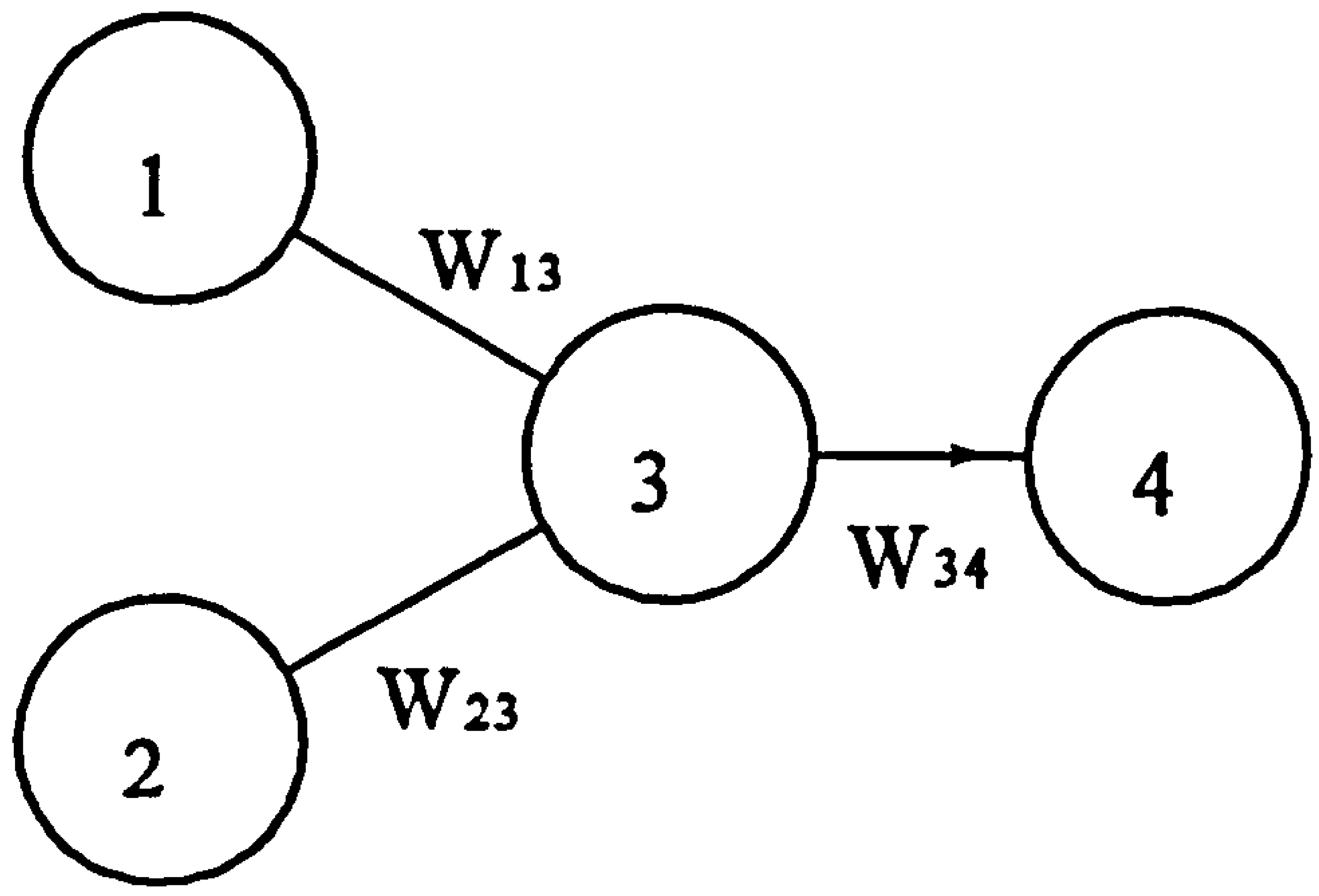
<u>Subtraction of two numbers is possible with an ANN</u>

6.4.2.2 Multiplication of two numbers using an ANN

Multiplication is the elementary building function of any model or simulator involving flow. Most systems can usually be represented as a system of compartments or zones. The dynamics of the system is modelled by flows between these regions. The reactor variables modelled as flows could include temperatures, pressures or masses. The basic requirement of a multiplying ANN is to take two numbers as inputs and produces an output of their product. The simple ANNs developed in the sections above only use additions to produce outputs, any required multiplication is introduced by the weighting between nodes. An ANN of this form cannot handle direct multiplication of two terms so a different ANN internal structure is required.

Consider the following three basic ANN structures, in order of complexity.

1) <u>One Hidden Node</u>



Where

W_{xy} = Weight between Nodes X & Y

X_a = Input A

I_x = Input to Node X

O_x = Output to Node X

Assume Node 4 has a linear transfer function.

Fig 6.32: One Hidden Node Multiplying ANN

Considering Node 3

$$I_3 = X_1 . W_{13} + X_2 . W_{23}$$
$$\therefore O_3 = f_3(X_1 . W_{13} + X_2 . W_{23})$$

If the threshold function f_3 is a half square ie $f_3(x) = \frac{1}{2}x^2$

$$\therefore I_4 = \frac{1}{2} W_{34} (X_1 W_{13} + X_2 W_{23})^2$$
$$= \frac{1}{2} W_{34} (X_1^2 W_{13}^2 + 2 X_1 X_2 W_{13} W_{23} + X_2^2 W_{23}^2)$$
$$= \frac{1}{2} X_1^2 W_{13}^2 W_{34} + X_1 X_2 W_{13} W_{23} W_{34} + \frac{1}{2} X_2^2 W_{23}^2 W_{34}$$

The required output from Node 4 is in terms of $X_1.X_2$, the terms involving just X_1 and X_2 need to be zero. Equating terms of X_1, X_2 and $X_1.X_2$, gives:

$$W_{13}^2 W_{34} = 0 \quad ...... (1)$$
$$W_{23}^2 W_{34} = 0 \quad ...... (2)$$
$$W_{13} W_{23} W_{34} = 1 \quad ...... (3)$$

From Equation (1) either $W_{13} = 0 \quad or \quad W_{34} = 0$

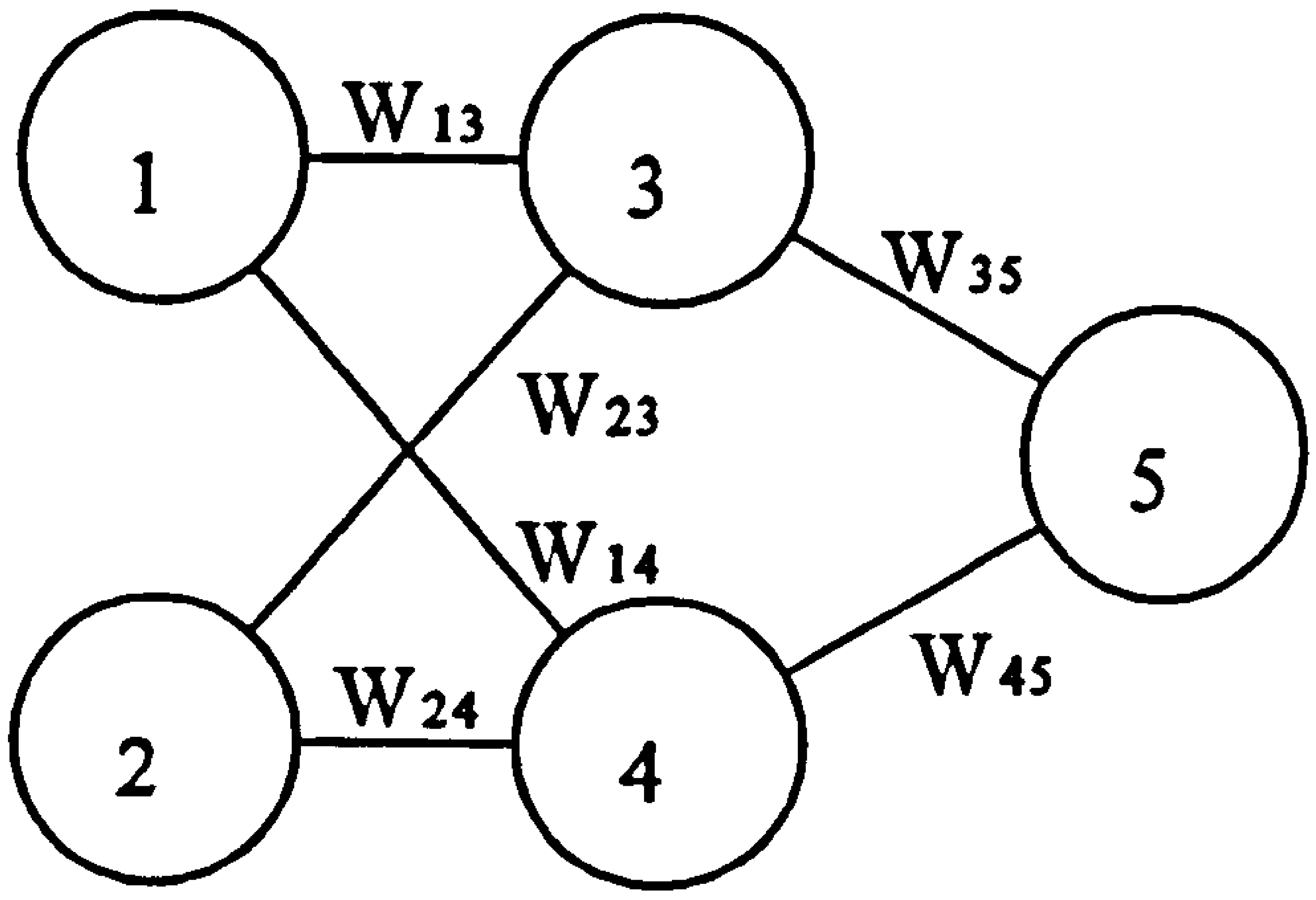
From Equation (2) either $W_{23} = 0 \quad or \quad W_{34} = 0$

Any of the above results cannot give the required result in equation (3). ie

$$If\ W_{13} = 0, W_{23} = 0\ or\ W_{34} = 0$$
$$Then\quad W_{13} W_{23} W_{34} \neq 1$$

Result: An ANN with one hidden node cannot be used as a multiplying unit

2) Two Hidden Nodes



Where

W_{xy} = Weight between Nodes X & Y

X_a = Input A

I_x = Input to Node X

O_x = Output from Node X

Fig 6.33: Two Hidden Node Multiplying ANN

Consider Nodes 3 and 4

$$I_3 = X_1 . W_{13} + X_2 . W_{23}$$
$$\therefore O_3 = f_3(X_1 . W_{13} + X_2 . W_{23})$$

$$I_4 = X_1 . W_{14} + X_2 . W_{24}$$
$$\therefore O_4 = f_4(X_1 . W_{14} + X_2 . W_{24})$$

Again if the threshold function of f_3 and f_4 is a half square ie $f(x) = ½x^2$

$$\begin{aligned} I_5 &= \frac{1}{2}W_{35}(X_1W_{13} + X_2W_{23})^2 + \frac{1}{2}W_{45}(X_1W_{14} + X_2W_{24})^2 \\ &= \frac{1}{2}W_{35}(X_1^2W_{13}^2 + 2X_1X_2W_{14}W_{23} + X_2^2W_{23}^2) \\ &+ \frac{1}{2}W_{45}(X_1^2W_{14}^2 + 2X_1X_2W_{14}W_{24} + X_2^2W_{24}^2) \end{aligned}$$

Again the required output is in terms of $X_1.X_2$, the terms involving just X_1 and X_2 need to be zero. Equating terms of X_1, X_2 and $X_1.X_2$, gives:

$$W_{13}^2W_{35} + W_{14}^2W_{45} = 0 \quad \ldots\ldots (1)$$
$$W_{23}^2W_{35} + W_{24}^2W_{45} = 0 \quad \ldots\ldots (2)$$
$$W_{13}W_{23}W_{35} + W_{14}W_{24}W_{45} = 1 \quad \ldots\ldots (3)$$

From Equation (1), either $W_{13} = W_{14}$ $and\, W_{35} = -W_{45}$

or $W_{13} = -W_{14}$ $and\, W_{35} = W_{45}$

or W_{13} or $W_{35} = 0$ $and\, W_{14}$ or $W_{45} = 0$

From Equation (2), either $W_{23} = W_{24}$ $and\, W_{35} = -W_{45}$

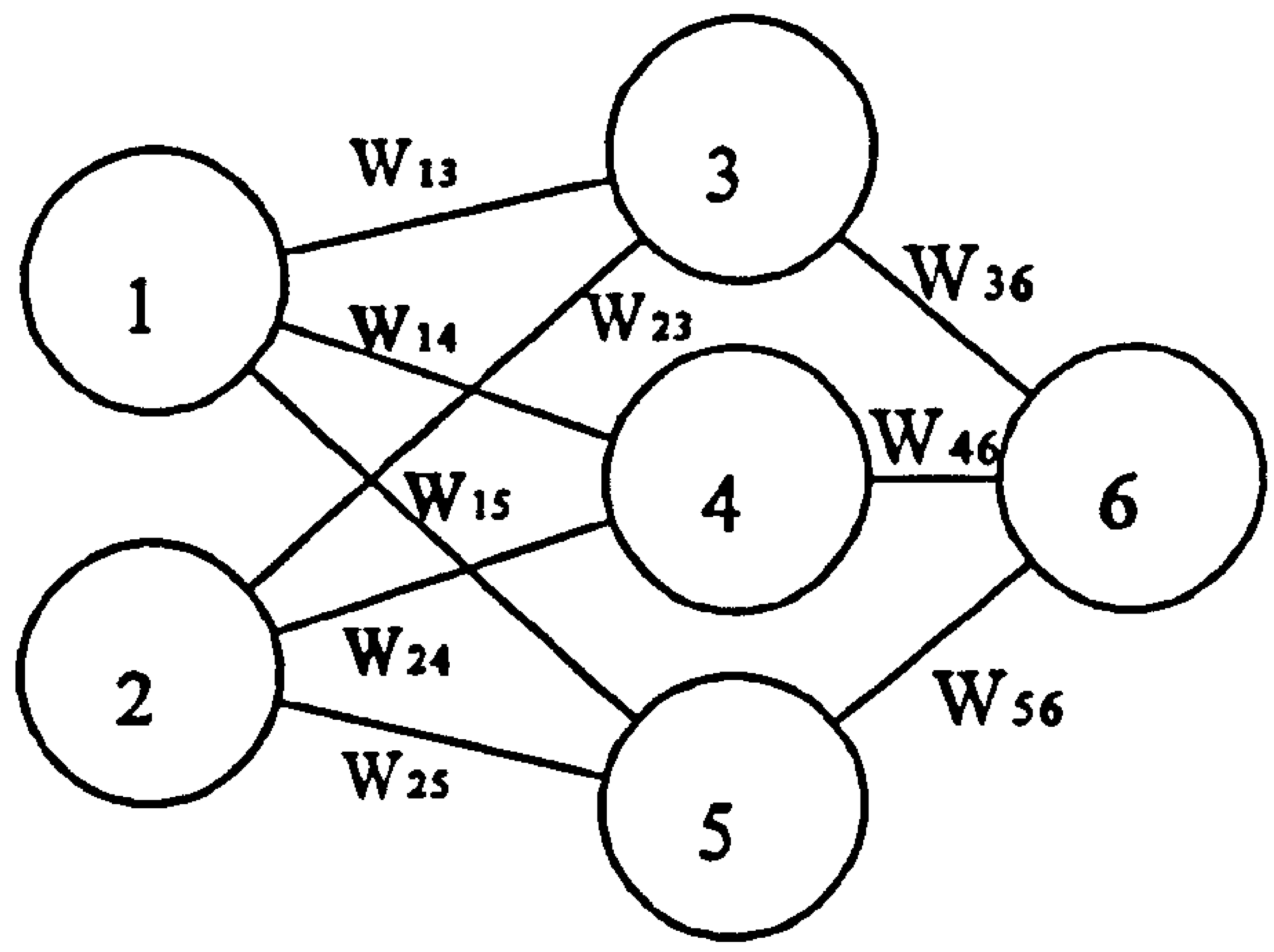
or $W_{23} = -W_{24}$ $and\, W_{35} = W_{45}$

or W_{23} or $W_{35} = 0$ and W_{24} or $W_{45} = 0$

If either W_{35} or $W_{45} = 0$ the problem reduces to a one hidden node ANN which cannot be used as a multiplying unit as shown in the previous case. None of the other pairs of weights can be zero as one of the inputs will be removed from the ANN creating a trivial example. All other combinations of weights do not permit Equation (3) to be correct

Result: An ANN with two hidden nodes cannot be used as a multiplying unit

3) Three Hidden Nodes



Where
W_{xy} = Weight between Nodes X & Y
X_a = Input A
I_x = Input to Node X
O_x = Output from Node X

Fig 6.34: Three Hidden Node Multiplying ANN

Considering Node 3

$$I_3 = X_1 \cdot W_{13} + X_2 \cdot W_{23}$$
$$\therefore O_3 = f_3(X_1 \cdot W_{13} + X_2 \cdot W_{23})$$

Similarly
$$O_4 = f_4(X_1 W_{14} + X_2 W_{24})$$
$$O_5 = f_5(X_1 W_{15} + X_2 W_{25})$$

$$\begin{aligned} O_6 &= W_{36}O_3 + W_{46}O_4 + W_{56}O_5 \\ &= W_{36}f_3(X_1W_{13} + X_2W_{23}) + W_{46}f_4(X_1W_{14} + X_2W_{24}) + W_{56}f_5(X_1W_{15} + X_2W_{25}) \end{aligned}$$

Again if the threshold function of f_3, f_4 and f_5 is a half square ie $f(x) = \frac{1}{2}x^2$

$$\begin{aligned} 2O_6 &= W_{36}(X_1W_{13} + X_2W_{23})^2 + W_{46}(X_1W_{14} + X_2W_{24})^2 + W_{56}(X_1W_{15} + X_2W_{25})^2 \\ &= X_1^2W_{13}^2W_{36} + 2X_1X_2W_{13}W_{23}W_{36} + X_2^2W_{23}^2W_{36} \\ &\quad + X_1^2W_{14}^2W_{46} + 2X_1X_2W_{14}W_{24}W_{46} + X_2^2W_{24}^2W_{46} \\ &\quad + X_1^2W_{15}^2W_{56} + 2X_1X_2W_{15}W_{25}W_{56} + X_2^2W_{25}^2W_{56} \end{aligned}$$

Again the required output is in terms of $X_1.X_2$, the terms involving just X_1 and X_2 need to be zero. Equating terms of X_1, X_2 and $X_1.X_2$, gives:

$$W_{13}^2 W_{36} + W_{14}^2 W_{46} + W_{15}^2 W_{56} = 0 \quad ......(1)$$
$$W_{23}^2 W_{36} + W_{24}^2 W_{46} + W_{25}^2 W_{56} = 0 \quad ......(2)$$
$$W_{13}W_{23}W_{36} + W_{14}W_{24}W_{46} + W_{15}W_{25}W_{56} = 1 \quad ......(3)$$

From Equation (1), say $W_{13}=W_{14}$ and $W_{36}=-W_{46}$

then $$W_{15}^2 W_{56} = 0 \quad ......(4)$$

Similarly from Equation (2), if $W_{24}=W_{25}$ and $W_{56}=-W_{46}$

then $$W_{23}^2 W_{36} = 0 \quad ......(5)$$

From (4) either $W_{15} = 0 \text{ or } W_{56} = 0$

And from (5) either $W_{23} = 0 \text{ or } W_{36} = 0$

If either $W_{56}= W_{36}=0$, then the problem resolves to a two hidden node ANN which has previously been shown not to be a multiplying unit.

$$\therefore W_{15} = W_{23} = 0$$

Putting these results into Equation (3) gives

$$0 + W_{14}W_{24}W_{46} + 0 = 1$$

say $W_{14} = W_{24} = W_{46} = 1$

The structure of the ANN becomes

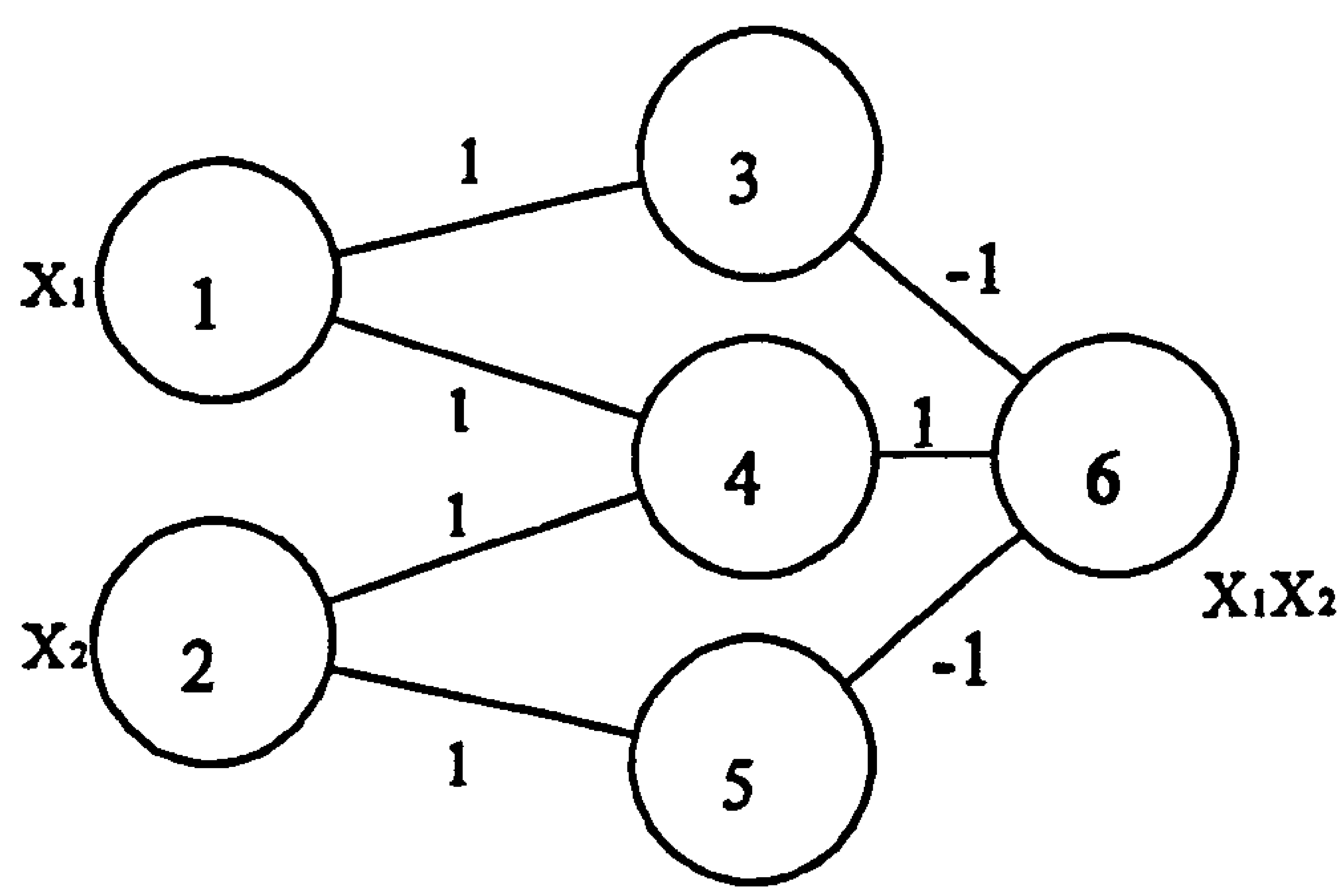


Fig 6.35: Structure of Multiplying ANN

Check

Input to Node 3 = X_1 ,

$\therefore$ Output from Node 3 = $0.5(X_1)^2$

Input to Node 4 = $X_1 + X_2$

$$\text{Output from Node 4} = 0.5(X_1 + X_2)^2$$
$$= 0.5(X_1^2 + 2X_1X_2 + X_2^2)$$

Input to Node 5 = X_2,

$\therefore$ Output from Node 5 = $0.5(X_2)^2$

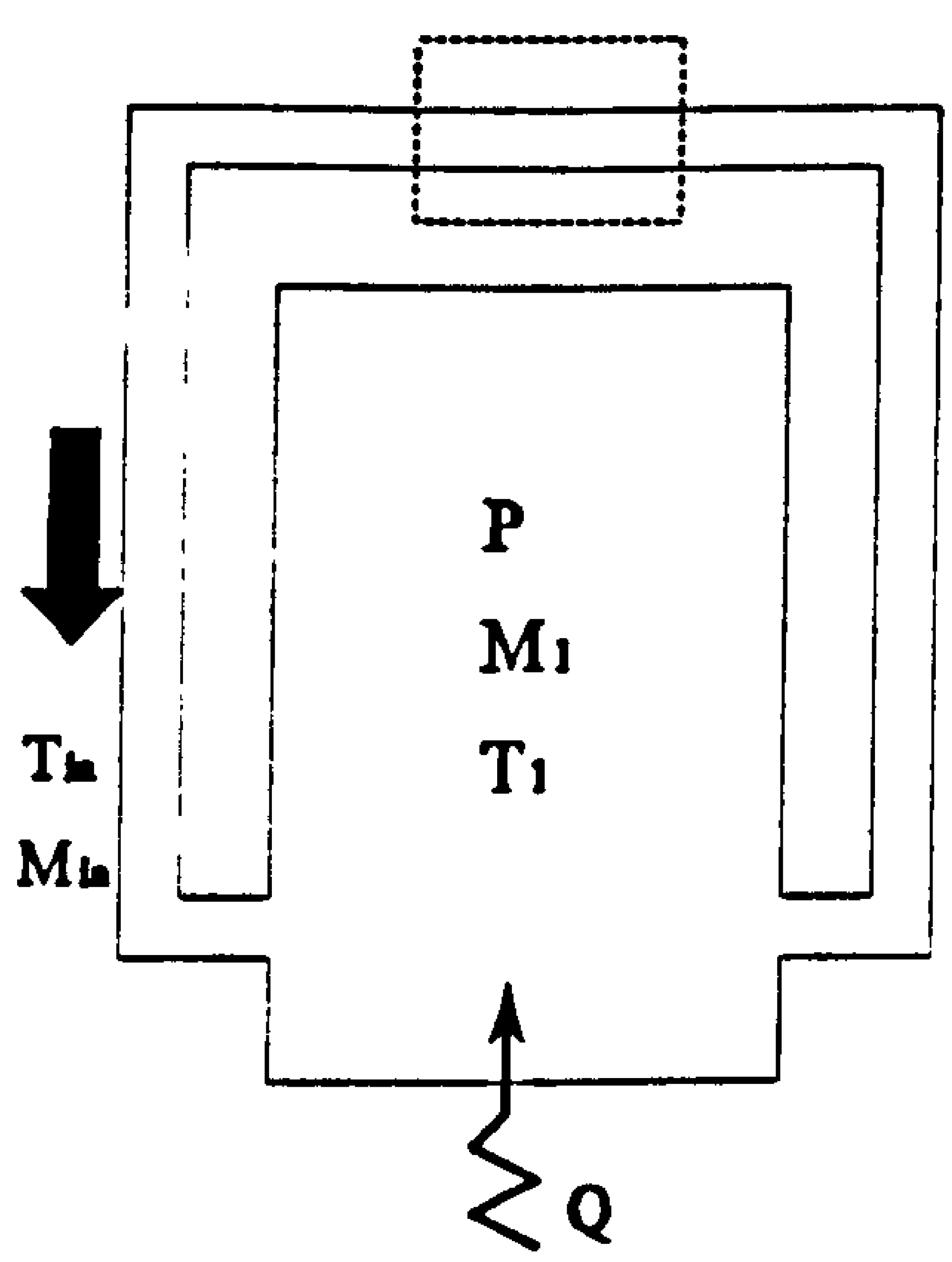
$$\text{Output from Node 6} = -0.5X_1^2 + 0.5(X_1^2 + 2X_1X_2 + X_2^2) - 0.5X_2^2$$
$$= X_1X_2$$

Result: An ANN with three hidden nodes can be used as a multiplying unit

The elementary mathematical functions of addition and multiplication have now been developed. This toolbox can now be used to investigate modelling the PWR primary circuit with an ANN. A simple, one compartmental model of the system is considered first. The full circuit will be considered later in this chapter.

6.5 One Compartment PWR Model

Consider a simple one compartmental model of a reactor as shown below in Figure 6.36. The main body of the model represents the reactor pressure vessel, containing the core, while the loop is a steam generator and piping. The system is assumed to be closed with a constant mass of liquid. The steam generator removes heat from the system so controlling the value of T_{in}



Where: P = Pressure in vessel,

M_1 = Mass of liquid in vessel,

T_1 = Temperature of liquid in vessel,

T_{in} = Temperature of incoming liquid,

M_{in} = Rate of flow of liquid mass,

Q = Heat into the system.

Fig 6.36: One Compartment Model of a Nuclear Reactor

The conservation of energy equation to calculate the next pressure vessel temperature for this system is:

$$T_1^{(k+\Delta t)} = T_1^{(k)} + \frac{Q\,\Delta t}{C_p\,M} + \frac{\Delta t}{M}\left(M_{in}\left(T_{in}^{(k)} - T_1^{(k)}\right)\right)$$

Where: C_p = Specific Heat of liquid in reactor pressure vessel,

Δt = Time step between (k+1) and k.

The intention is to develop an artificial neural network to model this system and calculate the next temperature of the vessel. The network will be an exact equivalent of the system. The first two terms of the equation are simple additions with the constant terms, Q, C_p and M_1, being modelled by the weightings between the ANN nodes. The third term contains a non-linearity, a multiplication, which can now be modelled using the ANN structure previously developed. The initial adaption of the multiplying section of the ANN model is shown below, in Figure 6.37. The constant term $\Delta t/M_1$ is represented by the weightings in links W_{36}, W_{46} and W_{56}.

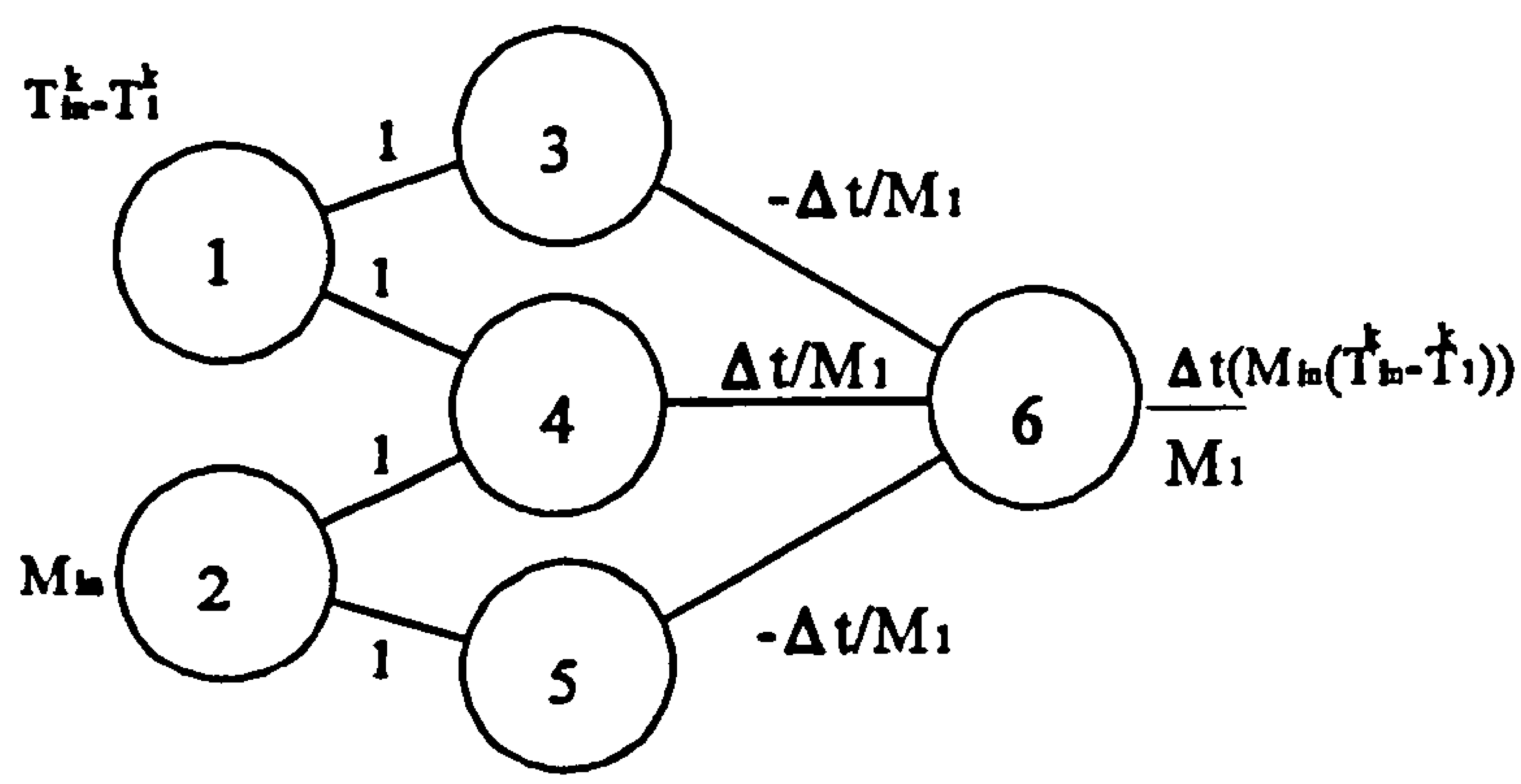


Fig 6.37: Multiplying Element of Simulation Model

This network can be further refined by replacing Node 1 with an ANN subtraction The multiplication is now modelled by the following network, Figure 6.38.

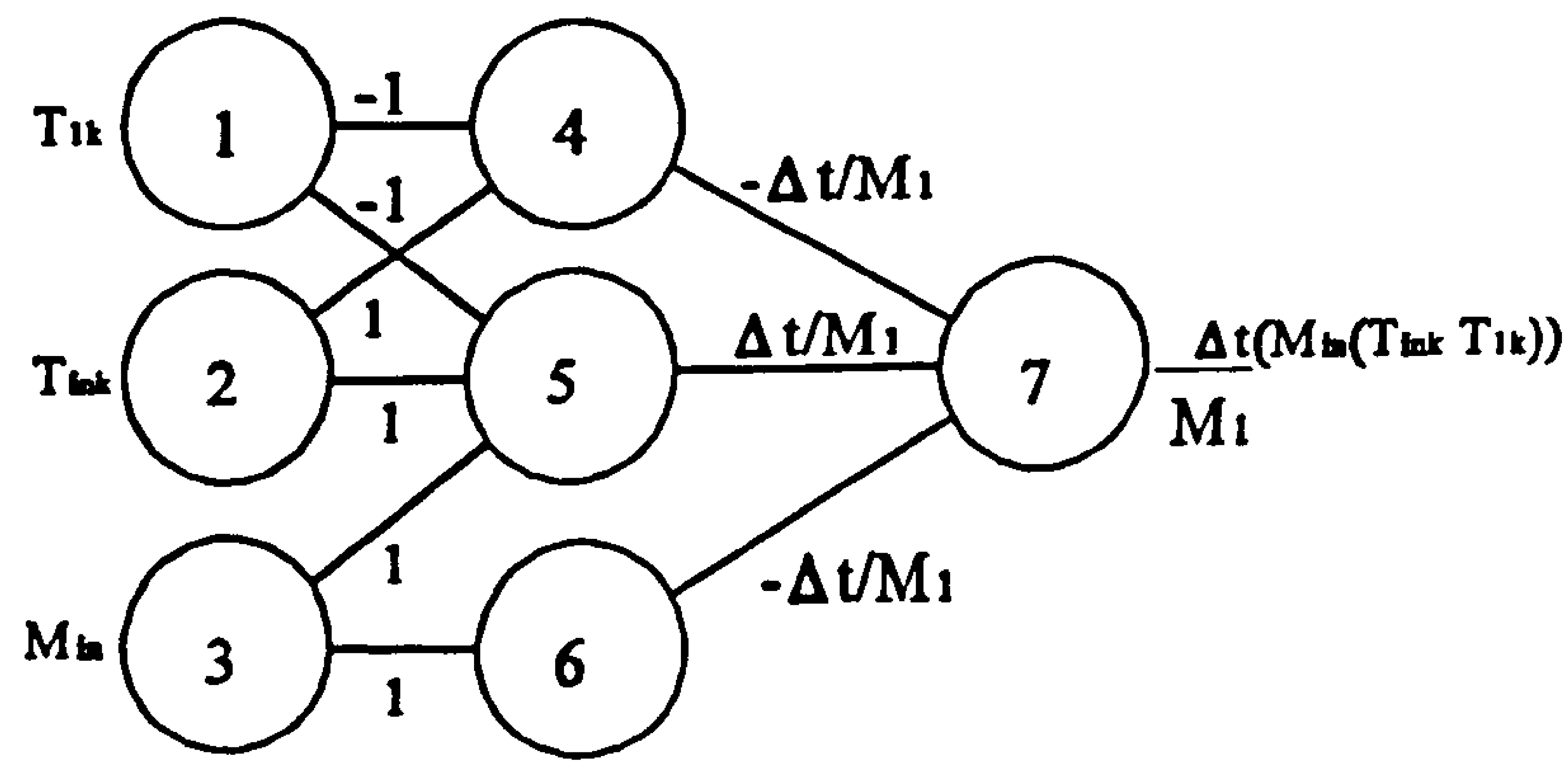


Fig 6.38: Refined Model Multiplying ANN

Combining this network with the first two terms of the energy equation produces the ANN model, shown in Figure 6.39, for the temperature flow in the one compartmental model. The central part of the network models the multiplication while the addition of the first two terms is included in the weights and summation in the output node.

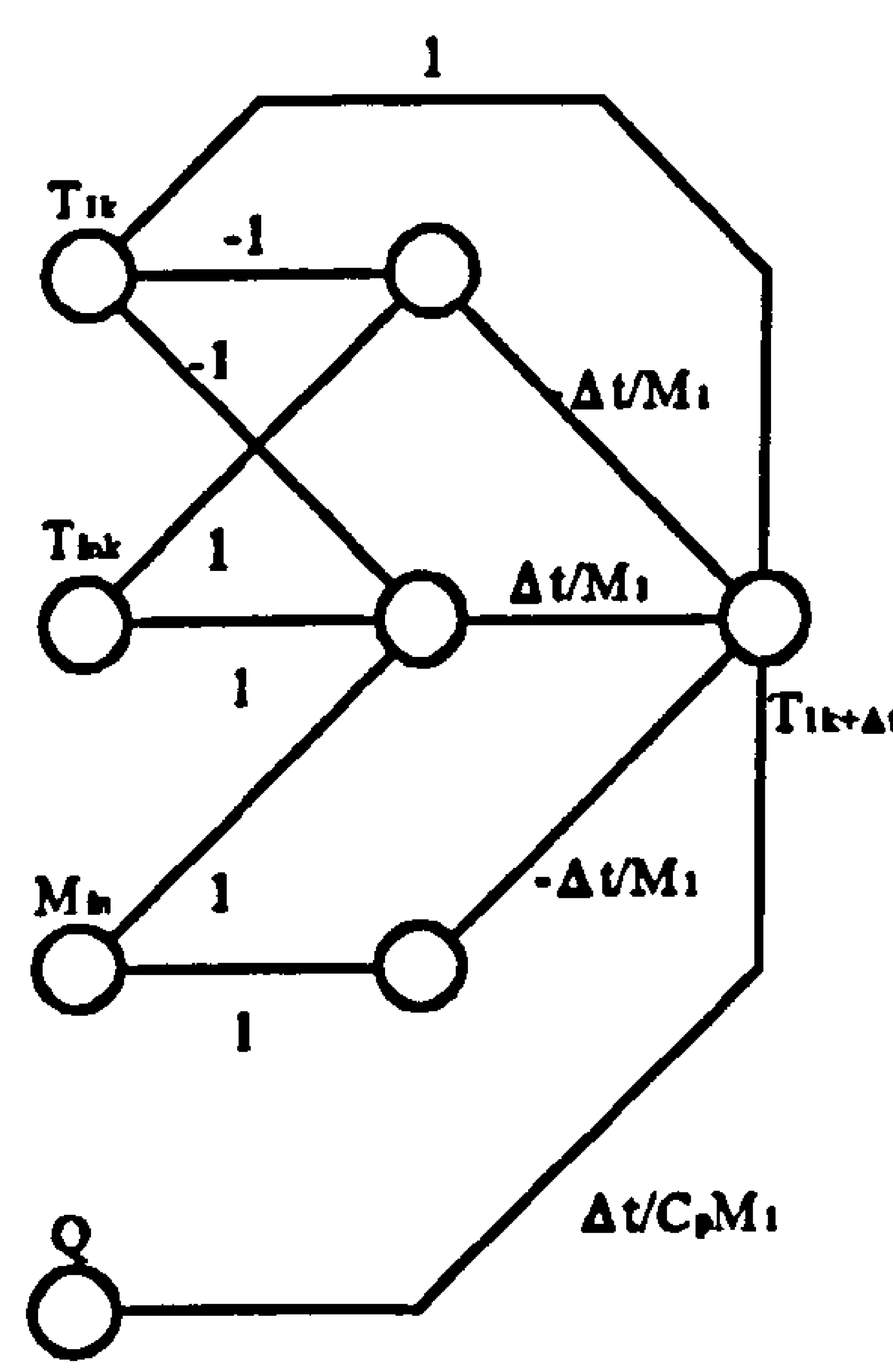


Fig 6.39: ANN Representation of One Compartment PWR Model

This network was originally produced on a spreadsheet program but proved to be very unwieldy in operation. The ANN was coded into a C program to enable testing of the models performance and comparison with the computer simulation to be performed practically. The full code listing is given in Appendix L. The program used a feedback loop in which each calculated value for T_1 was used as the input for Node 1 in the next iteration. A series of different conditions were presented to the model and run until a steady state was reached. To ascertain an approximate final value of this condition consider the following:

$$T_1^{(k+\Delta t)} = T_1^{(k)} + \frac{Q\,\Delta t}{C_p\,M} + \frac{\Delta t}{M}\left(M_{in}\left(T_{in}^{(k)} - T_1^{(k)}\right)\right)$$

For Steady State conditions $T_1^{k+\Delta t} = T_1^k$

$$\frac{Q}{C_p M_1} + \frac{1}{M_1}\left(M_{in}\left(T_{in}^k + T_1^k\right)\right) = 0$$

$$M_{in}\left(T_{in}^k + T_1^k\right) = -\left(\frac{Q}{C_p}\right)$$

$$T_{in}^k + T_1^k = -\left(\frac{Q}{C_p M_{in}}\right)$$

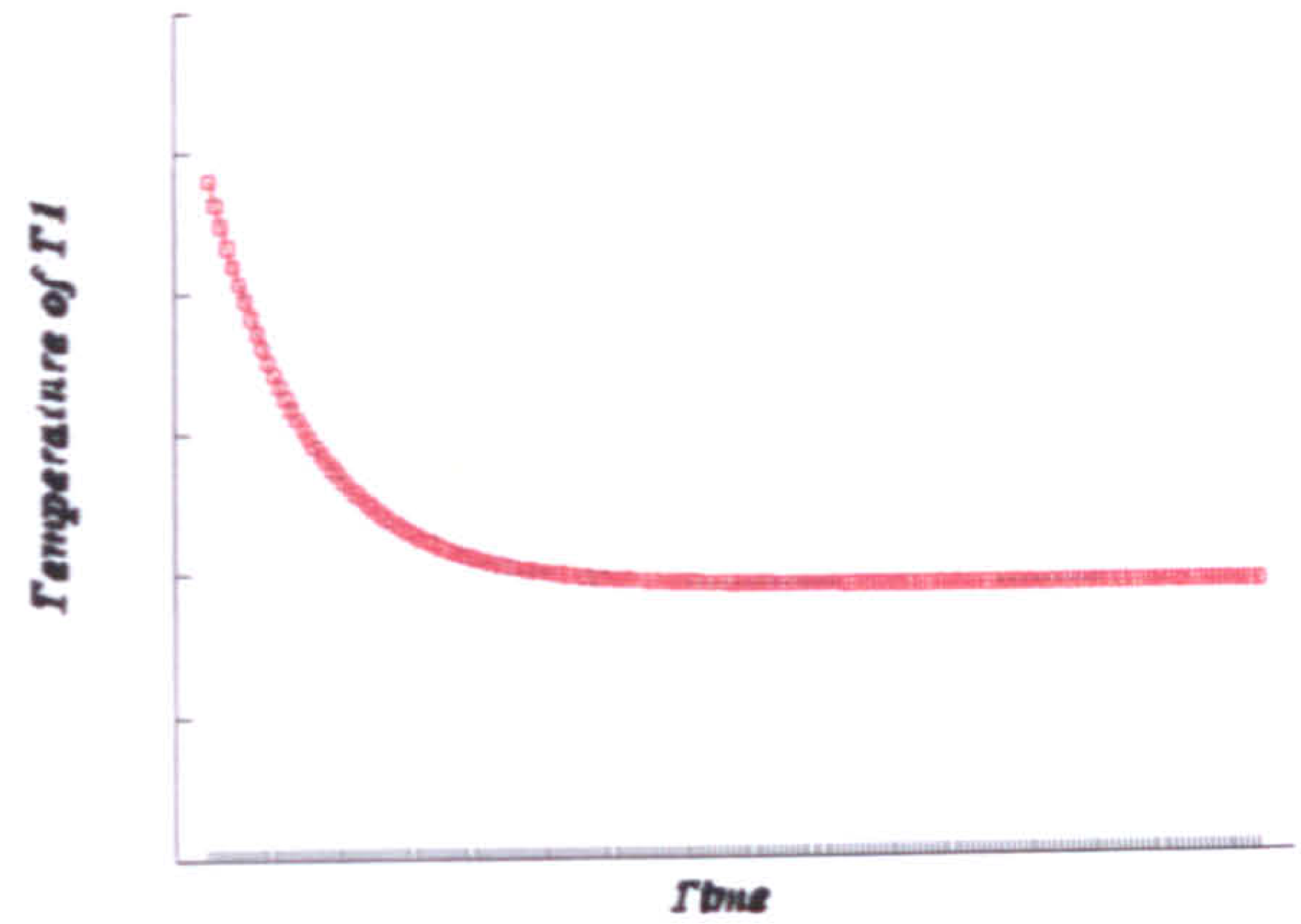
Let the final compartment temperature be T_1^{final}

Suitable variable values for testing the network were obtained for a Westinghouse PWR (Todreas and Kazmi, 1990). These are given below in Table 6.5.

Variable	Value
Inlet Temp., T_{in}	286°c
Outlet Temp., T_1	324°c
Core Flow Rate, M_{in}	17400 Kg/s
Core Power Level, Q	3800 MW
Core Volume, M_1	3.06 x 10^5 Kg

Table 6.7: Test Values for One Compartment Model

Using these values the following results shown in Figs 6.40 to 6.44, were obtained.



Case 1: Half Power

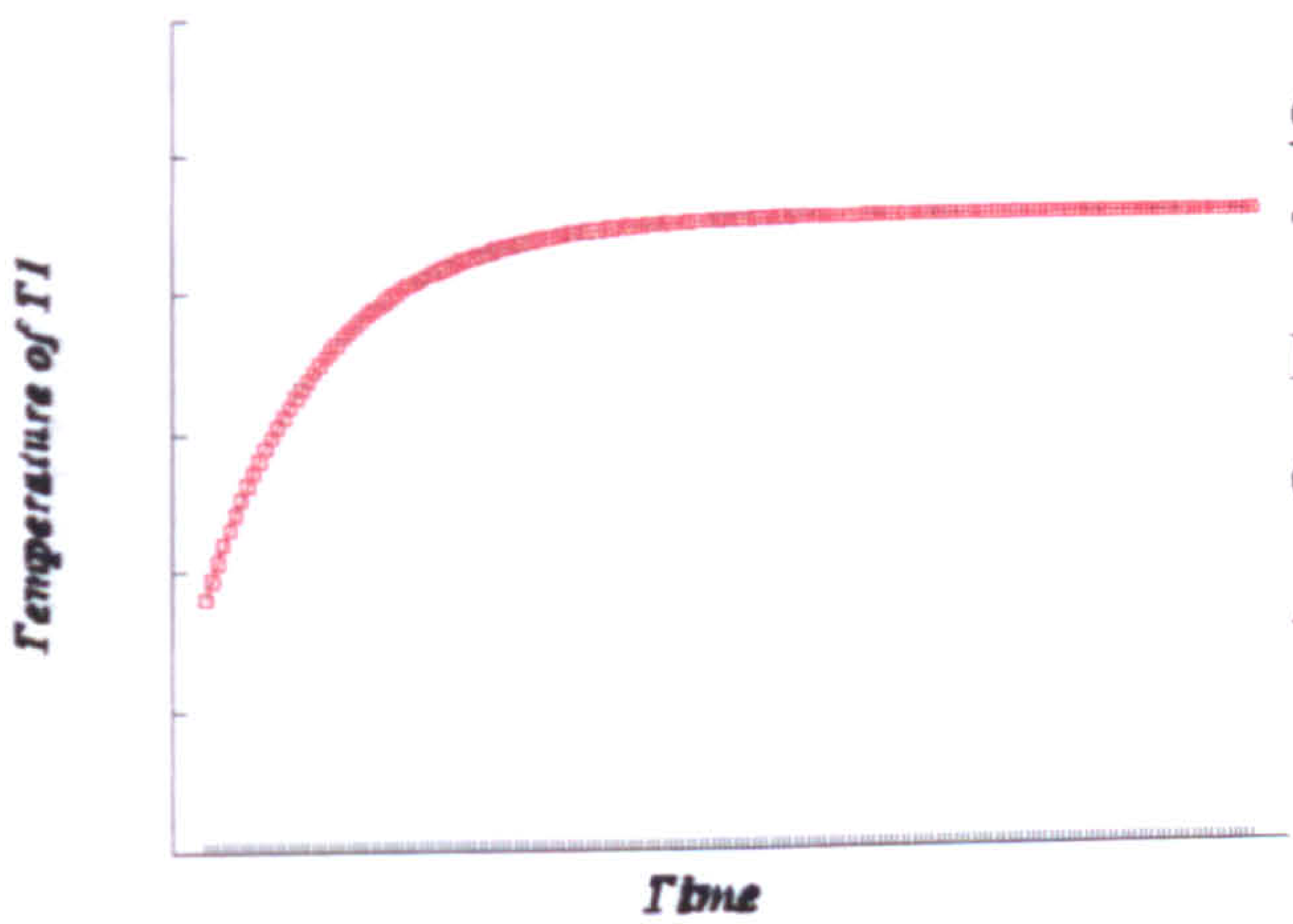
$T_1 = 324°c$, $T_{in} = 286°c$,

$M_{in} = 17400$ Kg/s, $Q = 2.0 \times 10^9$

Calculated $T_1^{Final} = 309.46°c$,

ANN $T_1^{Final} = 309.46°c$

Fig 6.40: Half Power Condition



Case 2: Reduced Flow, Small Leak

$T_1 = 324°c$, $T_{in} = 286°c$,

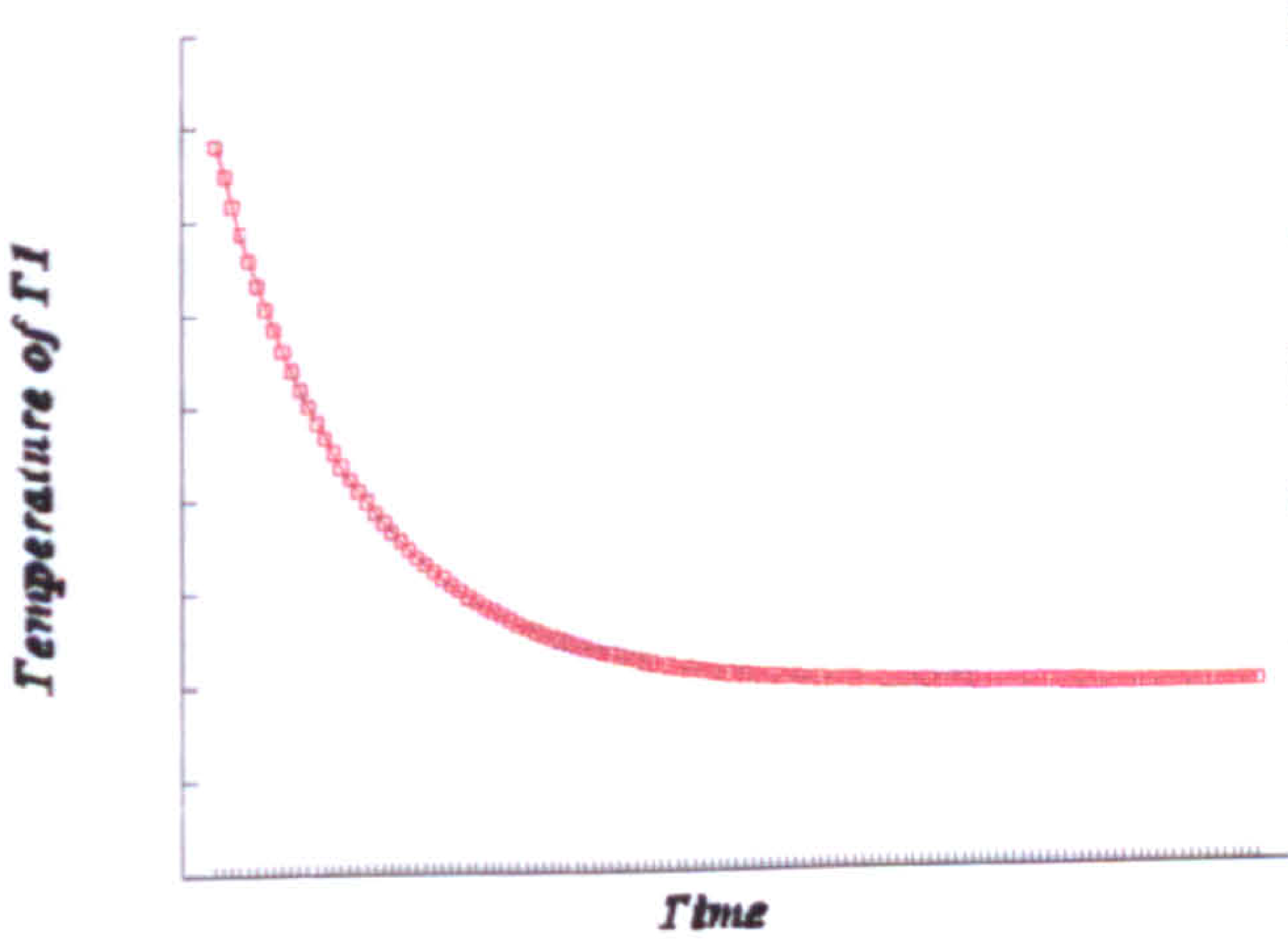
$M_{in} = 15000$ Kg/s, $Q = 3.8 \times 10^9$

Calculated $T_1^{Final} = 337.70°c$

ANN $T_1^{Final} = 337.70°c$

Fig 6.41: Reduced Flow Condition

Case 3: Reduced Incoming Temperature



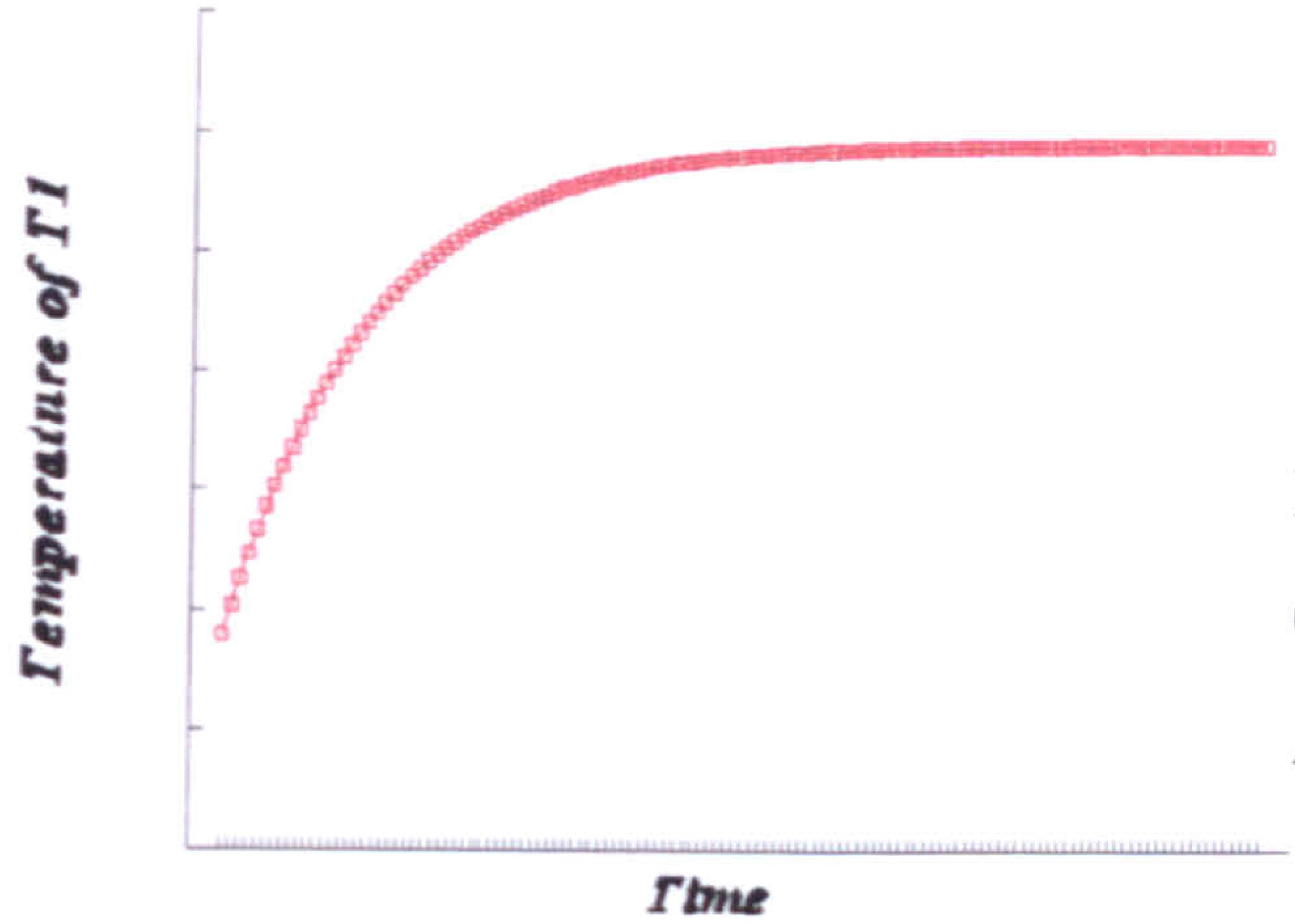
$T_1 = 324°c$, $T_{in} = 250°c$,

$M_{in} = 17400$ Kg/s, $Q = 3.8 \times 10^9$

Calculated $T_1^{Final} = 294.57°c$

ANN $T_1^{Final} = 294.57°c$

Fig 6.42: Reduced Temperature Condition



Case 4: Increased Incoming Temperature

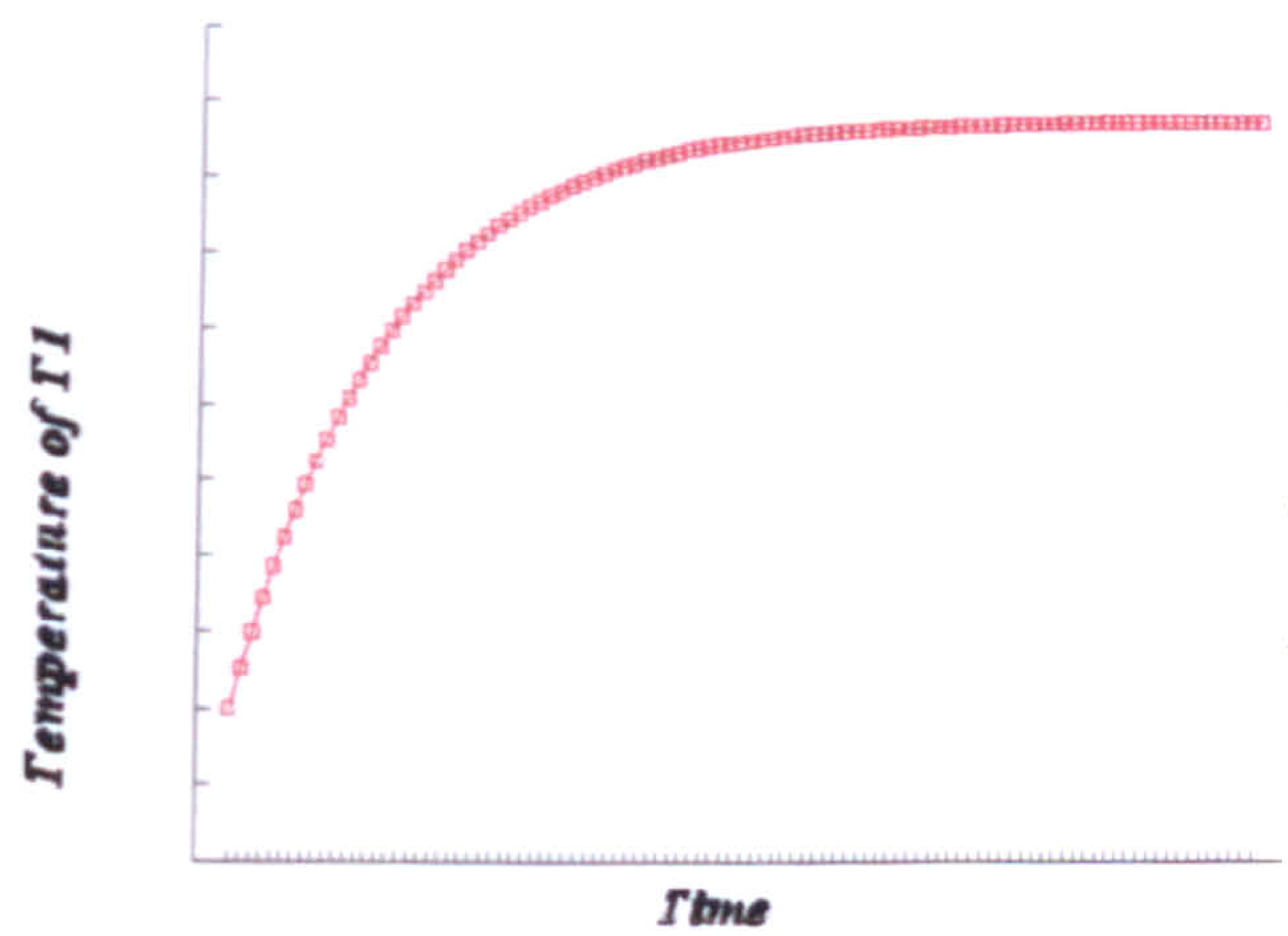
$T_1 = 324°c$, $T_{in} = 300°c$,

$M_{in} = 17400$ Kg/s, $Q = 3.8 \times 10^9$

Calculated $T_1^{Final} = 344.57°c$

ANN $T_1^{Final} = 344.57°c$

Fig 6.43: Increased Temperature Condition



Case 5: Increased Flow

$T_1 = 324°c$, $T_{in} = 286°c$,

$M_{in} = 20000$ Kg/s, $Q = 3.8 \times 10^9$

Calculated $T_1^{Final} = 324.78°c$

ANN $T_1^{Final} = 324.77°c$

Fig 6.44: Increased Flow Condition

These results were very close to that obtained from the simulator program. As the ANN is a direct equivalent of this system the results should be identical. For all cases, with the exception of Case 4, the actual and predicted values of T_1 are correct to 2 decimal places. The slight variation can be explained by the handling of numerical values by the compiler of the computer program. A feedback technique was used to input the next value of T_1 into the program. With this method even very small differences can gradually build up and create a larger errors. The results obtained warranted further investigation in this method of prediction.

This network is a direct equivalent of the equation but still uses the transfer function developed for the multiplying network. Namely:

$$O_j = 0.5(\Sigma_{i=1}^{3}(I_i \cdot W_{ij}))^2$$

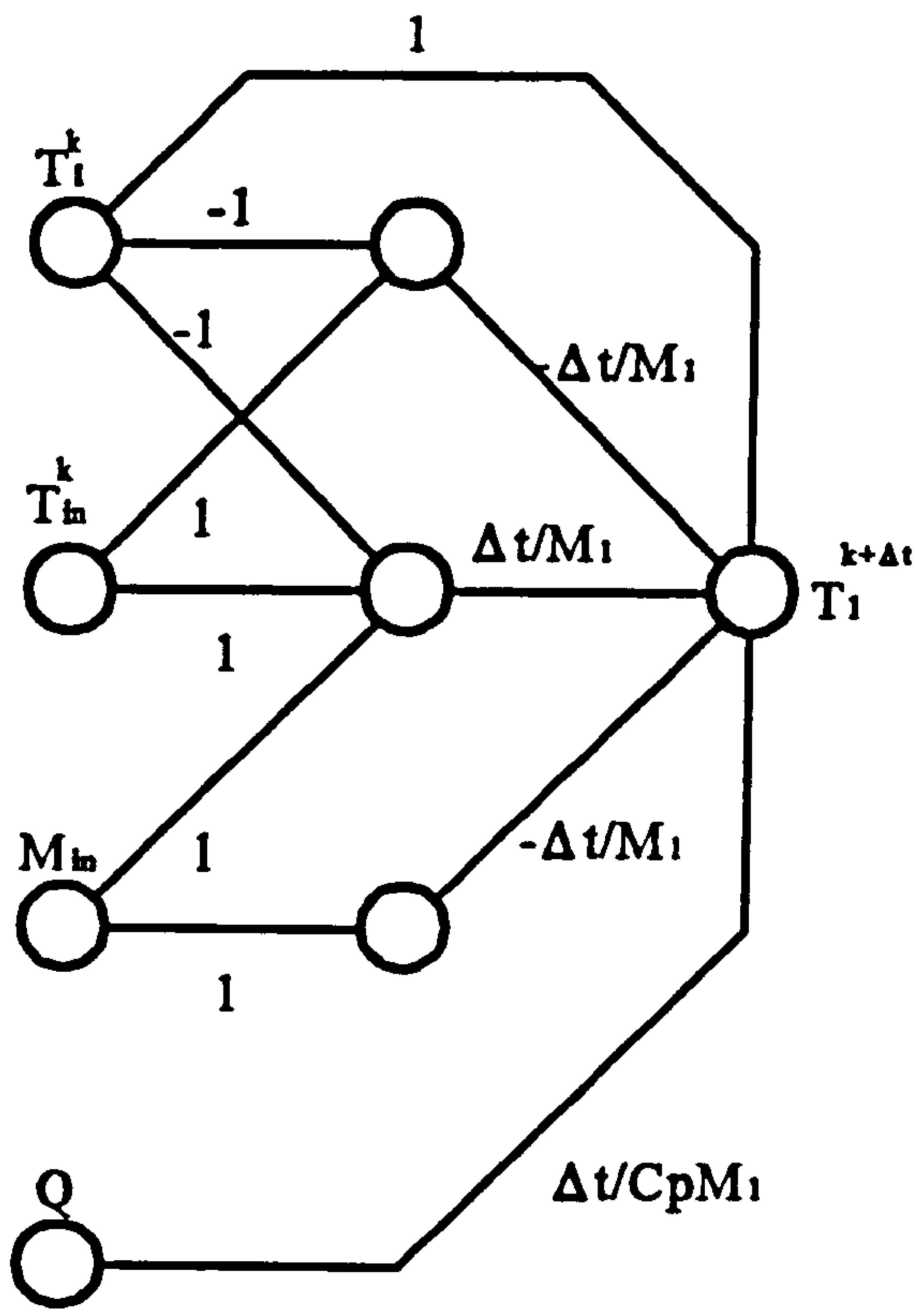
Where: O_j = Output at node j,

I_i = Input from node i,

W_{ij} = Weighting between Nodes i and j.

This feature is a useful theoretical tool for understanding the internal mechanism of the network but it is not a standard function. No saturation level is defined so the node output theoretically has no limit. This feature is undesirable for a practical ANN system as the lack of a maximum level for the node output enables possible chaotic, unseen situations to occur. A threshold limit will cap these extreme values to a known level and so prevent the simulation producing wildly inaccurate outputs. These unwanted cases could occur if the network was required to predict a transient scenario with catastrophic conditions, such as a large leak, when unrealistic values for the variables may occur, for example negative masses or extremely large temperatures. In the case of the PWR the transfer function saturation models physical limits such as the pipe diameter and the volumes of regions.

A series of neural networks were developed to investigate replacing this transfer function with standard transfer functions. Firstly, the half square elements of the network were replaced by ANNs trained to square a single input. A second ANN was then developed that multiplied two inputs and so could replace the entire multiplying section of the network. Lastly, an ANN was trained to perform the entire function of the equation and calculate the change in temperature. In all cases the result from one calculation, T^{k+1} was fed back into the model as the input to determine the next temperature, T^{k+2}. These three cases are shown, along with the original ANN structure, in the following set of diagrams, Figures 6.45 to 6.48. The known value of the weightings for node links are shown. The remaining values will be determined by each training process.



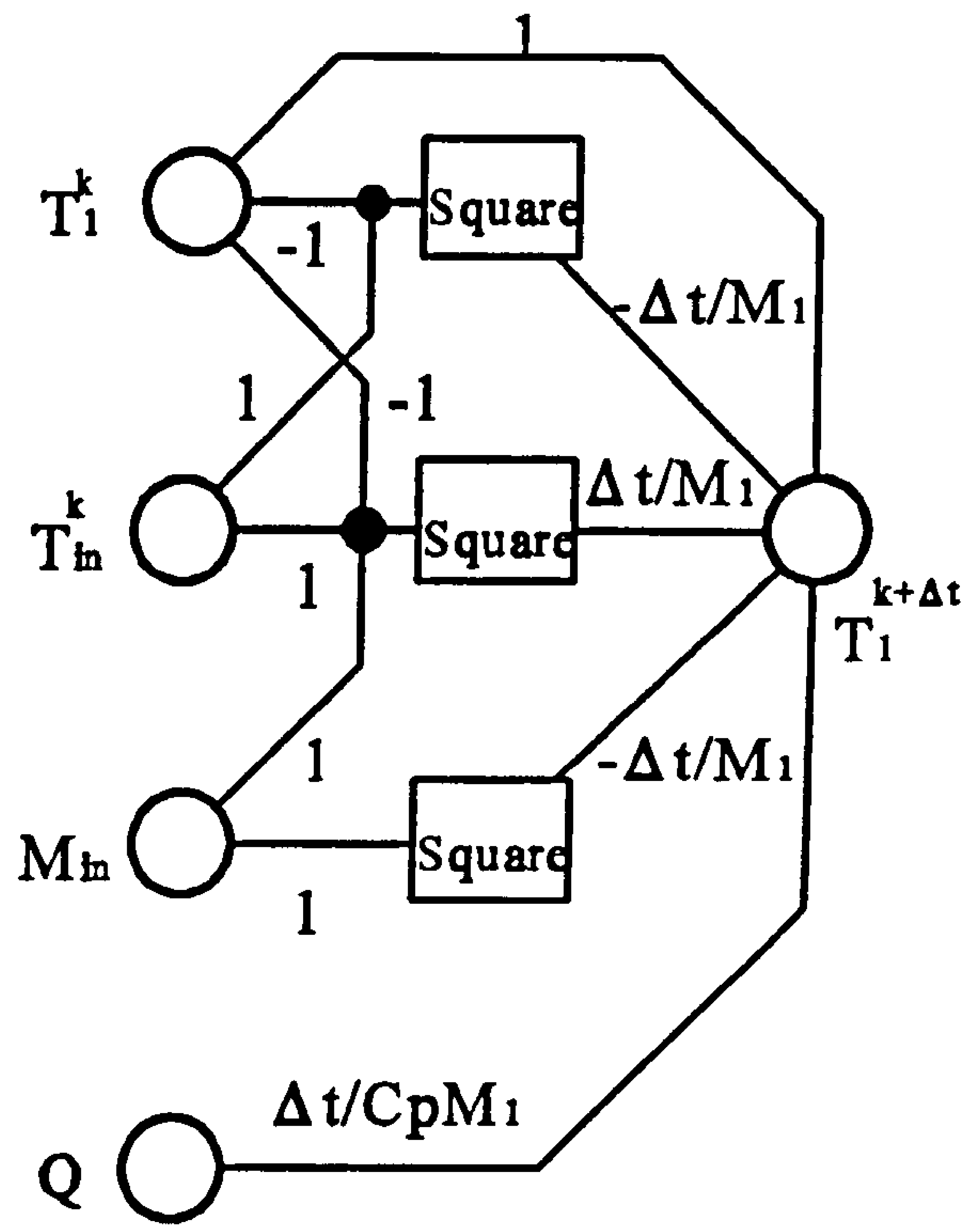
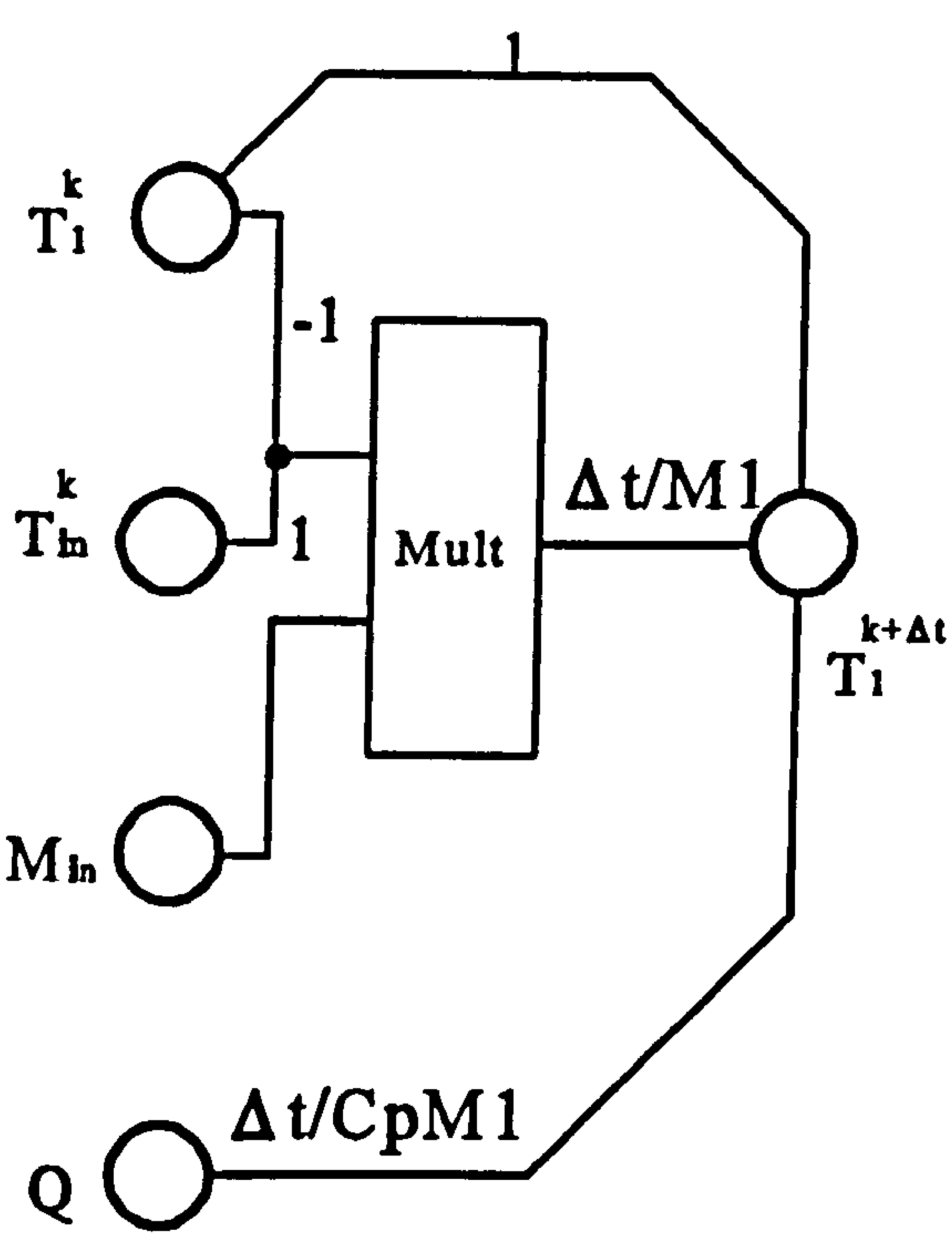


Fig 6.45: ANN of 1CV Model

Fig 6.46: ANN with Squaring ANN



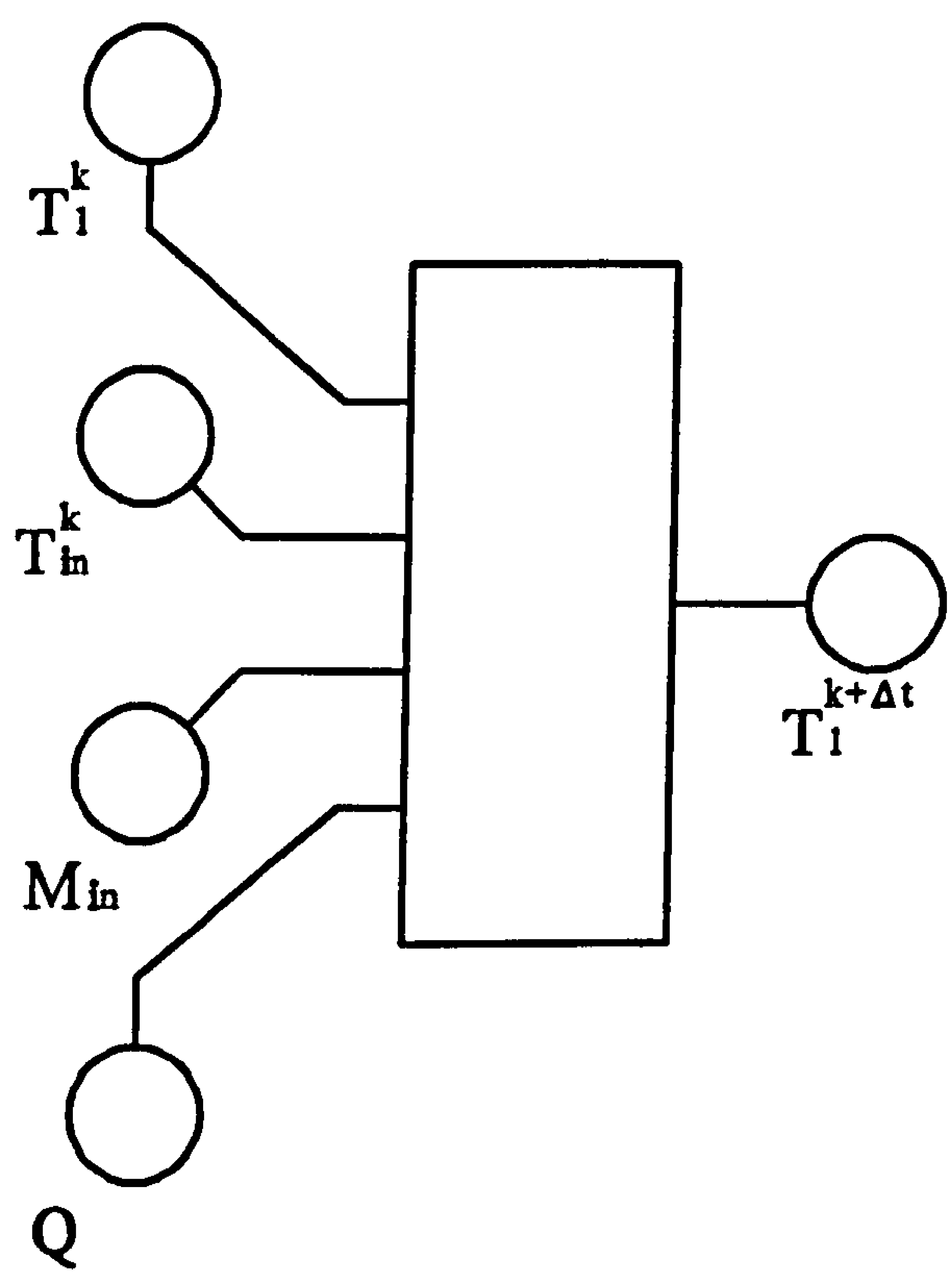


Fig 6.47: ANN with Multiplying ANN

Fig 6.48: ANN with Trained ANN

This gradual replacement of the non-standard component of the simulator ANN enabled a study of the different forms of ANN to be compared. The final output of each ANN output could also be investigated to determine any relationship between accuracy and standardisation. The first two ANNs would be composed of a composite structure; standard, trained ANNs inside a heuristically developed ANN.

The first two combined networks, Figs 6.45 and 6.46, were still direct equivalents of the system equations. These embedded ANNs were trained to perform simple mathematical functions, squaring and multiplying. The data sets for training these ANNs were therefore lists of squares and multiples respectively.

The squaring ANNs consisted of a single input, of the number to be squared, and one output node of the resulting square. Two sets of squaring ANNs were developed to investigate the process of squaring a number using an ANN. The first was trained to square the positive integers from 1 to 20 while the second set of ANNs considered the squares of the integers from -20 to 20. The ANN data set in the first case consisted of 20 cases while the second contained 41 examples. The main results are given in Table 6.6, below. The full results are given in Appendix M.

Filename	Architecture	Threshold	RMS Error
square2a.nnd	1-2-1	Tanh	0.0153
square7a.nnd	1-3-1	Sine	0.0296

Table 6.8: Results from Training Squaring ANNs

The graphs of these ANNs with the training data are shown in Figs 6.49 and 6.50.

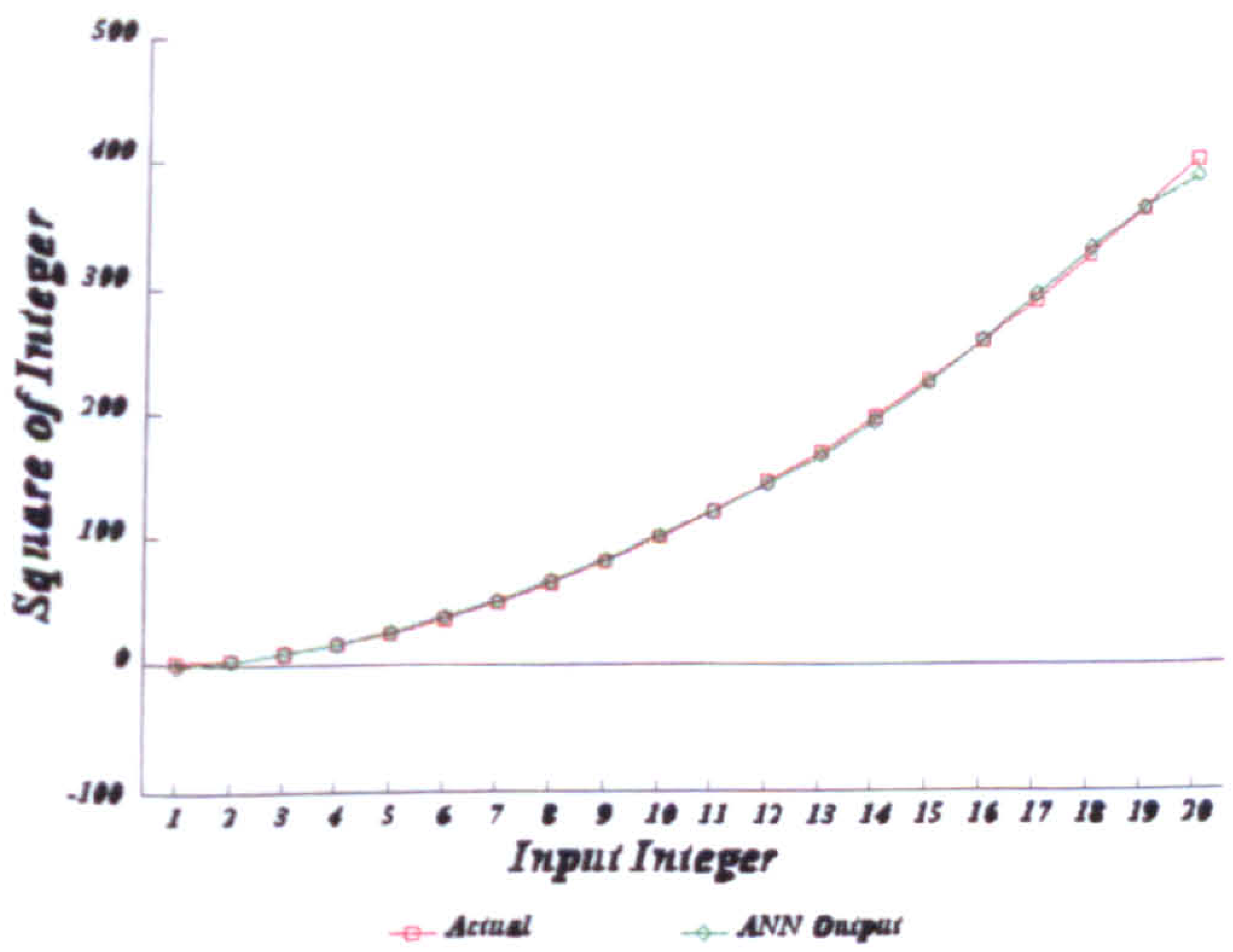


Fig 6.49: Results of Best Squaring ANN for Positive Integers < 20

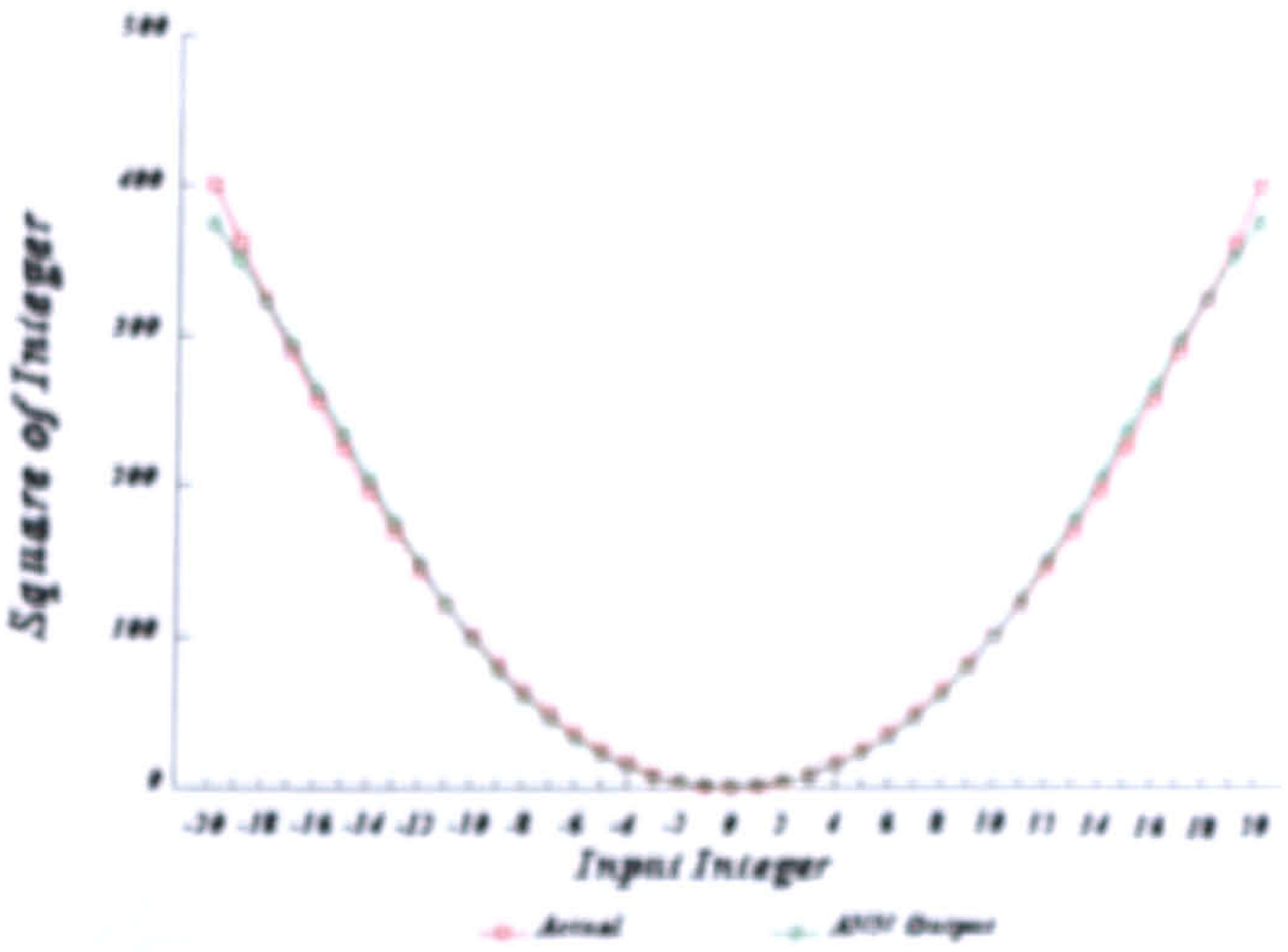


Fig 6.50: Results of Best Squaring ANN for all Integers $-20 \le x \le 20$

These graphs show that most error in the ANN output is at the extremities of the curves. For the most accurate results when using the squaring ANN the input would require scaling to the middle section of the input range.

The multiplying ANNs consisted of two input nodes, for each number, and a single output to contain the product of the inputs. Again a series of ANNs were developed to explore the gradual change from squaring to multiplying a range of integers by ANN. The first networks replaced the single input of the squaring ANN with two identical positive inputs. The second group of ANNs used both positive and negative identical integer values for inputs. The final networks used an input set produced by varying both inputs in the integer range of 0 to 10. This method produced a training data set of 111 cases. In each case the information was divided into training and test sets in the approx ratio of 2:1. The full results for the multiplying ANNs are given in Appendix N. The results from the best ANN, for the last stage, are given in Table 6.9.

Filename	Architecture	Threshold	RMS Error
mult19.nnd	2-3-1	Sine	0.0225

Table 6.9: Results from Training Multiplying ANNs

A comparison of the output from this ANN with the actual data is shown in Figure 6.51.

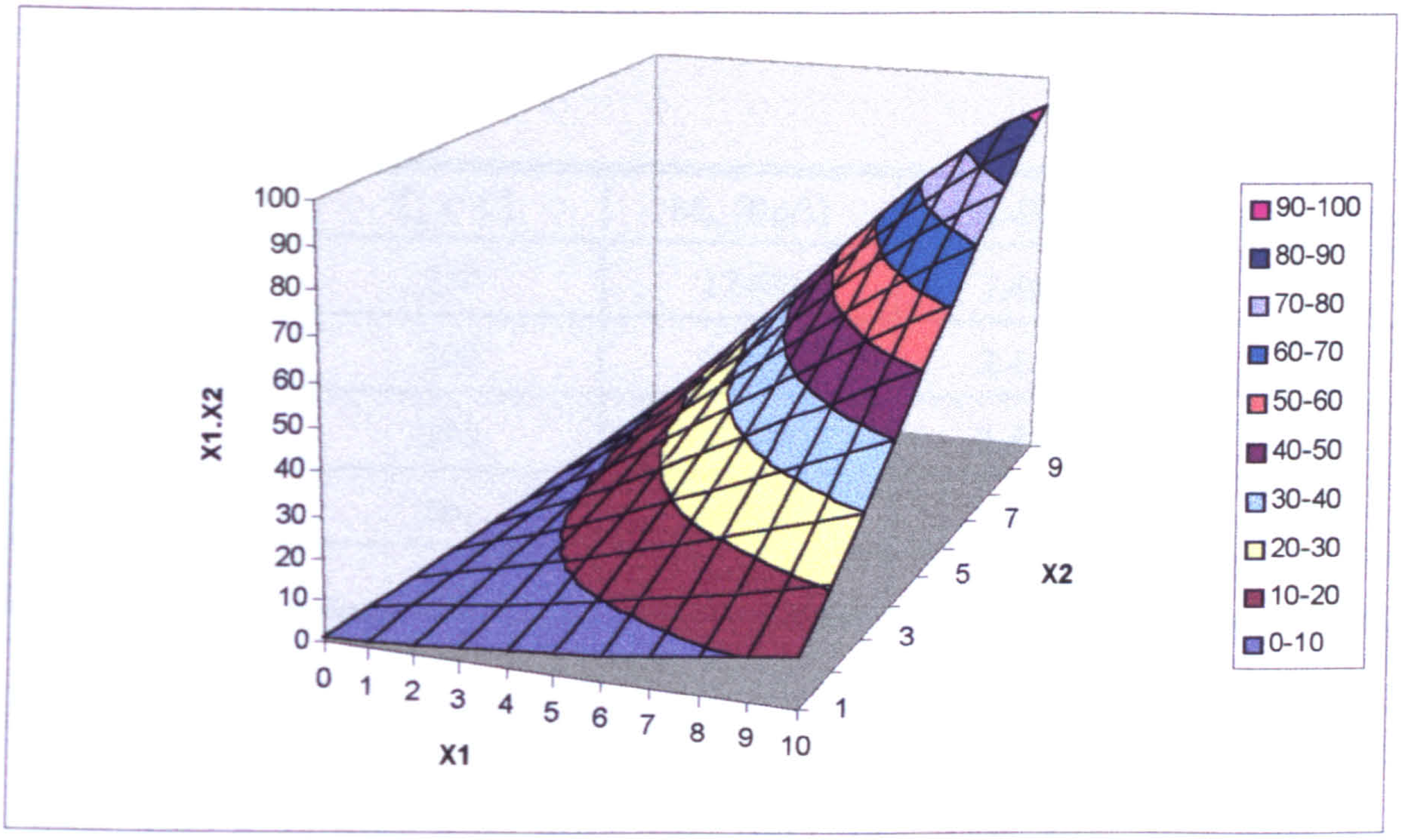


Fig 6.51: Results of Best Multiplying ANN

The results again show that the product of values at the extremities of the considered range are least accurately calculated by this ANN. This characteristic will need to be considered in further investigations. A possible method of reducing the impact of this aspect could be to scale the ANN input values to lie in the most accurately calculated region.

The third form of ANN to be developed, a complete 1CV model, required data from actual plant operating conditions to be able to produce realistic training and test sets. In order for the ANNs to develop relationships between the input variables the data sets need to reflect the form of these associations. Simple mathematical formula cannot be used to create simplified data sets, as in the above investigations, because the ANNs are required to perform a more complicated task, namely the modelling of a 1CV model of the PWR. The relationships between the variables are therefore more involved. Furthermore it is not possible to consider every possible scenario for the model. The set of combinations of all possible cases, even for a simple system, is large and an ANN training set designed to reflect these would also be sizeable. A set of scenarios were produced that, although not necessarily a true reflection of the actual operating situations, gave a guide to the relationships between the variables. A simulator model was used to model each of these situations with plant data from the Westinghouse PWR (Todreas and Kazmi, 1990). Initially each case was modelled until a steady state condition was achieved. This initial data set is given below in Table 6.8, the whole set consisted of 860 entries which were randomly given below, in Table 6.13, details on all ANNs are given in Appendix O.

Filename	Architecture	Threshold	RMS Error
temp13f.nnd	4-6-1	Sine	0.0177

Table 6.13: Result for 1CV PWR ANN with ΔT_1 as Output

This network was translated into a C program module and incorporated into a feed back program. This program also included the best ANNs previously developed for squaring and multiplying the input variables to the 1CV PWR module for comparison. The program listing is given in Appendix P. A series of tests were performed, using values typical of a civil nuclear reactor. Two examples of the results obtained are shown below.

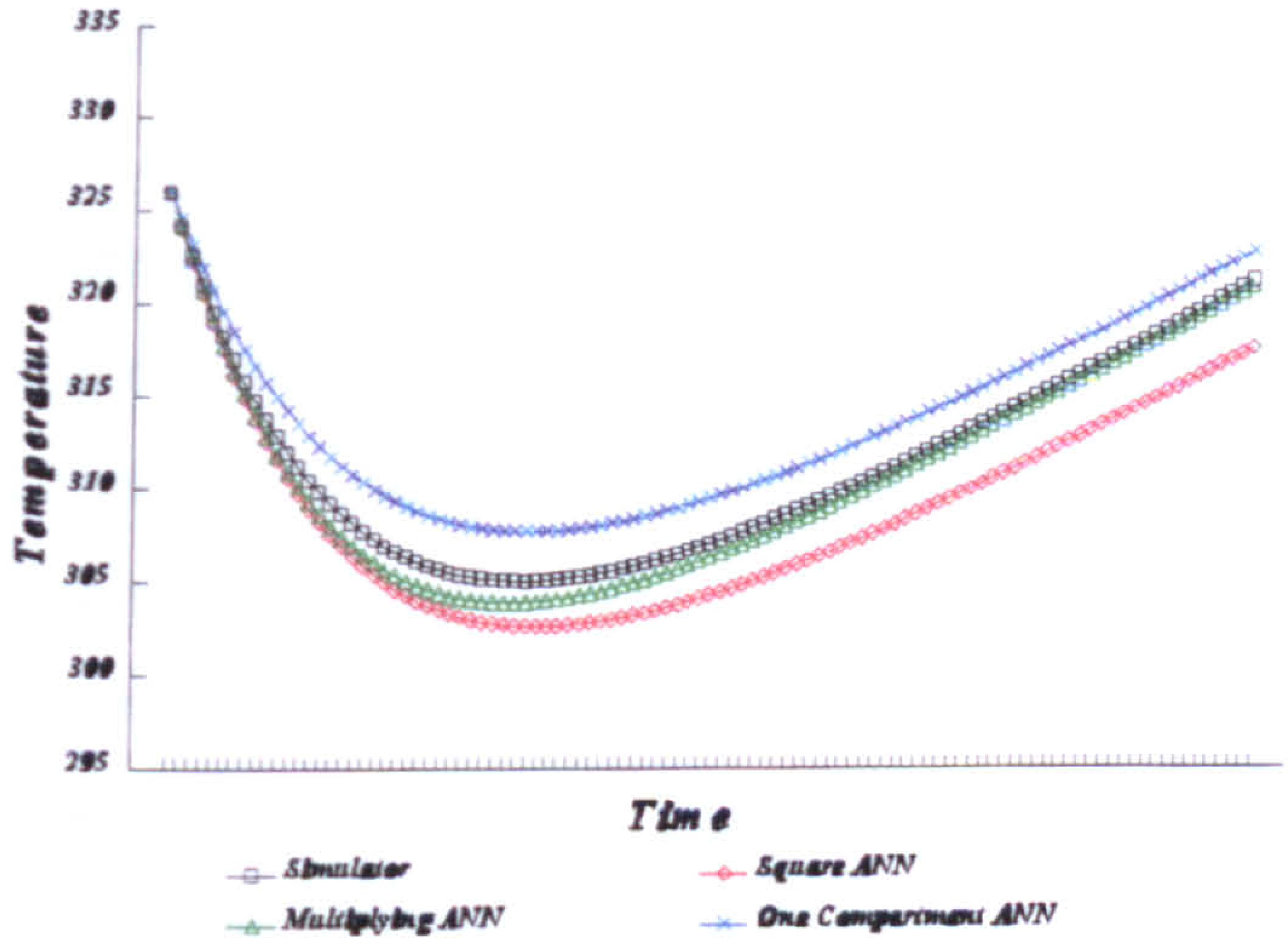


Fig 6.52: Results of Transient with Decreasing Flow and Heat

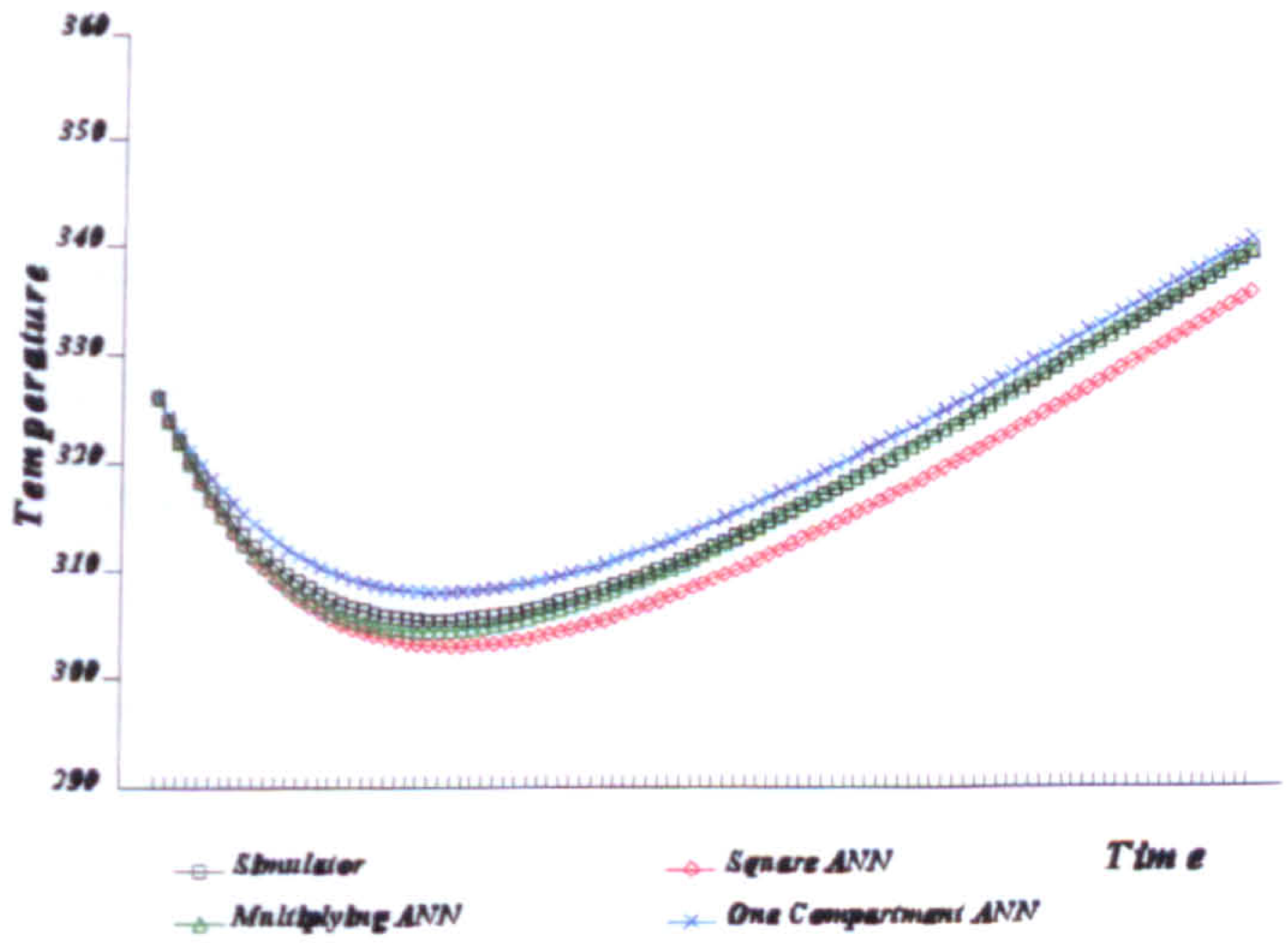


Fig 6.53: Results of Transient with Increasing Flow and Heat

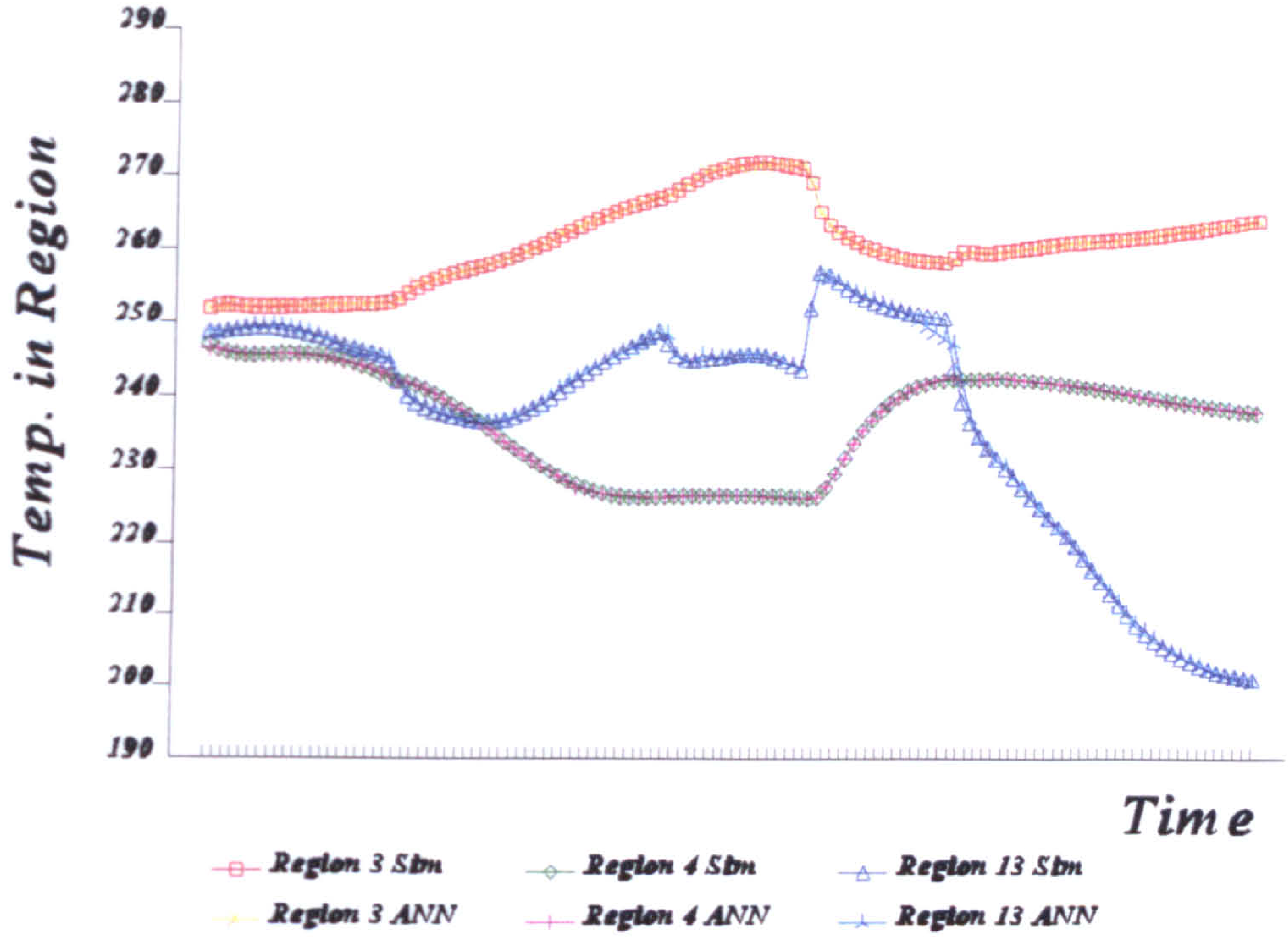


Fig 6.54: Temperature Changes for Steam Generator Loop

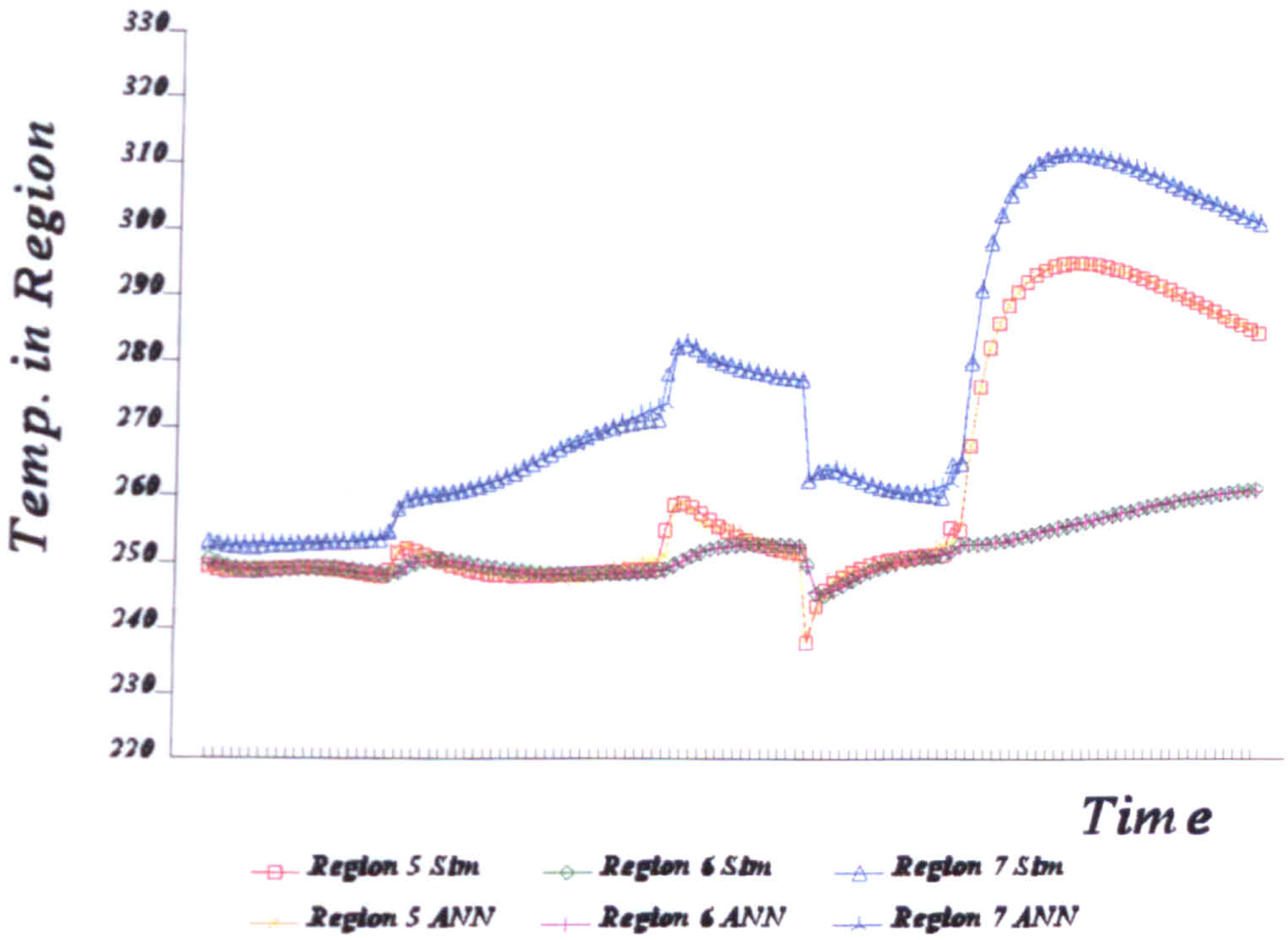


Fig 6.55: Temperature Changes for Reactor Pressure Vessel

The results show that the twenty-five region direct equivalent ANN model is a very good representation of the PWR primary circuit. The few minor differences can be attributed to the smoothing some of the sudden rates of change in regional temperature.

Filename	Architecture	Threshold	RMS Error
pwr4g.nnd	5-6-1	Sine	0.0563

Table 6.15: Result for Primary Circuit ANN Module

An independent data set was created to evaluate this ANN. The range of variable values used are given below, in Table 6.16.

Data Set	Initial T_1^k (°C)	T_{in}^k (°C)	Normalised M_{in} (Kg/s)	Q (W)	Normalised M_1 Kg
pwr_cv4.nna 101 Cases	250	250 → 200	0 → 1	0 → 5×10^6	0 → 1

Table 6.16: Data Set for Testing Primary Circuit ANN Module

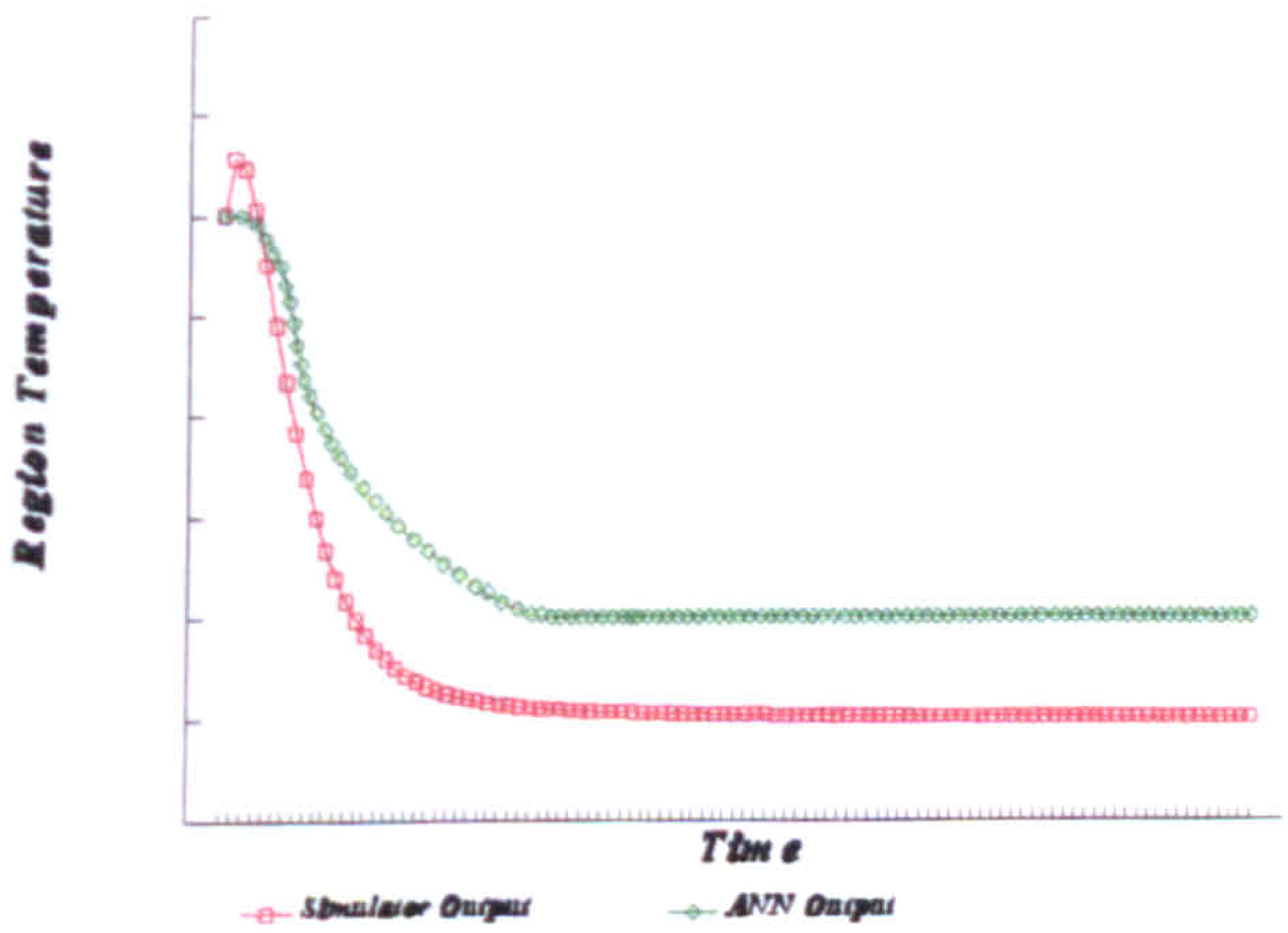


Fig 6.56: Independent Test Set Results

The results from this ANN are not very satisfactory. The curves produced are discernable as part of the transient being tested but the values are not precise enough for realistic inclusion in an advisory system. This approach requires refining to produce better predicted temperatures.

One method of improving the predictive capability of the primary circuit trained 1CV model was to consider the immediate history of the 1CV model, to include variable information

This ANN has a lower RMS error value than previously, however when the same data set from Table 6.16 was used to evaluate this ANN. The results obtained are shown below, in Figure 6.57.

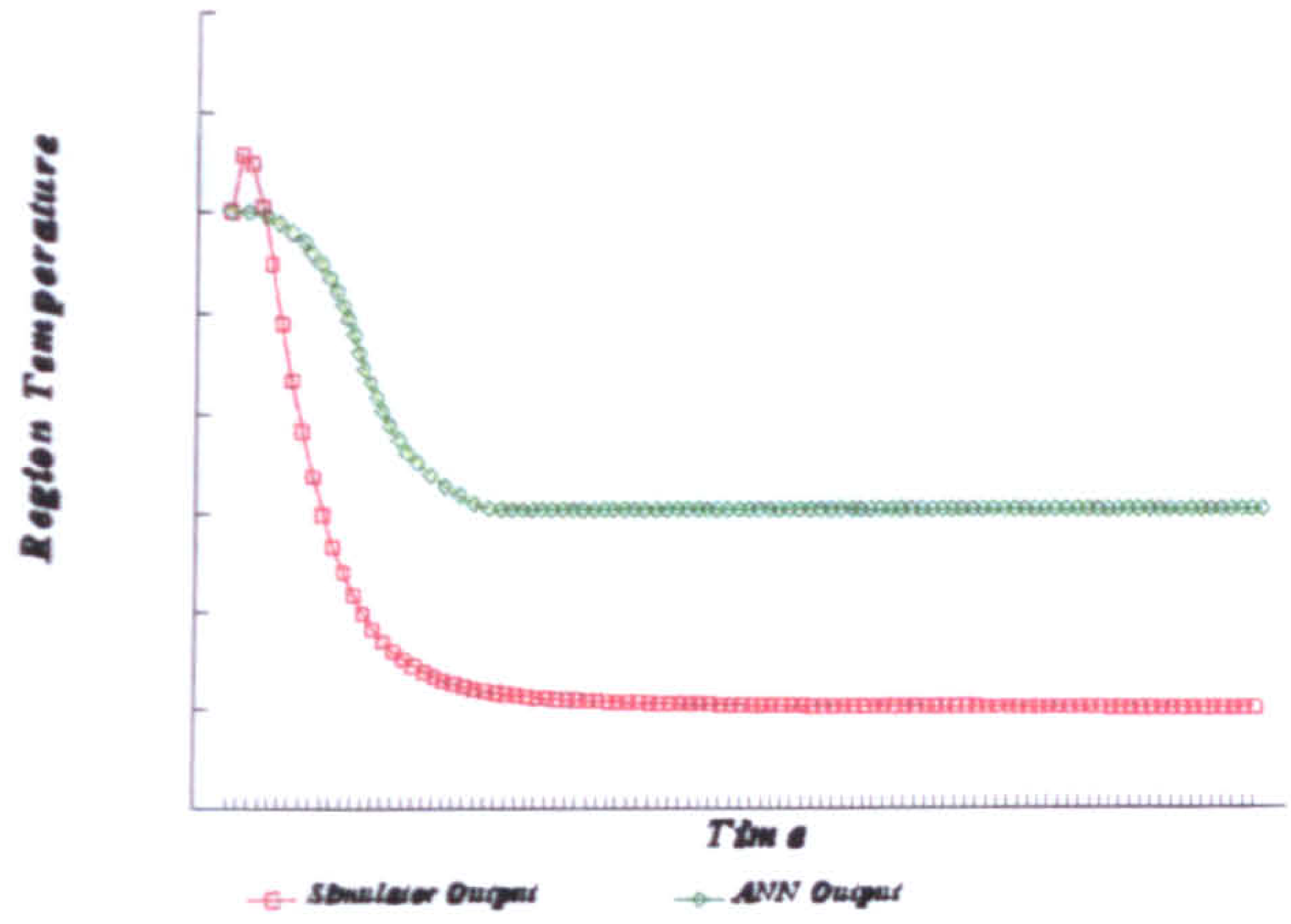


Fig 6.57: Result from Two Time Step ANN

These results are considerably worse than even the initial ANN from this section. This may be due to over fitting of the training data and the extra complexity in the ANN causing additional, unnecessary errors. The implicit inclusion of regional mass appears to complicate the predictive ANN. Previously the mass of each region was coded as a component of the final weighting between the ANN hidden layer and the output node. In contrast the last two investigations have introduced the regional mass as a main ANN component with a status equal to other variables such as temperature. This approach has meant that the ANNs have developed additional relationships between these variables which do not enhance the predictive ability of an ANN on an independent test set. The first approach to this dilemma was to retain regional mass as an ANN input but to only connect this node directly to the output node of the ANN. This structure would hopefully duplicate the direct equivalent model and so produce more accurate temperature predictions. The following two figures illustrate the revised architecture.

A new ANN architecture was developed in which the regional mass was retained as an input but only connected to a second hidden layer of nodes. A diagram of this arrangement is shown below. The number of nodes shown in the hidden layers are for illustration and do not necessarily represent the optimum ANN.

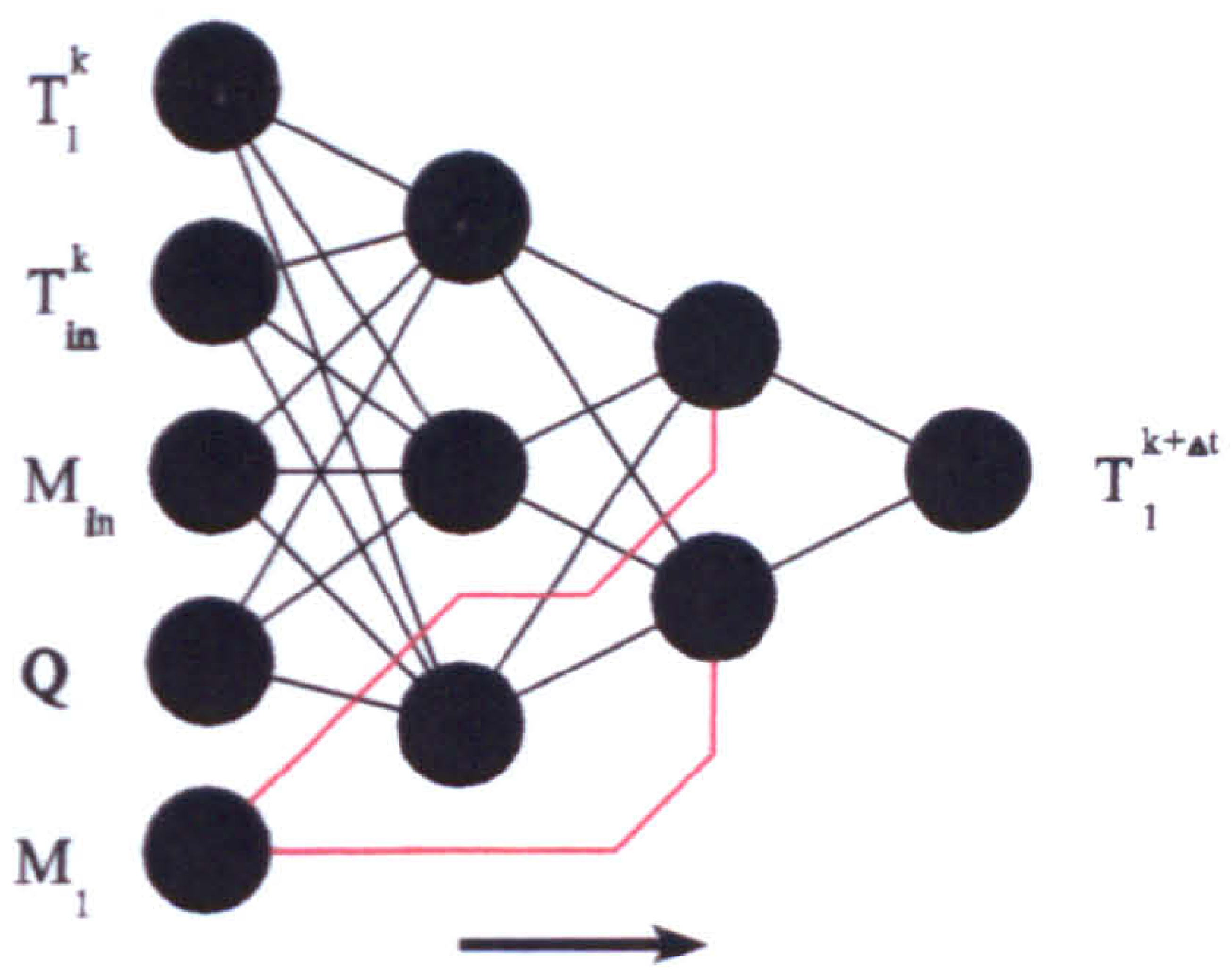


Fig 6.59: Revised Two Hidden Layer ANN structure

A series of ANNs were trained with various combinations of nodes in the two hidden layers. The training and test sets, pwr_cv3.nna, developed in the initial investigation in this section was again used. The ANNs were trained each trained for 80,000 cycles with the test set being presented every 100 iterations for the last 20,000 cycles. The best network structure, in terms of RMS error, was saved. The full results for the ANNs developed for this data set are given in Appendix R. The details of the best ANN developed are given below.

Filename	Architecture	Threshold	RMS Error
pwr8c.nnd	5-7-3-1	Sigmoid	0.0252

Table 6.20: Result for Primary Circuit Two Hidden Layer ANN Module

The RMS errors for this set of ANNs were far lower than the previous investigations in this section. A true determination of the predictive ability of this ANN must be by using an independent test set. Until now the test sets have been produced by running the simulator program for various combinations of initial settings. These sets have not consciously been representative of any actual PWR conditions. This situation is not ideal and does not provide the ANN with a realistic test, even if the tests sets used could have been more difficult to predict. To rectify this situation it was decided to test an ANN based PWR primary circuit model with data sets of transient scenarios. The 25 region model, developed earlier in this section, was retained but with the above ANN, pwr8c.nnd, embedded in place of the direct equivalent used previously. Two PWR transients, a primary coolant leak and a downstream steam leak, were modelled with the simulation program. The resulting simulator output files were converted into a form suitable as an input to the ANN. These files were then presented to the ANN based PWR model. The predicted regional temperatures from the ANN together with the initial simulator output for two loops of the PWR primary circuit are shown below, in Figs 6.60 to 6.63.

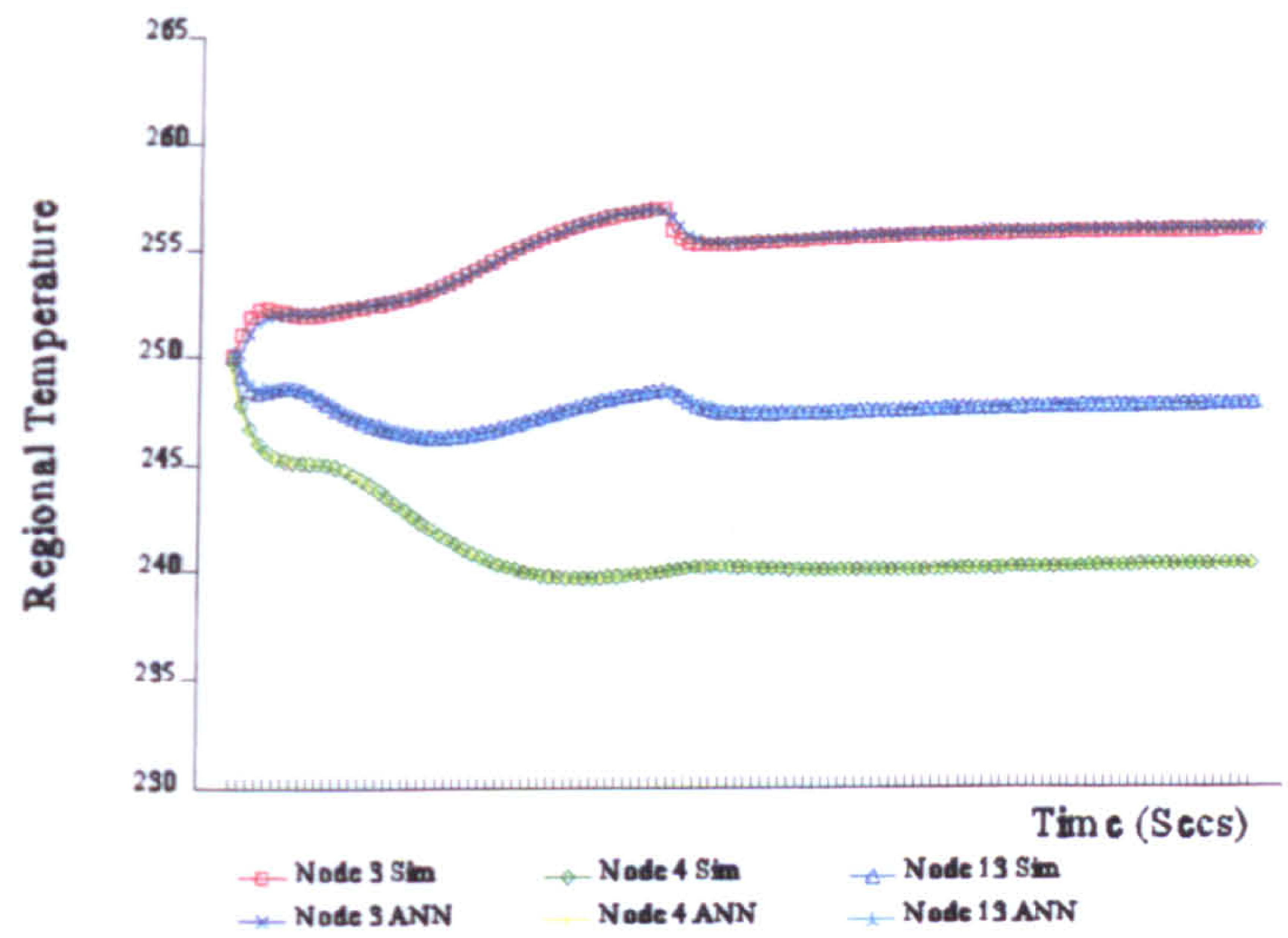


Fig 6.60: Steam Generator Loop Temperatures for Primary Coolant Leak

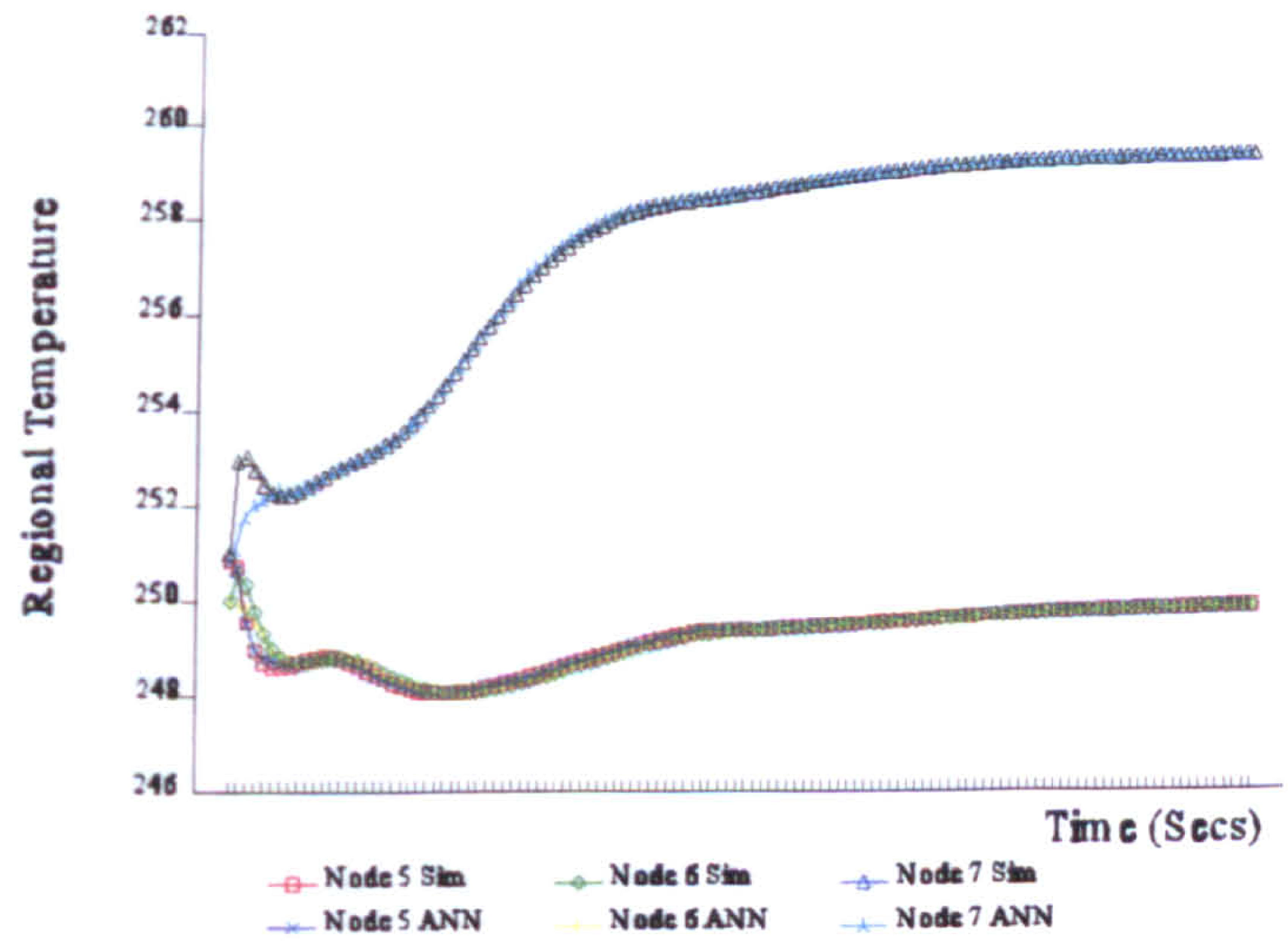


Fig 6.61: Reactor Pressure Vessel Temperatures for Primary Coolant Leak

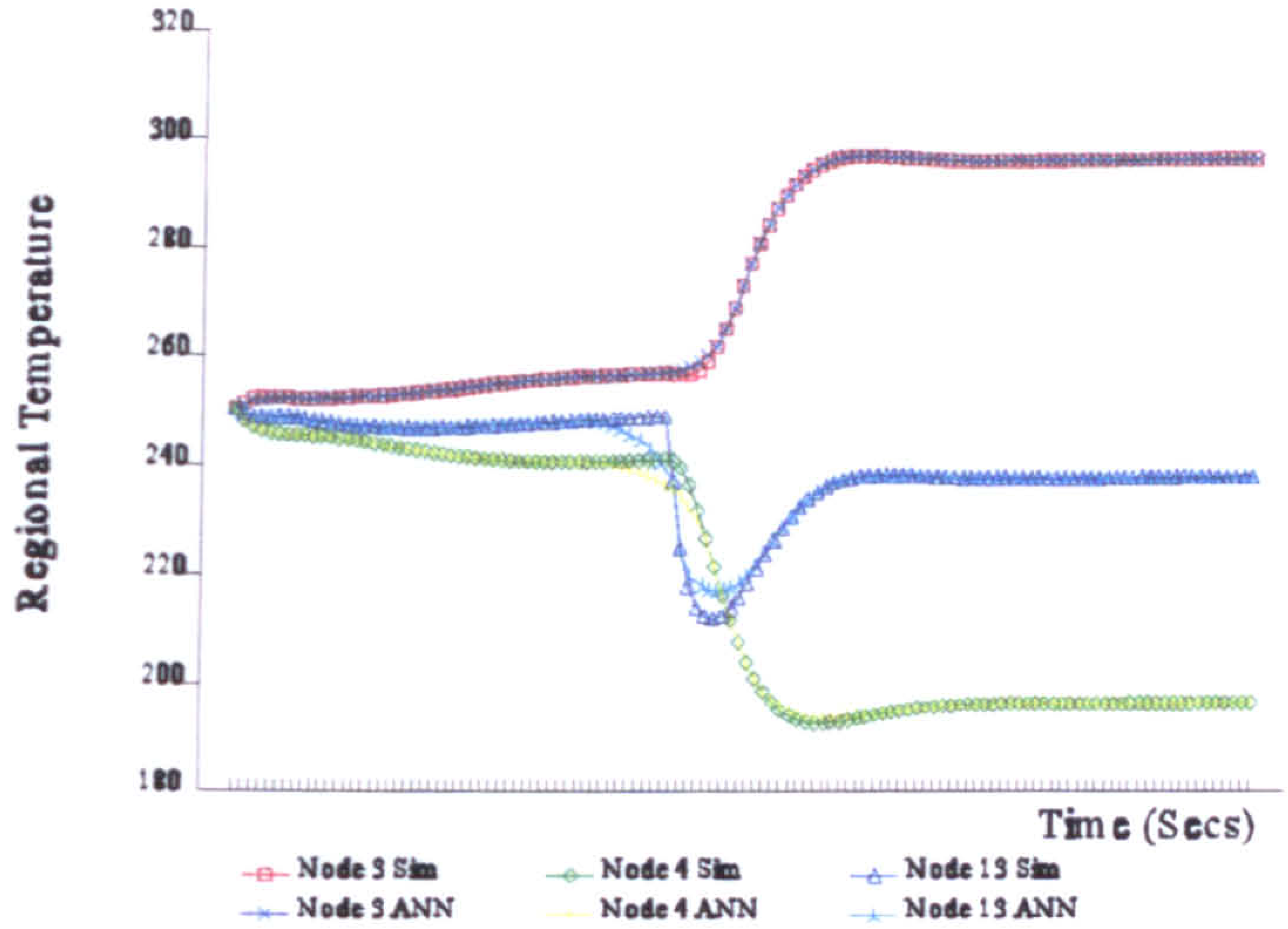


Fig 6.62: Steam Generator Loop for Downstream Steam Leak

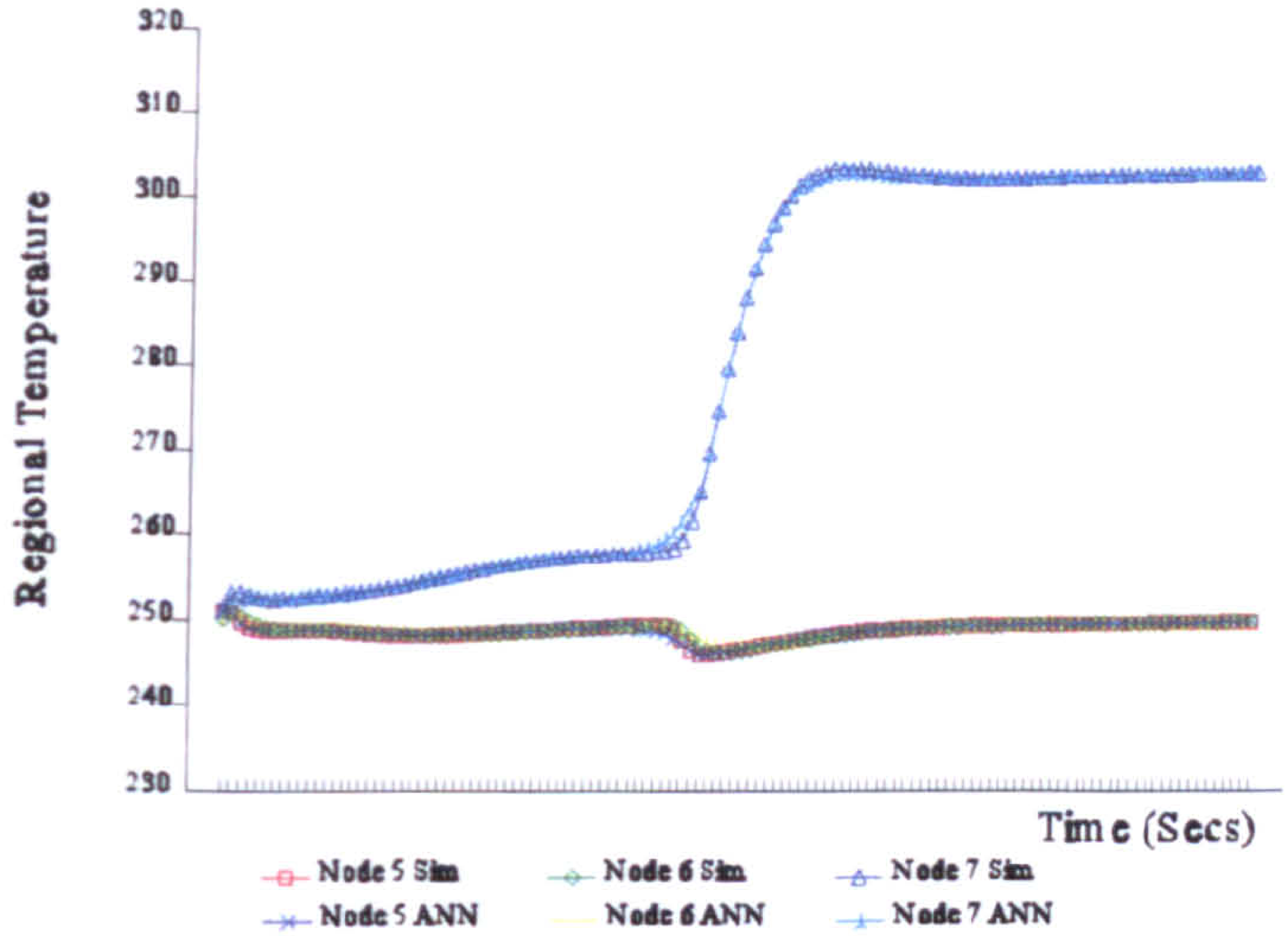


Fig 6.63: Reactor Pressure Vessel for Downstream Steam Leak

These results are a great improvement on those from the previous ANNs in this section. The regional temperatures are well predicted, the only discrepancies being the usual smoothing of acute gradient changes. The two hidden layer approach seems to provide a good solution to predicting PWR primary circuit regional temperatures. This ANN architecture appears to offer the best predictive capabilities in this section.

An iterative approach to the development of this ANN has produced some interesting results but it is possible that a similar result could have been obtained if the two hidden layer ANN had been adopted in the initial investigations. A single hidden layer was used to develop a simple mapping between the inputs and outputs.

structure, in terms of RMS error, was saved. The full results for the ANNs developed for this data set are given in Appendix R. The details of the best ANN developed are given below.

Filename	Architecture	Threshold	RMS Error
pwrsim1h.nnd	118-55-25	Sigmoid	0.0146

Table 6.22: Result for Single Entire Primary Circuit ANN

This ANN was evaluated on an independent test set consisting of a small coolant leak. Using a different sized transient was considered to be a stern test of the prediction ability of the ANN. However, the valve and throttle settings were not altered significantly from the training set. The results for two sections of the primary circuit are given below.

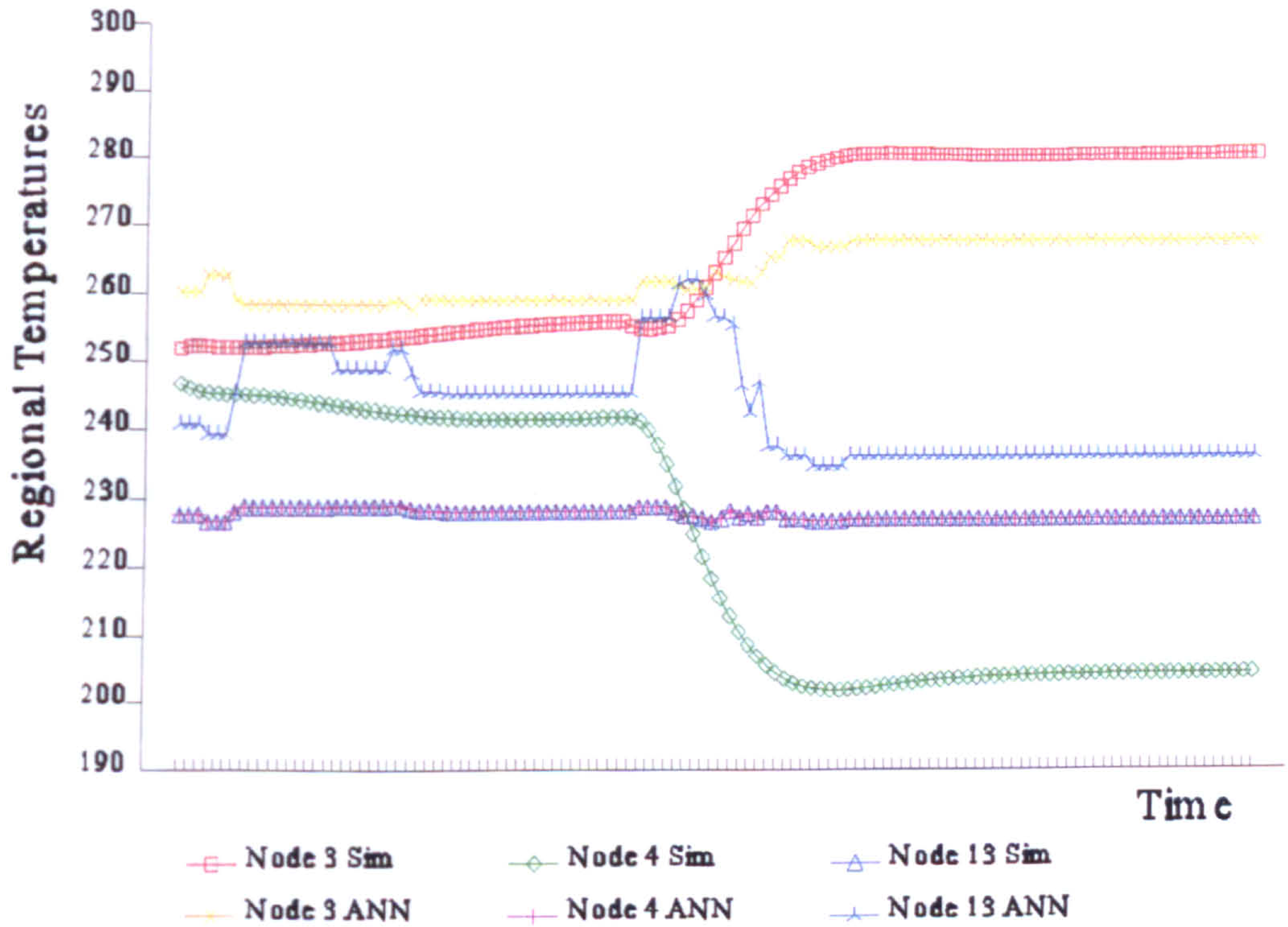


Fig 6.64: Prediction of Steam Generator Loop Temperatures

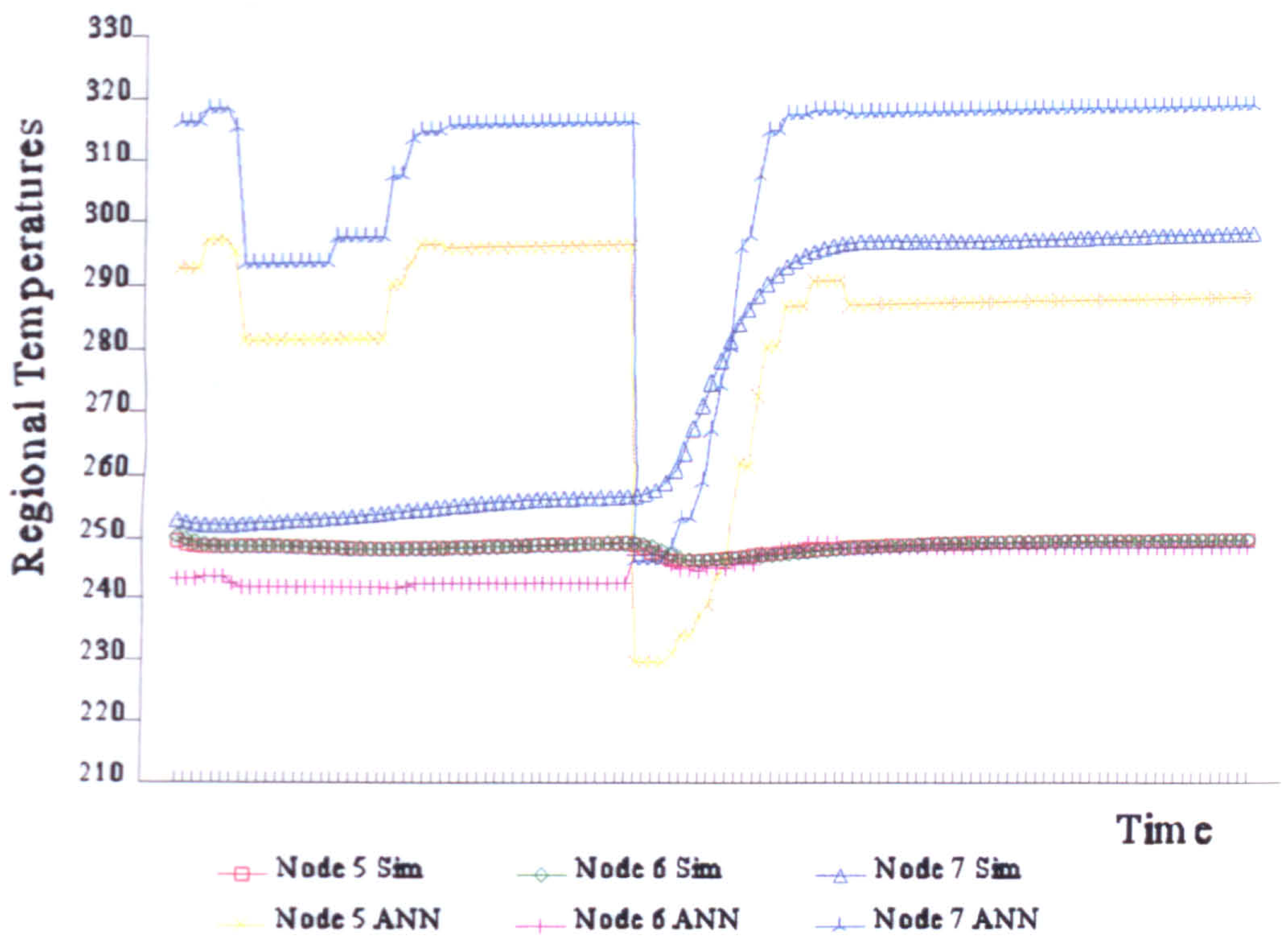


Fig 6.65: Prediction of Steam Generator Loop Temperatures

These results show that while the ANN can predict the general trends of the transient the regional temperatures are not accurately predicted. The relatively low RMS error obtained during the training shows that the data set can be learnt but perhaps either the ANN becomes overtrained or the solution space is too complicated to be accurately modelled. A second tests gave similar results so the size of the ANN and the task required appear to preclude this approach from serious consideration as the prediction component of the advisory system.

6.7 Discussion

The investigations discussed in this chapter arose from considering the number of transients that could be modelled by an ANN. Originally a number of different networks were envisaged for the prediction of reactor transients (Weller et al., 1995). Each ANN would be developed to predict a set of transients, grouped by common features or similar behaviour. However, it was soon realised that the grouping was very difficult. An alternative method would be to develop an ANN system capable of modelling many transients irrespective of their features. This approach has proved successful and the model divided into training and test sets.

Initial T_1 (°C)	T_{in} (°C)	M_{in} (Kg/s)	Q (MW)	No of Cases
326	250	17400	3.4×10^9	200
326	300	17400	3.4×10^9	151
326	286	12500	3.4×10^9	216
326	286	15000	3.4×10^9	161
326	286	20000	3.4×10^9	132

Table 6.10: Initial Data Set for ANN of 1CV PWR

A range of ANNs were developed with this data set. The full details are given in Appendix O. The best network was as follows

Filename	Architecture	Threshold	RMS Error
temp6.nnd	4-7-1	Sine	0.0177

Table 6.11: Result for ANN of 1CV PWR with Steady State Data Set

As seen in the last column of Table 6.10 this approach did not give equally sized data sets for each situation as the cases reached steady state at different times. Different sized sections of the data set could bias the ANN towards the situation with the larger contribution, the first or third scenarios in the above table. A comparison between this ANN and the simulator program for a simple scenario highlights the problem.

A refinement was to ramp the variables between two pre-defined limits. The range of the scenarios modelled is shown below in Table 6.12. The model was also run for a specified time of 100 time steps in order for each scenario to be represented by a data set of equal length and so avoid biasing the ANN. The simulator program was modified to enable the initial and final variable values to be initially entered and then linearly ramp between these two limits for the duration of the simulator run.

Data Set	Initial T_1^k (°C)	T_{in}^k (°C)	M_{in} (Kg/s)	Q (W)
temp11	326	286 → 300	17400 → 12500	$3.4\times10^9 \to 0$
(202 Cases)	326	286 → 250	17400 → 20000	$0 \to 3.4\times10^9$
temp12	326	286 → 300	17400 → 15000	$0 \to 3.4\times10^9$
(303 Cases)	326	286 → 250	17400 → 20000	$3.4\times10^9 \to 0$
	326	286 → 300	17400 → 12500	$3.4\times10^9 \to 3.6\times10^9$
temp13	326	286 → 300	17400 → 15000	$1.7\times10^9 \to 3.4\times10^9$
(303 Cases)	326	286 → 250	17400 → 20000	$3.4\times10^9 \to 1.7\times10^9$
	326	286 → 300	17400 → 12500	$3.2\times10^9 \to 3.4\times10^9$

Table 6.12: Ramped Values of Variables for ANN of 1CV PWR

Note: Each of the above rows consisted of 101 cases. The changes in variable values were all linear and concurrent. For example, in the last entry, in 101 steps the T_{in}^k was linearly ramped from 286°C to 300°C, at the same time M_{in} was linearly ramped from 17400 Kg/s to 12500 Kg/s and Q from 3.2×10^9 MW to 3.4×10^9 MW. The initial value of T_1 was 326°C and all future values were determined by the simulator program. For ANN training the entire data set was divided into training and test sets in the approx. ratio of 2:1. For example the data set temp12 was randomly divided, in the approx. ratio of 2:1, to produce a training set temp12tr.nna and a test set temp12te.nna.

A series of ANNs were developed using these data sets. Full details are given in Appendix O. In each case the data consisted of four inputs, one each for T_1^k, T_{in}^k, M_{in} and Q, and a single output. Initially the output was the compartmental temperature at the next time step, T_1^{k+1}, however it was found that during training the ANN gave a large weighting to the input T_1^k and this variable tended to dominate the network. The ANNs trained to use the previous value of T_1^k as the "Best Guess" for T_1^{k+1} and not consider the other variables to any great extent.

This arrangement is undesirable as although the results obtained were reasonable the ANNs were not establishing relationships between the input variables. The output was revised to contain the difference between T_1^{k+1} and T_1^k, ΔT_1. Using this new form of output a further series of ANNs were developed using the data set temp13 (ie training set temp13tr.nna and test set temp13te.nna). The details of the best ANN developed are

The results show that the ANN based predictions compare well to the simulator output. The lack of accuracy in the predictions from the squaring ANN model are due to the scaling factor applied to the network inputs. This resulted in some values being mapped to the extremities, the least accurate regions, of the output of the ANN square curve.

6.6 Modelling of PWR Primary Circuit

While the above model produced good results in modelling non-linearity in a single region far more information is required for the technique to be useful as a practical component of the proposed advisory system. The modelling of the primary circuit needs to be expanded for more detailed predictions to be produced. The work reported in this section considers developing a suitable ANN based model of the PWR primary circuit.

Two ANN systems are developed to investigate this requirement. The first considers enlarging the number of one compartment models, from the previous section, used to model the PWR primary circuit with twenty-five inter-connected key regions. Each one is then modelled by a one compartment model. The second approach retains the twenty five regions from the previous work but develops a single ANN for all regions. The results are compared and the compartmental modelling approach is found to be a significantly more accurate tool for the prediction element of the advisory system.

6.6.1 Compartmental Modelling of PWR Primary Circuit

The PWR primary circuit, as described in Appendix A, can be represented by twenty-five inter-connected key regions. An extension to the work from the previous section is to represent each of these with a one compartment ANN model. A system of such models , connected in the same manner as the regions in the primary circuit, could then be used to model PWR regional temperatures. The work initially uses the ANN direct equivalent of the energy equation but this is further refined to consider the single ANN model developed at the end of the previous section.

Some basic assumptions were made when considering the interconnections of the PWR model regions. The flow rate into each region was assumed to be the sum of the total upstream flows. Furthermore the input temperature of a region was assumed to be the flow weighted average of the corresponding up-stream temperatures.

The change of temperature was sequentially determined for each region the entire system was then updated at the end of each time step. Initially this time step was one second, the time interval for the initial 1CV model, however this setting was too coarse for visualizing the behaviour of the reactor. The time interval was therefore adjusted to 0.1 seconds. A computer program of this system was developed. The direct equivalent network was coded once and this module was used for the calculations of all regional temperatures. Arrays for the range of possible regional masses, flow rates and heating rates were defined at the beginning of the program and referred to for the relevant information at each iteration. This information was either used as an input to the ANN, as in the case of T_1^k, T_{in}^k, M_{in} and Q, or as weightings to the links in the case of M_1. Individual regional values were also initially defined by referring to the relevant array entry. The calculated temperatures were used as inputs for T_1^k in the next time step together with the values from the look up tables. In this way dynamic situations such as the opening of emergency cooling valves, throttle position changing and leak sizes could be modelled in the system. Alternatively a changing leak size could be represented by a mathematical expression calculated at each time step.

The resulting computer program was capable of modelling a large number of transient scenarios. The opening or closing of valves was accomplished by defining the time of change in the initial set up for each operation. However a more sophisticated approach would have consisted of an entry from the computer keyboard to assimilate a real time change in settings. A number of transients were modelled ranging from normal operating conditions to large LOCAs. The results for these tests were compared to that from the simulator program. The temperatures changes for two key areas of the PWR primary circuit for the same transient are given below in Figures 6.54 and 6.55. A full listing of the computer program is given in Appendix Q. The transient shown is not a typical fault transient but was constructed by providing inputs to produce a significant variation in plant parameters.

This work showed that it was possible to construct an ANN model for the PWR primary circuit. Again the model is artificial with a bespoke threshold function and unique coding for each regional in the circuit. In the above work the mass of liquid in each node is implicitly included as part of the weighting between the ANN hidden and output node. This results in additional and undesirable calculations and data manipulations that could slow down an operational advisory system. It is also restrictive for modelling scenarios where the regional mass of liquid drastically changes, for example downstream of a LOCA. The implicit inclusion of regional liquid mass is considered next.

The first investigation simply included regional mass as a new input in a variant of the 1CV model. An attraction to this approach is that a range of leaks, changing masses, in the system can be easily considered. A range of mass sizes could be included in the ANN training data. This set need not include every possible size of mass, indeed such a set would be unwieldy if it could be constructed. However, a representation of a range of masses could enable a suitable ANN to develop implicit relationships between the inputs and so enable the prediction of the future regional temperature for a vast range of masses. The ramping method developed in the previous section was retained for training data generation.

The data sets for developing these ANNs were produced using the 1CV model from the previous section. The data generation computer program, from the previous section, was modified to include mass in both the output file and ramping features. The new ANN structure consisted of five inputs for T_1^k, T_{in}^k, M_{in}, Q and M_1, and one output for ΔT_1^{k+1}. The following table, Table 6.14, gives the range of values used to produce the ANN training data sets.

Data Set	Initial T_1^k (°C)	T_{in}^k (°C)	Normalised M_{in} (Kg/s)	Q (W)	Normalised M_1 Kg
pwr_cv.nna 2525 Cases	250	2	Various	Various	Various
pwr_cv1.nna	250	250 → 300	0 → 1	0 → 5×10^6	0 → 0.33
102 Cases	250	250 → 200	1 → 0	5×10^6 → 0	0.33 → 0
pwr_cv3.nna	250	250 → 300	0.67 → 1	0 → 5×10^6	0 → 1
508 Cases	250	200 → 300	0 → 1	1×10^6 → 5×10^6	1 → 0
	250	200 → 300	1 → 0	0 → 5×10^6	0 → 1
	250	200 → 300	1 → 0	0 → 5×10^6	1 → 0
	250	200 → 300	1 → 0	5×10^6 → 0	0 → 1
	250	200 → 300	1 → 0	5×10^6 → 0	1 → 0
	250	300 → 200	0 → 1	5×10^6 → 0	0 → 1
	250	300 → 200	0 → 1	5×10^6 → 0	1 → 0
	250	300 → 200	0 → 1	5×10^6 → 0	0 → 1
	250	300 → 200	0 → 1	5×10^6 → 0	1 → 0

Table 6.14: Ramped Values of Variables for Training Primary Circuit ANN Module

The first set, pwr_cv.nna, was an initial investigation to determine the scale of the problem. The full primary circuit model was used to generate the data set as all nodes returned to steady state conditions. A second, smaller training data set, pwr_cv1.nna, only considered two scenarios ramping of all variables. The third set, pwr_cv3.nna, was developed from the results of ANNs trained with the first two data sets. A range of test scenarios were created with a combination of variable rampings. This set was used to develop a number of possible ANNs. For ANN training each of the above data set was divided into training and test sets in the approx. ratio of 2:1. The data set pwr_cv.nna was randomly divided to produce a training set pwr_cvtr.nna and a test set pwr_cvte.nna. The full results for all ANNs produced are given in Appendix R. The details of the best ANN developed with pwr_cv3.nna data is given below.

from two time steps as inputs for the ANN. The variables were still ramped between limits relevant to the PWR primary circuit. To generate the training data sets the ICV program was modified to record previous values together with the latest data. Consequently the ANNs developed with this data set, pwr_cv6.nna, consisted of ten inputs and one output, again for the change in T_1.

Data Set	Initial T_1^h (°C)	T_{in}^h (°C)	Normalised M_m (Kg/s)	Q (W)	Normalised M_1 Kg
pwr_cv6.nna	250	200 - 300	0 - 1	$0 - 5*10^4$	0 - 1
507 Cases	250	200 - 300	0 - 1	$1*10^4 - 5*10^4$	1 - 0
	250	200 - 300	1 - 0	$0 - 5*10^4$	0 - 1
	250	200 - 300	1 - 0	$0 - 5*10^4$	1 - 0
	250	200 - 300	1 - 0	$5*10^4 - 0$	0 - 1
	250	300 - 200	0 - 1	$5*10^4 - 0$	0 - 1
	250	300 - 200	0 - 1	$5*10^4 - 0$	1 - 0
.	250	300 - 200	0 - 1	$5*10^4 - 0$	0 - 1
	250	300 - 200	0 - 1	$5*10^4 - 0$	1 - 0

Table 6.17: Ramped Values of Variables for Training Primary Circuit ANN Module

A range of ANNs were developed using this data. The entire set was randomly divided into training and test sets in the approximate ratio of 2:1. Each ANN developed was trained for 100,000 cycles with the test set being presented every 100 iterations for the last 20,000 cycles. The best network structure, in terms of RMS error, was saved. The full results for the ANNs developed for this data set are given in Appendix R. The details of the best ANN developed are given below.

Filename	Architecture	Threshold	RMS Error
pwr6f.nnd	10-7-1	Sigmoid	0.0268

Table 6.18: Result for Primary Circuit ANN Module

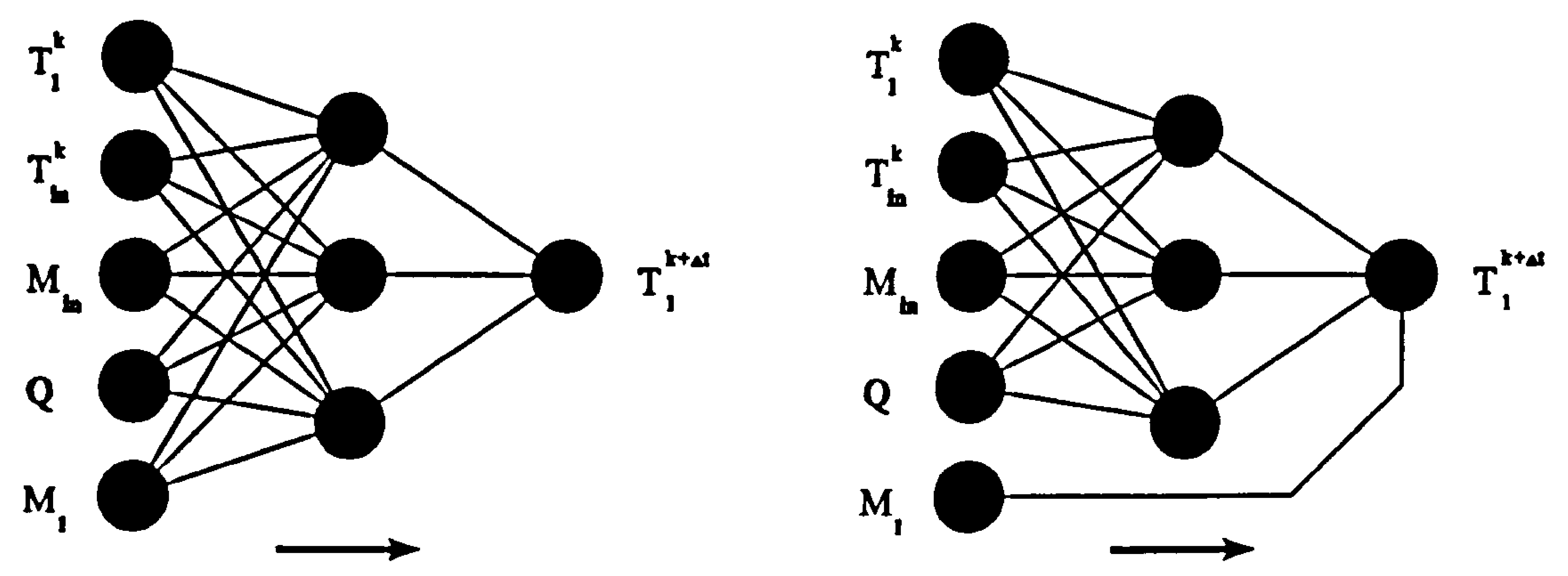


Fig 6.58 a & b: Diagrams of ANNs with Different Internal Connections

A range of ANNs were developed using the structure shown in Fig 6.58b and the previously defined test set pwr_cv3.nna. Again the ANNs were trained each trained for 80,000 cycles with the test set being presented every 100 iterations for the last 20,000 cycles. The best network structure, in terms of RMS error, was saved. The full results for the ANNs developed for this data set are given in Appendix R. The details of the best ANN developed are given below.

Filename	Architecture	Threshold	RMS Error
pwr7h.nnd	5-2-1	Sigmoid	0.0543

Table 6.19: Result for Primary Circuit ANN Module

This result was disappointing with a higher RMS error than the two time step ANN. The number of nodes in the hidden layer of this ANN was unexpectedly low. Examination of the inter-nodal weightings showed that those between the regional mass and the output of the hidden nodes changed very little from their initial, random, setting. The backpropagation training process did not seem to optimise these connections. The previous work on the 1CV model showed that mass of liquid in the region was an important consideration for predicted regional temperature yet this ANN did not support this result. Re-examining the original ANN model identified that regional mass had an effect on the output from the hidden nodes so this must also be included in the PWR regional ANN.

6.6.2 Single ANN Modelling of PWR Primary Circuit

A second method of modelling the primary circuit of the PWR is to develop a single ANN to predict the regional temperatures in a single step. The inputs to this ANN would consist of a set of current regional temperatures, masses, flows, throttle settings and valve positions. The outputs would consist of the predicted regional temperatures for the next time period. The final stage of primary circuit modelling was the development of a single ANN to predict the change in nodal temperatures. A attraction of this approach is that the ANN development is relatively straight forward, however the larger ANN is liable to suffer from the drawbacks previously discussed.

As mentioned the ANNs developed for this task would have the full set of primary circuit variables as inputs and the predicted change in temperature for each node as output. These networks therefore consisted of 118 inputs and 25 outputs. The training data was produced using a full simulator program which used the original energy equation, from Chapter Five, to determine the regional temperatures. The effect of changes in valve settings were also considered in the training data. All the valves were included in the simulator system and set to an initial state. During the transient the valve states could be changed either at defined times or when a set of criteria were satisfied, for example the emergency systems. These changes in valve states were reflected in the corresponding system flows and their affect on the node temperature. Information from three transient scenarios were included in the training data. The type and number of time steps included are given in the following table.

Transient Type	No. of Time Steps
Steady State	120
Primary Coolant Leak	120
Downstream Steam Leak	120

Table 6.21: Single ANN Training Set Components

The Primary Coolant and Downstream Steam leaks were represented by medium sized leaks. The resulting data set was divided into training and test sets in the approx ratio of 2:1. A series of ANNs were trained on this data. Each was trained for 80,000 cycles with the test set being presented every 100 iterations for the last 20,000 cycles. The best network consisting of a two hidden layer ANN has shown that a number of transients can be modelled by this system and accurate predictions made for regional temperatures. The strength of the technique is that each compartment predicts its own next temperature independently. The variables from upstream compartments are only considered at the next calculation and then in terms of the four inputs to the ANN.

The initial requirement for the predictive element of the advisory system was for a fast accurate method of predicting a wide range of PWR primary circuit transients. The above compartmental system may be a method of achieving that goal. Some extreme scenarios may still require a bespoke ANN but the majority of situations could be modelled by a refined version using the above concepts.

Throughout this chapter reference has been made to a simulator program or to established energy conservation equations. An advisory system could be constructed based solely upon this methods however the time required to accurately calculate each iterative regional temperature was felt to be a disadvantage. Although the ANN based solution developed may not prove to be as accurate as the discrete equations the gain in operating speed is felt to offer significant benefits. Once trained the ANN solution provides instantaneous results in operation.

A second observation from these investigations is that the selection of members in the training and test sets is also of importance. A balanced set of cases is essential for the development of a sensible ANN. An ANN trained solely upon steady state conditions can hardly be expected to accurately predict a range of LOCAs. The other extreme of including every possible transient scenario is not only impractical but would lead to unwieldy ANNs. A compromise was developed for this work. The ANN training sets used consisted of combinations of PWR variables ramped between their limits and the resulting effect on the remaining variables. This approach certainly resulted in accurate predictive ANNs but may not be the optimal method of producing training data. It is possible that in the solution to the question posed at the beginning of the chapter a further question on the optimum number of training cases that should be included in the training set needs to be considered.

The compartmental ANN approach has a wider appeal outside the nuclear industry. It is easily modified for any multi-compartment model providing that the links between compartments can be defined. A method of training such a system is presently under investigation. If successful it would allow modelling of compartments whose variables are problematic to measure.

6.8 Summary

The ANN prediction of PWR variables is further examined in this chapter. The question of the number of PWR transients that can be modelled by an ANN, posed at the end of Chapter 5, is addressed. An empirical approach proves unsatisfactory due to the interdependencies of the variables concerned. A modelling approach however proves far more successful. A direct equivalent of a simple PWR circuit is first developed and refined as a non-standard transfer function is initially used. This is an exact equivalent for the energy equation in the simulator program. Furthermore it is proven to be the minimum structure that can model the equation. A number of these basic elements are combined to consider the entire PWR primary circuit. This structure is successfully tested on actual PWR transient scenarios. This system should be capable of predicting all the fault conditions the simulator can model. The ANN version consisting of 25 basic elements to model the PWR primary circuit model is therefore selected for inclusion in the advisory system.

Chapter

7 Diagnosing PWR Fault Conditions

7.1 Introduction

This chapter reports on applying Artificial Neural Networks (ANNs) to diagnose the condition of a Pressurised Water Reactor (PWR). The result of this work is envisaged as the top layer of the proposed operators advisory system. As outlined earlier, in Chapter 4, it is intended that PWR data are input into a suitable ANN system in order to determine the current state of the plant and, if found to be in a fault condition, to ascertain the cause of the abnormality. The results from this diagnostic ANN are then to be conveyed to the operator and also used as inputs for the predictive ANN structure developed in the previous chapter.

The diagnostic work reported in this chapter is a different application of ANNs, compared to the predictive ANNs of the preceding two chapters. The diagnostic ANN is a classification tool that, once trained, uses the non-linear relationships between its variables to determine to which category a particular set of inputs belong. The trained predictive ANN is a multi-dimensional surface mapping that endeavours to fit a given input set to a suitable region of this surface and so establish the next value, or set of values, for that surface. The two approaches are intrinsically different, both in the training data and the outputs required. The form of the output from the diagnostic ANN can be binary. An output node is used for each possible condition and the coding used for the presence or absence of that condition, although in practice a range of tolerance is usually defined for the final outcome. The predictive ANN generally has a continuous multi-valued output for the future value of each variable. The outcome from this ANN may be used as the value for the next time step, it may even be used as an input for the ANN in a feed back loop. These differences in requirements and outputs preclude the development of a single ANN to perform both tasks.

The remainder of this chapter begins with the introduction of some practical considerations for using ANNs for diagnostics. The development of an ANN system for diagnosing a set of PWR conditions is then reported. The results of the investigation are then compared with a second system developed to perform the same task. The performance of these approaches is examined and discussed with the various merits identified. The chapter concludes with some general comments on diagnosing the condition of a complex system with ANNs. The main result from this work is that the developed ANN successfully diagnoses a selection of key PWR transients.

7.2 Practical Considerations

The diagnosing of PWR condition is a classification problem for which an ANN approach is ideally suited. Several such applications were reported in Chapter 2, all of which were deemed to be successful for their particular tasks. The inputs to such a system would be a set of values for plant variables and the output would be the corresponding classification of the PWR characteristic under consideration.

It is important that a realistic set of possible output classes be established. An ANN needs to be trained to diagnose between a sensible group of possible conditions. The inclusion of an unrelated output could cause the training process to create invalid relationships between the variables, it may even cause the ANN to totally fail to converge during training. Using the PWR as an example, a bad set of output conditions would include a diagnosis on the state of a turbo generator when the inputs and remaining output conditions all relate to the primary circuit.

A second key requirement for a classification ANN is that the structure of the ANN be balanced. An ANN designed to distinguish between a large number of possible conditions requires a realistically sized input set to establish the classification. For a extreme example, an ANN designed to classify between twenty possible conditions would probably not be very accurate if only two variables were used as inputs. In general an ANN would, at a minimum, require a similar number of inputs as outputs.

The training data also needs to be balanced to avoid biasing the ANN towards a particular outcome. Each possible condition needs to be equally represented in the training data. An output that dominates the training data will bias the ANN towards that output. For example, consider an ANN designed to determine if a PWR is either operating correctly or in fault, that is one of two possible outputs. If the training set was simply created by collecting data for a period of operation the number of normal readings would greatly outnumber the number of fault readings. An ANN trained on this data set would correspondingly be weighted towards the normal condition. Any abnormalities would therefore probably be incorrectly diagnosed as normal operation, with possible serious consequences. If instead the training set contained an equal number of normal and fault outputs the ANN would be balanced and capable of more accurate diagnosis.

There are situations when it may be advantageous to bias the data towards a particular outcome. For example, a primary coolant leak is one of the more serious fault transient that can occur in a PWR. Every occurrence of such a condition should be identified even at the expense of a similar transient being incorrectly diagnosed as a primary coolant leak. For these situations a higher number of examples of primary coolant leaks could be included in the training data set. The number of cases are not required to dominate the training sets only to bias the training towards the more serious outcome.

A second problem is highlighted by this simple example, namely the availability of data. The number of operational occurrences of a reactor fault are very rare indeed so a balanced ANN training set would be correspondingly small and possibly result in an ANN that fails to converge. In the PWR application this potential problem is resolved by using a computer model to simulate the reactor and collect data for scenarios that are seldom, if ever, experienced on real plant. Other complex environments may not be easily modelled and balanced data collection could remain a serious problem.

The accuracy of the diagnosis from an ANN is also an important consideration. A backpropagation ANN is trained to minimise the error between the actual output and a target output in the training set. Even for well trained ANNs there remains a small error, indeed if the output perfectly matched that of the training set the ANN would probably be over trained and only able to classify members of the training set. This residue error can manifest itself in one of two ways, it can be evenly distributed throughout the ANN or it can all be associated with just a few nodal weights. The latter possibility can easily result in incorrect classifications even if a threshold level is used for both membership and non-membership of each class of output. A corollary to this idea is that data points not correctly identified by a set of different ANNs could be classified as outliers.

A second ANN trained on the same data set would, because of the initial random nodal weightings, result in a different set of internodal weightings. Even if this second ANN had the same error value it is very unlikely that each network would give exactly the same results for a particular input although, with the threshold level, the final diagnosis may well be the same. This feature enables a more accurate ANN system to be constructed using different ANNs trained to perform the same task. If a number of such ANNs were constructed they could be used in parallel to produce a more accurate overall diagnostic system. Each ANN would use a set of values of PWR variables to perform an individual diagnosis of plant condition. The results from these ANNs would then be examined and the final diagnosis determined. The diagram, Figure 7.1, below illustrates the concept.

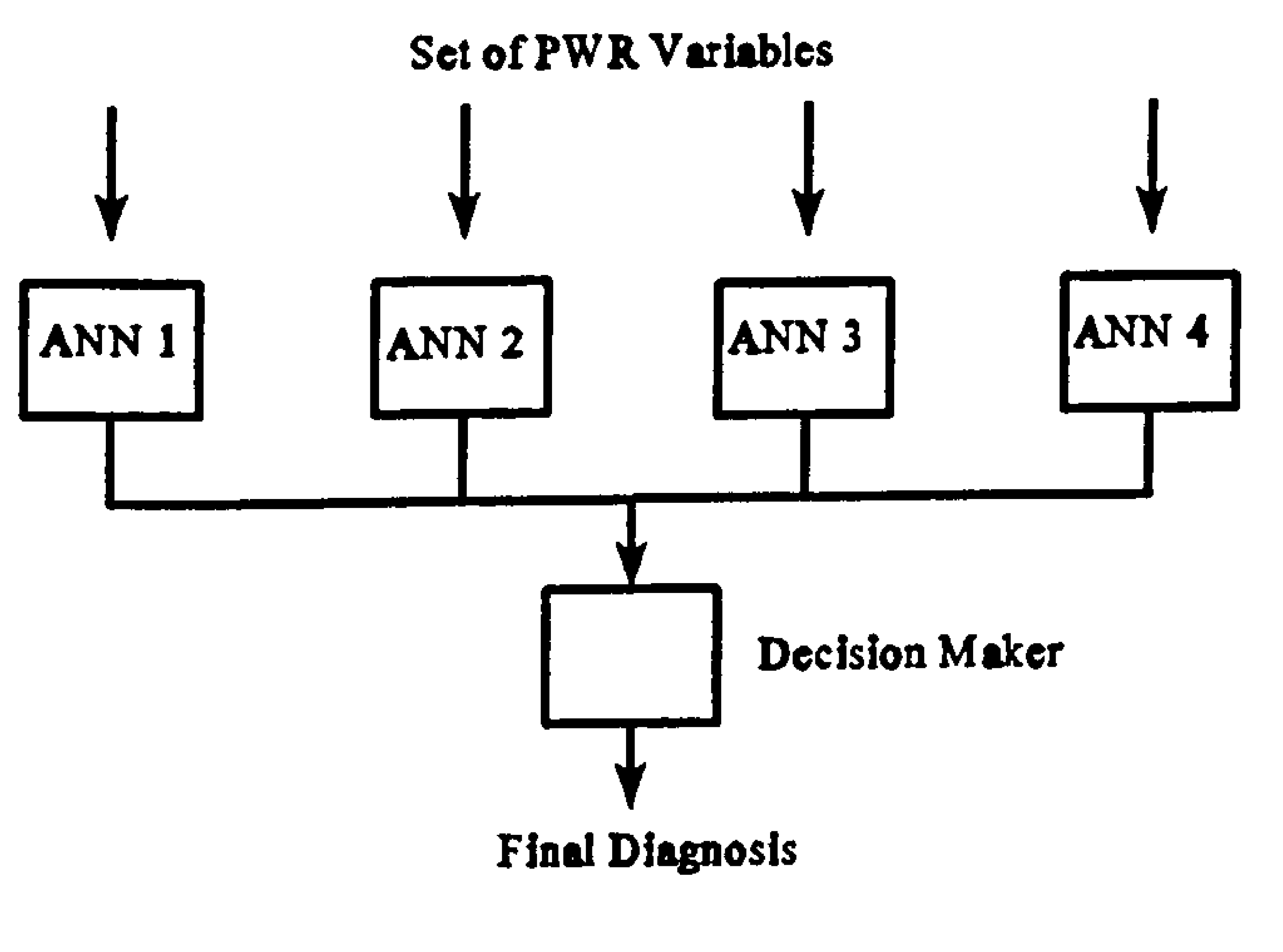


Fig 7.1: Multi-expert ANNs

The determination of the final output could be one, or more, of several methods.

1) A simple voting strategy, in which the outputs from each ANN are combined and the highest scoring output is used as the diagnosis.

2) Statistical methods such as Jordan (1994).

3) The outputs from the ANNs could be used as inputs to a further AI technique, another ANN or an expert or fuzzy system.

The diagnostic ANN structure can be extended to produce a hierarchy of diagnostic tools.

Each lower level could be developed for more detailed diagnosis. The outcome from one level of the hierarchy would be an input in a lower level and used to obtain further insight into the nature of the transient, for example location of leak. The following diagram illustrates the concept.

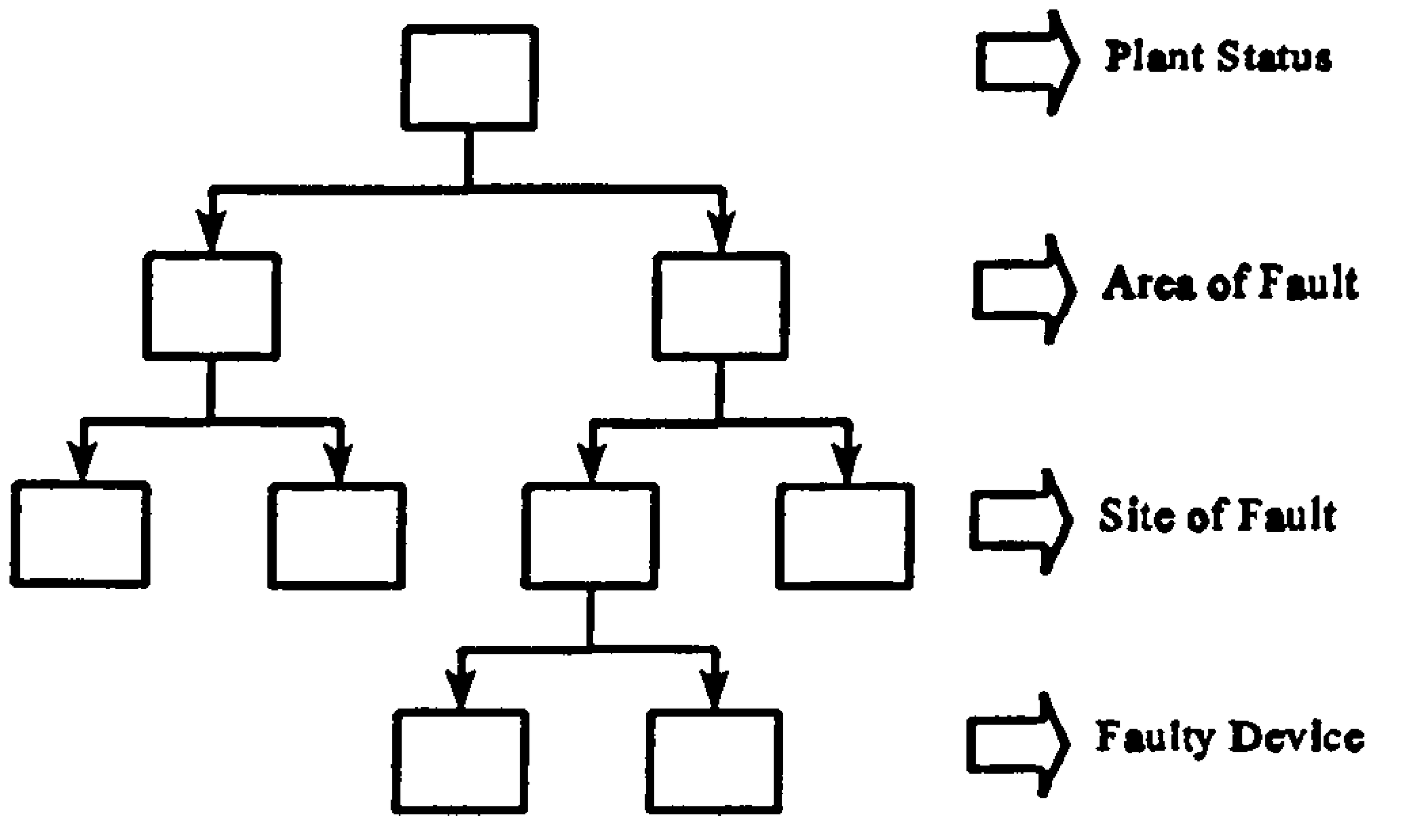


Fig 7.2: Hierarchy of Diagnostic ANNs

This approach may be restricted by the availability of plant data. This limitation may be caused by both practical and economical considerations. It is not always possible to position instrumentation in all areas of a PWR, the reactor core for example. A leak in a pipe may be very accurately located but the cost of installing and positioning the required number of sensors may prohibit implementation. A further limitation to the degree of diagnosis that could be performed is the availability of ANN training data. The computer simulator program would require extensive modification to produce such detailed leak information.

The confidence that should be applied to the ANN diagnosis of PWR condition depends on several important considerations. The accuracy of the data being input into the ANN is affected by the precision of the instrumentation used to record the data, noise from surrounding systems and interference collected by the wiring connecting the instruments to the computer running the advisor program. The nature of ANN training also results in an implicit error always being present in any output node. This is reflected by the value of the outputs not being purely binary, a '1' or '0', but as a range between these limits. To compensate for all these possible discrepancies diagnostic ANN outputs are usually post-processed to obtain a binary result. An upper limit is defined above which the output is a '1'. Similarly a lower limit is defined for a '0' output. For these investigations the upper limit has been set as 0.85 and the lower limit as 0.15. The confidence given to the results of the diagnostic ANNs is further discussed later in the Chapter, in Section 7.4.

For a diagnostic ANN to be useful in the advisory system it would be required to produce a rapid accurate assesment of the PWR condition. Two classes of ANN features, dynamic and static, can be identified as effecting ANN accuracy. The main dynamic condition is the ANN training algorithm. The static conditions are members of the training set, the number of time steps of data and the learning parameters of the algorithm, such as transfer function and momentum factor. A successful ANN would require an optimum combination of these factors, a suitable combination of static conditions together with an acceptable set of inter-nodal wieghtings, established by the training process. An iterative approach is adopted for this purpose where a set of ANN structures are trained for a selected combination of static features.

7.3 Implementation

This section describes the work undertaken to develop a diagnostic ANN. The approach adopted and the results obtained are presented and discussed. It was intended to develop the best possible diagnostic ANN for the defined task and within the limitations of data availability. A method of achieving this could be by varying both the number of time steps of data and variables used for inputs until a suitable ANN is developed. The starting point would be a training set of a large input data set of PWR variables at one time step. If this format proved unsatisfactory the training data would be modified to include more time steps of information and a different number of variables.

A complex system such as a PWR primary circuit has a large number of possible faults and transients that could occur. An ANN developed to diagnose all of these in detail would correspondingly be large and with all the associated problems of training large ANNs. A more practical approach would be to consider a smaller set of generic transients and develop an ANN to diagnose these classes of faults. A set of important PWR transients was identified to use in all the ANNs. This set was designed to include the most serious conditions and also to provide some scenarios with similar characteristics in order to compare the ability of an ANN to differentiate between alike outputs.

The first phase was to identify the transient conditions on which the ANNs would be trained. These transients would be the outputs from the ANNs. While there are potentially a large number of fault conditions that may occur, they can be classified into one of six generic forms.

1) Normal operating conditions

2) Primary coolant leak

3) Downstream steam leak

4) Throttle opening transient

5) Rod drop

6) Group Drop

The normal operating conditions category is required to identify the "no fault" situation. A separate, dedicated output is used instead of considering a zero output from all the remaining fault condition nodes to signify normal conditions. A separate output allows a positive identification of the reactor condition to be made as opposed to implying normal conditions from a lack of identification of the other transients, for example an actual transient may exist but is not satisfactorily identified by an ANN without a normal condition output, the outputs would all be zero but the PWR is definitely not operating normally. The addition of the extra output clarifies the situation because although no particular transient is identified the normal operating node output should also be zero indicating an unidentified transient condition.

As already stated a primary coolant leak poses a serious threat to a PWR and should be identified as quickly and accurately as possible to avoid even more serious consequences. The downstream steam leak is also a serious transient condition. Although not directly associated with the primary circuit the best indications of the occurrence of this form of leak are given in the primary circuit. The downstream steam leak is similar in appearance to a normal operating condition of throttle opening. This transient was therefore included in the list in order to compare the accuracy of identification between the two conditions. Two reactivity problems were included in the transient list. A single rod drop, the lowering of a control rod into the core, and a group drop, the rapid lowering of a set of rods into the core during a partial scram, were included in the list to provide further comparisons of similar transient conditions.

7.3.1 Using Full Data Set for ANN Development

A PWR simulator program was then used to model the selected transients and produce a data set of PWR variables. This program was run for 120 seconds with recordings every second. The required transient was instigated after an initial stabilising period when the reactor settled to steady state conditions. The full data set consisted of 67 variables. The inputs consisted of temperature in the reactor regions, throttle, rod, pump and valve settings and various PWR parameters. The nature of these variables were either real numbers, binary valued or a percentage of the maximum available value. The set of variables selected are easily measured and recorded in an actual PWR. The following table, Table 7.1, shows the form of the variables used for the training data.

Number of Inputs	Type of input	Form of Input
25	Nodal Temperature	Real Number
4	Throttle Setting	Binary
4	Rod Position	Percentage
12	Flow Rate Settings	Binary
11	Valve Settings	Binary
1	Tav	Real Number
1	Neutron Population	Real Number
1	Pressuriser Pressure	Real Number
1	Pressuriser Level	Real Number
6	Power Levels	Real Number
1	Start Up Rate	Real Number

Table 7.1: Details of Input Data Set

These results were used to produce all the training and test sets for subsequent ANN development to ensure consistency of data. The simulator outputs were concatenated and modified by the addition of a suitable output set and to reflect the number of time steps required to produce ANN training and test sets. The ANN output set consisted of one output for each of the six conditions. A binary coding was used with a '1' signifying the presence of the transient and similarly a '0' for the absence. The full data set was randomly divided into independent training and test sets in the approximate ratio of 2:1.

A series of ANNs were then trained and tested on these data sets. The backpropagation learning algorithm was used with 10% gaussian noise for the training together with various combinations of transfer function and momentum rate. Each ANN was trained for 80,000 cycles with testing every 100 cycles with the independent test set, the best network being saved. The best ANN in terms of RMS error was then used for more detailed tests and comparisons. The full results are given in Appendix M.

The details of the first training set are given below, in Table 7.2.

Training Set	Normal	TOT	PCL	DSL	RD	GD
test1.nna	120	120				

Table 7.2: Details of First Training Sets

Key: TOT - Throttle Opening Transient, PCL - Primary Coolant Leak, DSL - Downstream Steam Leak, RD - Rod Drop, GD - Group Drop.

The training set, test1.nna, consisted of two outputs, normal operating and throttle opening, to gain some initial experience on the expected performance of the ANN training. An ANN consisting of one hidden layer of 20 nodes produced a RMS error of 0.1897 when trained of test1.nna. The plots of the respective ANN outputs for each case are shown in figures 7.3 and 7.4.

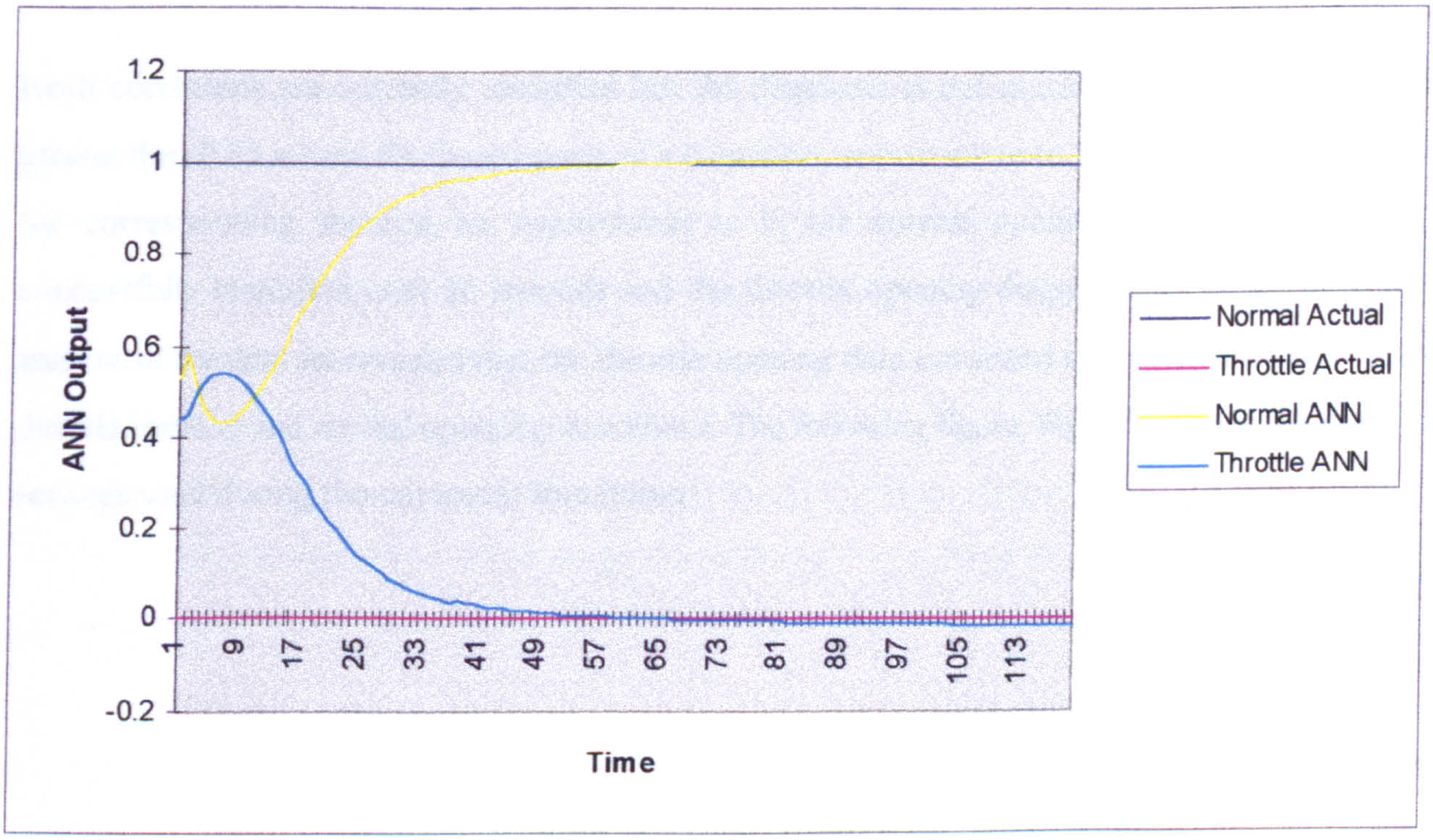


Fig 7.3: Normal Operating Conditions

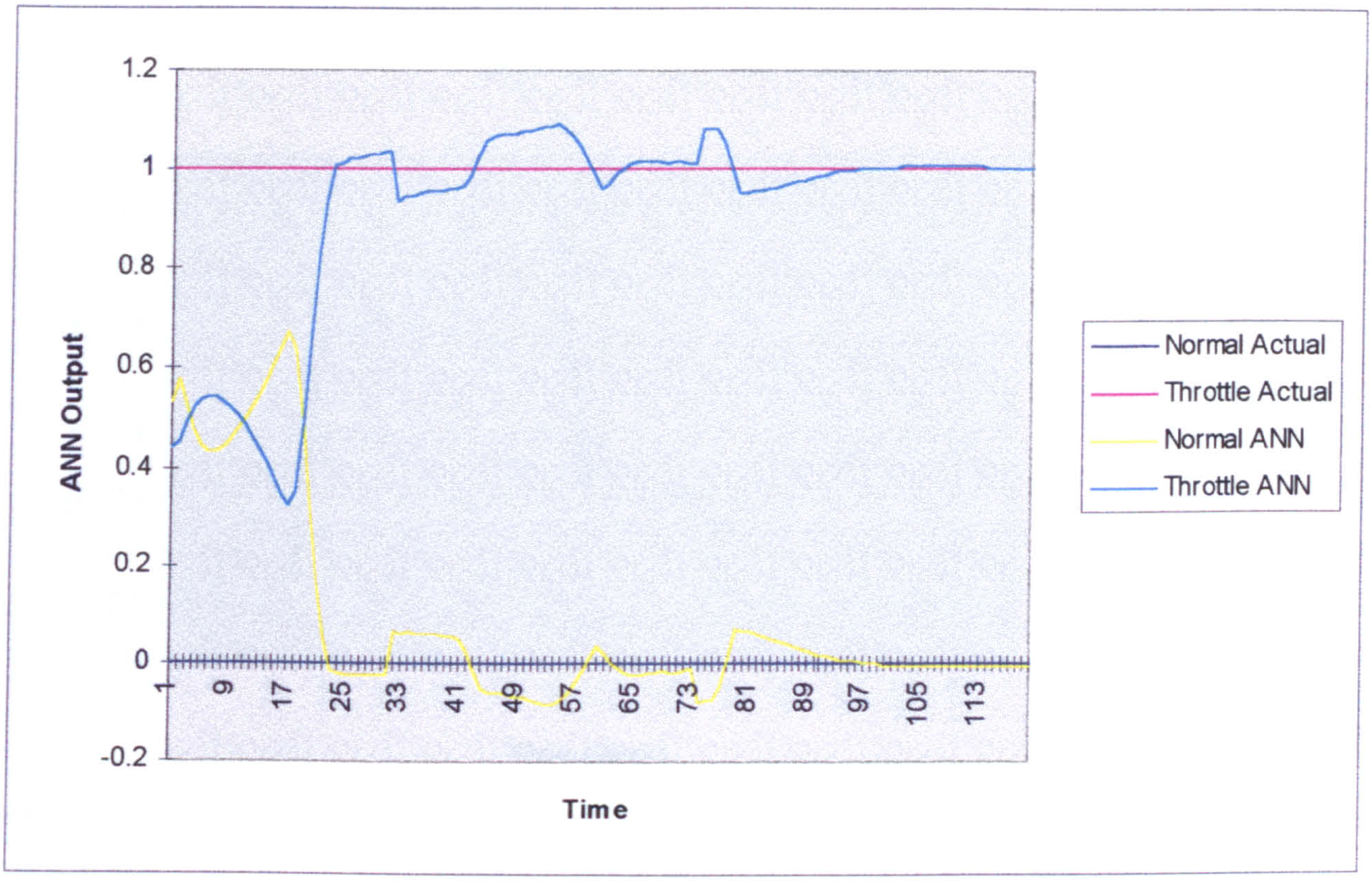


Fig 7.4: Throttle Transient

Both conditions are correctly identified but the diagnosis is not quick. If a threshold level of greater than 0.85 is used for the existence of a condition, approximate to 1, and less than 0.15 for the corresponding absence, an approximate to 0, the normal operating conditions are not successfully identified until 26 seconds and the throttle opening diagnosed in 21 seconds. An analysis of the data set revealed that the throttle opening data consisted of a combination of both throttle opening and normal operating conditions. The following figure, Fig 7.5, shows the throttle settings used during the computer simulation.

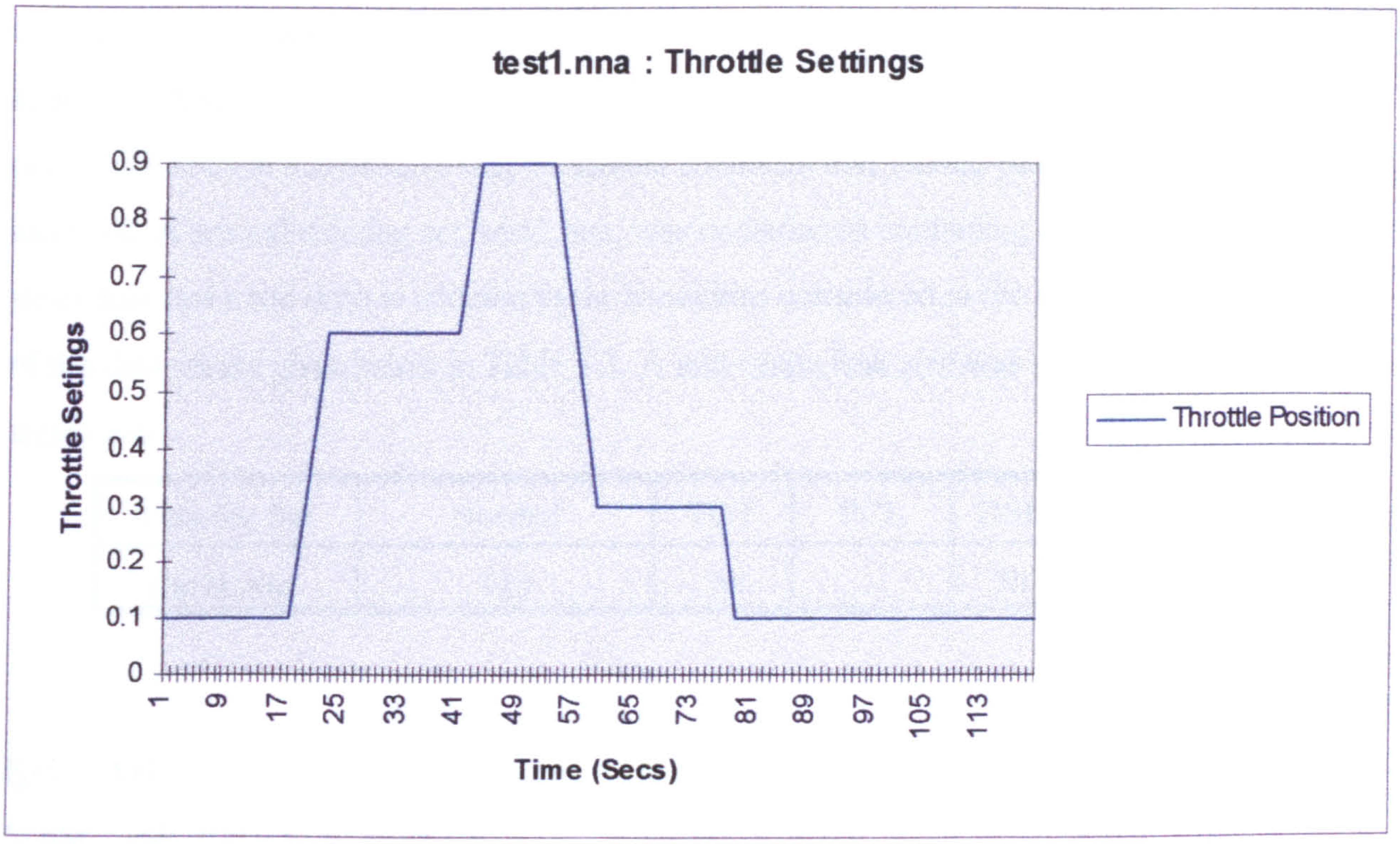


Fig 7.5: Transient Throttle Settings

The ramped periods designate changing throttle settings whilst the level sections signify constant throttle setting. Both situations were classified as a throttle opening transient in the ANN output set. During training the ANN could have established the relationship that if the throttle setting is greater than 0.1, the initial throttle position used for the normal operating conditions, then the PWR was in a throttle opening condition. This would be an incorrect relationship, the actual relationship should be developed for when the throttle setting is changing. This characteristic will be addressed in the future training sets when only the periods of changes in throttle settings will be classified as a throttle opening transient. Periods of decrease in throttle setting were also classified as a throttle transient. This approach resulted in a reduced number of cases that were classified as throttle opening. To compensate for this a further number of different simulation runs were made in which the throttle position was changed at different rates and times.

A further alteration to the data sets was to classify each period prior to the instigation of the leak as normal operating conditions. However this modification created a bias towards the reactor settling period as the same values for the PWR variables occurred for each transient condition and so duplicated the information in the training set. These stabilising periods in the datasets were therefore removed with the exception of the normal operating conditions and so avoiding biasing the ANN output towards normal conditions.

Overall the results were considered to be acceptable enough to encourage further work with this approach. A series of ANNs were developed with extra conditions included. By developing the diagnostic ANN in this iterative way a transient condition that caused poor results could be easily identified. A second training set, test2.nna, was constructed containing data from a downstream steam leak and a rod drop in addition to the conditions considered in the first data set. The details of this data set are given below in Table 7.3. A mid-range leak size was used for the downstream steam leak.

Training Set	Normal	TOT	PCL	DSL	RD	GD
test2b.nna	120	70		70	68	

Table 7.3: Details of Diagnostic Training Sets

Key: TOT - Throttle Opening Transient, PCL - Primary Coolant Leak, DSL - Downstream Steam Leak, RD - Rod Drop, GD - Group Drop.

A series of backpropagation ANNs with differing structures were trained with this training set for 80,000 cycles with testing every 100 cycles with the test set, the best network being saved. The best network had a RMS error of 0,1373. The full data set was then presented to this ANN to explore the accuracy and distribution of error. The results are given in Figures 7.6 to 7.9.

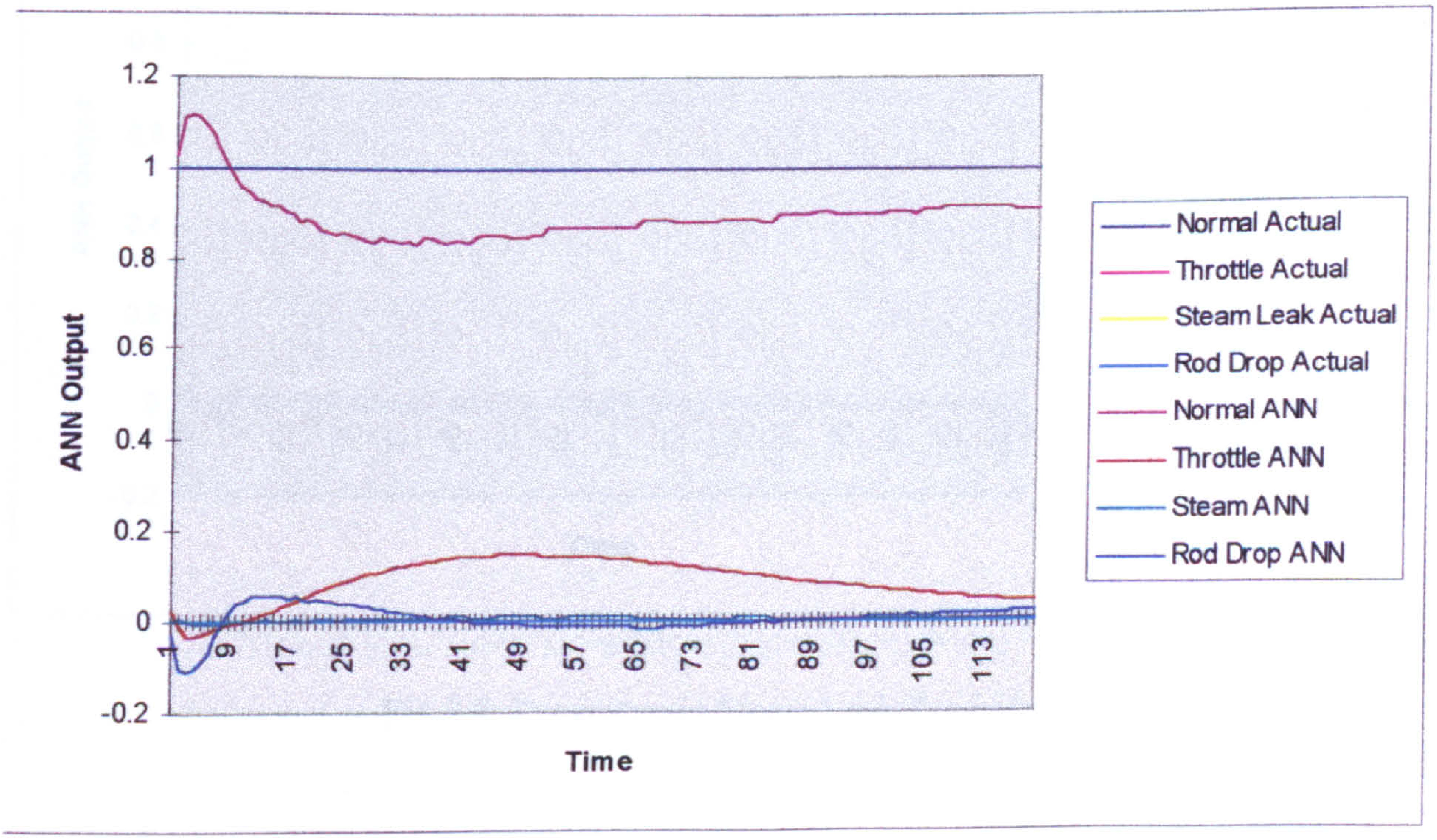


Fig 7.6 Normal Operating Conditions

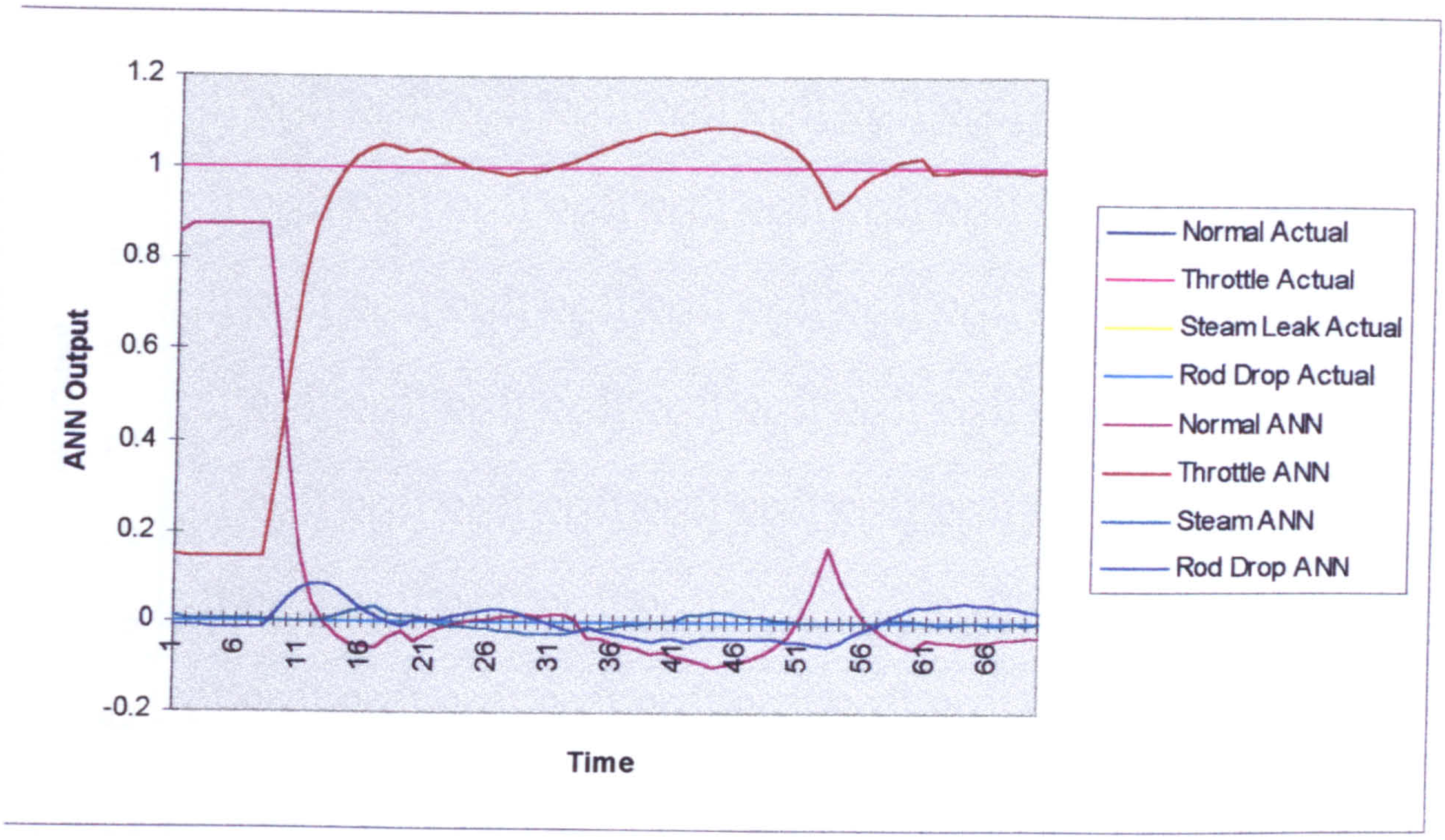


Fig 7.7: Throttle Transient

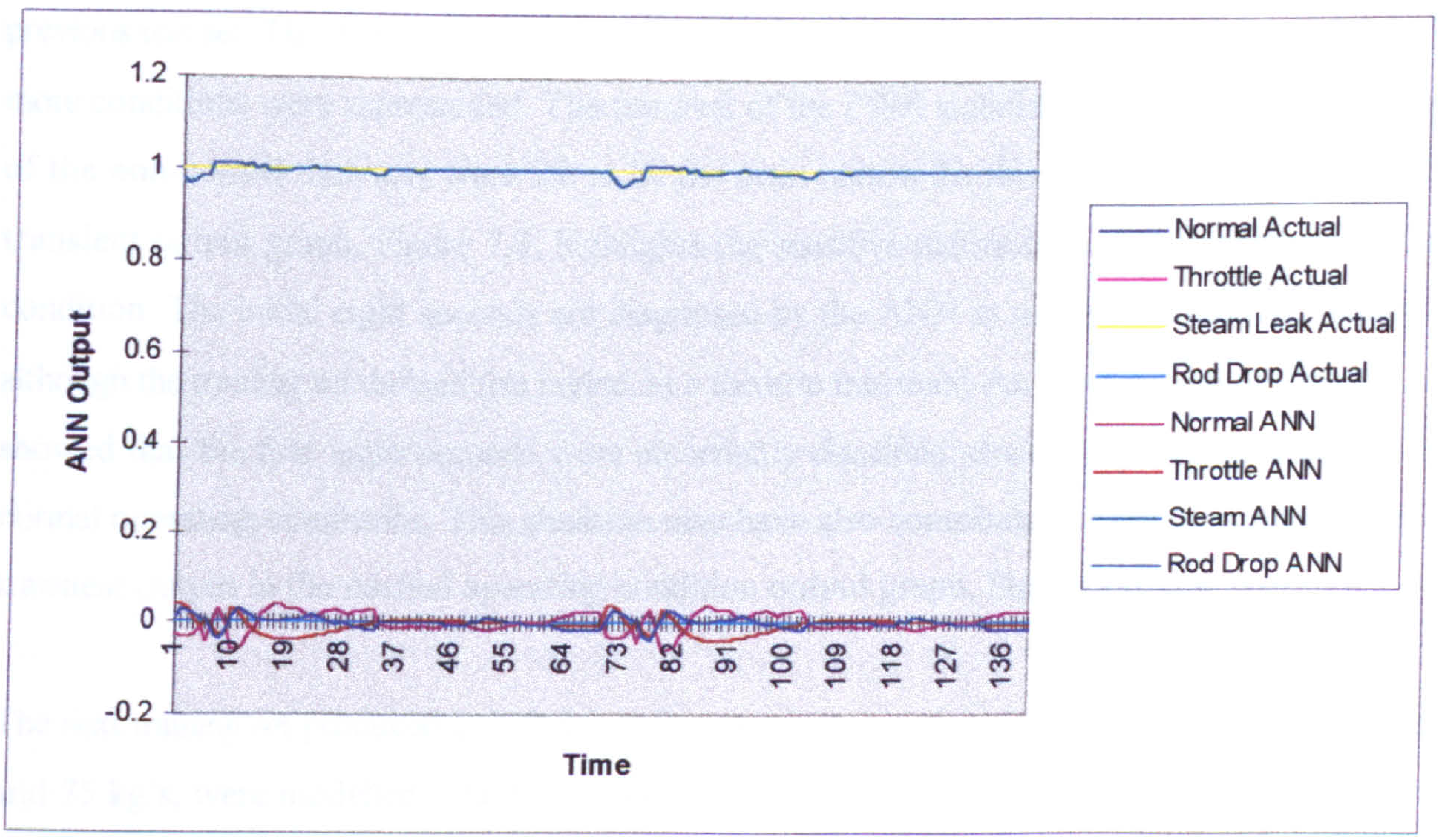


Fig 7.8: Downstream Steam Leak Transient

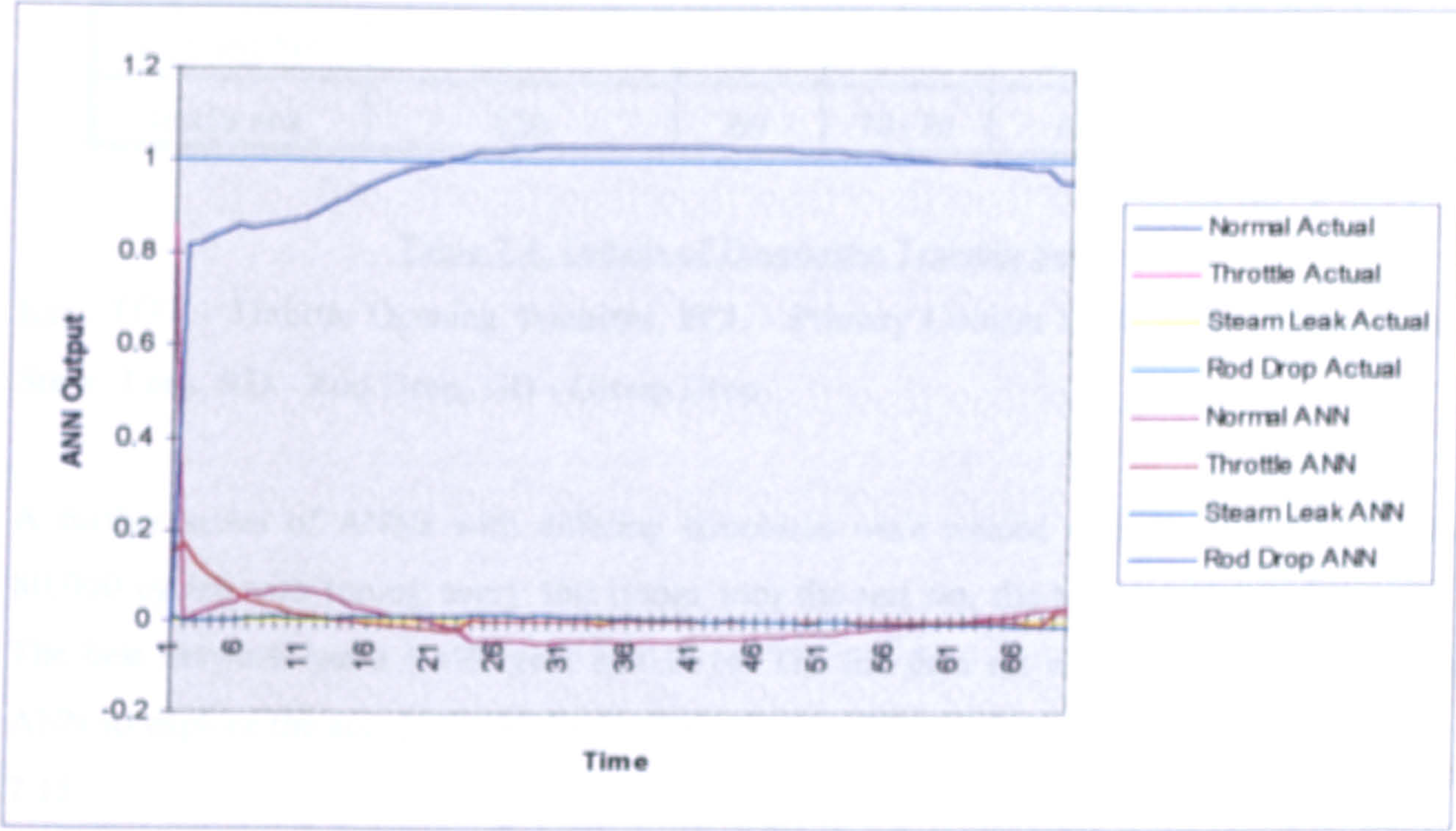


Fig 7.9: Rod Drop Transient

Each output is correctly identified and in a faster time than the best ANN trained with the previous test set. The RMS error was considerably lower than the previous network even though more conditions were represented. The removal of the PWR stabilising period from the portion of the non-normal data sets were felt to be the main reason for this improvement. The throttle transient output graph, Figure 7.7, highlights the sensitive nature of diagnosing this particular condition. The initial eight seconds are diagnosed by the ANN as normal operating conditions, although the training set defined this period as a throttle transient. An examination of the data set showed that the first eight seconds were incorrectly classified as a throttle transient instead of normal operating conditions. This situation may have also contributed to the noticeable throttle transient output in the normal operating condition output graph, Figure 7.6.

The next training set produced included data from a primary coolant leak. Two leak sizes, 25 kg/s and 75 kg/s, were modelled. The two sizes were included to explore the ability of the ANN to extrapolate of other leak sizes and to investigate the biasing of the ANN towards the more serious condition. The size of each data set members is given below in Table 7.4.

Training Set	Normal	TOT	PCL	DSL	RD	GD
test5a.nna	120	69	70+70	68	68	

Table 7.4: Details of Diagnostic Training Sets

Key: TOT - Throttle Opening Transient, PCL - Primary Coolant Leak, DSL - Downstream Steam Leak, RD - Rod Drop, GD - Group Drop.

A further series of ANNs with differing structures were trained with this training set for 80,000 cycles with testing every 100 cycles with the test set, the best network being saved. The best network had a RMS error of 0,0516. The full data set was then presented to this ANN to explore the accuracy and distribution of error. The results are given in Figures 7.10 to 7.15.

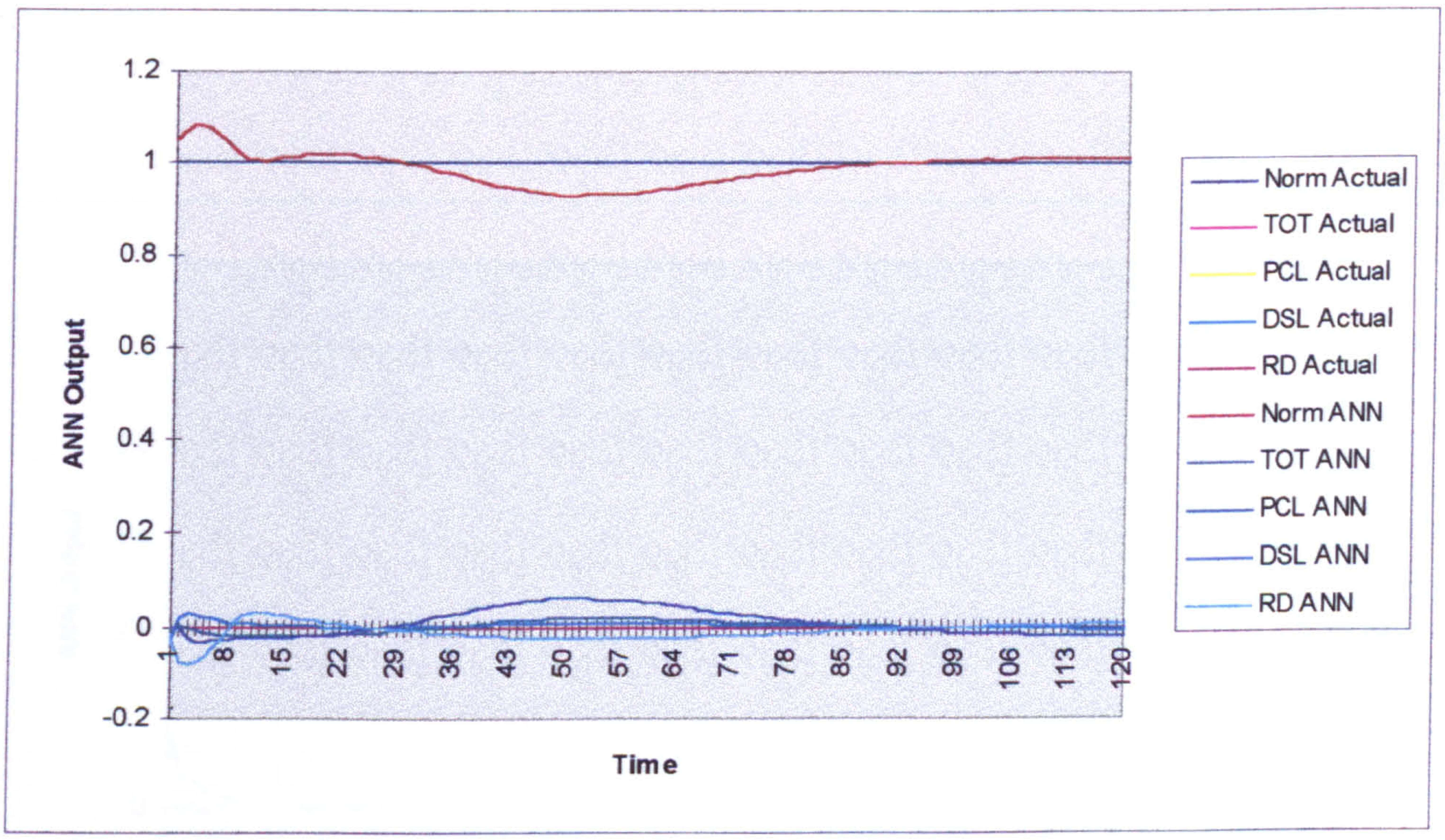


Fig 7.10: ANN Result for Normal Operating Conditions

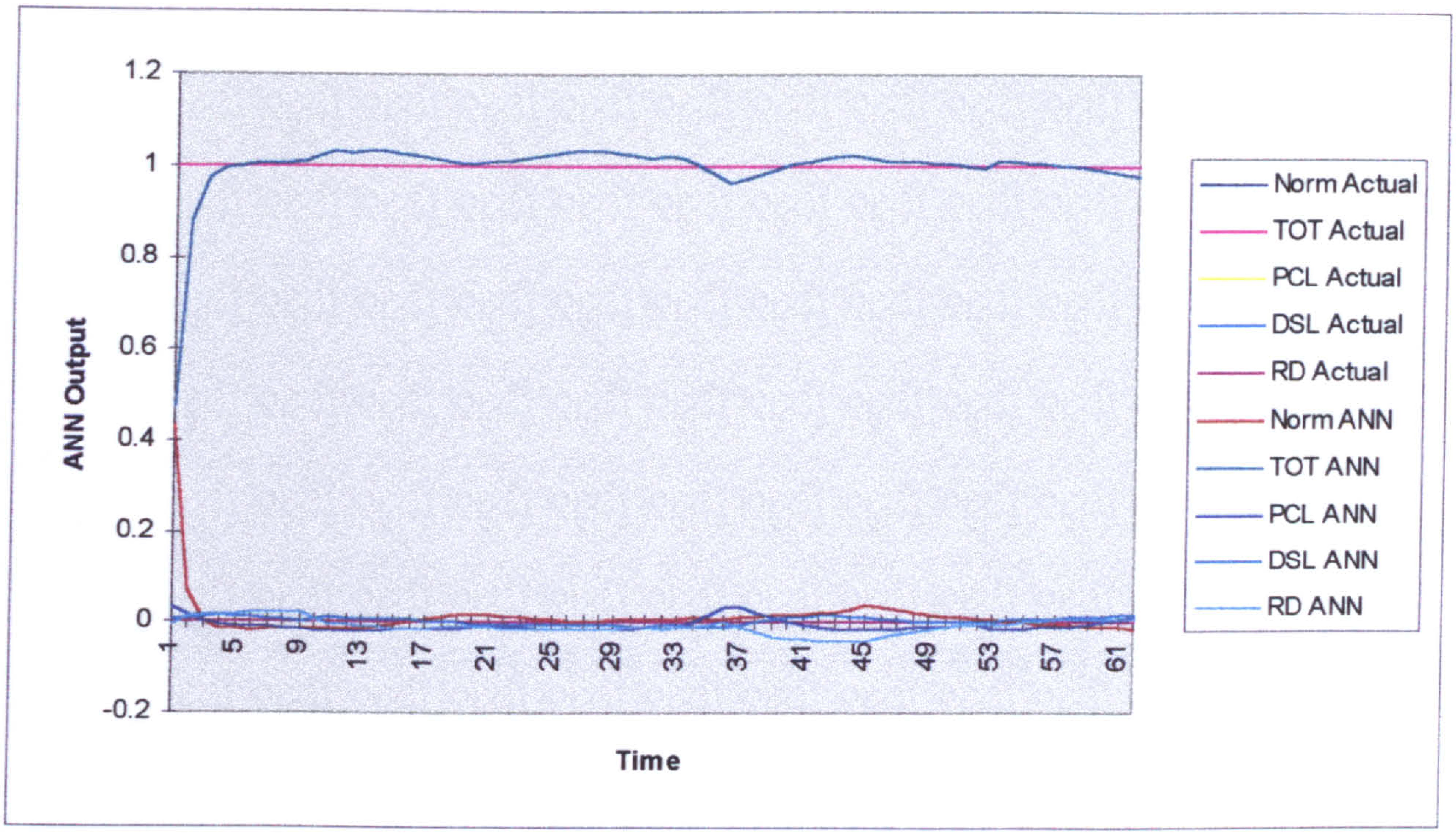


Fig 7.11: ANN Result for Throttle Transient

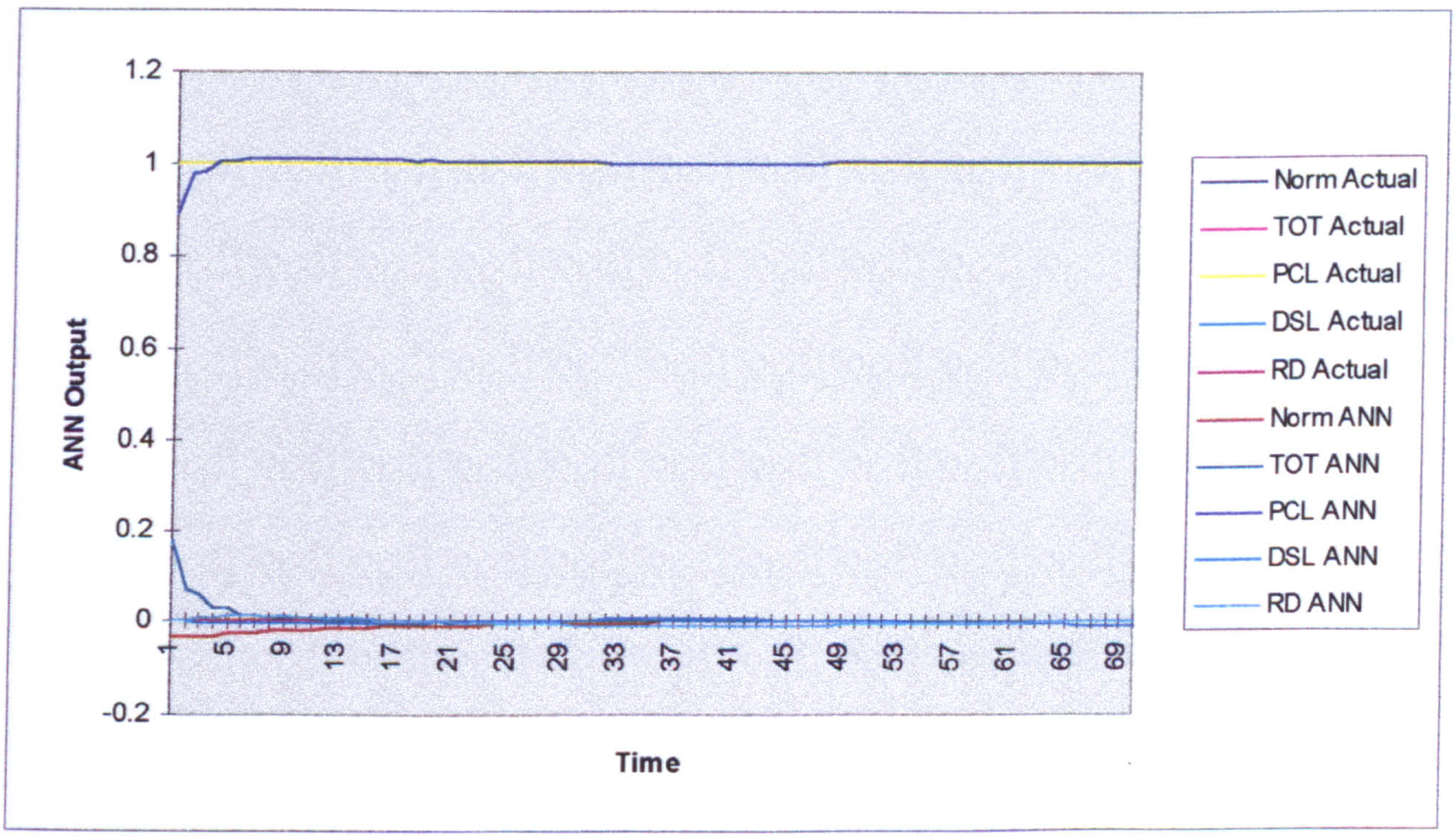


Fig 7.12: ANN Result for Large Primary Coolant Leak

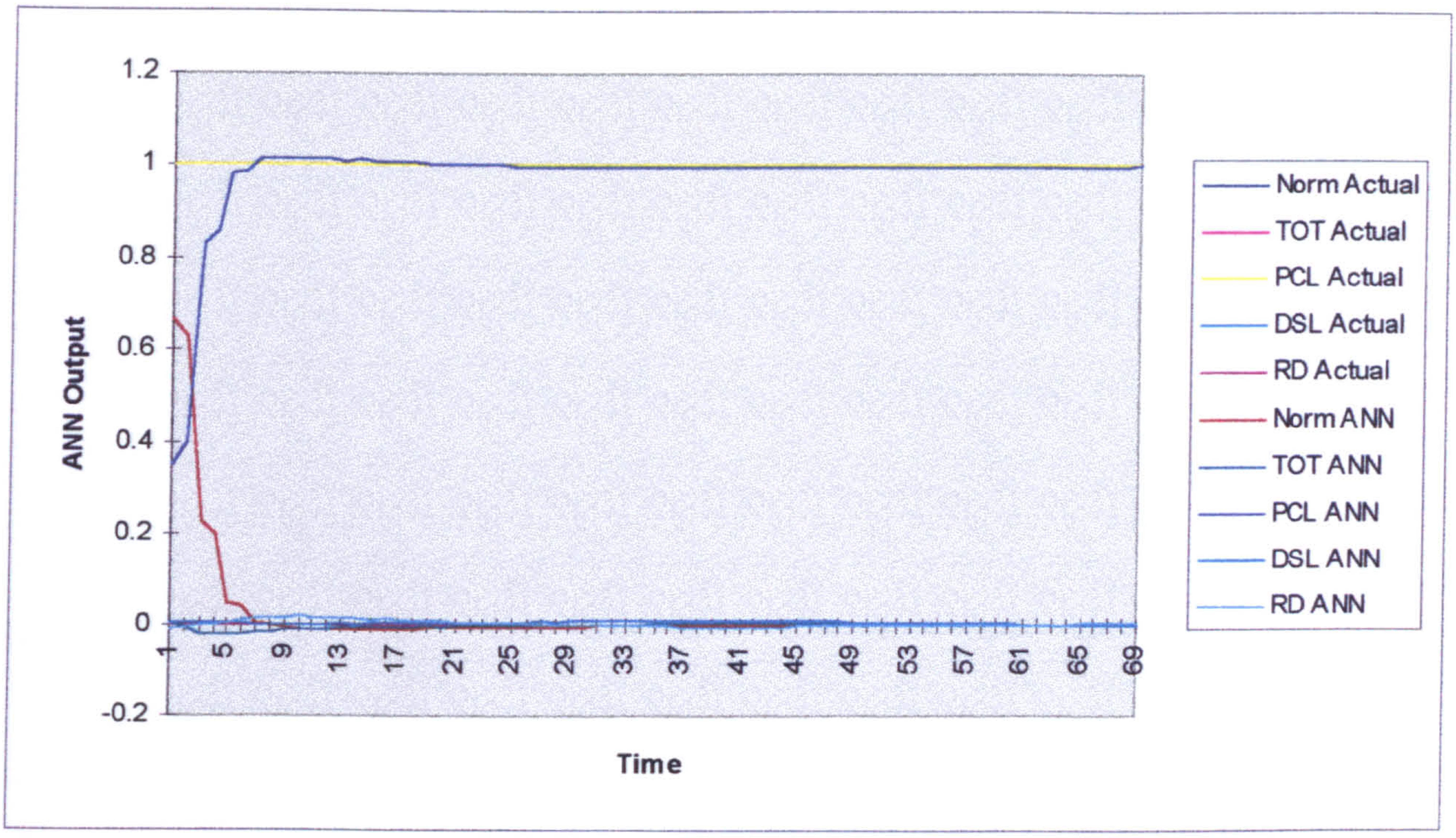


Fig 7.13: ANN Result for Small Primary Coolant Leak

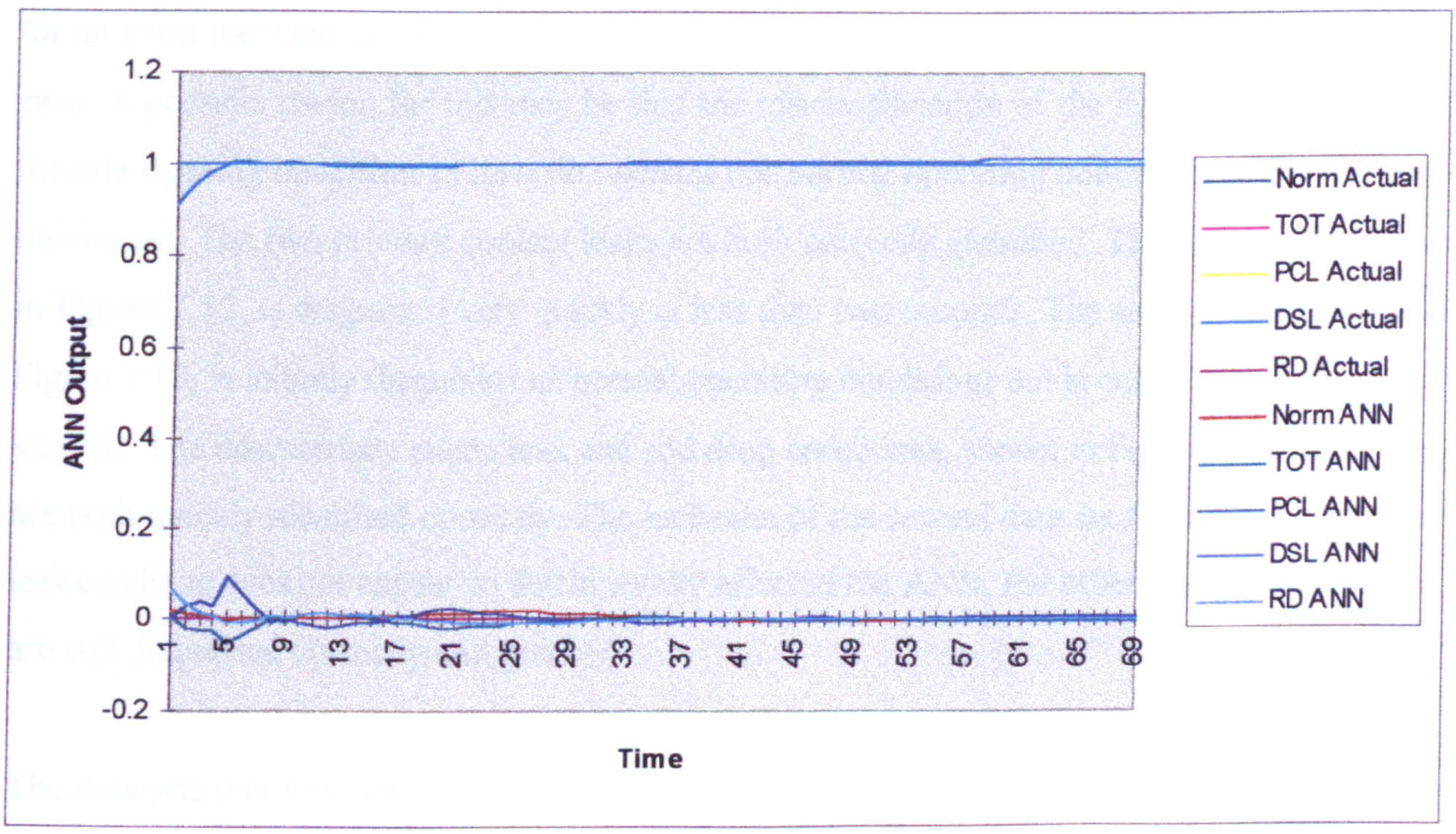


Fig 7.14: ANN Result for Downstream Steam Leak

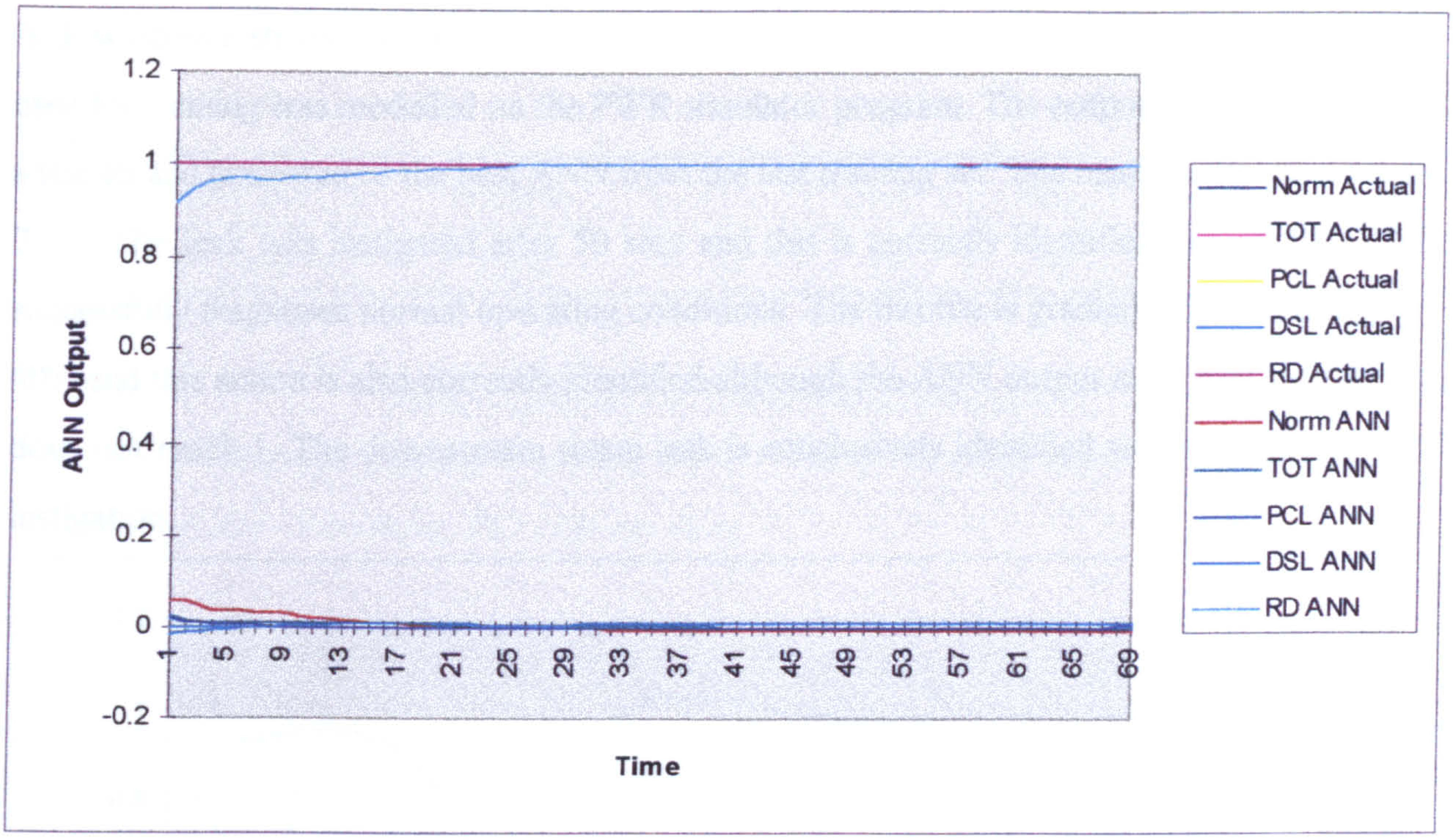


Fig 7.15: ANN Result for Rod drop Transient

This result is a significant improvement on the previous data set. The training set contains data for an extra transient condition yet the RMS error for the best ANN is lower than the previous case. A possible reason for this may be that the misclassification of the first data points of the throttle opening condition as throttle opening not normal operating conditions in the previous training set. The two primary coolant leaks are both correctly identified. The larger leak, shown in Figure 7.12, is diagnosed very quickly in less than two seconds. The smaller leak, shown in Figure 7.13, is initially diagnosed as normal operating conditions but is correctly identified in 5 seconds. The downstream steam leak and rod drop conditions, shown in Figures 7.14 and 7.15, are both quickly identified correctly. The inclusion of the second data set for a primary coolant leak condition does not appear to detrimentally affected the ANN, the other transient conditions are still diagnosed correctly and quickly.

The data sets that have been used to test the ANNs to date have only been the training and test data used to train the ANNs. This situation is very artificial and does not give a true guide to the ANNs ability to diagnose data from a condition scenario unknown to the ANN. The ANN may have been trained to correctly identify the transients conditions used for training but cannot diagnose new data. Whilst the ANN so far developed have not considered a full set of transient conditions and therefore permit extensive testing on new transient scenarios, a test with a scenario not used for training would give a guide to the generality of the ANN.

A downstream steam leak with a smaller leak size and throttle settings other than the transient used for training was modelled on the PWR simulator program. The output was converted into a test set and presented to the best ANN from the last training set. The result is shown in Figure 7.17. The leak was instigated after 50 secs and this is correctly identified. Initially the ANN successfully diagnoses normal operating conditions. The throttle is gradually opened to approx 40% and this action is also correctly identified although the ANN output node for this transient does not reach 1. The downstream steam leak is conclusively identified within two seconds of instigation.

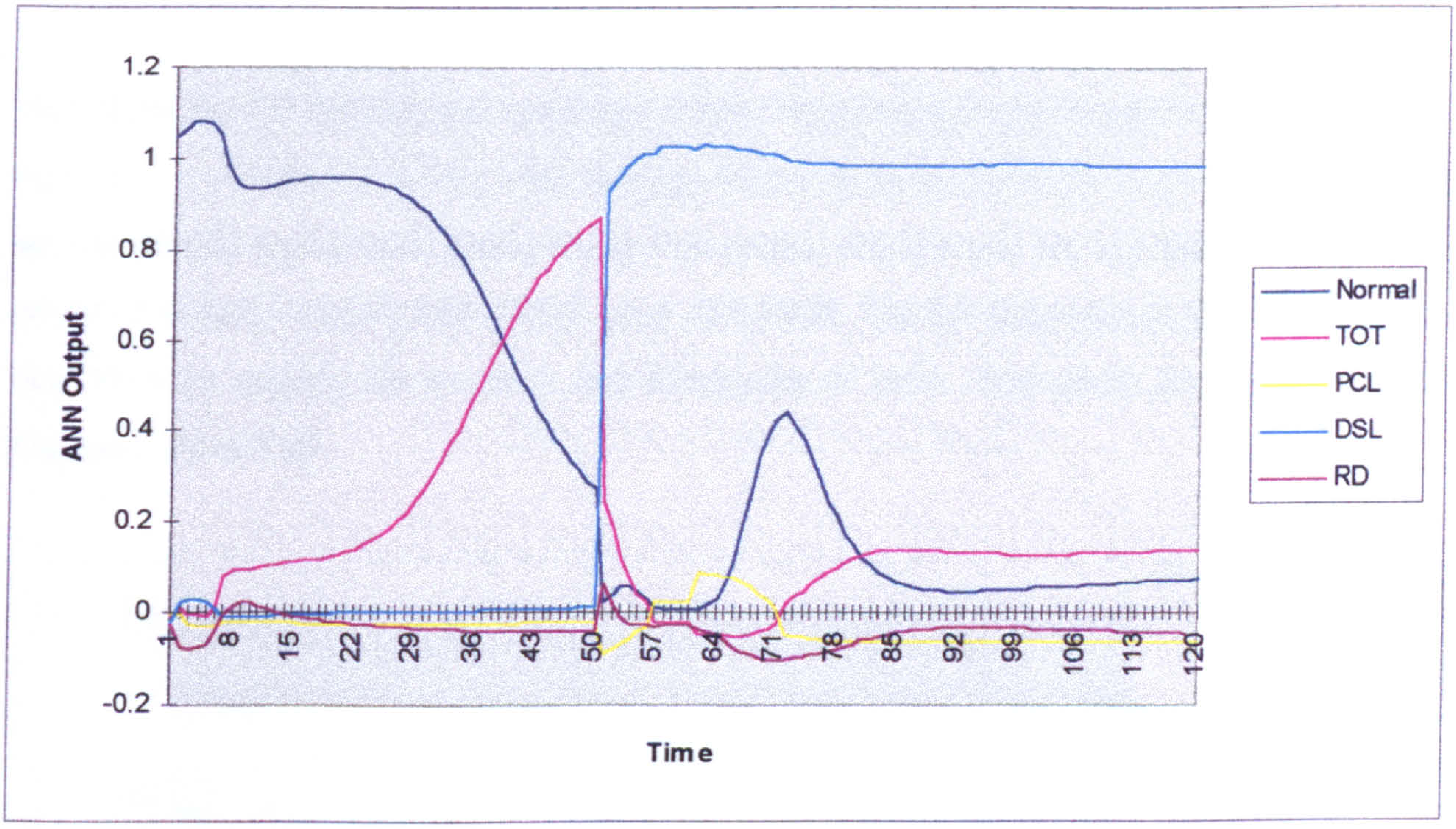


Fig 7.16: Untrained Downstream Steam Leak

While this test shows that the diagnostic ANN can successfully diagnose an unlearnt transient scenario it is by no means conclusive. A more complete range of conditions would require testing on a suitable ANN to gain confidence in the generalisation and accuracy of the diagnosis. It would be more profitable to perform this testing once an ANN with a complete range of conditions is included in the output set. This test does however, show the ability of the ANN to successfully diagnose multiple conditions. Further tests to investigate this feature will also be made with an ANN developed with the full output.

The next set of tests were made with new ANN training and test sets. A group drop transient was modelled on the PWR simulator and the results were included in the data sets. All of the selected transient conditions were now represented in the data set. A single example of the primary coolant leak was included for comparison with the previous investigation.

Training Set	Normal	TOT	PCL	DSL	RD	GD
test13.nna	120	69	70	70	70	70

Table 7.5: Details of Diagnostic Training Sets

Key: TOT - Throttle Opening Transient, PCL - Primary Coolant Leak, DSL - Downstream Steam Leak, RD - Rod Drop, GD - Group Drop.

As before the full data set was randomly divided into training and test sets in the approximate ratio of 2:1. A series of ANNs, with varying internal structures, were trained with this training set for 80,000 cycles with testing every 100 cycles with the test set, the best network being saved. The best network had a RMS error of 0,0868. The full data set was then presented to this ANN to explore the accuracy and distribution of error. The results are given below in Figures 7.17 to 7.22.

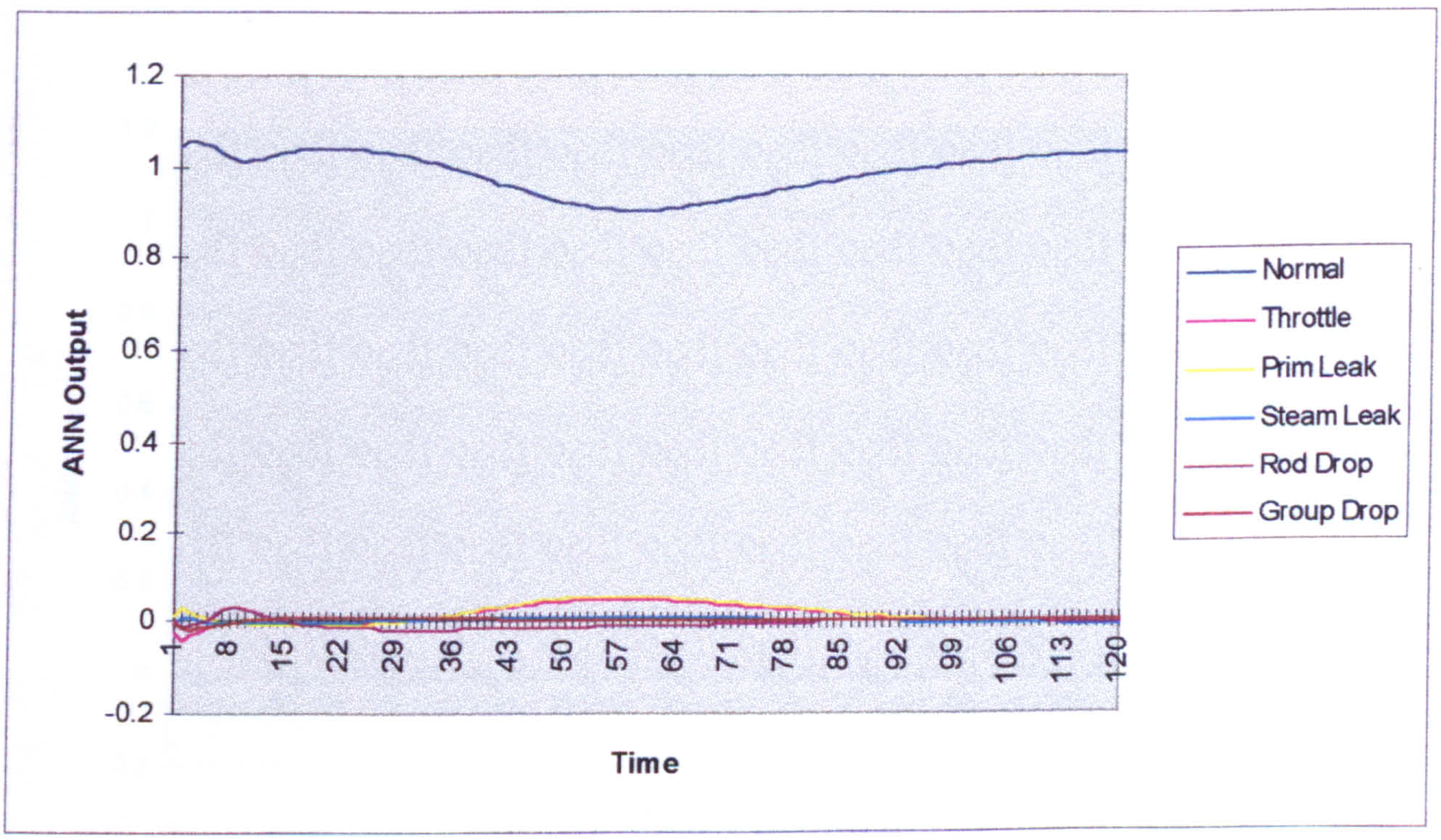


Fig 7.17: Normal Operating Training Set

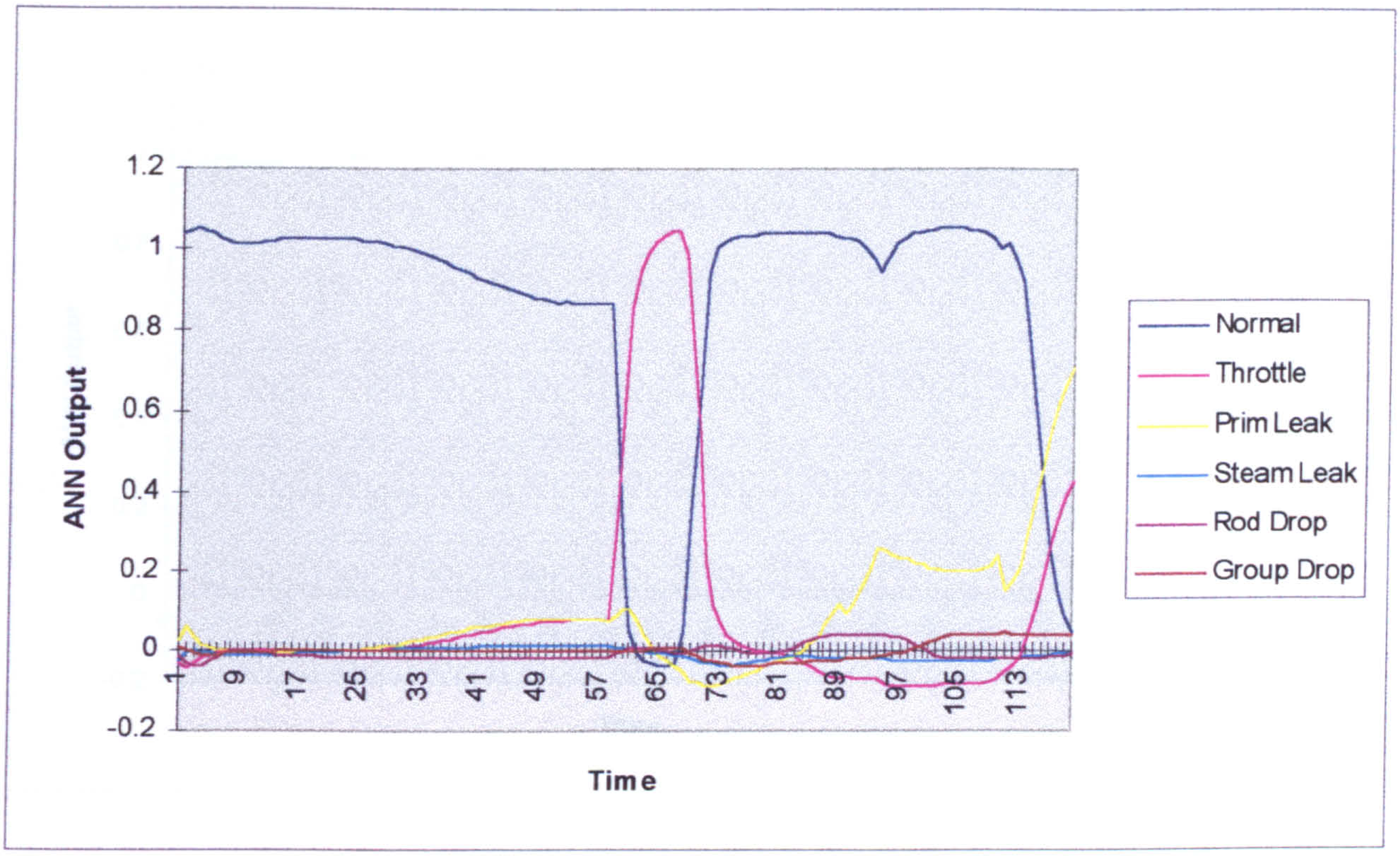


Fig 7.18: Throttle Transient Training Set

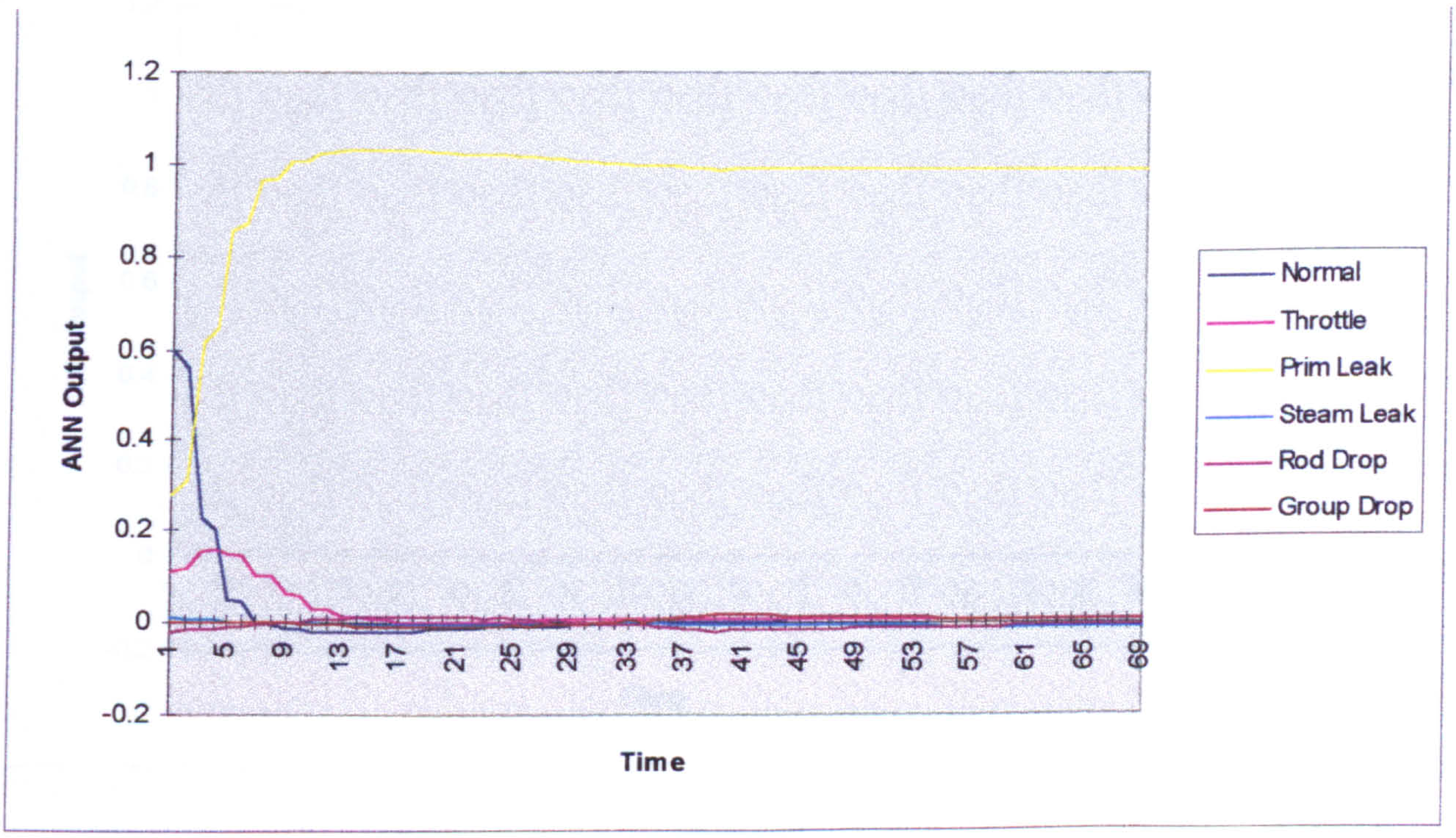


Fig 7.19: Primary Coolant Leak Training Set

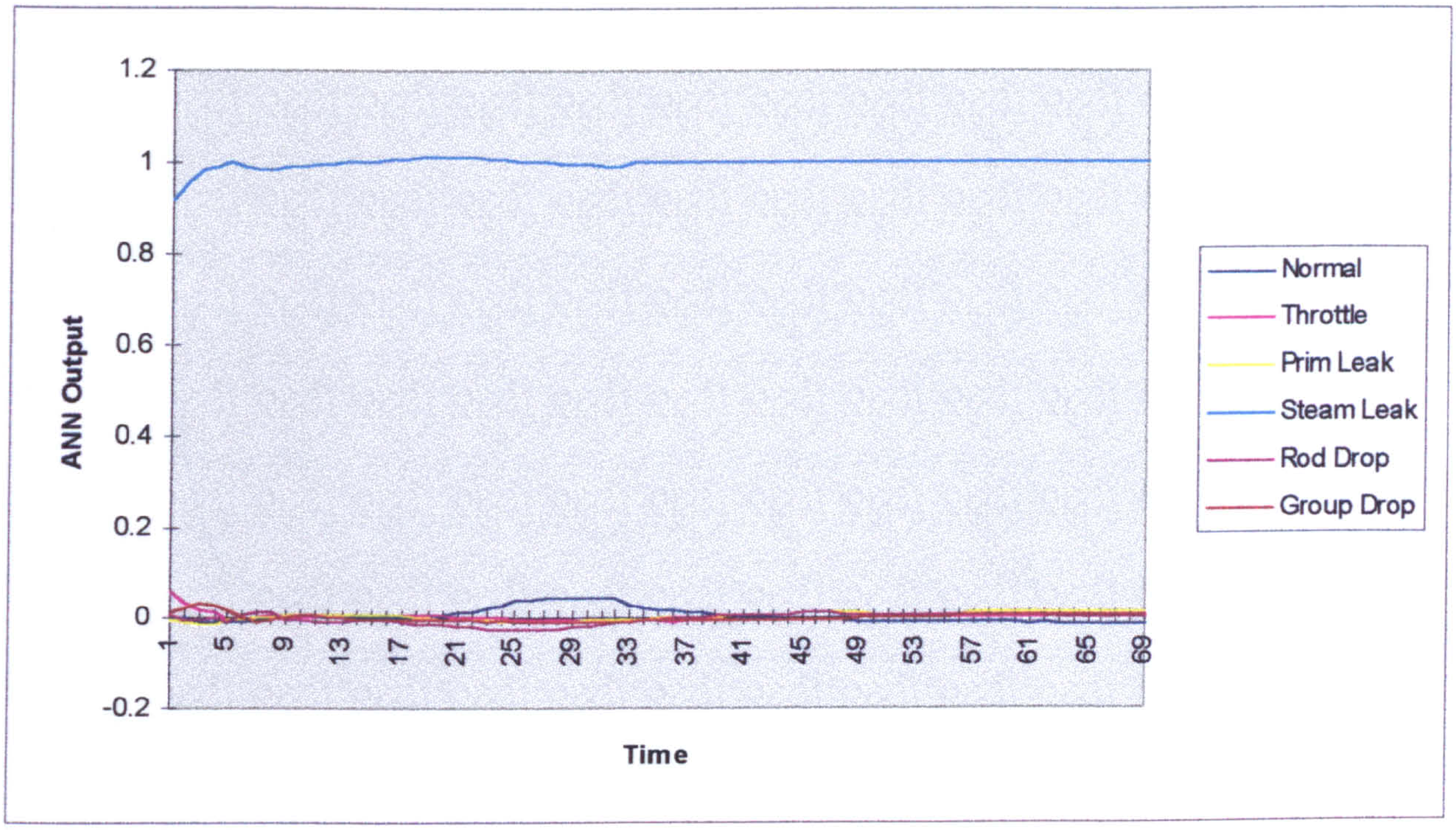


Fig 7.20: Downstream Steam Leak Training Set

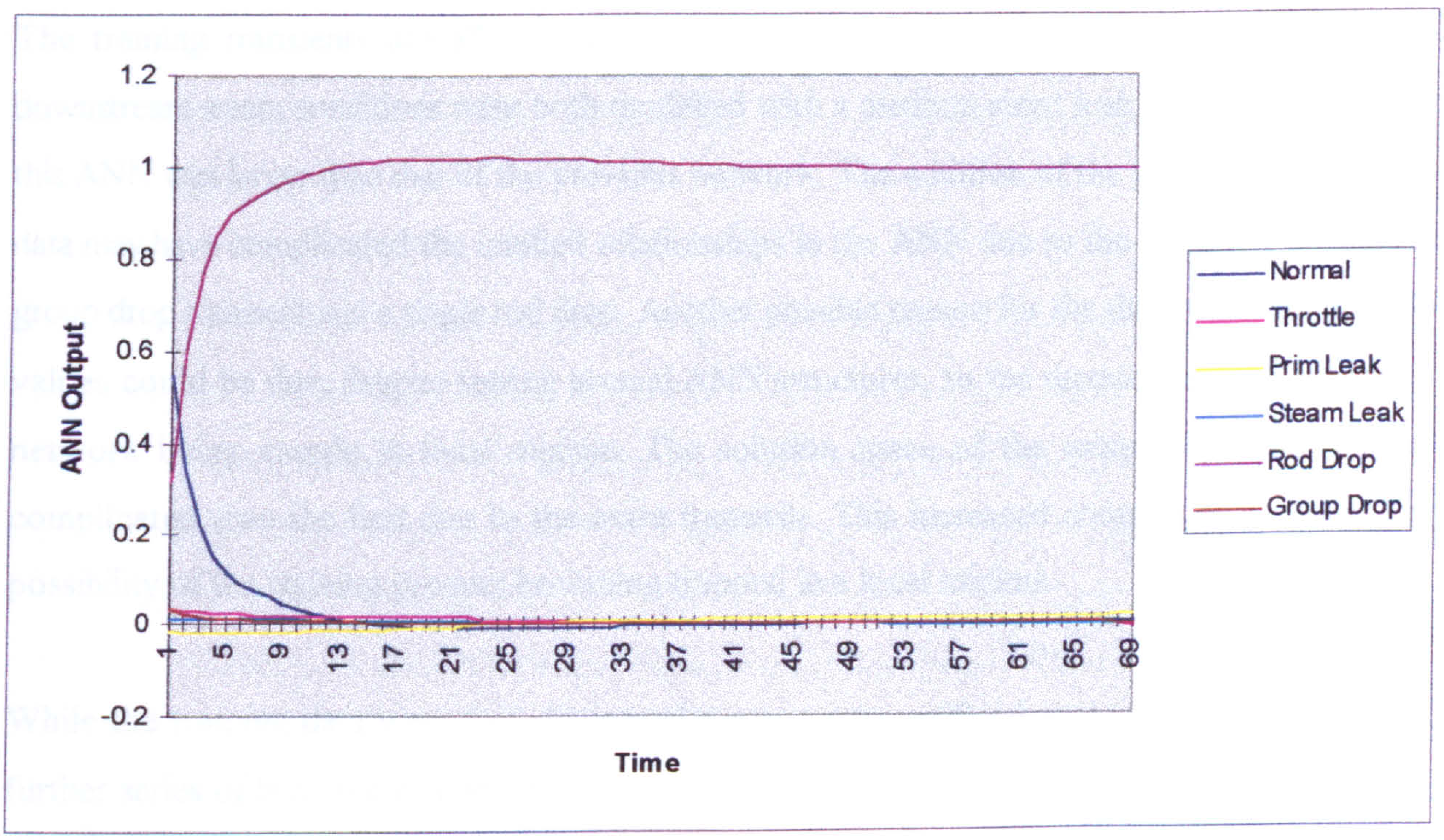


Fig 7.21: Rod Drop Training Set

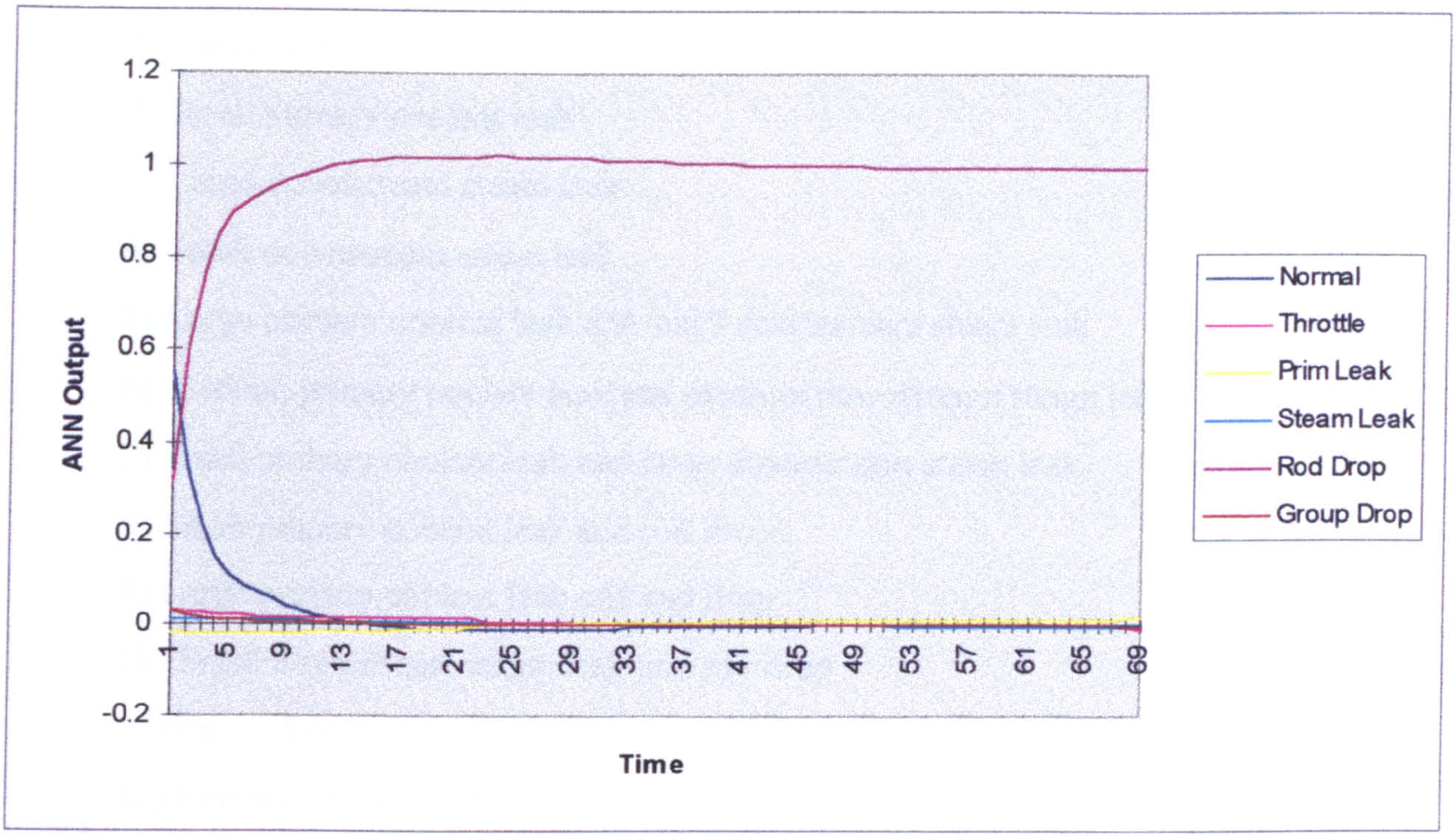


Fig 7.22: Group Drop Training Set

The training transients are all diagnosed quickly and accurately. The primary coolant and downstream steam conditions were both modelled with a medium sized leak. The RMS error for this ANN was larger than that of the previous network. The addition of the group drop transient data may have complicated the implicit relationships in the ANN due to the similarity between a group drop transient and a single rod drop. Another possible reason for the difference in the RMS values could be due, despite testing several ANN structures, to the second, more complicated network being caught in local minima. The solution space of the second data set is more complicated than the first due to the extra transient. This increased complexity increases the possibility of the training process becoming trapped in a local minima.

While the training data were well diagnosed it was a very artificial evaluation of the ANN. A further series of tests were conducted to evaluate the ANN response to new transient data, not used for training. New scenarios were modelled on the PWR simulator including different leak sizes, throttle settings and multiple transient conditions. The simulator output was modified for use as an ANN input set. No output terms were included in these data sets. The following tests were conducted.

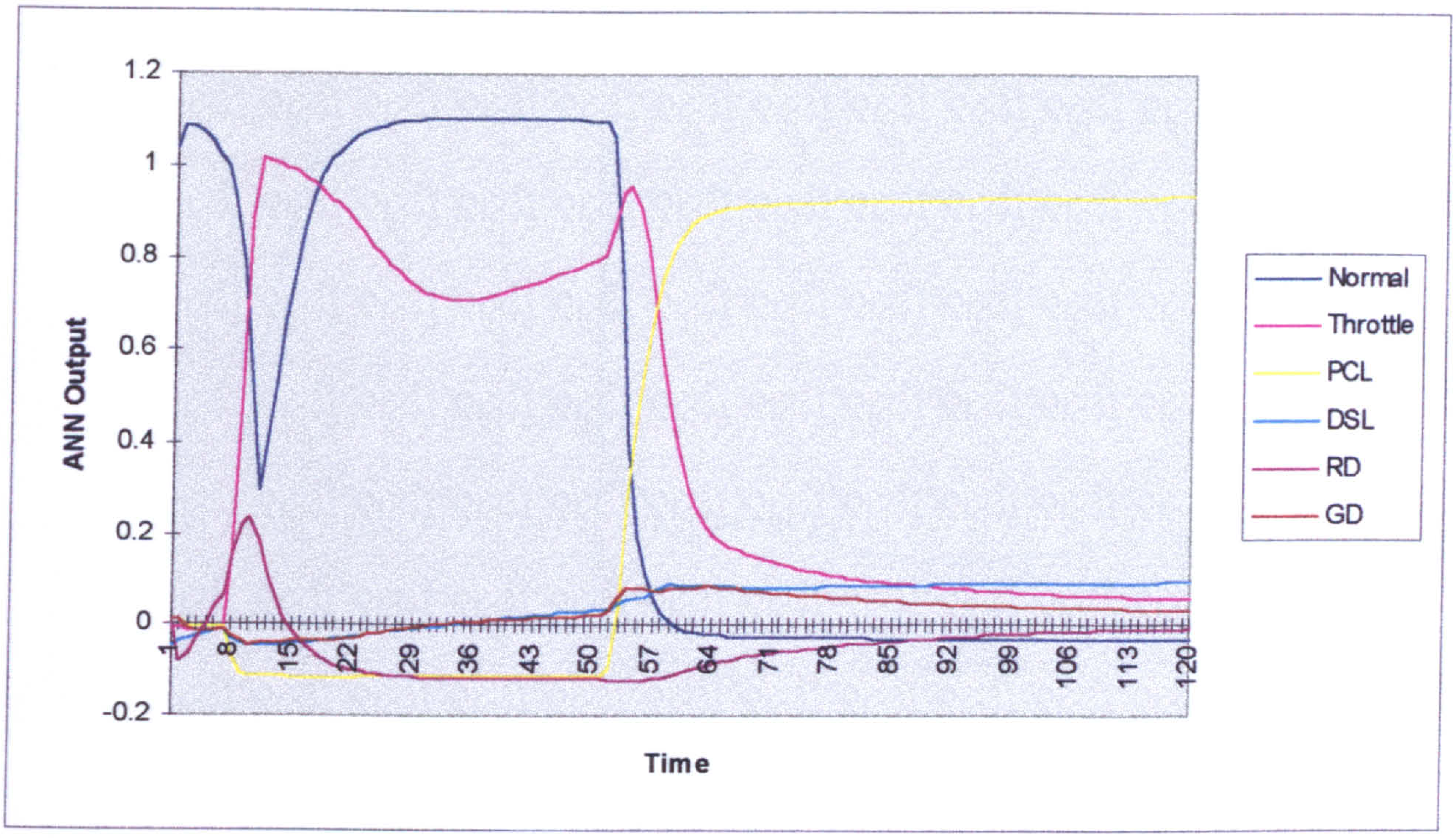
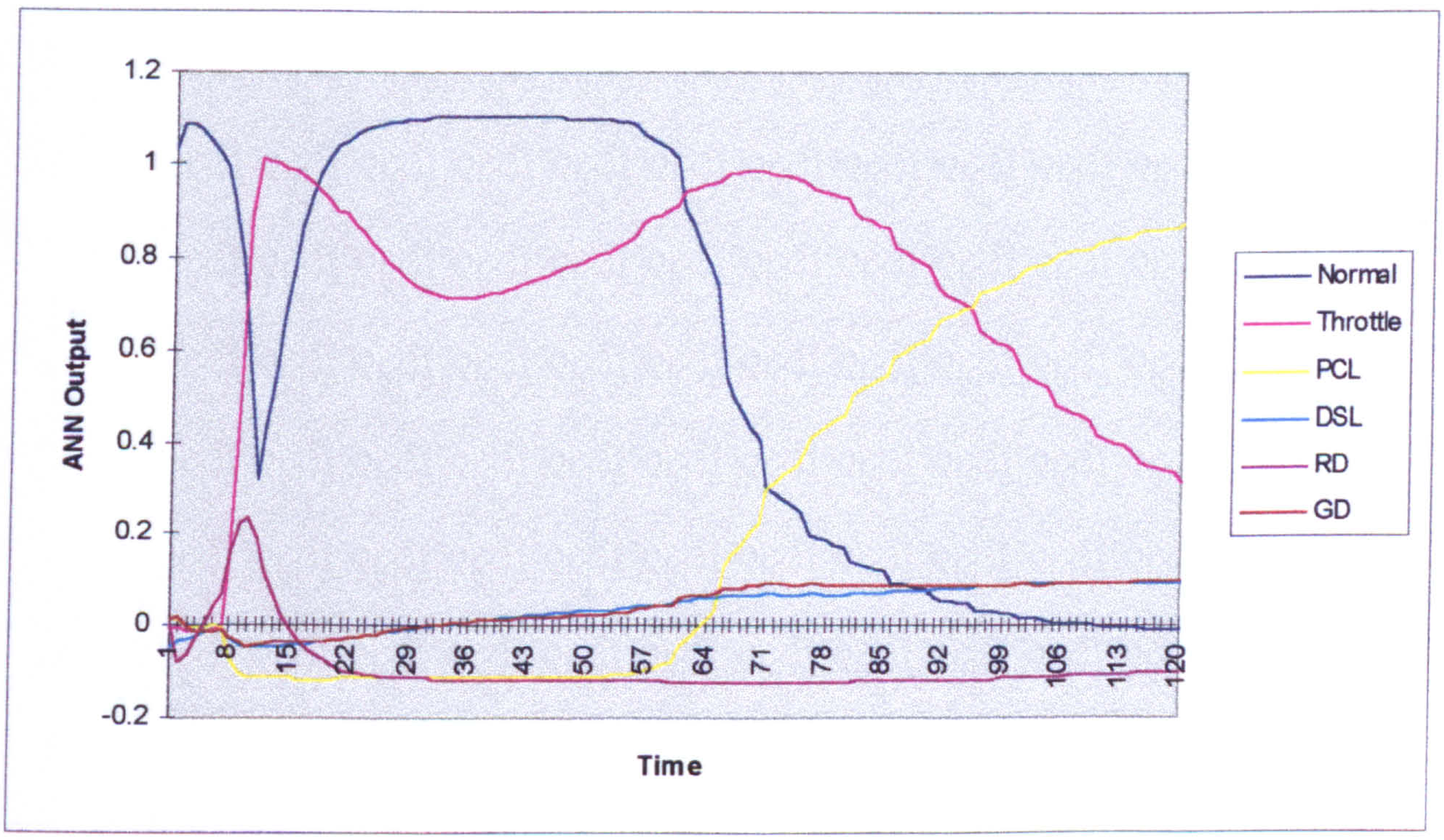


Fig 7.23: ANN Results for Large Primary Coolant Leak



7.24: ANN Result for Small Primary Coolant Leak

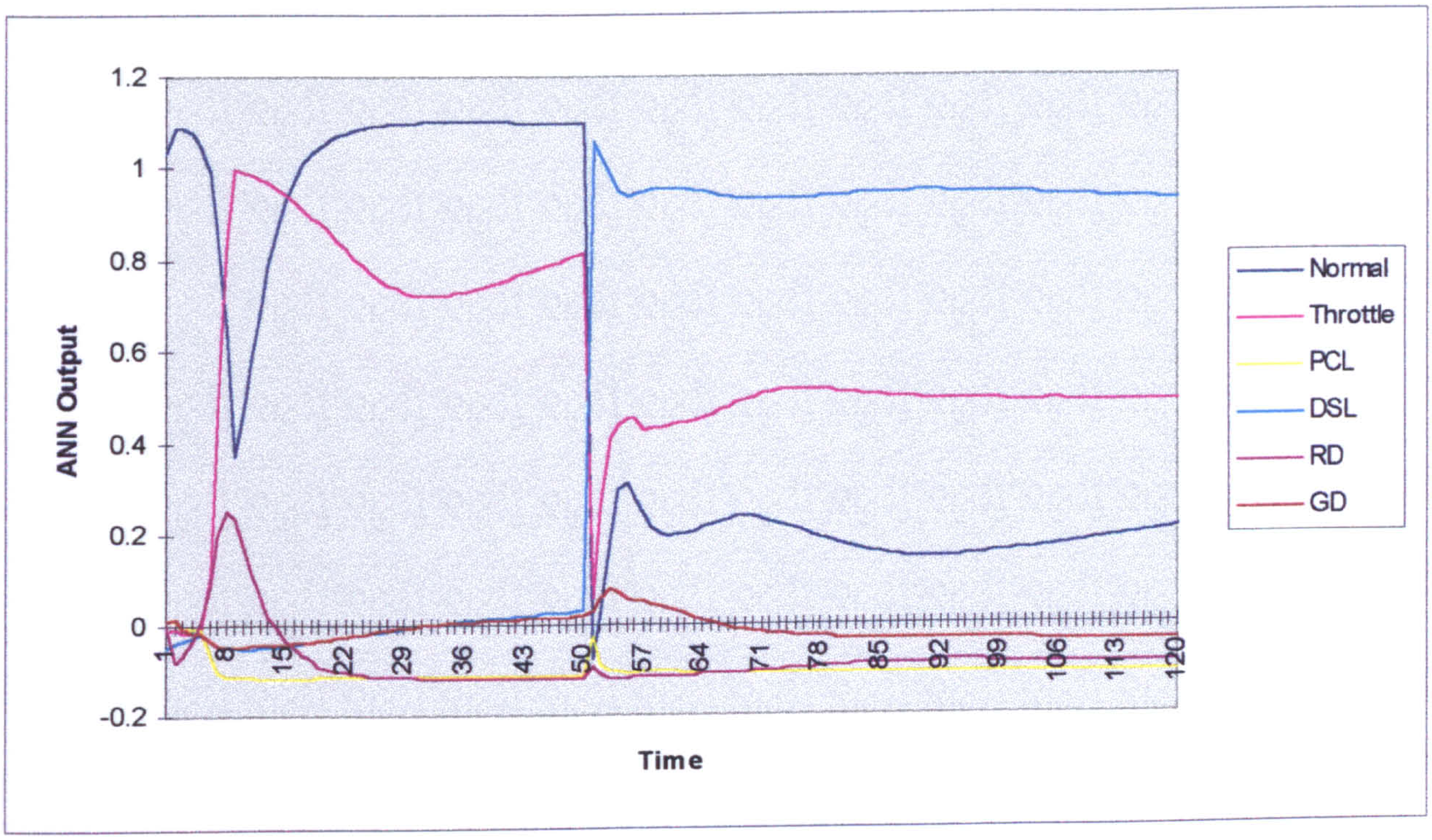


Fig 7.25: ANN Result for Large Downstream Steam Leak

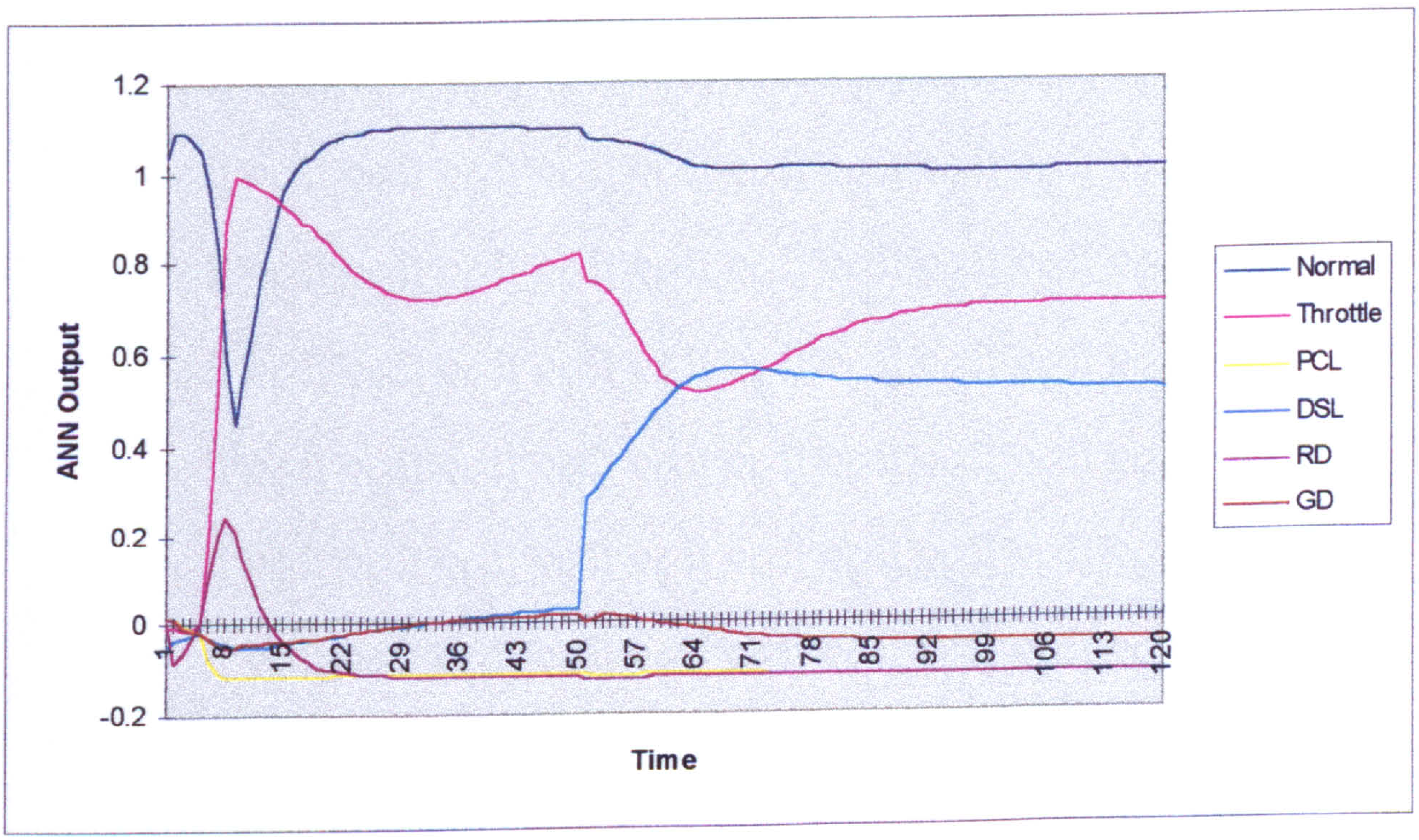
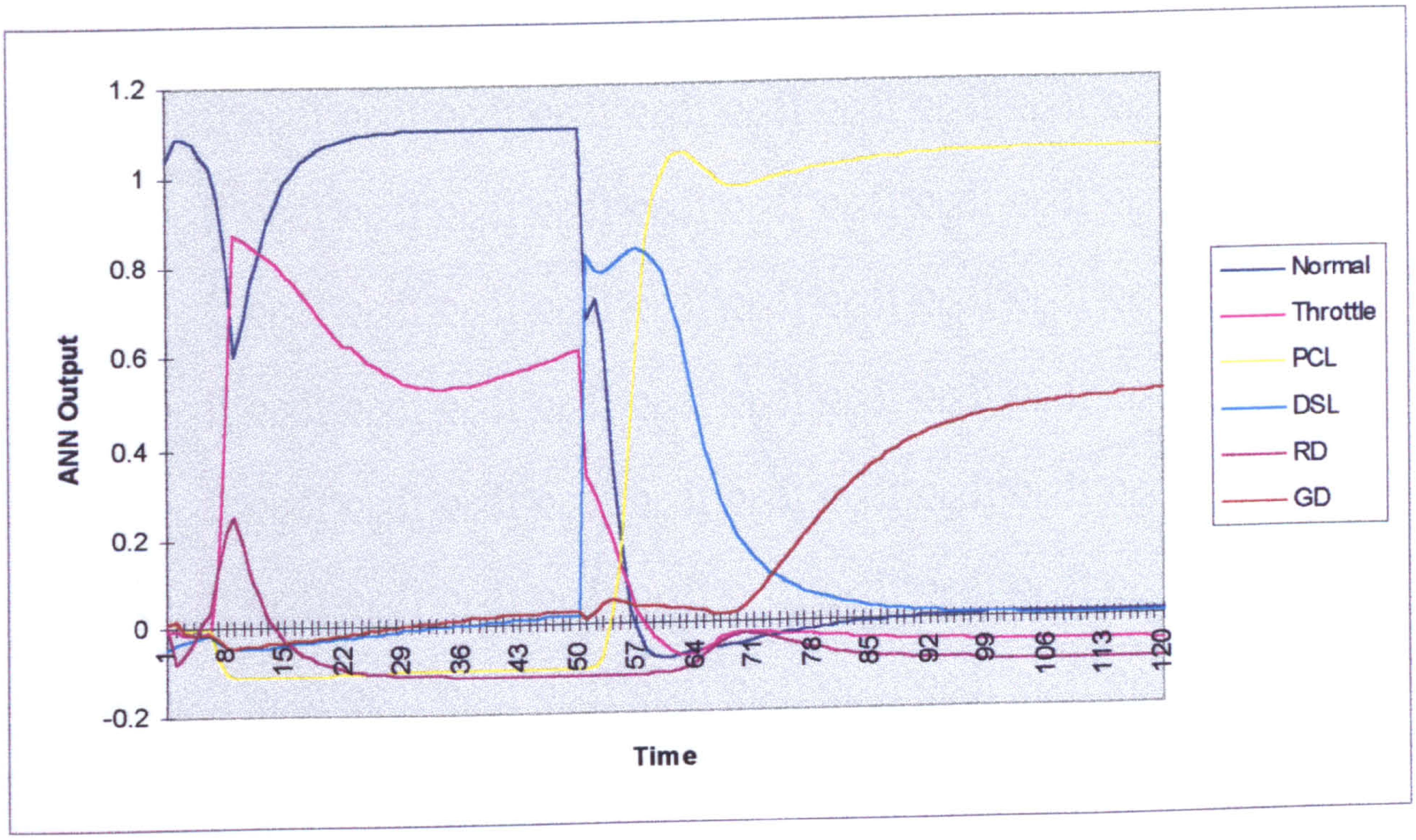


Fig 7.26: ANN Result for Small Downstream Steam Leak



7.27: ANN Result for Large Coolant Leak and Small Steam Leak

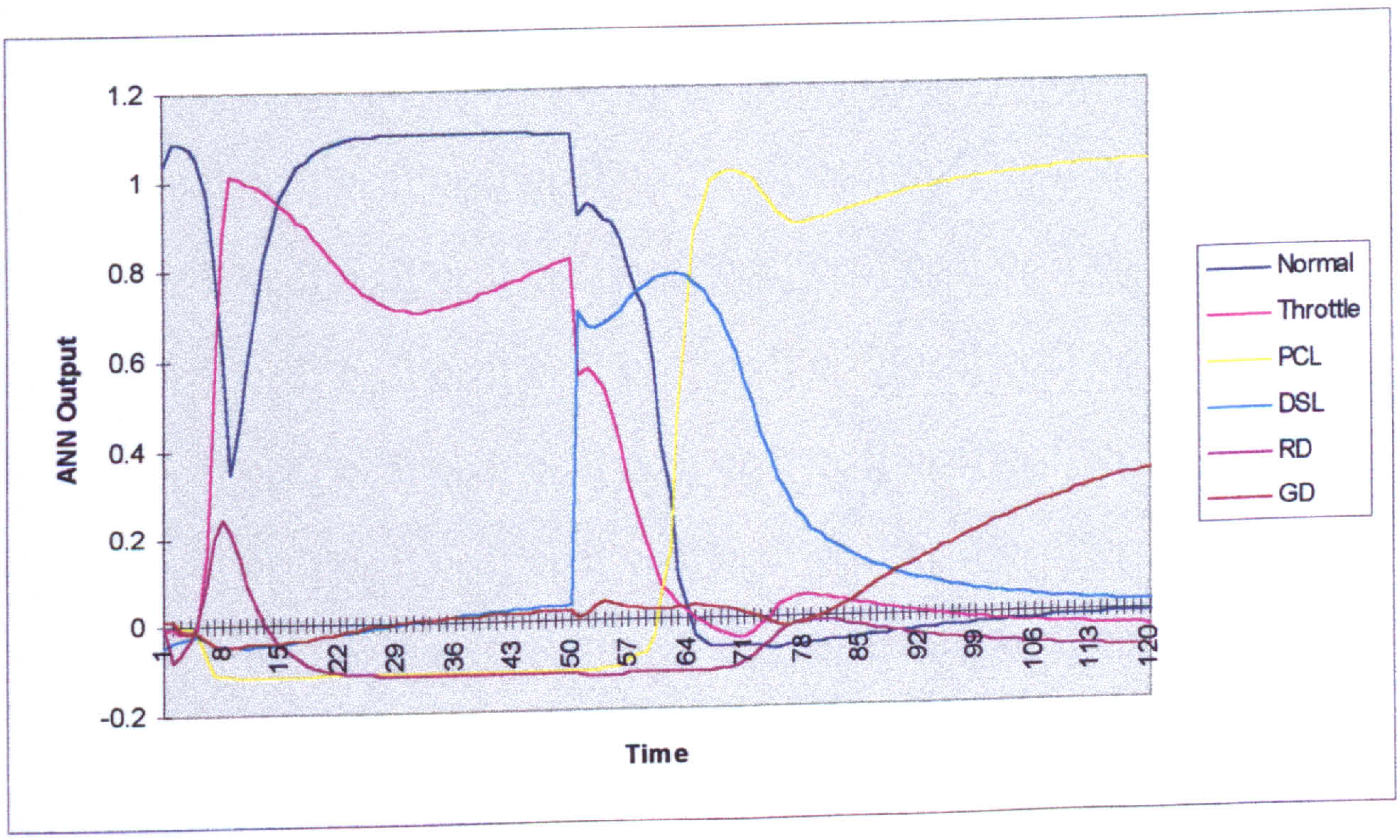
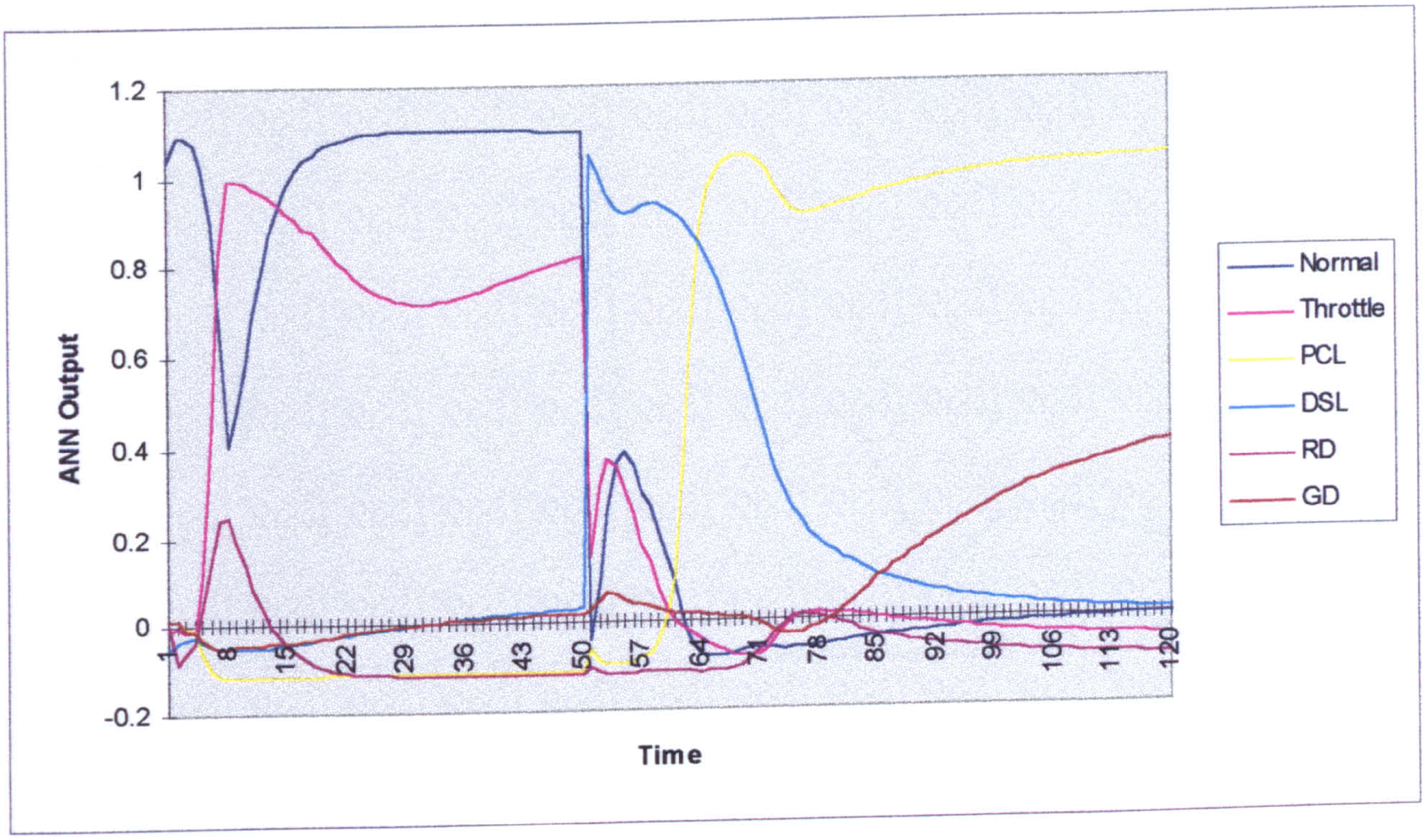
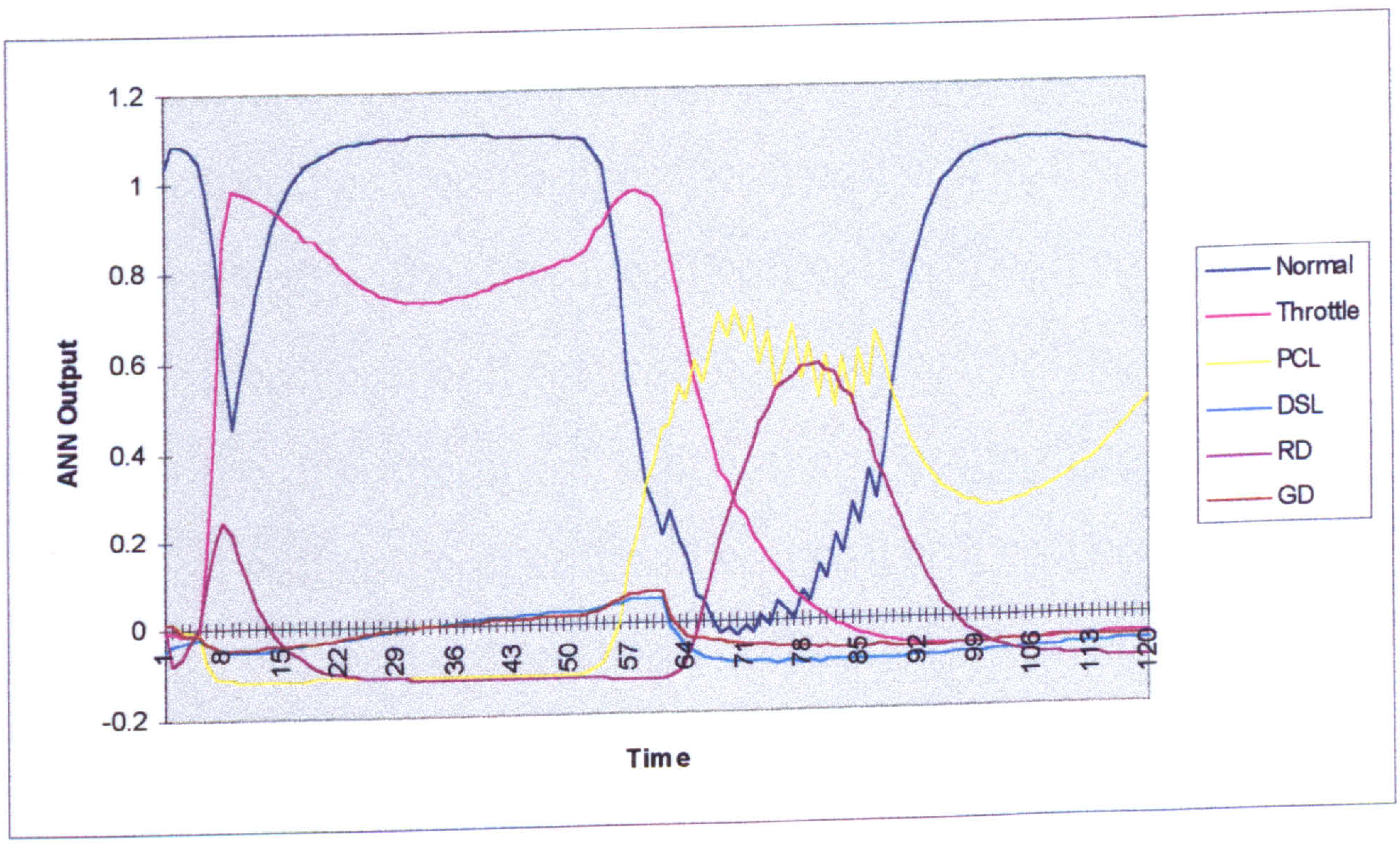


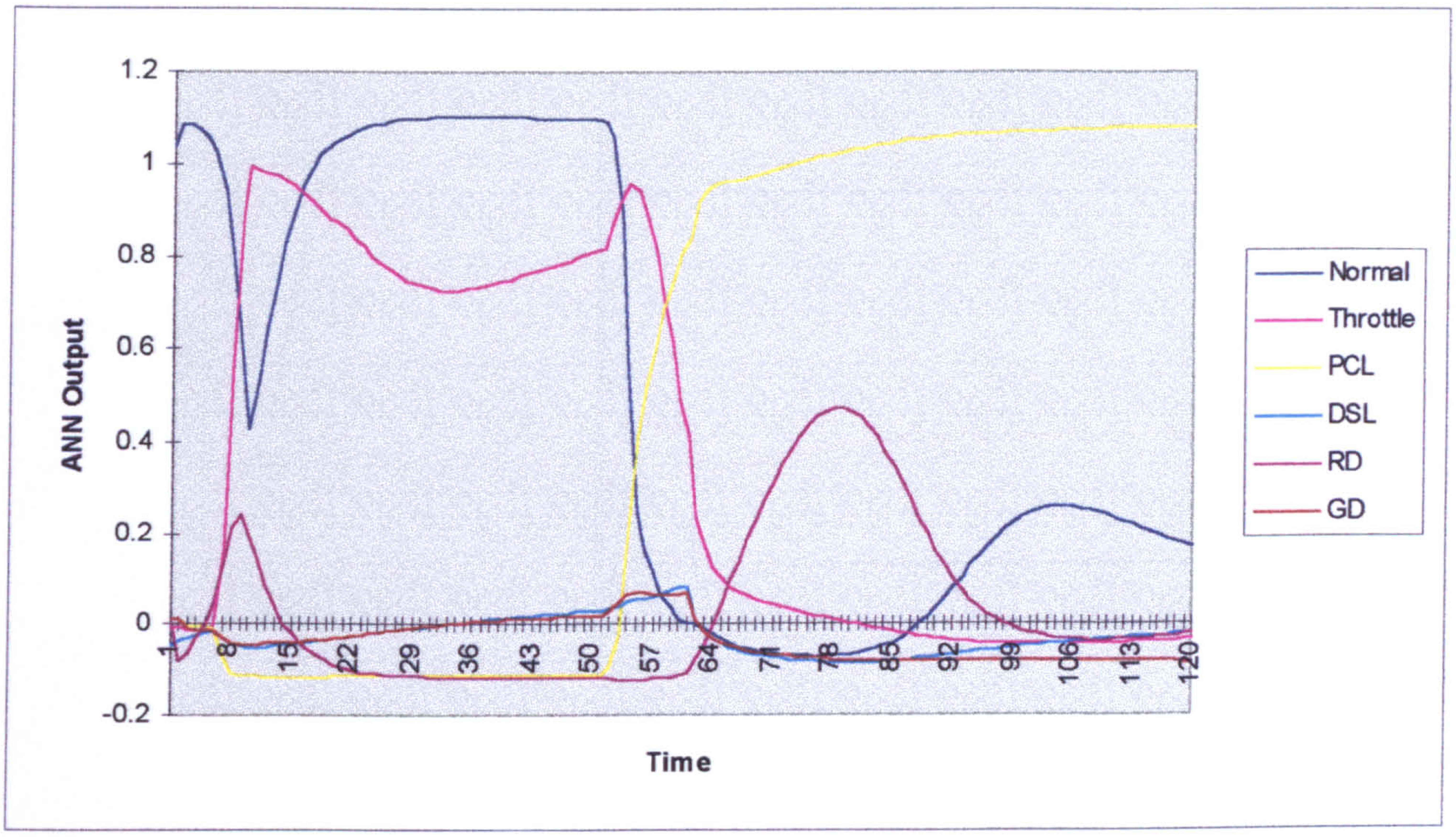
Fig 7.28: ANN Result for Medium Coolant Leak and Medium Steam Leak



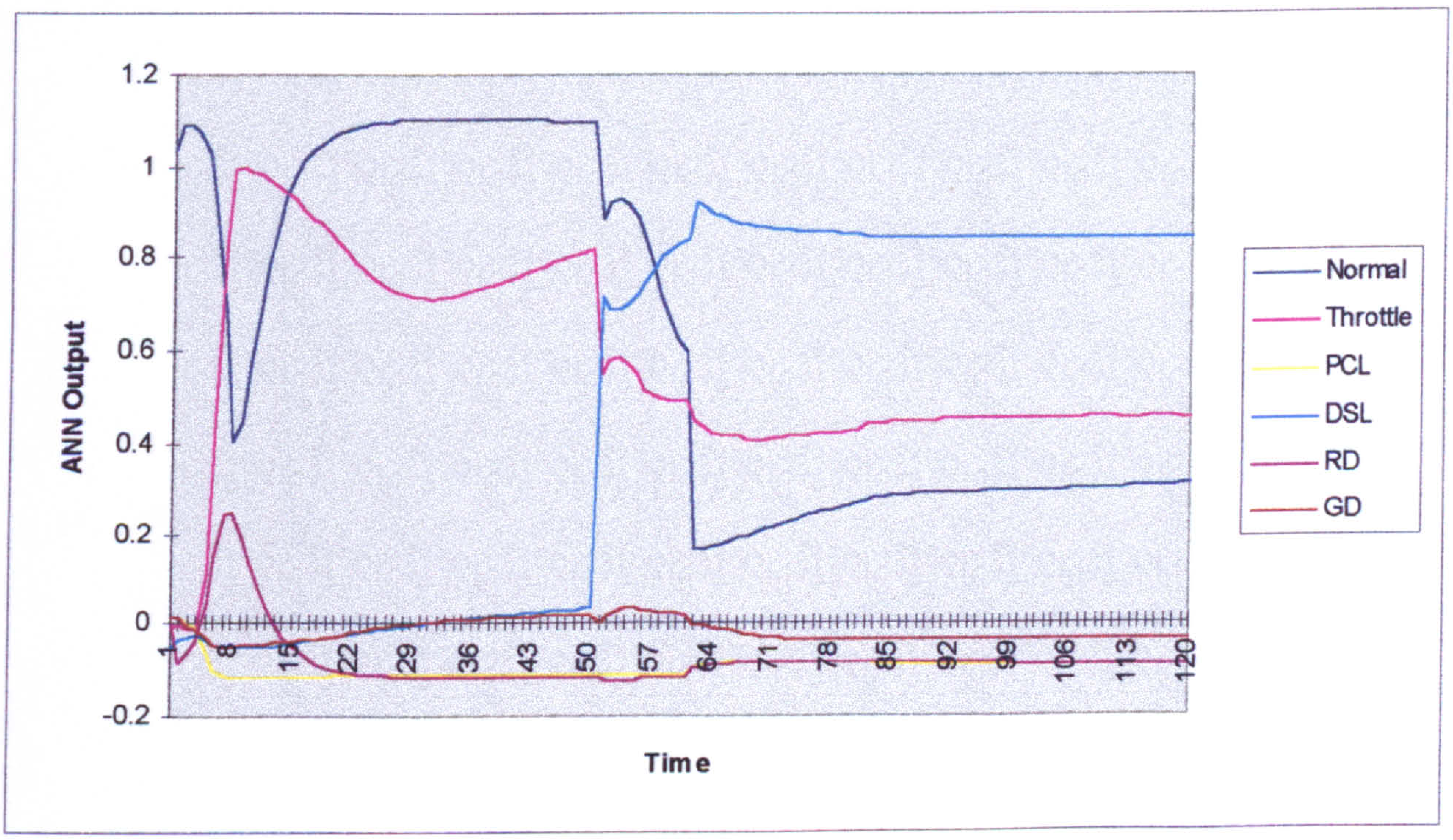
7.29: ANN Result for Small Coolant Leak and Large Steam Leak



7.30: ANN Result for Small Coolant Leak and Rod Drop



7.31: ANN Result for Large Coolant Leak and Rod Drop



7.32: ANN Result for Small Steam Leak and Rod Drop

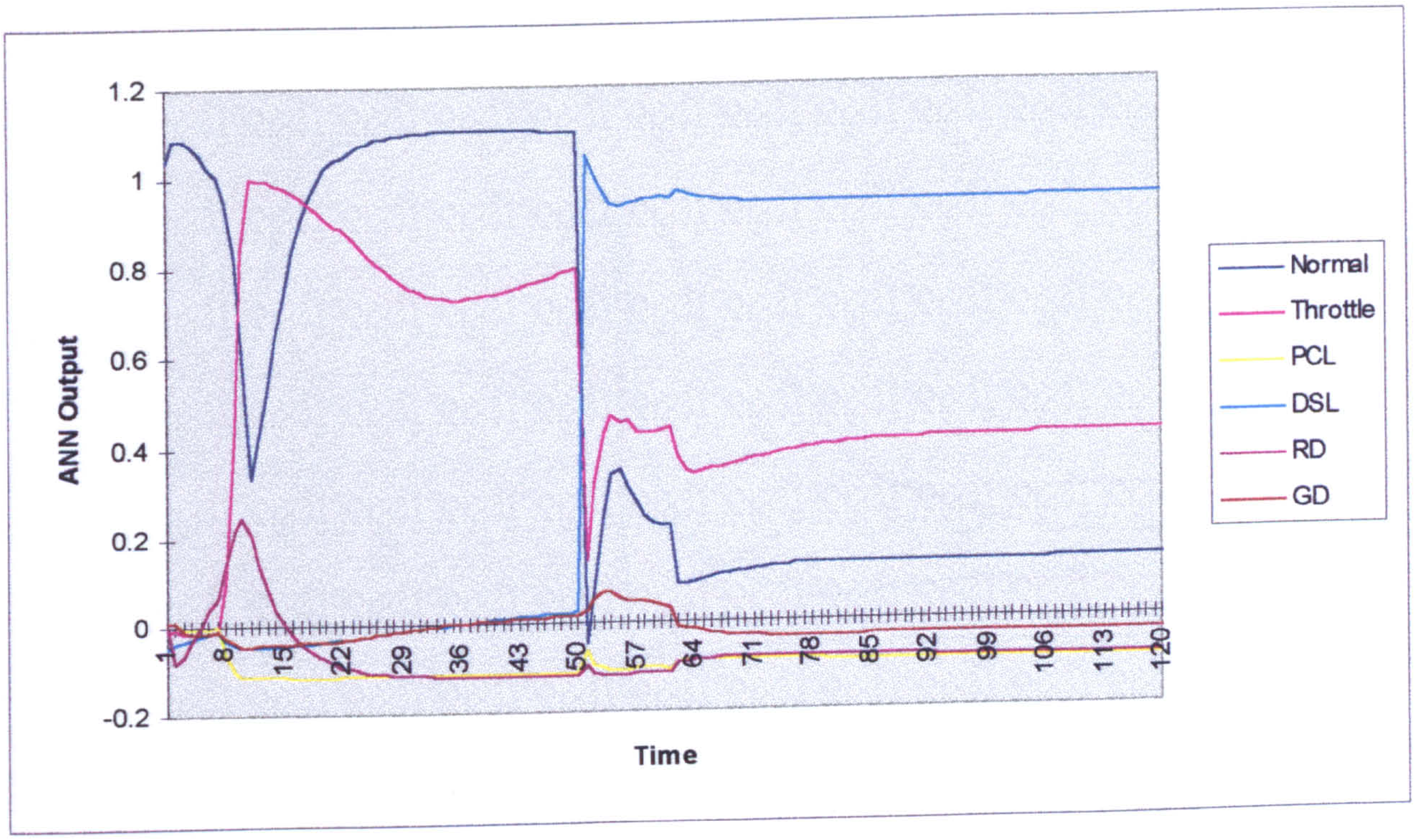


Fig 7.33: ANN Result for Large Steam Leak and Rod Drop

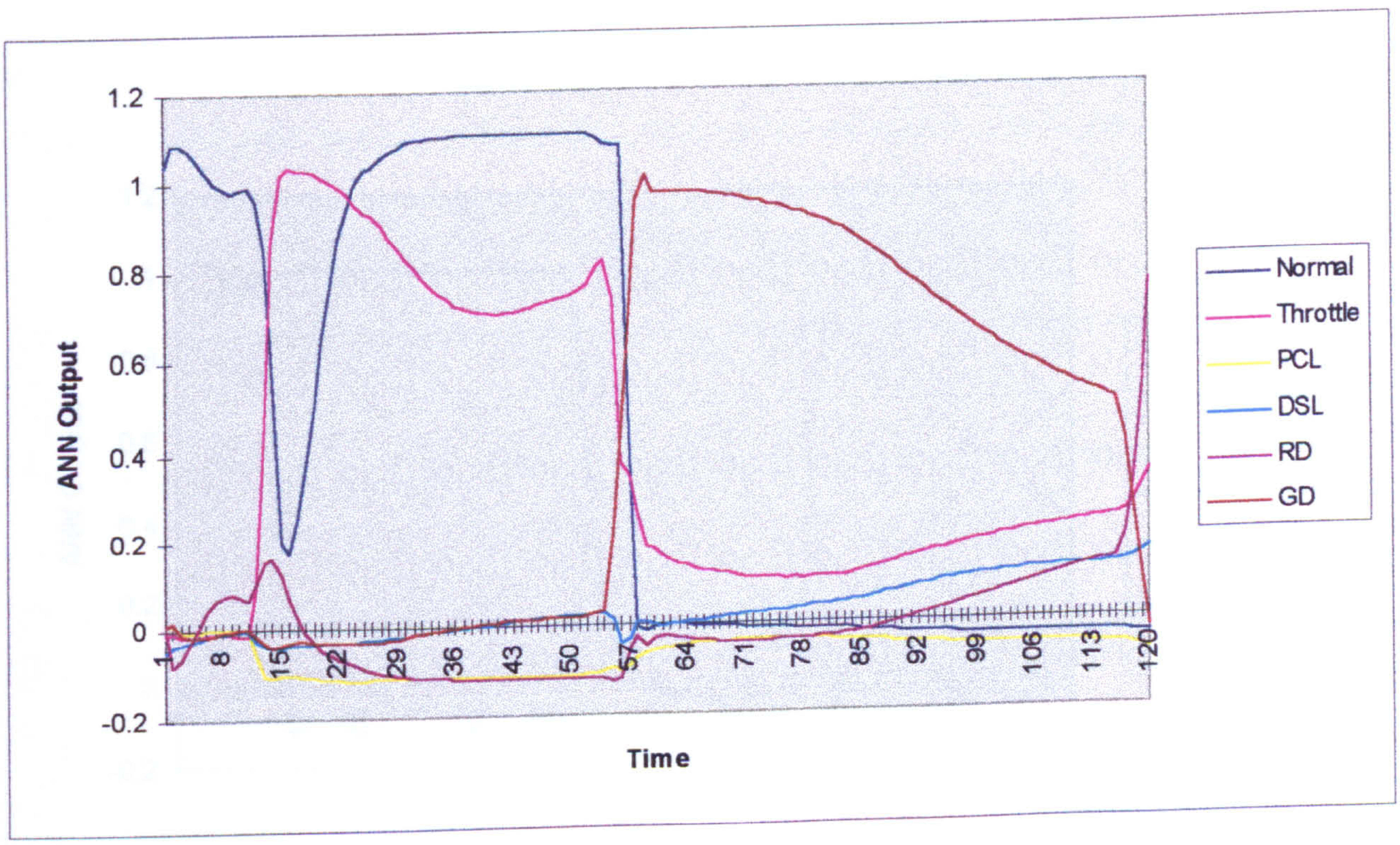


Fig 7.34: ANN Result for Small Coolant Leak and Group Drop

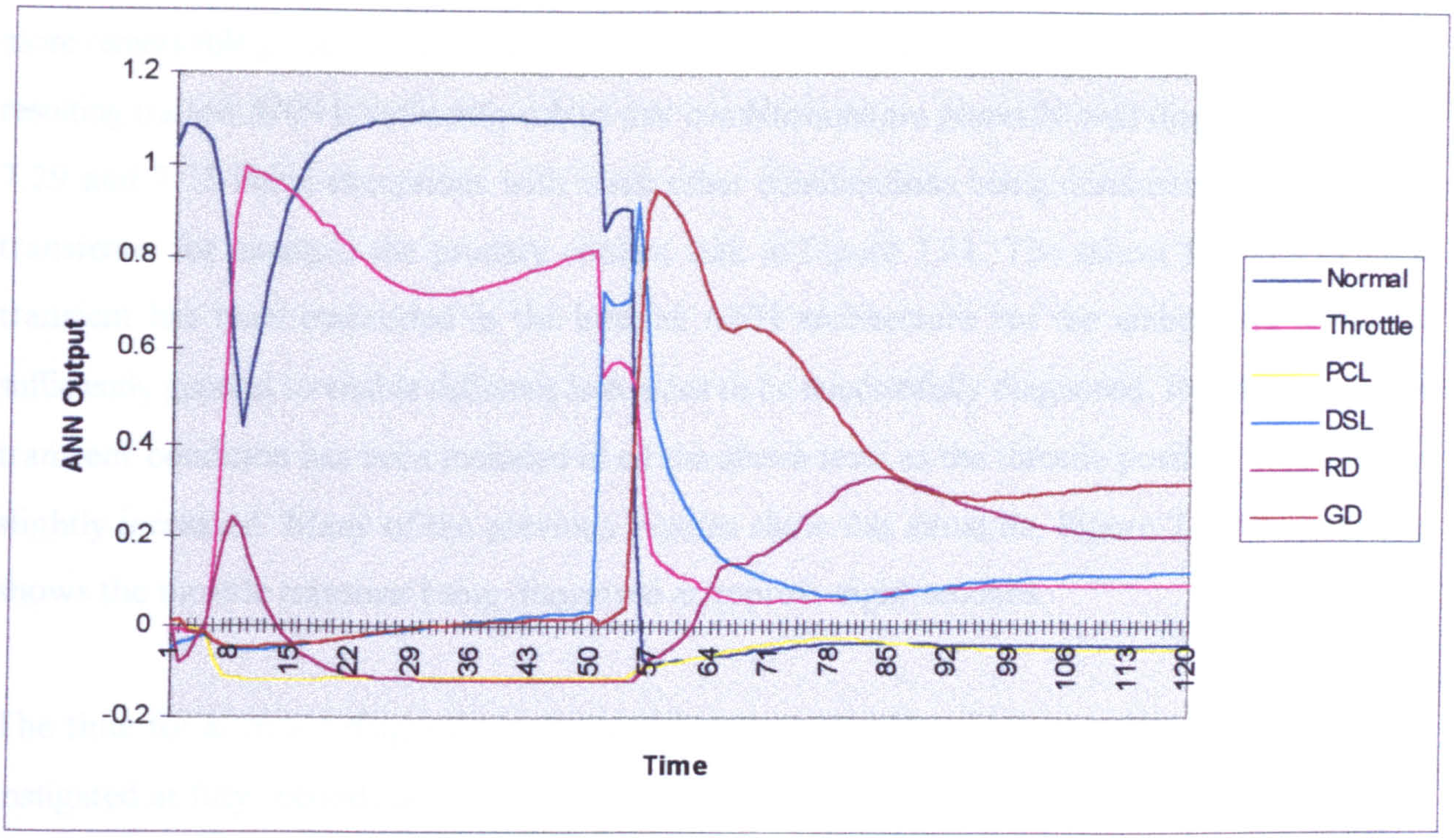


Fig 7.35: ANN Result for Small Steam Leak and Group Drop

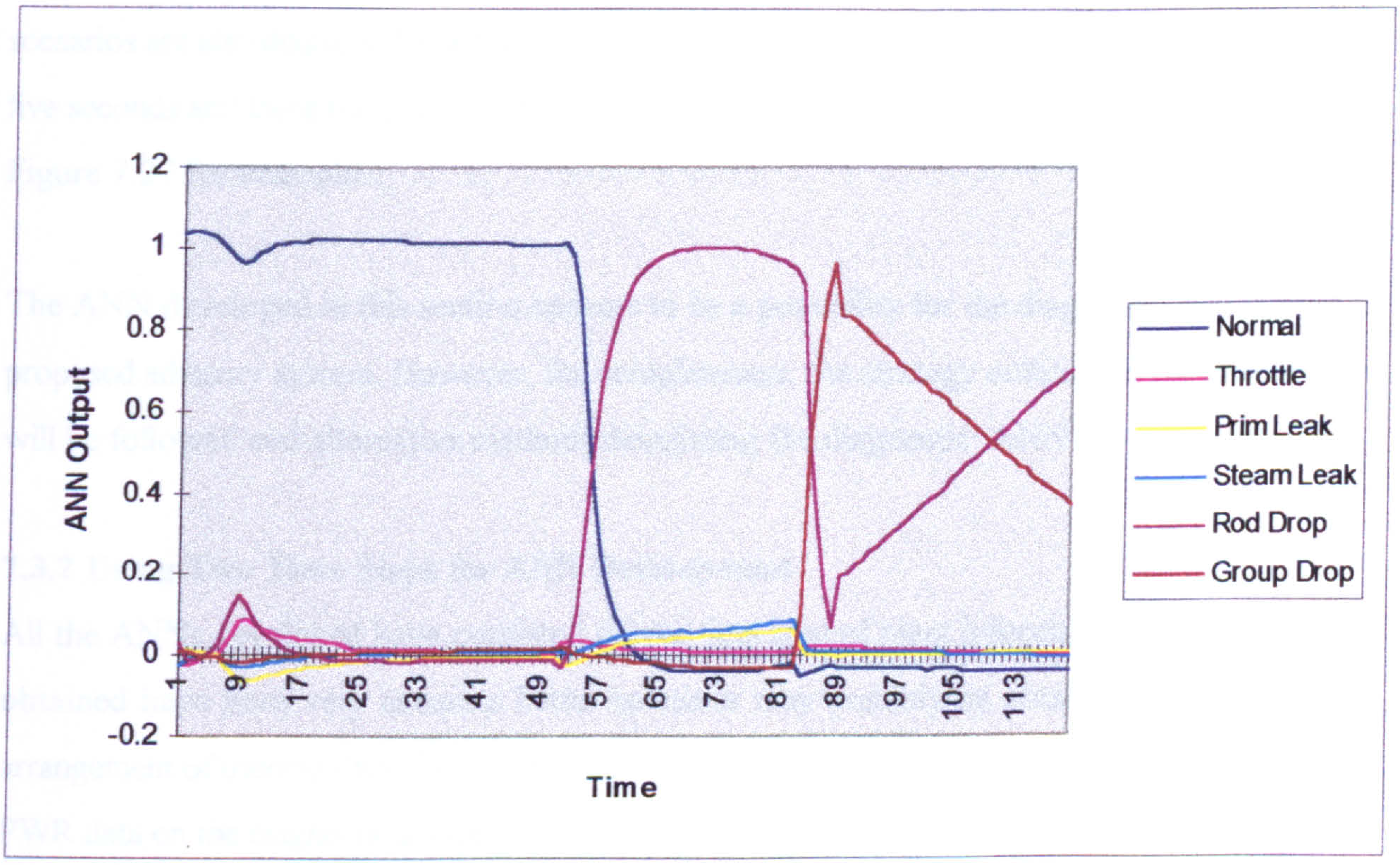


Fig 7.36: ANN Result for Group Drop and Rod Drop

Generally these results are very encouraging. All the single transient conditions have been correctly diagnosed for a range of leak sizes. The results for combinations of transients are even

A series of ANNs were trained with this data. In each case the training data was presented for 80,000 cycles with testing every 100 cycles with the test set, the best network being saved. The full results obtained are given in Appendix M. The best ANN developed had a RMS error of 0.1015. This value is higher than the best one time step ANN previously developed. A reason for this may be that during training the larger, more complex, network was more prone to become trapped in a local minima and despite several attempts a more optimum solution was not found. A sample of the results obtained by presenting the training data to this ANN are given below in Figure 7.37 to 7.39.

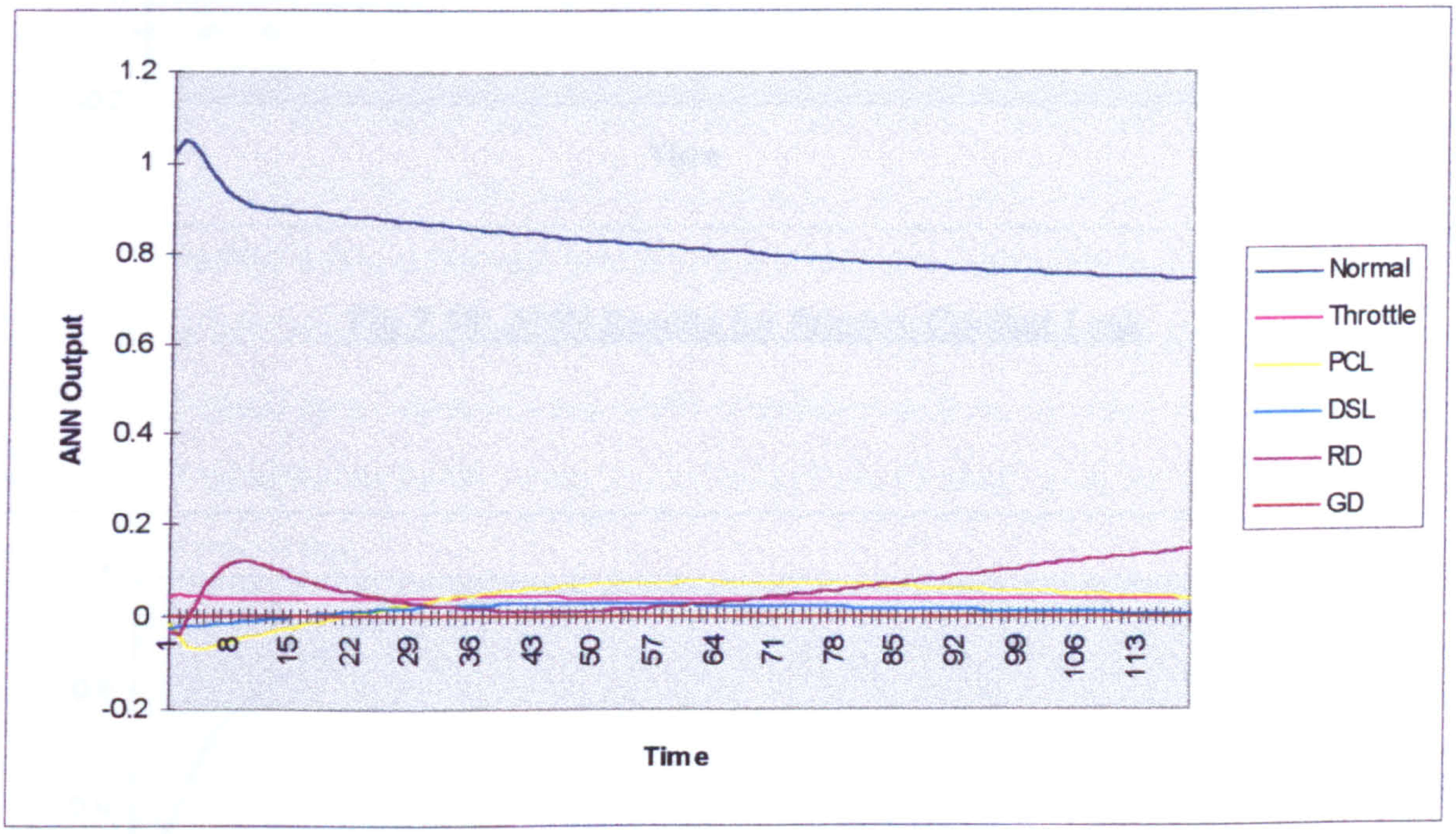


Fig 7.37: ANN Results for Normal Operating Conditions

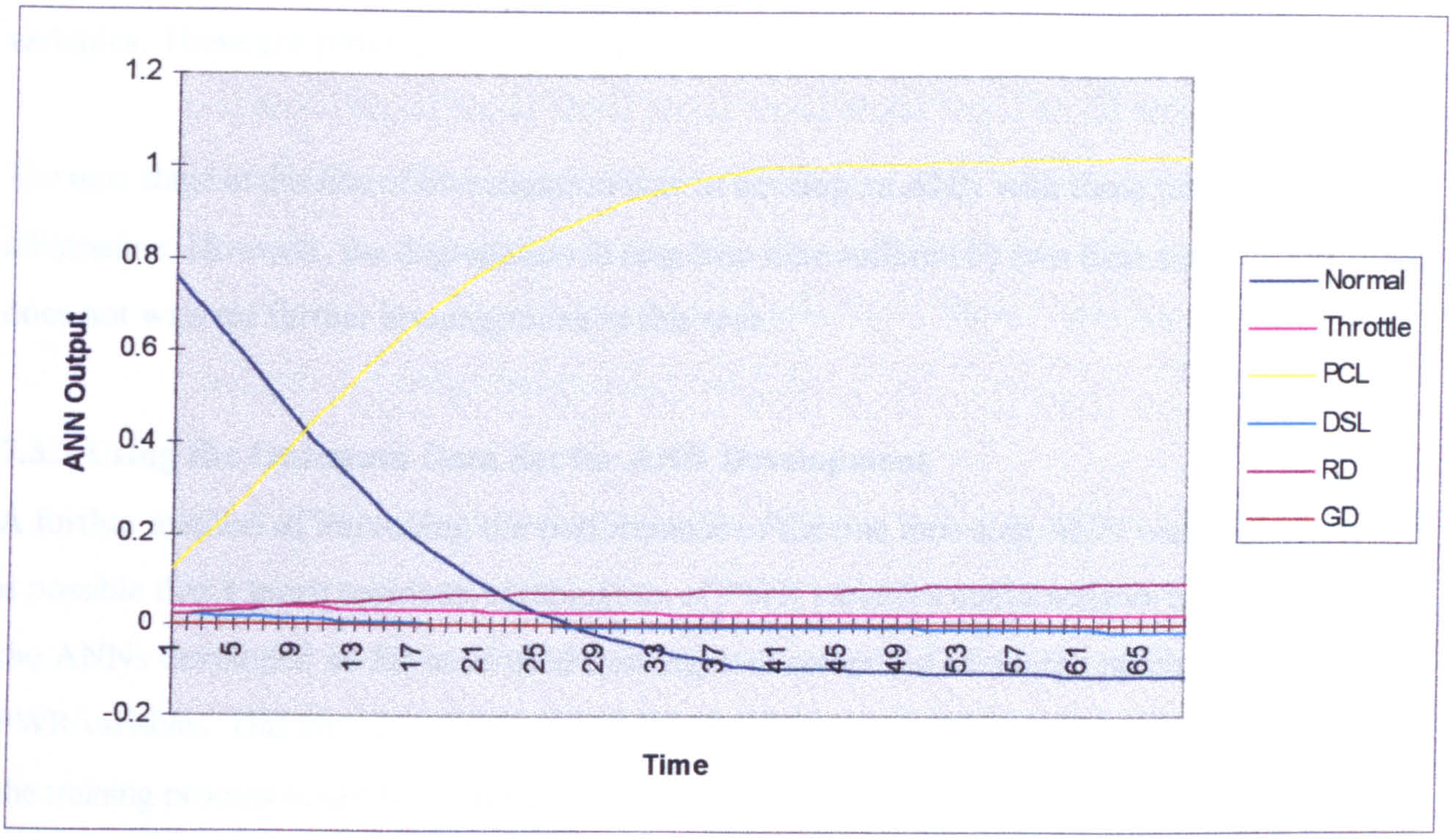


Fig 7.38: ANN Results for Primary Coolant Leak

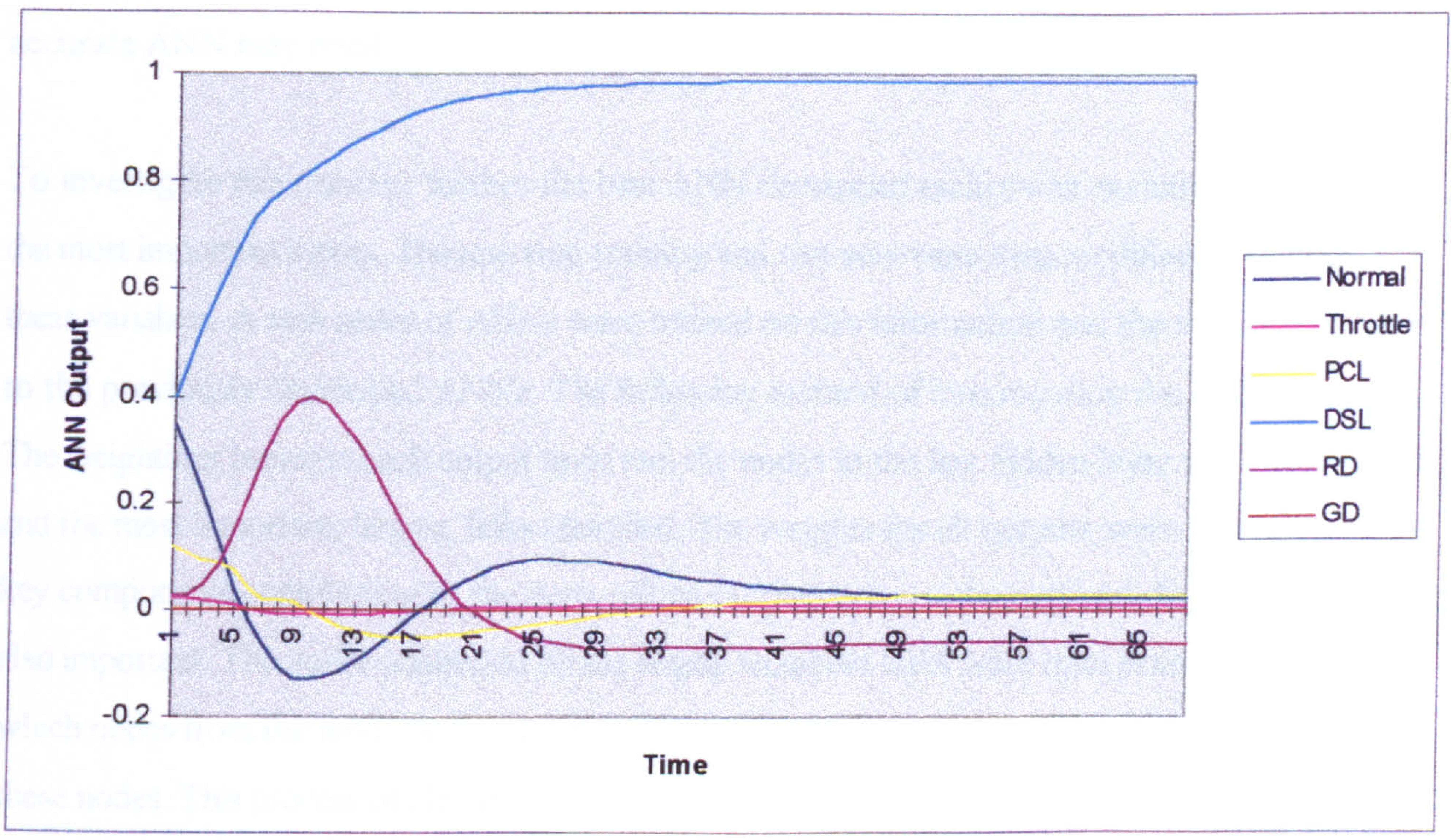


Fig 7.39: ANN Results for Downstream Steam Leak

The above results show that while the transients are correctly identified the time to do so is noticably slower than the one time step ANN. The inclusion of an extra time step has produced a far more complicated network with many additional internal relationships between the plant that were key inputs for a diagnostic ANN. The set consisted of three region temperatures, the pressuriser pressure, the pressuriser level and the reactor start up rate. These six variables were found to have a far greater relevance on the ANN output than the remaining variables so the subjective nature of the method is reduced. The training data set was modified to only contain these six variables as inputs, the outputs remained as before. This set was randomly divided into training and test sets in the approximate ratio of 2:1.

A further series of ANNs were trained on this data. As before the training data was presented for 80,000 cycles with testing every 100 cycles with the test set, the best network being saved. The full results obtained are given in Appendix M. The best ANN developed had a RMS error of 0.2414. This value is far higher than both the best one step ANN RMS of 0.0868 for the full data set and the best two time step ANN RMS error of 0.1015. Although the most important input variables were used for developing the ANN this result indicates that the other variables also contribute to the accuracy of the diagnosis. The diagnosis accuracy is still acceptable as shown in the following sample of output graphs, Figures 7.40 and 7.41.

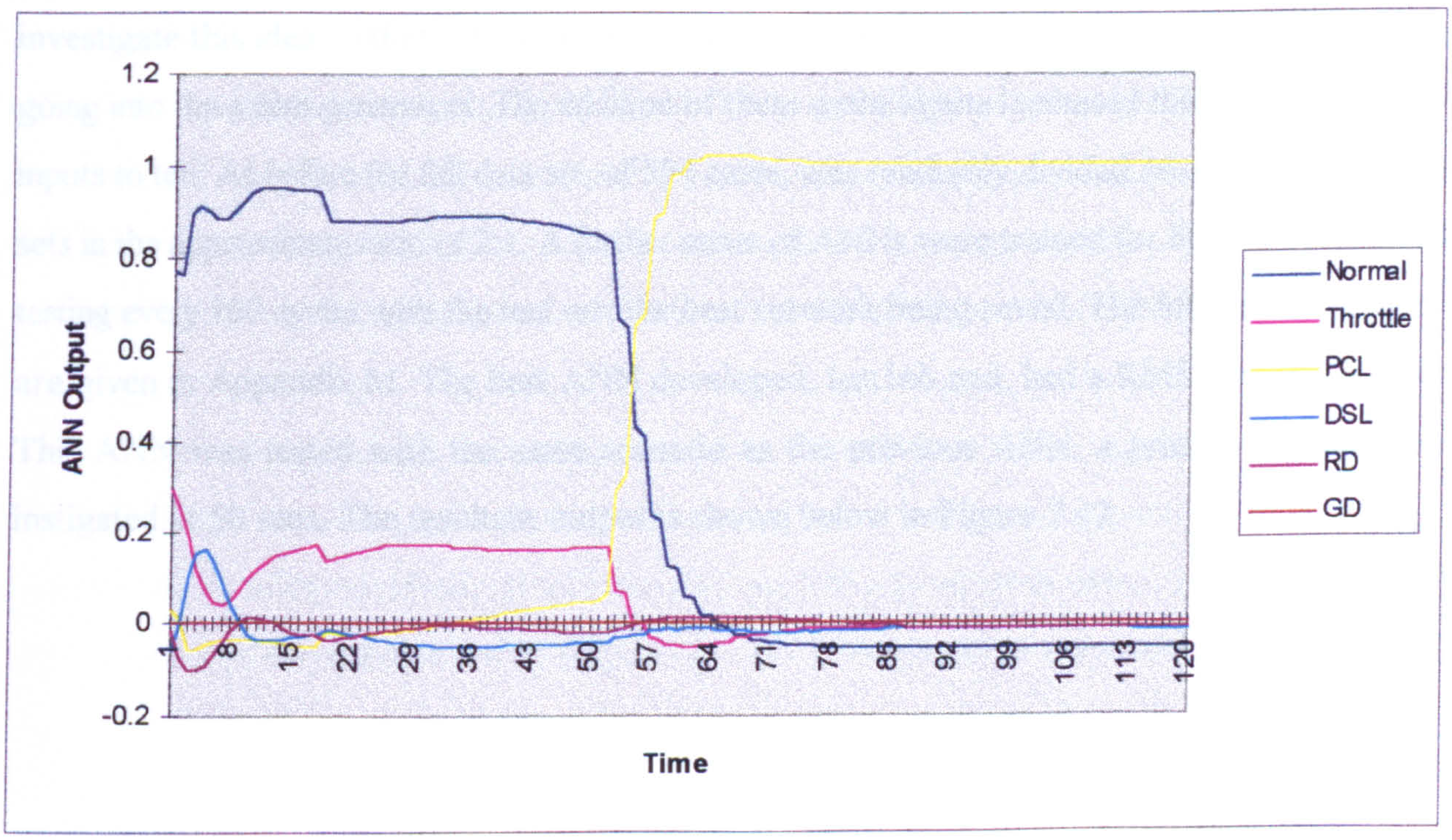


Fig 7.40: ANN Results for Primary Coolant Leak

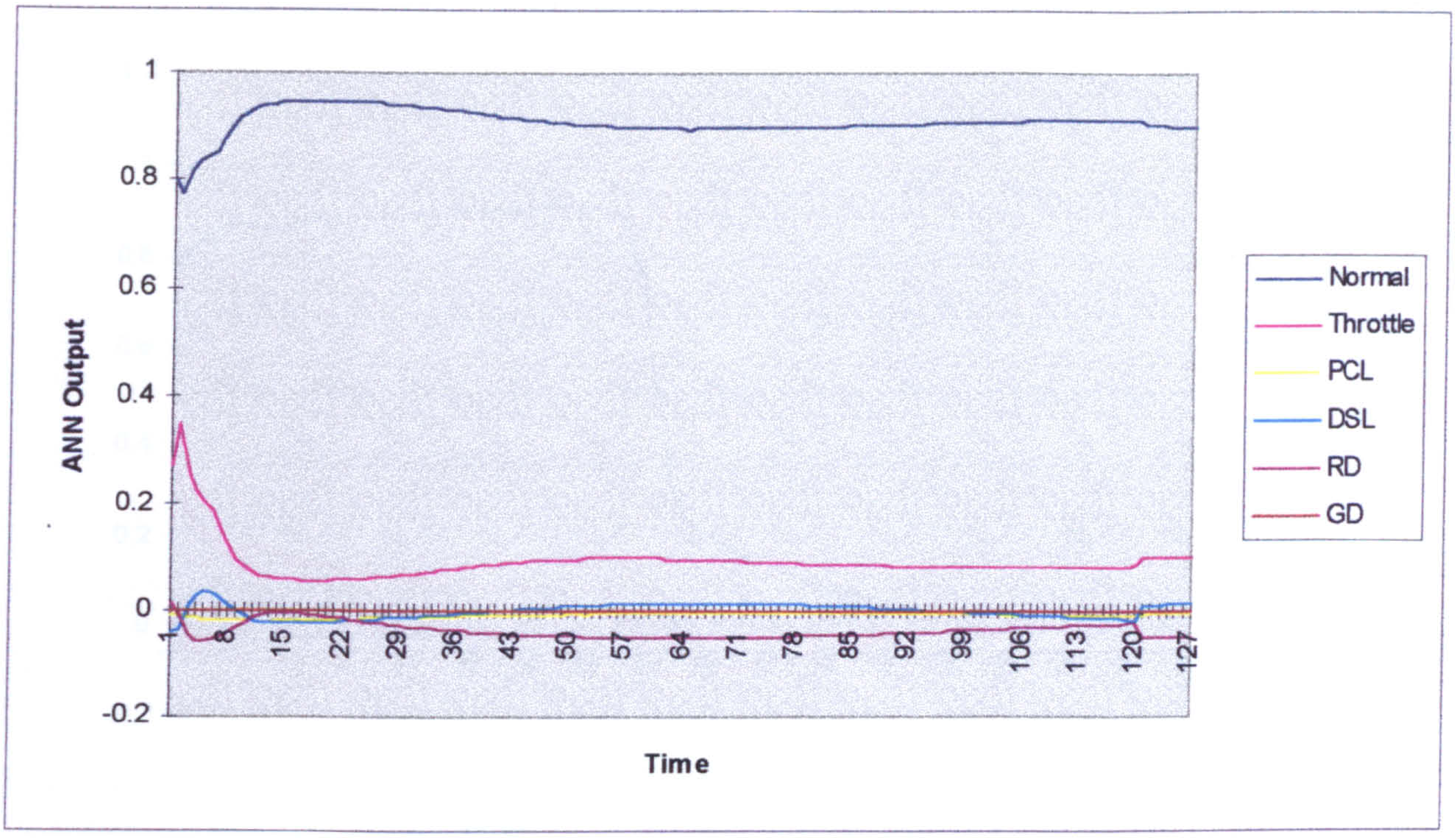


Fig 7.41: ANN Results for Normal Operating Conditions

The next four most important input variables were then included into the training data sets to investigate this idea further. These inputs were additional regional temperatures and the energy going into the steam generators. The addition of these extra inputs increased the number of ANN inputs to ten. As before the full data set, of 554 cases, was randomly divided into training and test sets in the approximate ratio of 2:1. A further series of ANNs were trained for 80,000 cycles with testing every 100 cycles with the test set, the best network being saved. The full results obtained are given in Appendix M. The best ANN developed, test16b.nnd, had a RMS error of 0.2530. This ANN was tested with the same scenario as the previous ANN, a primary coolant leak instigated at 50 secs, The resultant output is shown below in Figure 7.42.

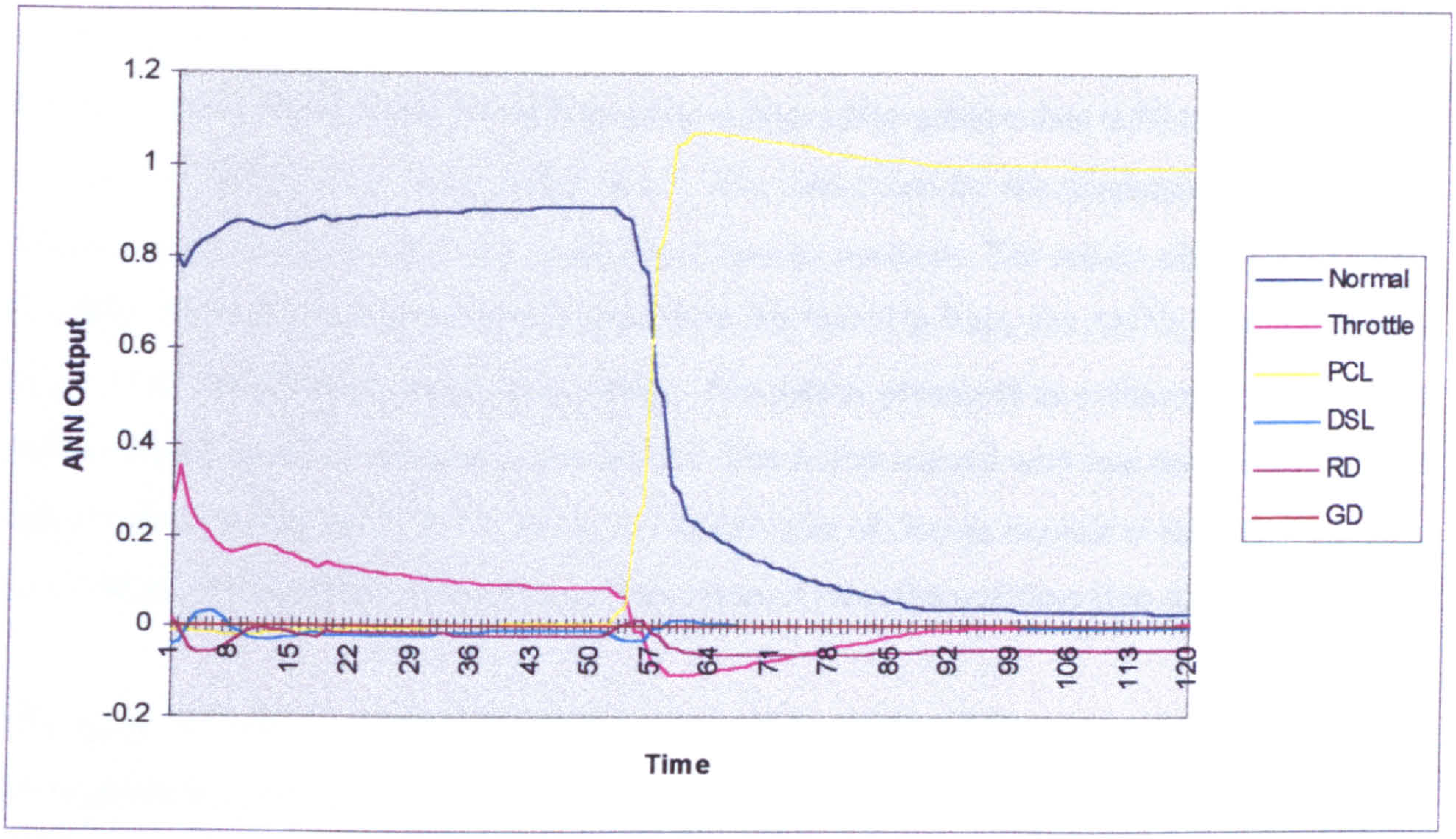


Fig 7.42: ANN Results for Untrained Transient Test

This result was an improvement over the previous ANN but, while producing an accurate diagnosis, was still was not as rapid a diagnostic tool as the full data set ANN. This process could be continued and the next most important input variables could be included in the training data and a further series of ANNs trained but the results do not suggest that a significantly better diagnostic tools would result. The investigation was therefore halted at this stage.

An accurate and rapid diagnostic ANN has been developed for the identification of a set of key PWR transients. Several approaches have been explored to obtain the optimum diagnostic tool. These have included combinations of input data set and time steps of data. The best diagnostic ANN developed consisted of a single time set of the full data set of measurable plant variables.

7.4 Discussion

The results obtained in the previous section are now discussed in more general terms. The developed diagnostic ANNs are then examined and contrasted with a similar ANN developed for the same task (Weller et al., 1997b).

In general the accuracy of all the diagnostic ANNs was very good. No transient conditions were

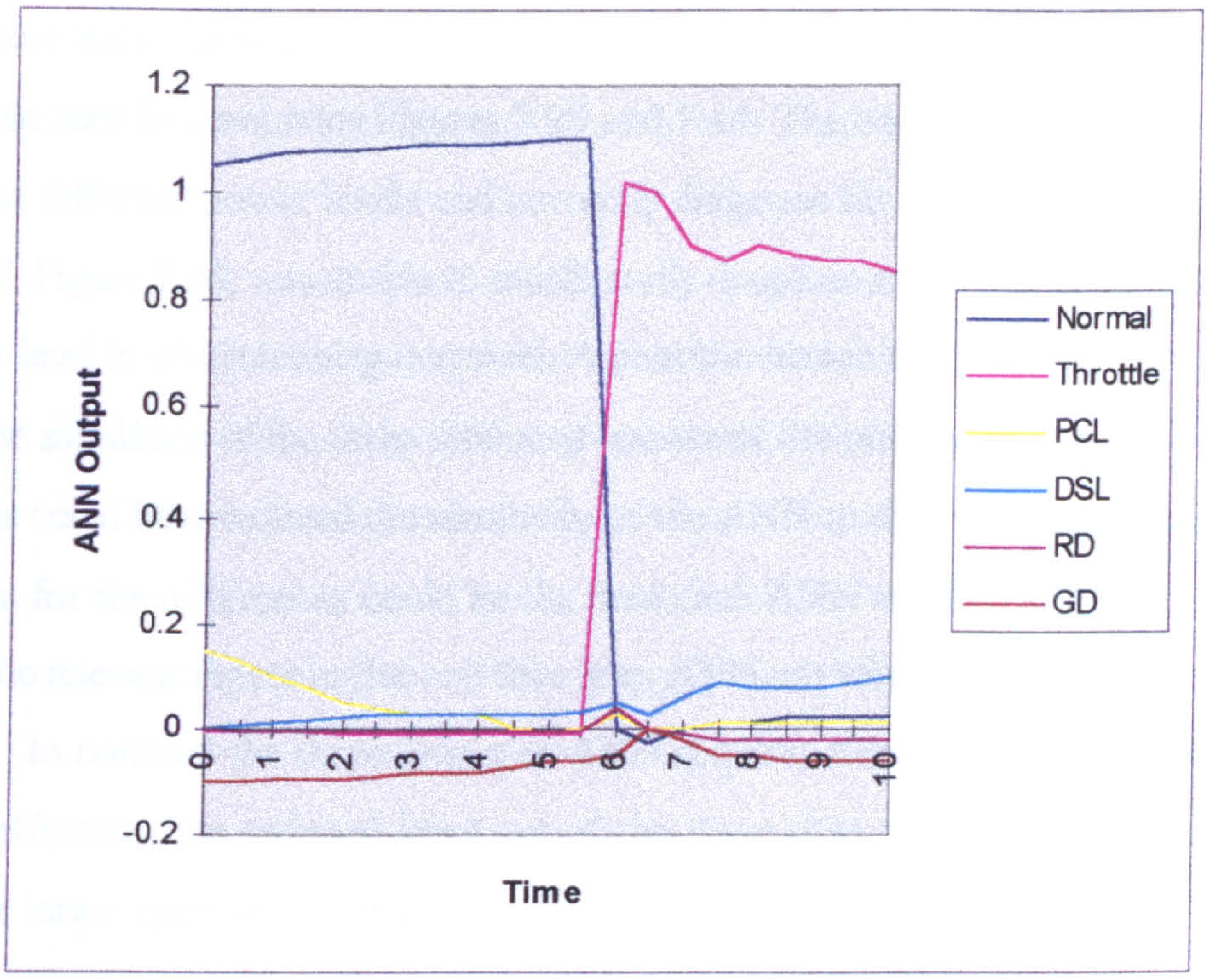


Fig 7.43: Three Time Step ANN Results for Throttle Opening Transient

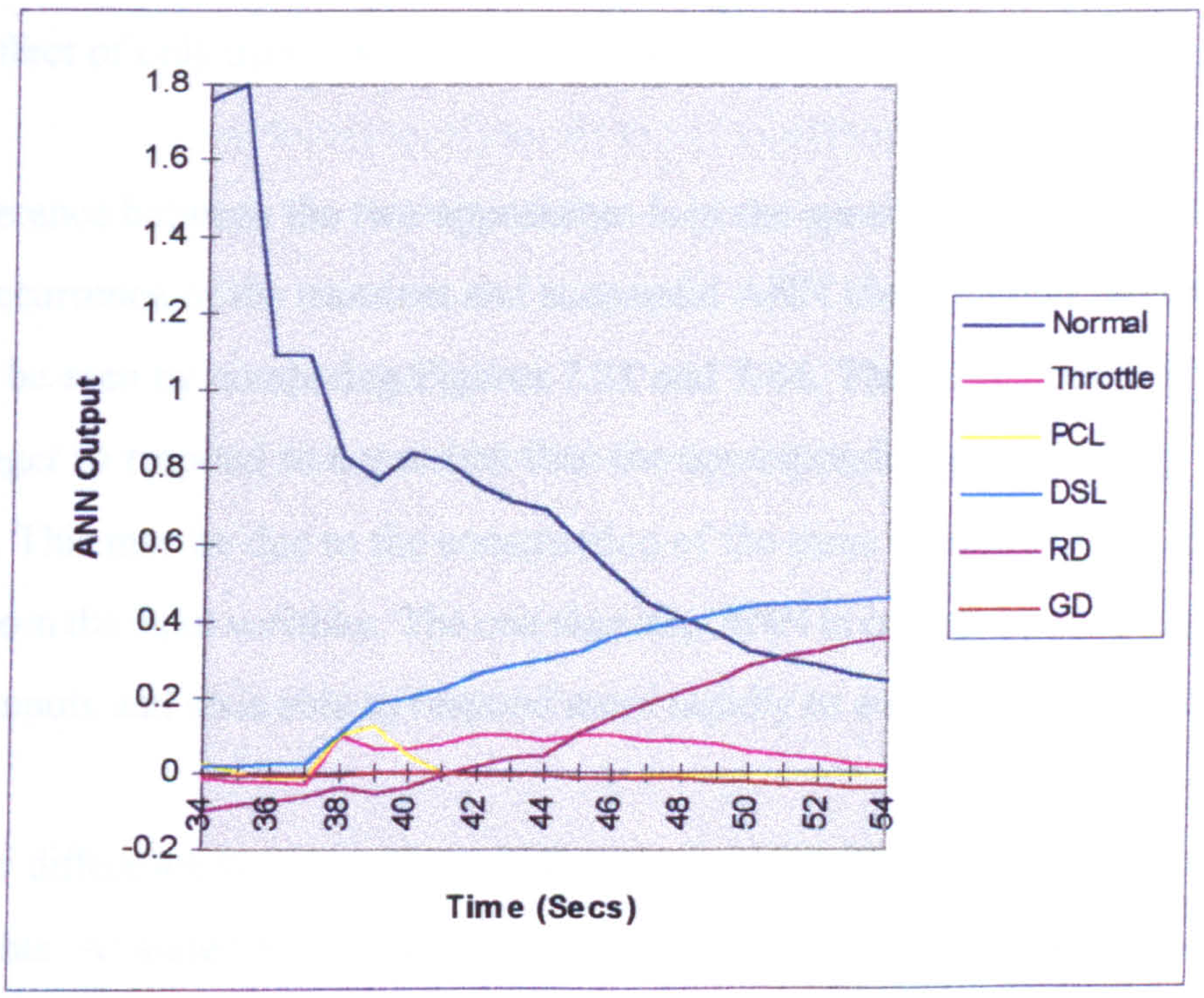


Fig 7.44: Three Time Step ANN Results for Downstream Steam Leak at Full Power State, 60% Reactor Power

The comparison between the best ANNs from the two investigations show some significant differences. These indicate that an ANN trained with a single time step of a large data set may provide the better diagnostic system.

more remarkable given that the training data consisted entirely of single transient conditions. The resulting trained ANN is sufficently robust that combinations are generally well diagnosed, Figures 7.29 and 7.35 being exceptions with some other combinations being dominated by one of the transients, for example the primary coolant leak in Figure 7.31. The salient features of each transient has been embedded in the internal ANN architecture but the embedding has been sufficiently general to enable different leak sizes to be successfully diagnosed. In reality an extra transient condition has been included in all the above tests as the throttle position was initially slightly increased. Many of the previous Figures show this situation, Figure 7.32 for example shows the throttle transient being diagnosed at approx. eight seconds.

The time for accurate diagnosis is also very encouraging. The single transient conditons were instigated at fifty seconds and in all cases these were reasonably quickly diagnosed. The size of transient was reflected in the time for diagnosis. A large primary coolant leak, as shown in Figure 7.23, is correctly diagnosed in about ten seconds whereas a small primary coolant leak, Figure 7.24, is not correctly diagnosed until seventy seconds after occurrence. A human operator would experience similar difficulties and could take a similar period to respond. The two transient scenarios are also diagnosed in a respectable time. The second transient was instigated at seventy-five seconds and these transients, if identified, are done so in a similar time to the single condition, Figure 7.27 for example.

The ANN developed in this section appears to be a possibility for the diagnostic element of the proposed advisory system. However, for completeness, the strategy outlined in the introduction will be followed and alternative methods developing the diagnostic ANN will be investigated.

7.3.2 Using Two Time Steps for ANN Development

All the ANNs developed have consisted of one time step of plant information while the results obtained have been very accurate better solutions may possibly be obtained from a different arrangement of training data. Tests were conducted to investigate the effect of two time steps of PWR data on the diagnostic ability of an ANN. The data sets from the last test, test13.nna, were modified to include two sets of 69 variables as inputs, 138 inputs in total. The 6 output nodes used in the previous tests were retained. The concatenation of data files reduces the number of examples in the training set by one example. This new data set was randomly divided into training and tests in the approximate ratio of 2:1.

variables. These are proving to be counter productive to the final result of the output nodes.

The next stage in this line of investigation was to develop an ANN with three time steps of input information. However, the degregation in response time suffered by two time steps of input data does not warrent further investigations in this area.

7.3.3 Using the Optimum Data Set for ANN Development

A further method of improving the performance of the one time step ANN was investigated. It is possible that a more optimum combination of PWR variables could achieve better results. All the ANNs developed so far have used training data composed of a large number of recordable PWR variables. This arrangement may not be optimal as some of the implicit relationships during the training process could be detrimental to the accuracy of the final ANN. A particular variable may not be important to the diagnostic process yet the links between this input and the rest of the ANN structure still have a weighting which may become very small during the training process but will never reach zero. Each of these little weightings adds to the final error of the trained ANN. If these superfluous inputs were pruned out of the network training data a better, more accurate ANN may result.

To investigate this concept further the best ANN developed earlier was examined to determine the most important inputs. The one step training and test sets were then modified to only contain these variables. A new series of ANNs were trained on this information and the results compared to the previously developed ANNs. The following method of interrogating the ANN was used. The weightings between each output node and the nodes in the last hidden layer were considered and the most important, largest, links identified. The weights for all outputs were inspected as the key components contributing to the zero valued outputs not just the required diagnosis node are also important. The nodes connected by the largest weighted links were then examined to identify which nodes from the next, earlier, hidden layer that contributed major largest weighted links to these nodes. This process of identifying the most important nodes is continued backwards through the ANN until the input layer is reached and the key PWR variables determined. The range of the inputs does not have a bearing on this approach as the ANN design software package used normalises each input so all variables presented to the ANN are in the range 0 to 1. Whilst this method is very subjective it should give a guide to the most important variables.

The results of applying this method to the best one step ANN identified a set of six plant variables incorrectly classified although the results from some of the poorer performers were close to the threshold levels. These were pre-defined as an output value greater than 0.85 to be classified as a 1 and less than 0.15 to be classified as a 0. The time taken for the diagnosis was also good, certainly comparable and in many cases faster than an operator. The results obtained for single time step ANNs are more surprising as, apart from the Start Up Rate, the ANNs were not trained on or given any reference to rates of change. The data is presented as a discreet value with no mention of the previous values for each variable. The ANNs trained with two time steps of PWR data are the exception and had a concept of variable rate of change implicit in their structure but their diagnostic capabilities were not an improvement over the one time step ANNs.

The same techniques for ANN development were employed for a separate continuation investigation into the diagnostic ability of ANNs (Weller et al., 1997b). For this work the nature and contents of the input set was defined by a domain expert. A set of twelve PWR variables were selected and these were used to construct training data sets. A further contrast was the use of three time steps of plant information for the ANN input. The same six transient conditions were used for outputs. The ANNs therefore consisted of thirty-six inputs and six binary outputs. As before the transients were modelled on a PWR simulator. The resulting data file was modified into an ANN input set and randomly divided into training and test sets. An extensive range of ANNs were developed with these data sets, the best one, with a RMS error of 0.014, was used for further testing on a similar range of scenarios as before. A sample of the ANN outputs obtained are shown in the following set of diagrams, Figures 7.43 and 7.44. The threshold criteria for classifying an output was set at greater than 0.9 for an output of 1 and less than 0.1 for an output of 0. The RMS error result for the best of these ANNs was considerably better than those of the previous ANNs.

The first of these is the response to changes in the power level of the PWR. An example of this difference can be seen by comparing Figures 7.25 and 7.44. The one step ANN, Figure 7.25, was able to consider different power levels and correctly diagnose the transient condition. The three time step ANN, Figure 7.44, was unable to satisfactorily diagnose transients at power levels other than the power level in which training occurred. A possible reason for this difference could be for the whole of the simulation of the three time step transients the reactor heaters were modelled as on. This feature could have reduced the sensitivity of the ANN to different power levels. Another possible reason for the differences could be the time each ANN takes to respond to a change in power level. The relevant inputs in the one time step ANN are affected immediately by a change in power level. In contrast the three time step ANN requires three time steps for all the related inputs to be affected. The reduced input set of the three time step ANN may also not be as sensitive as the large input set of the one time step ANN.

A second difference is the ANN responses to throttle transients. This topic has been considered earlier in this chapter but the comparisons between the throttle output in Figures 6.19 and 6.43 illustrate the effect of only using the actual changes in throttle position to identify the transient.

A further difference between the two approaches is in the speed of response, the elapsed time between the occurrence of the transient and successful ANN identification. An example of this difference can be seen by comparing Figures 7.21 and 7.44. The three time step ANN, Figure 7.44, takes longer to respond to a transient than the corresponding one time step ANN, shown in Figure 7.21. This may be due to the construction of the three time step ANN involving rates of change between the input variables. The one time step ANN in comparison only utilises discreet values for the inputs and so is able to respond more rapidly to a changing environment.

The last major difference between the two forms of ANNs is in the accuracy of diagnosing multiple transients. As stated earlier the occurrence of such conditions on actual plant is so rare that the diagnosis of them is of more academic than practical interest. An example of this difference can be seen by comparing Figures 7.33 and 7.44. The one time step ANN, Figure 7.33, is considerably better at diagnosing all multiple transient conditions than the three time step ANN, Figure 7.44. The main reason for this may be in the construction of the initial training data sets. If the data for each transient condition was not edited to remove the initial normal operating period or throttle opening information the ANN may lose some specificity on training. The ANN possessed a better RMS error than the one time step ANN but any generality was lost with the complex nature of the training data. The possible reliance of the three time step ANN on rates of change between a small set of variables may not be as robust as the large single data set of the one time step ANN.

The latter work also investigated a extra refinement in that the best ANN was interfaced with the simulator to provide an on-line and real time diagnosis of the PWR condition. The ANN was coded as a module of the simulation computer program and operated in parallel with the simulator. As a transient was initiated on the simulator the ANN provided a continuous diagnosis. The results of this work (Weller et al., 1997b) showed that all single transient conditions were correctly identified. Furthermore the time for diagnosis was quicker than that for an operator. Additionally some combinations of transients were also correctly diagnosed. The success of this work indicates that the on-line operators advisor introduced in Chapter 4 could be a practical consideration.

All of the ANNs produced a reasonable diagnostic system. Every ANN converged during training and produced sensible results with the test data. The results were not necessarily optimum, a slightly better ANN may have been obtained by extra training but the improvement would not have been significant. None of the ANNs developed were considered particularly bad. A reason for this may be that the training data was well organised and the outputs realistic. Furthermore, a PWR may be a well organised system which can be modelled by a smooth, continuous, multi-dimensional surface although some drastic rates of change may occur, especially during large Loss Of Cooling Accidents (LOCAs).

All the ANNs were trained on single fault conditions, a downstream leak or a rod drop for example, yet the trained ANN was able to successfully diagnose combinations of these conditions without having been presented with the combination during training.

Throughout the investigation the ANN with the lowest RMS error was considered as the best ANN for each particular task. As described in section 7.2 this is not always the case. An ANN with a large number of little inaccuracies would be a better diagnostic ANN than one with a large error providing small errors within upper limit for correct output. However the randomly selected test data, although not totally independent of the training process, was felt to be a sufficient test for diagnostic accuracy. While this approach may not have selected the absolute best ANN for further investigation, the ANNs chosen were sufficiently accurate to justify the selection criterion.

The biasing of the ANN training data appears not to be a very sensitive process. In all the ANNs developed the data for normal operating conditions outweighed each example of the other transients. There were 120 cases of normal operating data compared to 70 cases of each of the remaining conditions. This asymmetrical loading did not seem to have an adverse effect on the performance of the ANNs. The biasing of the training data towards particular transients, for example a primary coolant leak, similarly did not produce enhanced diagnostic capabilities for the selected condition. The final training data set only contained one example of each transient with a medium sized leak, the exception was the throttle transient which contained a number of throttle opening and closing scenarios to produce an equal number of cases.

A final, trivial observation of the ANN output graphs is the almost symmetrical nature of the values. Several of the graphs exhibit symmetry about a horizontal line approximately through the ANN output of 0.5, Figure 7.36 for example. This symmetry may as a result of the rapid diagnosis produced by the ANNs. As one output condition is less likely to be the cause of the transient a second condition correspondingly gains in likelihood output. The overall total value of the system is approximately unity.

7.5 Conclusions - Diagnosing condition of complex systems

From this investigations an ANN would seem to be an ideal tool for diagnosing the condition of a complex system. The results obtained for a wide range of scenarios have been both accurate and quickly obtained, including multiple transients. The speed of diagnosis compares very favourably with that of a human operator.

As reported in Chapter 2 the use of ANNs to diagnose the condition of a nuclear reactor is not a new application. However, few of the reported systems have considered a PWR. The results of this chapter are therefore a contribution to rectifying that void.

Several different approaches to developing a diagnostic ANN have been investigated. In terms of speed of response, accurate diagnosis of previously untrained conditions and flexibility of response the input set containing a single time step of large PWR variables would appear to produce the better diagnostic ANN for the six conditions selected. It should be stressed that all the ANNs developed could have been used for a diagnostic system, the chosen approach only provided the best diagnostic ANN.

7.6 Summary

The work reported in this chapter has been concerned with developing a diagnostic ANN for determining the condition of a PWR. Some practical ideas were first described. The method of producing ANNs was then introduced. Six key transient conditions, including normal operation, were selected. A PWR simulator was then used to model these scenarios. The output from the simulator was modified for use in an ANN development computer program. Several forms and structures of ANN were then considered and produced. The best of which contained an input set of sixty-seven PWR measurable variable. This ANN was tested on a wide range of untrained transient conditions including different leak sizes and multiple faults. The results show a good generality of diagnosis as nearly all of these scenarios were correctly and promptly diagnosed. The model was felt to be sufficiently robust and generalised to be used as the diagnostic element of the advisory system.

Chapter

8 Implementation

8.1 Introduction

This chapter discusses a possible implementation of the methods developed in the previous chapters. This stage combines the predictive and diagnostic Artificial Neural Networks (ANNs) elements to form the advisor for the operator of a Pressurised Water Reactor (PWR) as introduced in Chapter Four. Currently the two systems have been developed separately with very little interface between them. Whilst this approach was necessary to explore the details and understand each requirement, a practical operators advisor requires both systems to communicate and interface together. The diagnostic component of the structure providing a guide to the condition of the PWR and the predictive section reporting future values of plant variables while in that condition.

The ideas discussed in this chapter are theoretical in nature. A prototype operators' adviser has yet to be constructed. Some practical experience was gained from interfacing the three time step diagnostic system with a PWR simulator, as discussed in the previous chapter. This was still artificial as the ANN resided as a program on a parallel computer and shared data files with the main simulator program.

The remainder of the Chapter is as follows. The strategy behind such an implementation is introduced and discussed. A set of practical considerations for this proposal are then addressed. The chapter also includes some thoughts on possible future work and direction.

8.2 Strategy for Implementation

The strategy for implementation is that all components of the advisor must be interfaced together so that they appear as a single program with no user input required to enable the program to function or to initiate any features. The result from the diagnostic ANN is automatically passed to the predictive ANN. This program can be initially configured for user preferences, such as screen colours or layout, but the key elements such as the ANNs cannot be user adjusted.

An important component not previously considered in great depth is a further program to control and maintain the system. This would also be a self-contained computer program embedded in the overall advisor package. It is envisaged that the controller software would manage and perform the important tasks of maintaining a record of the results of the diagnostics and predictions. A second vital task that this module could perform is that of organising the signals received from the PWR and organising them into the correct form and order for presentation to the ANNs. This function could also include a check or estimate on the acceptability of the data being received from the plant. A diagram showing the envisaged functions for the advisor management program is shown below in Figure 8.1.

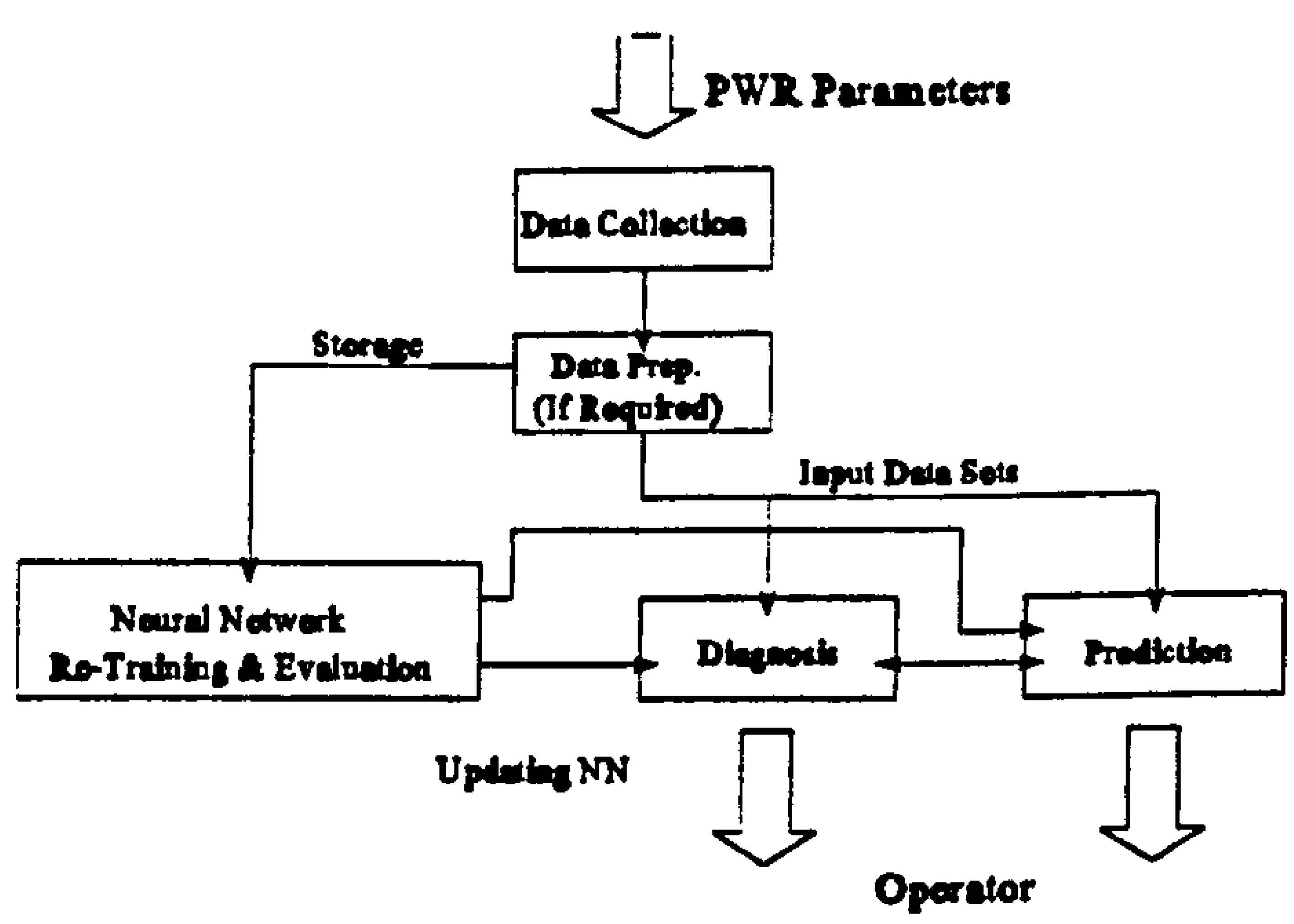


Fig 8.1 - Management Program Tasks

A new section shown in the above diagram is concerned with the updating of ANNs. Currently a neural network will not be changed once it has been developed. Training and test sets used in producing are established using data from simulators. In operation any situations that are different from these scenarios may be evaluated and the outcome determined by the implicit relationships within the ANN. However some unanticipated situations may occur that are outside even this framework and not included in the original data sets. A facility to update and, if possible, improve the performance of the advisor would be a necessary requirement of any practical implementation.

The training and test sets information could be stored in a database of conditions. During operation, each data set measured from the PWR could be compared to the contents of the database. If a similar information set is not currently part of the database the new details are added. After a predefined number of additions to the database or a predefined time period, the relevant neural network is retrained on an updated training set. The revised neural network is then implemented into the system. Initially it could operate in parallel with its predecessor but once its performance has been evaluated it could replace the old neural network if found to be an improvement. The following diagram, Fig 8.2, shows the proposed ideas.

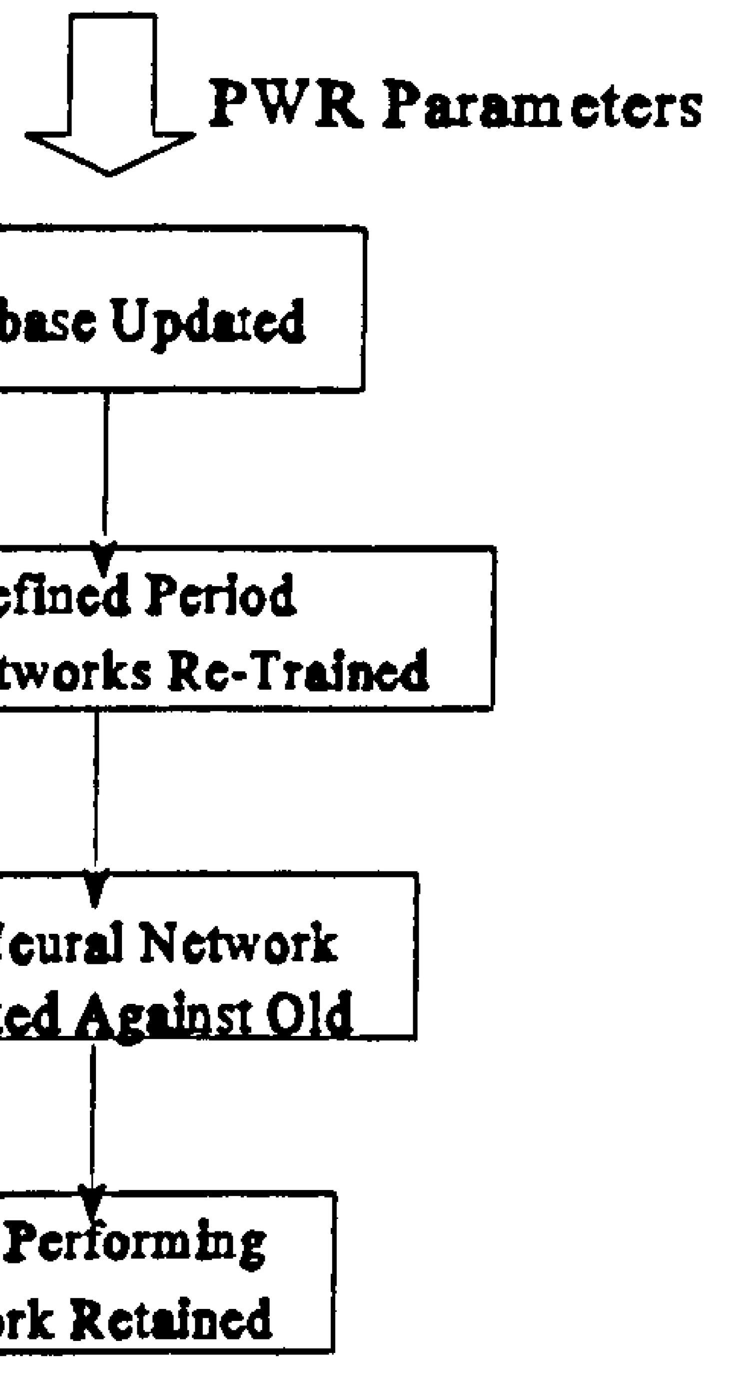


Fig 8.2 Proposed ANN Updating Sequence

It may not be possible to carry out the recording of a new condition in one stage. The data sets for the prediction element of the advisor will need to be stored temporarily until information on the actual plant condition is available and the outcome can be combined with the input set.

8.3 Practical Considerations

A realistic implementation of the operators advisor would consist of a software program, a computer with the facility to receive data from analogue and digital sources. The software program contains the embedded coding of the ANNs and the controlling software. Later versions of the advisor could utilise dedicated ANN circuitry but until this technology has become more accepted and usable a software solution is preferable.

The connection between the advisor and the source of data, either a simulator or a PWR, needs to be a high speed link to ensure real time monitoring. The required information is already available in the control room as inputs to the various displays currently monitored by the operator. The only additional connection therefore required would be between the computer with the installed advisor software and the control room instrumentation. The impact of such a system on safety, power loading and speed of data traffic would clearly need to be established.

The manner in which the output from the advisor is conveyed to the operator also requires investigation. Currently the ANNs merely produce a real valued output for both diagnosis and prediction. The method by which these results is represented to the operator is vital both in terms of the valuable of the advisor and the acceptability of the advisor by the operators as a useful tool. Many new fields of information representation, such as data mining, virtual reality or other Artificial Intelligence techniques could be explored to develop an optimum user interface.

The reliability and accuracy of the data being received by the advisor is an important consideration for a practical implementation. At present all the data has been produced with a PWR simulator and, although an element of noise was added during training, the information has been complete and accurate. The problem of data reliability is resolved in current PWR control rooms by an in-built redundancy of data measurement. Key variables are independently measured more than once and a comparison or voting technique used to determine the final value. A similar procedure could be adopted for the advisor. The duplicated readings could all be presented to the system and a method of resolution used to determine the variable value to be used for the ANNs. The techniques developed by Upadhyaya and Eryunek (1992) could be used to also give information on the accuracy of the readings.

One of the most important requirements of the operators advisor is that it operate in real time. The usefulness of the system as a tool for assisting the PWR operator is the speed of diagnosis and prediction. Good accuracy of the outputs is also an obvious criteria but this is a result of the ANN training and data selection. The accuracy of the ANN output is not a function of the speed of the system operation. Any practical implementation would need careful consideration of the operational quickness of each component and link in the system.

8.4 Summary

The chapter briefly examined a possible approach to a practical implementation of the operators advisor. Some of the envisaged problems were identified and potential solutions addressed. Some areas that require further investigation were highlighted and discussed. Although theoretical in nature the proposed approach should provide a sound basis for development of a usable tool.

many hierarchical layers, as introduced and discussed in Chapter Four. Instead a single ANN has been developed for each task. As considered earlier, in Chapters Four and Seven, the diagnostic feature could be expanded to produce more detailed results but this would require considerably more development of a PWR simulator program to produce sufficiently detailed training data. The optimisation of the ANN input nodes was also far simpler than initially envisaged. Chapter Four discussed the use of Genetic Algorithms for producing the best ANN input set. However the approach adopted for the predictive element defined the ANN input set and the diagnostic element used the full set of possible plant variables, although it is possible that some rationalisation is possible.

The ANNs that were finally developed were composed of relatively simple structures for the complexity of the tasks required of them. Most of the ANNs produced during the entire project contained only one or two layers of hidden nodes. The experiments with three hidden layers yielded no improvement on the results obtained with the simpler architecture. This result may be due to the nature of the PWR data which, although non-linear, is not chaotic. The solution space that the ANNs attempt to model is therefore well organised and this perhaps requires few hidden nodes to accurately map. The restricted scenario set considered during the project may have enhanced this observation but the conclusion is still believed to be valid.

A further general observation is that only information from one time step was required for both the elements of the advisory system. Originally a series of recent plant history was envisaged for input to the ANNs. However comparisons in ANN performance vs number of time steps for both predictive and diagnostic functions, reported in Chapters Five and Seven, reveal no advantage gained with the larger number of time steps. The converse is probably more likely as the resulting larger ANNs would be harder to train and require more information for learning.

It is intended that the techniques investigated are valid in applications in other critical, non-linear systems, such as patient monitoring in a hospital intensive care unit or share dealing in the financial markets. In such arenas it is believed that the results from this work could make an important contribution.

9.2.1 Discussion on Predicting PWR Variables

This work has shown that an ANN can be successfully used to predict PWR condition both one and many time steps ahead. The predictions are generally accurate enough for the PWR operator

Chapter

9

Conclusions

9.1 Introduction

This final chapter discusses the results of the previous chapters and assesses the achievements of the project.

9.2 Discussion

This work has shown that Artificial Neural Networks (ANNs) can be successfully used as the basis for a Pressurised Water Reactor (PWR) operators advisor. The ANNs developed during the project effectively performed their envisaged tasks, diagnostics and prediction. The advisor is capable of diagnosing a limited, though important, range of PWR conditions and then predicting future plant state based on that diagnosis. While these systems will require further refinement and evaluation for a prototype advisor to be constructed, the initial concept and grounding appears sound.

A surprising result from the investigations is that the proposed structure of the advisory system is quite simple, consisting merely of a hierarchy of two ANN structures. This result is quite different to the initial concept which consisted of a far larger number of ANNs configured into to assess the future state of the plant and gauge the results of changes made to the PWR. One sophisticated model, containing a single basic ANN element, can be used to predict a large range of PWR conditions. An ANN direct equivalent of a simple one compartment model has been developed, refined and shown to be a minimum structure. This is used to model PWR non-linearity for the prediction of plant temperature many time steps ahead and for a wide range of conditions.

The results and conclusions from the work on predicting PWR variables, reported in Chapters 5 and 6, can be summarised as follows:

- ANN technology has been shown to be a valid tool for predicting variables in a PWR
- A set of different forms of PWR variables suitable for use as ANN inputs have been identified
- The number of PWR transients that can be accurately predicted by an ANN was identified as a key consideration
- An exact equivalent of the energy conservation equation was developed to resolve the predicting of a number of transients problem
- This model was refined to include only standard ANN transfer functions
- The refined model was proved to be the minimum structure for this task
- A model of the PWR primary circuit was constructed using 25 of these basic ANN elements
- This model was shown to accurately predict PWR transients
- The structure is capable of predicting all the fault conditions that the simulator could model

9.2.2 Discussion on Diagnosing PWR Condition

A second ANN was successfully developed for the diagnosis of six key PWR transient conditions. These transients were successfully diagnosed at different power levels to those used for the training sets. Furthermore the developed ANN was sufficiently generalised to correctly identify transient scenarios not included in the training data. Combinations of transients, again not in training set, were also correctly diagnosed. From this work a key set of PWR variables were identified.

The results and conclusions from this work can be summarised as follows:

- A set of six key PWR transients has been identified
- A single time set of PWR variables is sufficient for accurate ANN diagnosis of plant condition
- Correspondingly a large set of PWR variables is better for accurate ANN diagnosis of plant condition compared to a selected set of perceived key variables.

- An ANN trained to identify single fault conditions can also correctly diagnose multiple transient scenarios, which were not included in the training data.

9.3 Summary

This project has shown that ANN technology can be used as a basis for the construction of a computer based PWR operators advisory system. Although further work and testing is required to develop these ideas into a usable prototype the ideas from this project will provide a sound framework.

References

Bartlett E.B., Basu A. (1991). A Dynamic Node Architecture Scheme for Backpropagation Neural Networks. Intelligent Engineering Systems Through Artificial Neural Networks, Page 101. American Society of Mechanical Engineers Press, New York.

Bartlett E.B., Uhrig R.E. (1992). Nuclear Power Plant Status Diagnostics Using an Artificial Neural Network. Nuclear Technology, Vol 97, Pages 272-281

Basu A., Bartlett E.B. (1994). Detecting Faults in a Nuclear Power Plant by Using Dynamic Node Architecture Neural Networks. Nuclear Science and Engineering 116, April 1994, Pages 313-325.

Bishop C. (1996). Neural Networks for Pattern Recognition. Oxford University Press. ISBN 0-19-853864-2.

Cheon S.W. et al. (1993). Applications of Neural Networks to a Connectionist Expert System for Transient Identification in Nuclear Power Plants. Nuclear Technology, Vol 102, Pages 177-191.

Caudill M. (1990). Expert Networks. Neural Network PC Tools, ed. R.C. Ederhart, R.W. Dobbins, Academic Press. ISBN 0-12-228640-5

Dasgupta et al. (1992). Designing Application-Specfic Neural Networks using the Structured Genetic Algorithm. International Workshop on Combinations of Genetic Algorithms and Neural Networks 1992 (COGANN-92), USA. Proceedings, Pages 87-96

Elias J.G. (1992). Genetic Generation of Connection Patterns for a Dynamic Artificial Neural Network. International Workshop on Combinations of Genetic Algorithms and Neural Networks 1992 (COGANN-92), USA. Proceedings, Pages 38-54

Gacem A. et al. (1990). A Neural Network to Extract Implicit Knowledge from a Nuclear Data Base. Transactions of the American Nuclear Society, Volume 62, Page 129.

Gia-Shuh Jang et al. (1993). Intelligent Stock Trading System with Price Trend Prediction and Reversal Recognition Using Dual-Module Neural Networks. Journal of Applied Intelligence, Volume 3 (1993), Pages 225-248.

Guo Z. et al. (1992). Using Genetic Algorithms to Select Inputs for Neural Networks. International Workshop on Combinations of Genetic Algorithms and Neural Networks 1992 (COGANN-92), USA. Proceedings, Pages 223-234.

Heger A.S. et al. (1989). Development of an Adaptive Interface for Nuclear Data Bases. Transactions of the American Nuclear Society, Volume 60, Pages 192-193.

Holbert K.E., Upadhyaya (1990). An Integrated Signal Validation System for Nuclear Power Plants. Nuclear Technology, Vol 92, Pages 411-427.

Holland J.H. (1975). Adaption in Natural and Artificial Systems. MIT Press,
ISBN 0-262-58111-6

Ikonomopoulos A. et al. (1994). Integration of Neural Networks with Fuzzy Reasoning for Measuring Operational Parameters in a Nuclear Reactor. Nuclear Technology, Vol 104, Pages 1-12

Jouse W.C. (1990). Neural Control of Temperature and Pressure During PWR Start-Up. Transactions of the American Nuclear Society, Volume 61, Pages 219-220.

Jouse W.C., Williams J.G. (1993). Safety Control of Nuclear Power Operations Using Self-Programming Neural Networks. Nuclear Science and Engineering, May 1993, Pages 42-54.

Karunanithi N. et al. (1992). Genetic Cascade Learning for Neural Networks. International Workshop on Combinations of Genetic Algorithms and Neural Networks 1992 (COGANN-92), USA. Proceedings, Pages 134-145.

Kim Han Gon et al. (1993). Pressurised Water Reactor Core Parameter Perdiction Using an Artificial Neural Network. Nuclear Science and Engineering, Jan 1993, Pages 70-76.

Kim Hyun-Koon et al. (1993). Neural Network Model for Estimating Departure from Nucleate Boiling Performance of a Pressurised Water Reactor Core. Nuclear Technology, Vol 101, Pages 111-121.

Lin, Bartel & Uhrig (1995). Predicting the Severity of Nuclear Power Plant Transients using nearest neighbours modelling optimized by Genetic Algorithms on a Parallel Computer. Nuclear Technology, Vol 111, July 1995, Pages 46-62.

Masters T. (1993). Practical Neural Network Recipes in C++. Academic Press. ISBN 0-12-479040-2

Miller L.F. et al. (1992). An Application of Neural Networks and Artificial Intelligence for In-Core Fuel Management. Transactions of the American Nuclear Society, Volume 66, Pages 108-109.

Minsky M.L, Papert S.A. (1988). Perceptrons: Expanded Edition. MIT Press, Cambridge, MA. 1988. ISBN 0-262-63111-3

Mulsant B.H. (1990). A Neural Network as an Approach to Clinical Diagnosis. M D Computing, Volume 7, Number 1, Pages 25-36.

Ohga Y. et al. (1993). Abnormal Event Identification in Nuclear Power Plants Using a Neural Network and Knowledge Processing. Nuclear Technology, Vol 101, Pages 159-167.

Parlos A.G. et al. (1992). Nonlinear Identification of Process Dynamics Using Neural Networks. Nuclear Technology, Vol 97, Pages 79-96.

Parlos A.G. et al. (1994). Empirical Model Development and Validation with Dynamic Learning in the Recurrent Multilayer Perception. Nuclear Technology, Vol 105, Pages 271-290.

Patterson W.C. (1987). Nuclear Power, Second Edition. Penguin Books, UK. ISBN 0-14-022499-8.

Potter M.A. (1992). A Genetic Cascade-Correlation Learning Algorithm. International Workshop on Combinations of Genetic Algorithms and Neural Networks 1992 (COGANN-92), USA. Proceedings, Pages 123-133.

Reid (1993). ABLAZE: The story of Chernobyl. Secker and Warburg Ltd. ISBN 0-436-40963-1

Rumelhart D.E., McClelland J.L. (1986). Parallel Distributed Processing. PDP Research Group. MIT Press. ISBN 0-262-63112-1

Refenes A.N. (1991). Constructive Learning and its Application to Currency Exchange Rate Forecasting. Neural Network Applications in Investment and Financial Services. Probus Publishing, USA.

Refenes et al. (1993). Currency Exchange Rate Prediction and Neural Network Design Strategies. Neural Computing and Applications, Volume 1 (1993), Pages 46-58

Refenes A.N. (1994). Measuring the Performance of Neural Networks in Modern Portfolio Management: Testing Strategies and Metrics. Conference on Adaptive Computing and Information Processing, London, UK. Proceedings, Volume 1, Pages 209-224.

Sang Ki Moon et al. (1994). Classification and Prediction of the Critical Heat Flux using Fuzzy Logic and Artificial Neural Networks. Nuclear Engineering and Design 150 (1994), Pages 151-161.

Shannon, Weaver (1971). The Mathematical Theory of Communcation, University of Illinois Press, Urbana, Illinois. 1971.

Summers R. et al. (1994). Cogent: An Expert Network for Medical Diagnosis. ACTA IMEKO XIII, Turin, Italy. Proceedings, Volume 2.

Swain I.D., Guttmann H.E. (1983). Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Applications, Sandia National Laboratories, Alburgrueque, New Mexico, August 1983.

Todreas N.E., Kazmi M.S. (1990). Nuclear Systems 1 Thermal Hydraulic Fundamentals. Hemisphere Publishing Corporation. ISBN 0-89116-935-0.

Tsoukalas L., Ikonomopoulos A., Uhrig R.E. (1994). Fuzzy Neural Control. Chapter 15 in Artificial Neural Networks for Intelligent Manufacturing. ed. C. Dagli. Chapman and Hall. ISBN 0-412-48050-6

Thompson A.C., Burch J.M. (1993). Determination of Leak Size and Site in a PWR System Using a Neural Network. International Conference on New Trends in Nuclear System Thermohydraulics. Pisa, Italy. Proceedings, Volume 1, Pages 583-591.

Uhrig R.E., Miller L.F. (1989). Use of Neural Networks for In-Core Fuel Management. Transactions of American Nuclear Society, Volume 59, June 1989, Page 59.

Uhrig R.E. (1991). Potential Applications of Neural Networks to the Operation of Nuclear Power Plants. Nuclear Safety, Volume 32, No.1, Pages 68-79.

Upadhyaya B.R., Eryunek E. (1992). Application of Neural Networks for Sensor Validation and Plant Monitoring. Nuclear Technology, Volume 97, Pages 170-176.

Weller P.R. (1993). Genetic Synthesis of Neural Network Architecture on a Local Area Network. MSc Thesis. Department of System Science, City University, London.

Weller et al. (1995). Using Hierarchical Neural Networks for Diagnosing and Predicting the Condition of a Nuclear Reactor. Proceedings of Session 12 (Technical Diagnosis and Nondestructive Testing). ICANN'95 (International Conference on Artificial Neural Networks), Paris, France. 9-13 October 1995.

Weller et al. (1997a). Using Artificial Neural Networks to Model Non-linearity in a Complex System. Proceedings of ICANNGA'97 (International Conference on Artificial Neural Networks and Genetic Algorithms), Norwich, UK. 2-4 April 1997.

Weller et al. (1997b). Using an On-line Artificial Neural Network to Diagnose Real Time Transient Conditions in a Critical Complex System. Proceedings of IEA/AEI-97 (International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. Atlanta, Georgia, USA.10-13 June 1997. Eds. Potter, Matthews, Ali. Gordon and Breach Science Pub. Pages 333-340.

Yearworth M., Sharpe P. (1992). Essential Methodologies: Neural Networks in Laboratory Medicine. A State-of-the-Art. An OpenLabs CEC-2 Public Report, July 1992, Pages 112-123.

Yukiharu Ohga et al. (1993). Abnormal Event Identification in Nuclear Power Plants using a Neural Network and Knowledge Processing. Nuclear Technology, Vol 101, Pages 159-167.

Appendix

A Pressurised Water Reactor

Introduction to Pressurised Water Reactors (PWRs)

This section gives a basic description of a PWR. The account is, of necessity, rather brief more detailed explanations are referred to in the bibliography.

The Pressurised Water Reactor (PWRs) is one of the world's most popular design for nuclear reactors. A simplified diagram, Fig A.1, showing the main components is given below.

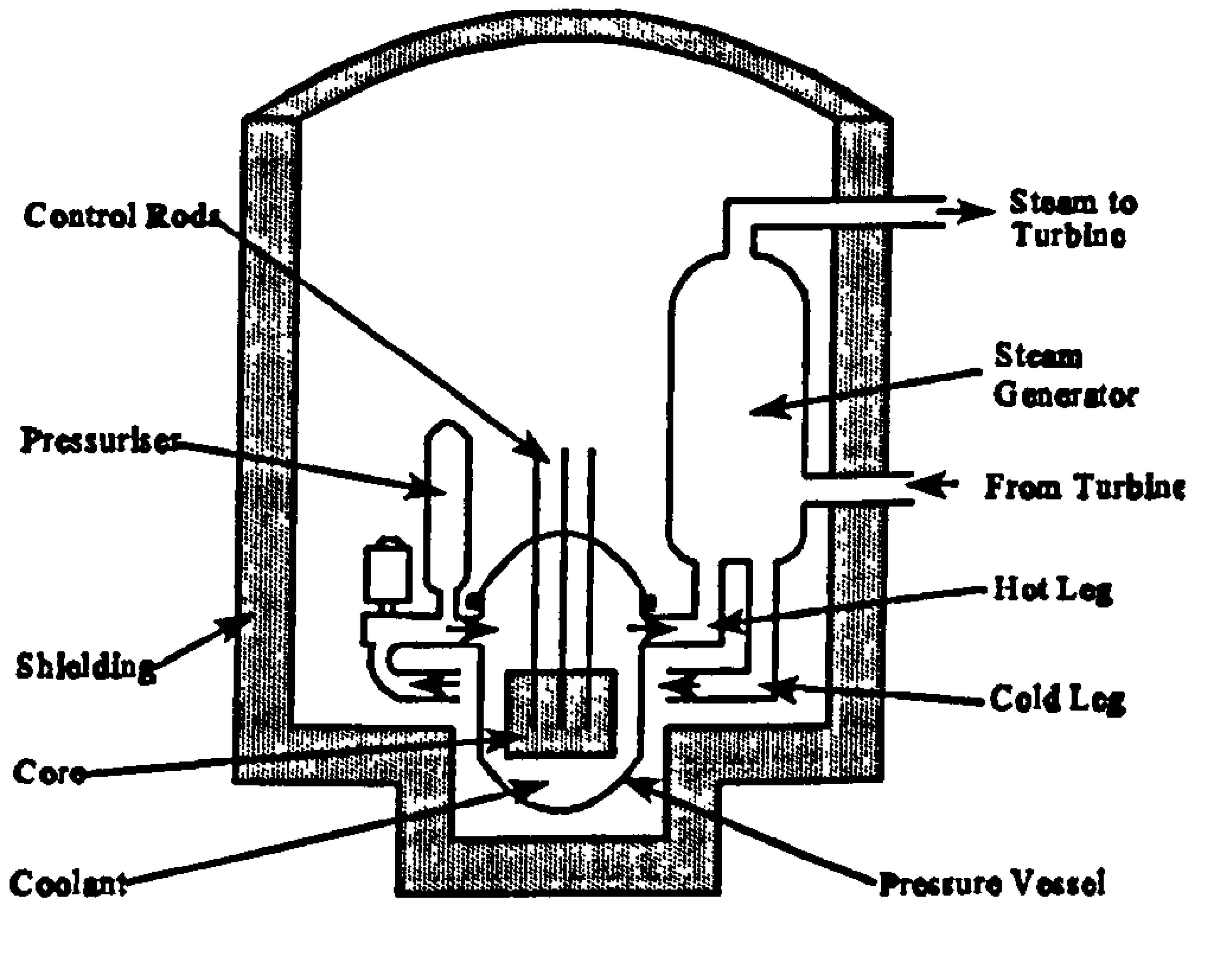


Fig A.1: Diagram of a PWR, (From Patterson)

The whole assembly is enclosed in heavy shielding. The main structure is a large pressure vessel made of welded steel. It has a lid which is secured to the vessel by a ring of heavy bolts. The pressure vessel contains the reactor core and the control rods. The core is composed of a lattice of fuel elements containing enriched Uranium 235. The remaining volume in the pressure vessel is filled with water under a pressure of 150 atmospheres. The water is used as coolant and a moderator. Heat is generated by the fission of the Uranium and is absorbed by the circulating water.

Water leaves the pressure vessel through pairs of heavy pipes welded to the vessel. One pipe carries heated water away from the core and is referred to as the `hot leg'. The other pipe carries water back to the core and is called the `cold leg'. Each reactor has two or more loops of these pairs of pipes. In some PWRs valves are included for isoslating particular areas of the primary circuit. The hot leg takes the heated pressurised water to a steam generator. The generator is a heat exchanger composed of many small tubes surrounded by water. These tubes are filled with the heated water from the core. The pressurised primary circuit water inside the tubes cannot boil, but the water in the shell surrounding the generator tubes does. The steam produced in this secondary circuit is then processed and used in a turbine to produce electricity. The cooler pressurised water is then pumped back to the pressure vessel through the cold leg. Any downstream steam leaks in the steam generator circuit have an important bearing on the primary circuit.

One of the pressurised water loops contains a pressuriser which evaporates a quantity of coolant to maintain the water pressure in the primary vessel. This component is also used to compensate for the unexpected pressure changes resulting from a transient or fault condition.

Control rods are mounted in the top of the pressure vessel. In normal operation groups of control rods are raised or lowered into the core to control the power level of the PWR. In an emergency these can be driven down into the core and control the fission process. This condition is known as a `scram'. It is possible for a single rod to become detached and drop into the core. This condition is referred to as a rod drop. Similarly, a sub-group of rods may become detached and drop into the core, a condition known as a group drop. This condition causes an inadvertant partial scram of the reactor.

Other very important components of a PWR are the safety systems. The pressurised nature of the system means that if a leak occurs in one of the primary pipes, a primary coolant leak, a large amount of the cooling water may be lost very quickly. Continued cooling must be maintained to prevent damage to the core. This type of fault is referred to as a "Loss Of Coolant Accident" (LOCA), and will be referred to throughout this report.

Appendix

B Introduction to Artificial Neural Networks and Genetic Algorithms

B.1 Introduction

This appendix introduces the two modelling techniques used in this research, artificial neural networks (ANNs) and genetic algorithms (GAs). The topics are only briefly described and further information can be obtained from the specialist books recommended in the last section of the appendix.

B.2 Artificial Neural Networks

An ANN is a form of pattern classifier that functions on similar principles to the human brain, although far simpler and smaller. They have been researched and investigated from the early days of the computer age. The principal component of an ANN is a node, the computer equivalent of the neuron in the biological brain. A number of these nodes are used to form a network and are usually arranged into layers. A number of ANN methodologies have been developed with different forms of nodal links and pattern processing. Each of these operate in separate ways and model various brain features. All the ANNs developed for this research project are fully connected feed-forward ANNs.

In a fully connected ANN each member of a layer is connected to all the members of both the preceding and succeeding layers with the computer equivalent of synapse and axonal paths of the brain. A simple ANN is shown in the following diagram, Fig B.1. The first layer is for the presentation of the pattern under consideration while the last layer contains the result of the ANN's action on the pattern. A pattern is presented to the input layer, processed by the hidden layers and the result produced at the output layer. The information flows forward from the input to the output layer. The figure represents a very small ANN, in practice they consist of more nodes in all layers. However, there is seldom a requirement for more than two hidden layers.

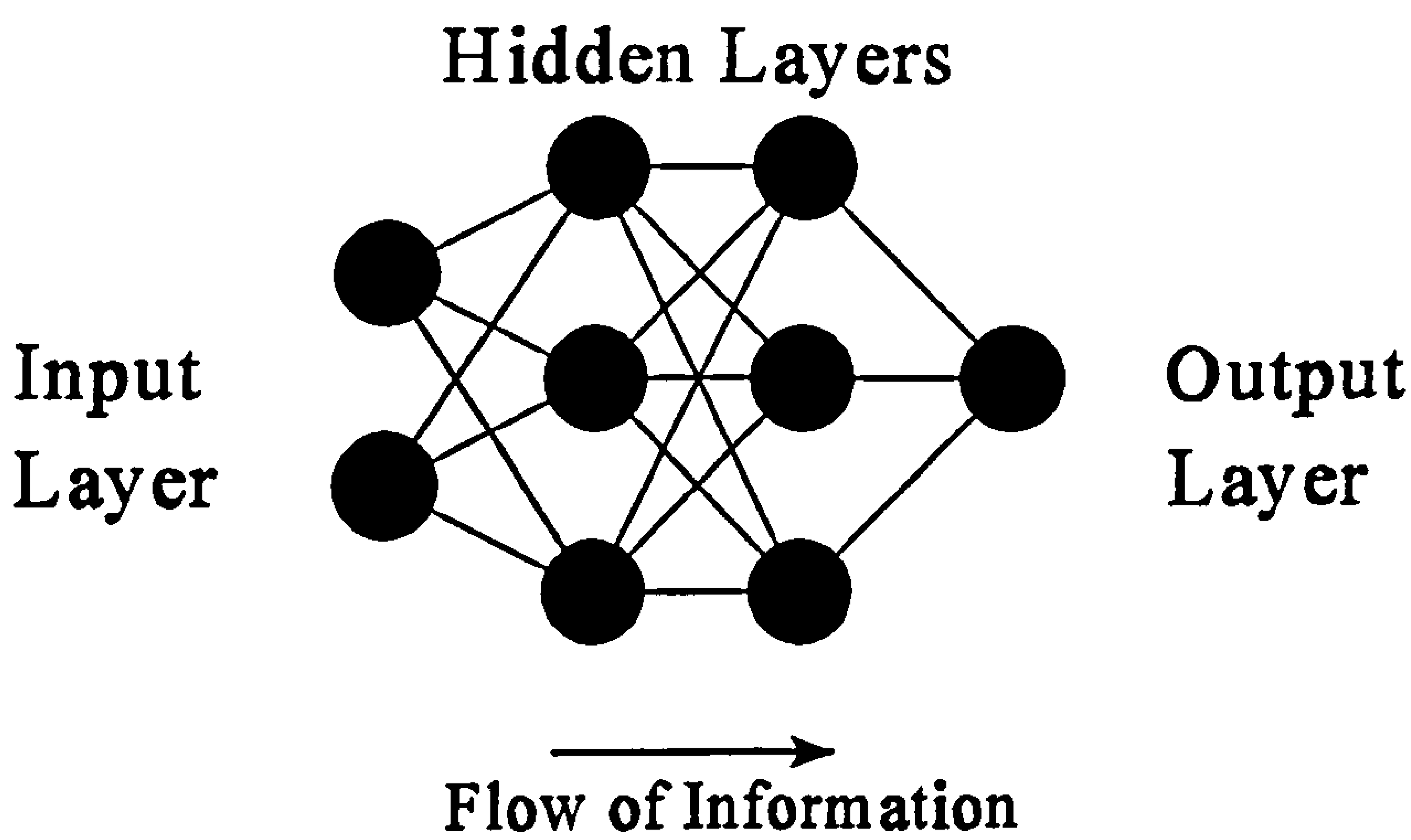


Fig B.1 Diagram of a simple ANN

The links between nodes have a weighting which is a factor by which the output from one node is multiplied before it enters the next node. The node performs as if it were a summation function, combining all of the individual input signals into a single value. With the exception of the input layer each node also has a transfer function which is applied upon the sum of the inputs. This process models the sodium/potassium chemical transfer process of excitation in the biological brain. The summation and transfer processes are shown in Figure B.2. The processed input signal is then sent as an input to the nodes in the next layer.

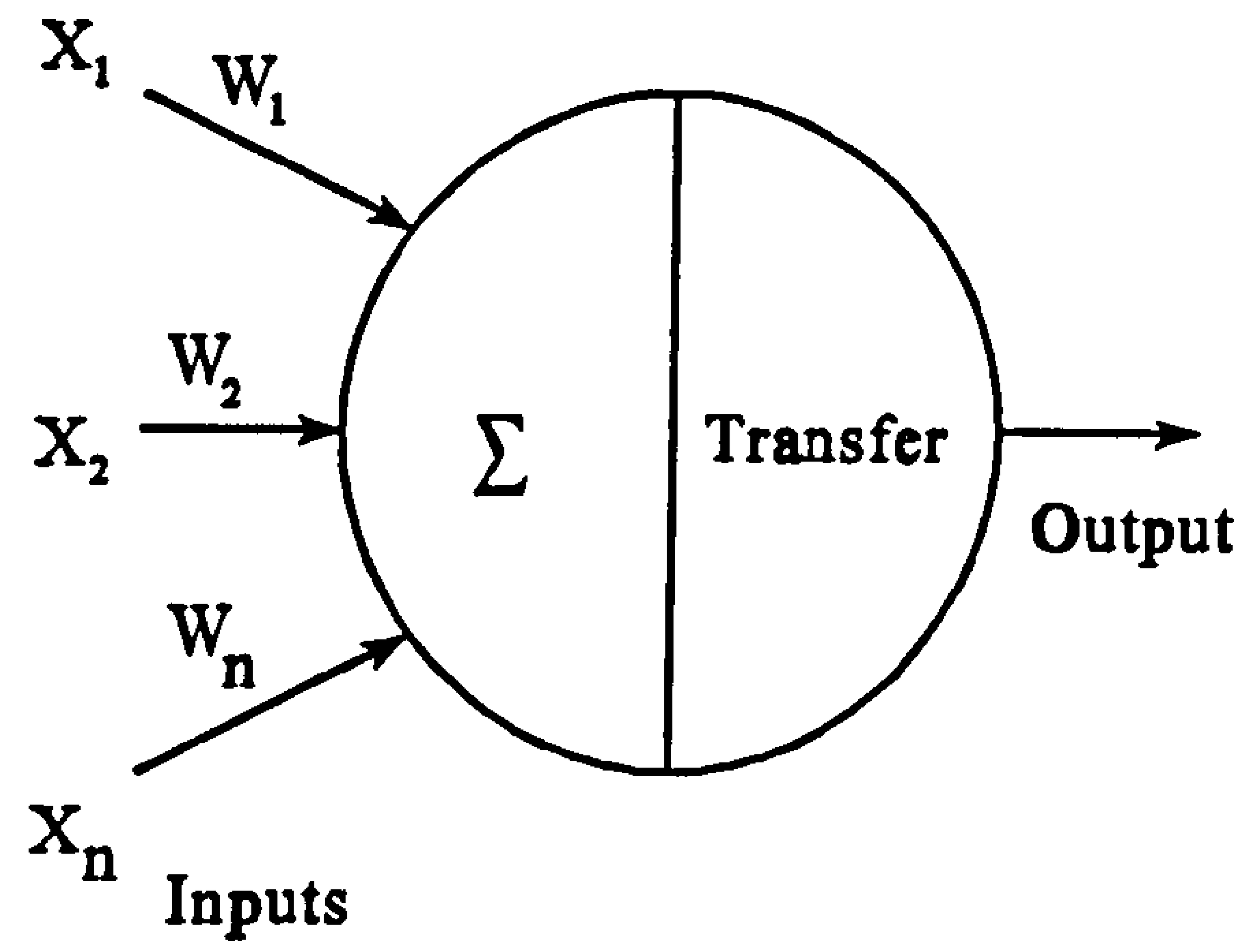


Fig B.2: Model of a Neuron

$$Output = f_{Transfer}\left(\sum_{k=1}^{n} X_k W_k\right)$$

Several transfer functions are used in this project, the main ones are:

Sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Hyperbolic Tangent function:

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

An important feature of neural networks is their facility to be taught to discriminate between given patterns. The ability to learn a complex set of classifier conditions or test patterns was one reason for the recent renewed interest in the subject. Through learning, the neural network creates implicit relationships between variables. Several ANN training algorithms exist but the procedure used for the majority of this research project is referred to as error back-propagation using the generalised delta rule.

To implement this algorithm an ANN structure needs to be created. The number of input and output nodes are usually problem defined and fixed during the training process. The number of hidden nodes and layers can be user selected. Various heuristic methods have been proposed but for the majority of this project the following equations from Masters (Page 177) have been used.

For One Hidden Layer: $N = (mn)^{1/2}$

For Two Hidden Layers: $N1 = mr^2$

$N2 = mr$

Where: m = number of output neurons, n = number of input neurons, $r = (n/m)^{1/3}$

The final item of training information consists of a set of inputs and the corresponding known output. This is a set of known inputs and their respective outputs. In practice this set is divided into two; the larger set is used to train the ANN while a second, smaller, independent set is used to test the training. The initial values for the internodal weights are randomly assigned. These values are then gradually changed to minimise the error between the actual ANN output and the desired output from the training set. This iterative process is detailed in the pseudocode below.

```
Procedure ANN
begin
   randomly assign weightings and threshold values
   while termination state not achieved do;
   begin
      for size of training set do;
      begin
         present a member of the training set to the network
         calculate the output from the ANN to this input
         calculate the error between ANN output and actual
         adjust weights and thresholds to minimise the error
      end
   end
end.
```

The method of adjusting the internodal weights is performed using a gradient descent rule. This algorithm is repeated for a number of cycles, usually thousands, or until the error is below a pre-defined value. The trained network can then be used to examine new conditions, including those with noisy or incomplete data.

For One Hidden Layer: $N = (mn)^{1/2}$

For Two Hidden Layers: $N1 = mr^2$

$N2 = mr$

Where: m = number of output neurons, n = number of input neurons, $r = (n/m)^{1/3}$

The final item of training information consists of a set of inputs and the corresponding known output. This is a set of known inputs and their respective outputs. In practice this set is divided into two; the larger set is used to train the ANN while a second, smaller, independent set is used to test the training. The initial values for the internodal weights are randomly assigned. These values are then gradually changed to minimise the error between the actual ANN output and the desired output from the training set. This iterative process is detailed in the pseudocode below.

```
Procedure ANN
begin
   randomly assign weightings and threshold values
   while termination state not achieved do;
   begin
      for size of training set do;
      begin
         present a member of the training set to the network
         calculate the output from the ANN to this input
         calculate the error between ANN output and actual
         adjust weights and thresholds to minimise the error
      end
   end
end.
```

The method of adjusting the internodal weights is performed using a gradient descent rule. This algorithm is repeated for a number of cycles, usually thousands, or until the error is below a pre-defined value. The trained network can then be used to examine new conditions, including those with noisy or incomplete data.

B.3 Genetic Algorithms

In evolution a parent's characteristics are carried forward to their offsprings by coded information contained in the gene. Each parent contributes a set of features; for example, brown eyes or blond hair. Some combinations of these attributes can enable members of one generation to be better equipped for their environment than were the previous generation. By natural selection and survival of the fittest these features can be enhanced and the species becomes more dominant.

A Genetic Algorithm is a recent (Holland, 1975) and robust alternative method of problem solving which uses the techniques from natural selection to search for and evolve the best solution. It requires little specific knowledge of the problem domain. This is achieved by considering the task as a "Black Box" and using the inputs and outputs to evolve the solution as shown in the following figure, Fig B.3. The inputs to the problem are formulated as a form of genetic code which is initially selected at random. These chromosomes are then tested on the given problem and an output determined for each code. This output gives the acceptability, or fitness, of the code.

Pairs of chromosomes are then randomly chosen to produce the next generation of possible solutions. The probability of a chromosomes selection is in proportion to its fitness. The fitter, more acceptable, codes therefore have a greater probability to contribute to the next generation but less fit members could also be selected although with less frequency. The new population is produced by exchanging part of the information of the selected codes, a process known as crossover. The simplest crossover technique is called one-point crossover and occurs when the parts of two parents chromosomes are swapped at a randomly selected point to make two children. The first child having the code of the first parent up to the cross over point and the code of the second afterwards. The second child has the reverse, the second parent's code to the crossover and then the first parent's code. For example, using a crossover point of 3, the two codes below are used to produce two members of the next generation.

Parent1 a1 a2 a3 | a4 a5 Child1 a1 a2 a3 | b4 b5
Parent2 b1 b2 b3 | b4 b5 Child2 b1 b2 b3 | a4 a5

These new codes produced are then themselves evaluated and then used to generate a new population of codes. This process is repeated for a number of generations.

The other important ingredient of genetic algorithms, again a technique observed in biological genetics, is mutation. Each element of a code, an allele, could be modified again using a random chance factor. This could also result in an improvement in the solution and avoiding the solutions becoming trapped in a local minima of the solution space.

The underlying principle is that by combining the best features of one generation, better, fitter solutions should exist in the succeeding one. The important feature of genetic algorithms is that through initially using random numbers a sound, sensible and sometimes unexpected solution can be arrived at.

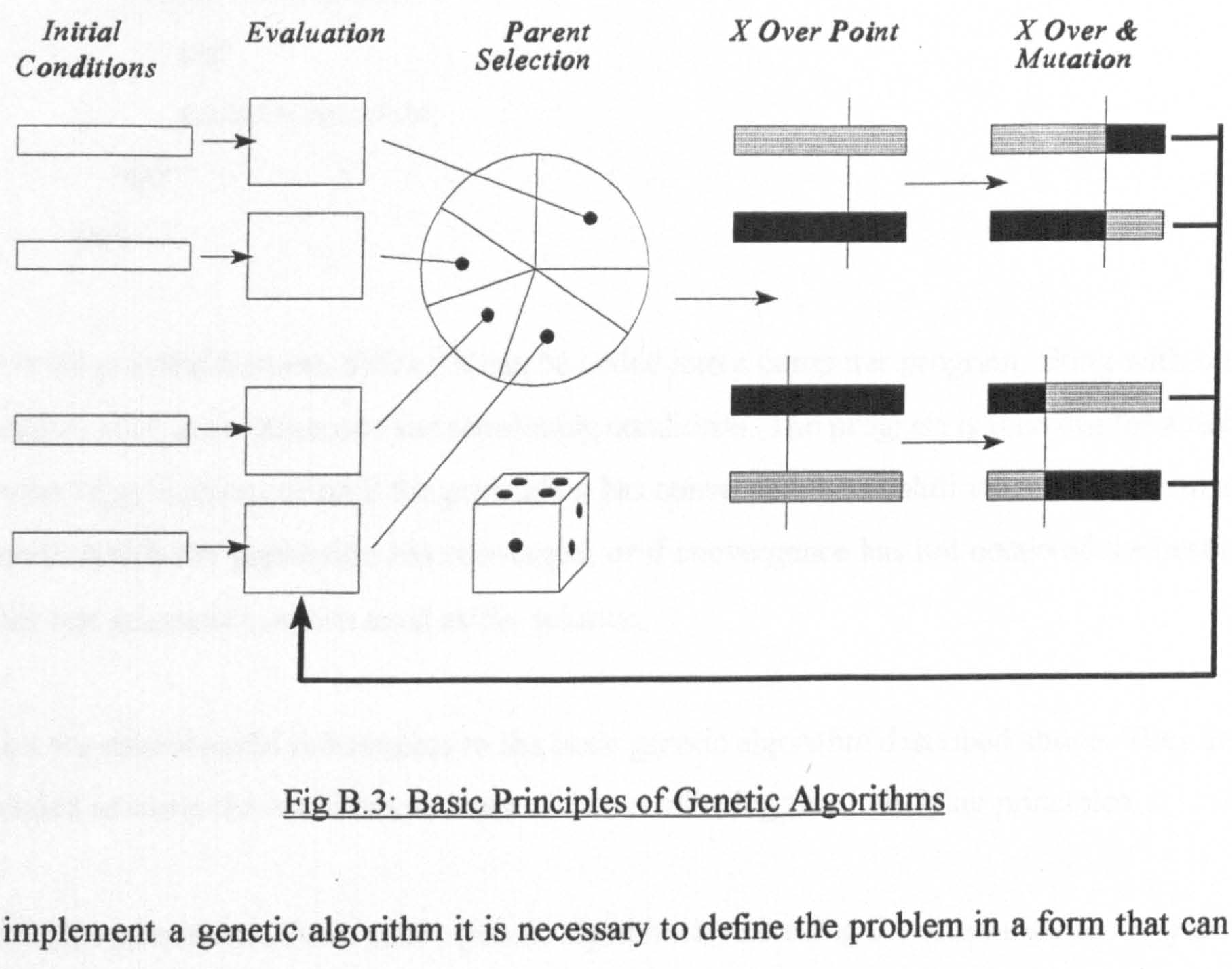


Fig B.3: Basic Principles of Genetic Algorithms

To implement a genetic algorithm it is necessary to define the problem in a form that can be understood by the technique. The first requirement is to code the possible solutions as a genetic string. Usually this is a binary representation but other codings are possible. A measure of suitability of the possible solutions, the fitness, also needs to be defined. This fitness is a mapping of each genetic string to a discrete value, with respect to the other members of the population, the best solutions have a better fitness. The steps are highlighted in the pseudocode below:

```
Procedure GA
begin
    randomly select initial population members;
    while termination state not achieved do;
    begin
        evaluate fitness of population members;
        for size of population do;
        begin
            randomly select two parents based on fitness;
            crossover their strings at randomly selected point(s) to produce new
    population members;
        end
        consider mutation;
    end
end.
```

Once the problem has been defined it can be coded into a computer program, along with several variables, such as mutation rate and terminating conditions. The program is then run for a defined number of generations or until the population has converged. The solution to the problem is the code to which the population has converged, or if convergence has not occurred the best code in the last generation is then used as the solution.

There are several useful refinements to the basic genetic algorithm described above. They are all intended to assist the evolution process while not affecting the underlying principles.

One of the potential problems with a genetic algorithm is that using the cross-over technique does not guarantee that the new generation will contain better members than the previous one. It is possible that the random selection of cross-over position joins elements of fit chromosomes in a retrograde combination. To prevent a new generation consisting of a population of less fit members the fittest chromosome of the previous generation is automatically carried forward to the new generation. This technique is known as an Elitist policy.

The one point crossover technique, described above, has a serious weakness. It is not good for preserving the fitness of chromosomes whose key alleles are situated at, for example, at either end of the string. Unless key arrangements of alleles, called schema, can be propagated the genetic algorithm has little chance of evolving to the best solution. A method of resolving this problem is to use two point crossover. In this method two points are randomly selected and the alleles exchanged between these two points. For example the following two codes, with crossover points of two and three, produce the two children shown with two point crossover.

Parent1 a1 | a2 a3 | a4 a5 Child1 a1 | b2 b3 | a4 a5

Parent2 b1 | b2 b3 | b4 b5 Child2 b1 | a2 a3 | b4 b5

If either of the points selected equals an end point on the chromosome then the crossover procedure reduces to the original one point crossover.

If the algorithm is close to convergence all the population members will have a similar fitness. The task of selecting the fittest members as parents is complicated as all the codes have a similar probability of being selected. A method of scaling is sometimes used to redefine the fitness values and recalculate the probabilities for selection based on these new values. There are two most popular scaling methods. The first is a simple fitness ranking order with the worst member of a population of n having a ranking 1 and the fittest a ranking of n. The second method defines a new base value, for example the worst member of the population, and all other chromosome fitnesses are expressed by their value over this base level.

In many applications a large population size is used and it is possible that many codes will be produced more than once. This arrangement is generally undesirable as the information is repeated and computational time is wasted calculating the same fitness value. A refinement to the crossover technique, called a 'no duplicates policy', is to compare a new code with its peers and produce another code if that chromosome already exists. This policy requires careful monitoring as possible new codes will become very scarce as the solutions converge to the final answer.

Some applications of genetic algorithms are for problems that do not easily lend themselves to a binary coding. An example is the class of problems are concerned with determining a minimal path between a set number of locations, traditionally referred to as the travelling salesman problem. The solutions to a genetic algorithm of this class of problem are coded as a possible route between the locations. A binary coding of such a problem would be rather unwieldy so a coding relating to the number of locations would be used. As an example, the code 3 1 4 2 would be translated as a journey starting at point 3 and travelling to point 2 via points 1 and 4 respectively. The fitness function for this type of problem would be the total distance travelled by following this route. Care needs to be taken with the crossover element of the algorithm to prevent the same location appearing more than once in a chromosome.

B.4 Further Reading

This appendix has only briefly introduced ANNs and GAs. For futher details the following books may help.

Eberhart R.C., Dobbins R.W. eds. Neural Network PC Tools, A Practical Guide
Academic Press, 1990, ISBN 0-12-228640-5

Goldberg D.E., Genetic Algorithms in Search, Optimization & Machine Learning
Addison Wesley, 1989, ISBN 0-201-15767-5

Haykin S. Neural Networks, A Comprehensive Foundation
Macmillan College Publishing, 1994, ISBN 0-02-352761-7

Mitchel M., An Introduction to Genetic Algorithms
MIT Press, 1996, ISBN 0-262-13316-4

Appendix

C

Appendix C

This appendix contains the full results of all the ANNs trained to predict Transient 1 in Section 5.2. The ANNs were all designed to predict the values of PWR variables one step ahead. Four sets of recent PWR variables were used as inputs the ANNs. These were one to four time steps. For each input set a range of one and two hidden layer ANNs were developed. The number of nodes in the hidden was varied from a maximum of 18 to a minimum of 2. The values in between were in steps of two nodes. In the case of two hidden layers both layers were varied between these limits. All the ANN training was repeated four times to avoid possible local minima. The value recorded in each case is the lowest RMS error for the training. As explained in Chapter 5 all ANNs were trained for 120,000 cycles of presentation of the training set and then a further 40,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing. In the following tables the best RMS error result for each ANN architecture is underlined while the best overall ANN for each hidden layer configuration is in bold.

One Time Step Input for One Step Prediction

Filename	Size of ANN	RMS Error			
1_1x1aa	18	0.0344	0.0345	0.0354	0.0347
1_1x1ab	16	0.0399	0.0361	0.0351	0.0342
1_1x1ac	14	0.0336	0.0349	0.0348	0.0341
1_1x1ad	12	0.0335	0.0336	0.0348	0.0340
1_1x1ae	10	0.0340	0.0342	0.0345	0.0353
1_1x1af	8	0.0346	0.0341	0.0359	0.0334
1_1x1ag	6	0.0336	0.0326	0.0343	0.0325
1_1x1ah	4	0.0326	0.0337	0.0334	0.0333
1_1x1ai	2	0.0381	0.0389	0.0381	0.0380
1_1x2aa	18, 18	0.0395	0.0399	0.0398	0.0406
1_1x2ab	18, 16	0.0397	0.0419	0.0390	0.0393
1_1x2ac	18, 14	0.0393	0.0399	0.0397	0.0392
1_1x2ad	18, 12	0.0392	0.0392	0.0391	0.0396
1_1x2ae	18, 10	0.0390	0.0390	0.0390	0.0390
1_1x2af	18, 8	0.0390	0.0389	0.0389	0.0389
1_1x2ag	18, 6	0.0386	0.0387	0.0386	0.0387
1_1x2ah	18, 4	0.0384	0.0387	0.0385	0.0385
1_1x2ai	18, 2	0.0386	0.0383	0.0384	0.0386
1_1x2aj	16, 18	0.0393	0.0392	0.0395	0.0393
1_1x2ak	16, 16	0.0392	0.0394	0.0396	0.0391
1_1x2al	16, 14	0.0394	0.0391	0.0390	0.0393
1_1x2am	16, 12	0.0396	0.0397	0.0390	0.0388
1_1x2an	16, 10	0.0390	0.0393	0.0389	0.0390
1_1x2ao	16, 8	0.0388	0.0388	0.0390	0.0386
1_1x2ap	16, 6	0.0391	0.0386	0.0385	0.0388
1_1x2aq	16, 4	0.0384	0.0383	0.0385	0.0381
1_1x2ar	16, 2	0.0386	0.0385	0.0384	0.0387

Filename	Size of ANN	RMS Error			
1_1x2as	14, 18	0.0395	0.0391	0.0392	0.0394
1_1x2at	14, 16	0.0394	0.0392	0.0389	0.0389
1_1x2au	14, 14	0.0388	0.0392	0.0390	0.0387
1_1x2av	14, 12	0.0388	0.0388	0.0389	0.0385
1_1x2aw	14, 10	0.0388	0.0387	0.0389	0.0390
1_1x2ax	14, 8	0.0388	0.0390	0.0389	0.0386
1_1x2ay	14, 6	0.0386	0.0387	0.0387	0.0385
1_1x2az	14, 4	0.0384	0.0383	0.0385	0.0381
1_1x2ba	14, 2	0.0385	0.0385	0.0384	0.0389
1_1x2bb	12, 18	0.0388	0.0389	0.0393	0.0391
1_1x2bc	12, 16	0.0391	0.0391	0.0391	0.0392
1_1x2bd	12, 14	0.0386	0.0387	0.0392	0.0390
1_1x2be	12, 12	0.0390	0.0387	0.0388	0.0388
1_1x2bf	12, 10	0.0385	0.0386	0.0388	0.0388
1_1x2bg	12, 8	0.0383	0.0389	0.0387	0.0384
1_1x2bh	12, 6	0.0383	0.0381	0.0384	0.0383
1_1x2bi	12, 4	0.0385	0.0382	0.0380	0.0378
1_1x2bj	12, 2	0.0388	0.0385	0.0386	0.0383
1_1x2bk	10, 18	0.0388	0.0387	0.0388	0.0392
1_1x2bl	10, 16	0.0386	0.0389	0.0387	0.0389
1_1x2bm	10, 14	0.0386	0.0386	0.0391	0.0384
1_1x2bn	10, 12	0.0384	0.0392	0.0389	0.0389
1_1x2bo	10, 10	0.0387	0.0385	0.0388	0.0385
1_1x2bp	10, 8	0.0385	0.0386	0.0383	0.0385
1_1x2bq	10, 6	0.0376	0.0378	0.0379	0.0379
1_1x2br	10, 4	0.0384	0.0377	0.0377	0.0387
1_1x2bs	10, 2	0.0383	0.0384	0.0385	0.0388

Filename	Size of ANN	RMS Error			
1_1x2bt	8, 18	0.0389	0.0388	0.0386	0.0387
1_1x2bu	8, 16	0.0386	0.0389	0.0390	0.0389
1_1x2bv	8, 14	0.0387	0.0388	0.0385	0.0385
1_1x2bw	8, 12	0.0382	0.0387	0.0383	0.0384
1_1x2bx	8, 10	0.0386	0.0388	0.0383	0.0385
1_1x2by	8, 8	0.0388	0.0386	0.0383	0.0385
1_1x2bz	8, 6	0.0386	0.0382	0.0385	0.0378
1_1x2ca	8, 4	0.0380	0.0382	0.0385	0.0377
1_1x2cb	8, 2	0.0385	0.0384	0.0381	0.0383
1_1x2cc	6, 18	0.0390	0.0389	0.0385	0.0385
1_1x2cd	6, 16	0.0389	0.0387	0.0384	0.0385
1_1x2ce	6, 14	0.0390	0.0389	0.0389	0.0381
1_1x2cf	6, 12	0.0383	0.0383	0.0381	0.0387
1_1x2cg	6, 10	0.0382	0.0382	0.0384	0.0385
1_1x2ch	6, 8	0.0381	0.0387	0.0384	0.0391
1_1x2ci	6, 6	0.0384	0.0377	0.0380	0.0387
1_1x2cj	6, 4	0.0381	0.0385	0.0380	0.0384
1_1x2ck	6, 2	0.0387	0.0384	0.0384	0.0383
1_1x2cl	4, 18	0.0390	0.0599	0.0389	0.0391
1_1x2cm	4, 16	0.0378	0.0387	0.0393	0.0386
1_1x2cn	4, 14	0.0383	0.0384	0.0393	0.0390
1_1x2co	4, 12	0.0391	0.0388	0.0382	0.0380
1_1x2cp	4, 10	0.0380	0.0386	0.0377	0.0385
1_1x2cq	4, 8	0.0383	0.0387	0.0384	0.0383
1_1x2cr	4, 6	0.0380	0.0378	0.0380	0.0384
1_1x2cs	4, 4	0.0383	0.0372	0.0376	0.0374
1_1x2ct	4, 2	0.0387	0.0386	0.0386	0.0384

Filename	Size of ANN	RMS Error			
1_1x2cv	2, 18	0.0583	0.0568	0.0556	0.0569
1_1x2cw	2, 16	0.0438	0.0545	0.0767	0.0581
1_1x2cx	2, 14	0.0614	0.0574	0.0576	0.0577
1_1x2cy	2, 12	0.0573	0.0551	0.0587	0.0422
1_1x2cz	2, 10	0.0473	0.0405	0.0581	0.0567
1_1x2da	2, 8	0.0558	0.0465	0.0413	0.0434
1_1x2db	2, 6	0.0388	0.0394	0.0430	0.0564
1_1x2dc	2, 4	0.0384	0.0416	0.0386	0.0461
1_1x2dd	2, 2	0.0433	0.0387	0.0781	0.0716

Two Time Steps Input for One Step Prediction

1_2x1aa	18	0.0359	0.0336	0.0348	0.0356
1_2x1ab	16	0.0359	0.0361	0.0378	0.0363
1_2x1ac	14	0.0364	0.0363	0.0346	0.0355
1_2x1ad	12	0.0345	0.0346	0.0363	0.0352
1_2x1ae	10	0.0363	0.0358	0.0349	0.0364
1_2x1af	8	0.0360	0.0347	0.0328	0.0360
1_2x1ag	6	0.0354	0.0373	0.0340	0.0354
1_2x1ah	4	0.0381	0.0356	0.0360	0.0351
1_2x1ai	2	0.0392	0.0391	0.0391	0.0395
1_2x2aa	18, 18	0.0436	0.0415	0.0426	0.0422
1_2x2ab	18, 16	0.0424	0.0428	0.0419	0.0408
1_2x2ac	18, 14	0.0446	0.0416	0.0416	0.0422
1_2x2ad	18, 12	0.0406	0.0418	0.0405	0.0410
1_2x2ae	18, 10	0.0410	0.0419	0.0405	0.0412
1_2x2af	18, 8	0.0400	0.0404	0.0410	0.0399
1_2x2ag	18, 6	0.0400	0.0403	0.0402	0.0406

Filename	Size of ANN	RMS Error			
1_2x2ah	18, 4	0.0403	0.0401	0.0399	0.0396
1_2x2ai	18, 2	0.0411	0.0450	0.0407	0.0394
1_2x2aj	16, 18	0.0412	0.0410	0.0414	0.0417
1_2x2ak	16, 16	0.0417	0.0409	0.0417	0.0419
1_2x2al	16, 14	0.0417	0.0422	0.0414	0.0429
1_2x2am	16, 12	0.0410	0.0417	0.0414	0.0410
1_2x2an	16, 10	0.0414	0.0408	0.0406	0.0412
1_2x2ao	16, 8	0.0409	0.0412	0.0401	0.0405
1_2x2ap	16, 6	0.0407	0.0400	0.0408	0.0403
1_2x2aq	16, 4	0.0411	0.0400	0.0395	0.0397
1_2x2ar	16, 2	0.0411	0.0406	0.0411	0.0405
1_2x2as	14, 18	0.0407	0.0424	0.0414	0.0416
1_2x2at	14, 16	0.0413	0.0414	0.0405	0.0417
1_2x2au	14, 14	0.0413	0.0412	0.0408	0.0410
1_2x2av	14, 12	0.0409	0.0404	0.0402	0.0416
1_2x2aw	14, 10	0.0409	0.0406	0.0404	0.0409
1_2x2ax	14, 8	0.0403	0.0403	0.0402	0.0410
1_2x2ay	14, 6	0.0401	0.0393	0.0395	0.0406
1_2x2az	14, 4	0.0402	0.0393	0.0401	0.0392
1_2x2ba	14, 2	0.0407	0.0410	0.0411	0.0405
1_2x2bb	12, 18	0.0408	0.0416	0.0408	0.0417
1_2x2bc	12, 16	0.0406	0.0417	0.0411	0.0412
1_2x2bd	12, 14	0.0407	0.0405	0.0408	0.0407
1_2x2be	12, 12	0.0409	0.0402	0.0406	0.0400
1_2x2bf	12, 10	0.0400	0.0399	0.0399	0.0403
1_2x2bg	12, 8	0.0395	0.0406	0.0401	0.0398
1_2x2bh	12, 6	0.0404	0.0394	0.0405	0.0393
1_2x2bi	12, 4	0.0402	0.0397	0.0391	0.0402
1_2x2bj	12, 2	0.0404	0.0389	0.0407	0.0404

Filename	Size of ANN	RMS Error			
1_2x2bk	10, 18	0.0423	0.0405	0.0409	0.0411
1_2x2bl	10, 16	0.0407	0.0401	0.0409	0.0400
1_2x2bm	10, 14	0.0413	0.0399	0.0405	0.0402
1_2x2bn	10, 12	0.0404	0.0405	0.0402	0.0407
1_2x2bo	10, 10	0.0404	0.0402	0.0409	0.0407
1_2x2bp	10, 8	0.0391	0.0395	0.0386	0.0398
1_2x2bq	10, 6	0.0397	0.0410	0.0395	0.0406
1_2x2br	10, 4	0.0393	0.0397	0.0393	0.0390
1_2x2bs	10, 2	0.0411	0.0392	0.0404	0.0401
1_2x2bt	8, 18	0.0406	0.0411	0.0404	0.0406
1_2x2bu	8, 16	0.0406	0.0398	0.0407	0.0409
1_2x2bv	8, 14	0.0402	0.0403	0.0399	0.0401
1_2x2bw	8, 12	0.0402	0.0403	0.0400	0.0399
1_2x2bx	8, 10	0.0396	0.0401	0.0391	0.0393
1_2x2by	8, 8	0.0392	0.0394	0.0396	0.0390
1_2x2bz	8, 6	0.0398	0.0386	0.0398	0.0399
1_2x2ca	8, 4	0.0392	0.0400	0.0403	0.0391
1_2x2cb	8, 2	0.0392	0.0411	0.0389	0.0391
1_2x2cc	6, 18	0.0404	0.0401	0.0397	0.0407
1_2x2cd	6, 16	0.0399	0.0402	0.0398	0.0397
1_2x2ce	6, 14	0.0396	0.0403	0.0399	0.0400
1_2x2cf	6, 12	0.0399	0.0393	0.0395	0.0394
1_2x2cg	6, 10	0.0395	0.0390	0.0392	0.0389
1_2x2ch	6, 8	0.0393	0.0391	0.0397	0.0383
1_2x2ci	6, 6	0.0385	0.0392	0.0390	0.0396
1_2x2cj	6, 4	0.0393	0.0399	0.0381	0.0386
1_2x2ck	6, 2	0.0404	0.0399	0.0402	0.0395
1_2x2cl	4, 18	0.0395	0.0395	0.0387	0.0404
1_2x2cm	4, 16	0.0396	0.0404	0.0405	0.0395

Filename	Size of ANN	RMS Error			
1_2x2cn	4, 14	0.0393	0.0400	0.0402	0.0386
1_2x2co	4, 12	0.0406	0.0396	0.0408	0.0393
1_2x2cp	4, 10	0.0391	0.0383	0.0386	0.0395
1_2x2cq	4, 8	0.0392	0.0394	0.0390	0.0392
1_2x2cr	4, 6	0.0399	0.0391	0.0393	0.0398
1_2x2cs	4, 4	0.0384	0.0391	0.0406	0.0388
1_2x2ct	4, 2	0.0403	0.0406	0.0402	0.0402
1_2x2cu	2, 18	0.0736	0.0563	0.0825	0.0777
1_2x2cv	2, 16	0.0753	0.0761	0.0795	0.0664
1_2x2cw	2, 14	0.0830	0.0734	0.0822	0.0739
1_2x2cx	2, 12	0.0683	0.0827	0.0802	0.0688
1_2x2cy	2, 10	0.0741	0.0789	0.0678	0.0505
1_2x2cz	2, 8	0.0792	0.0611	0.0588	0.0726
1_2x2da	2, 6	0.0714	0.0408	0.0813	0.0649
1_2x2db	2, 4	0.0715	0.0395	0.0785	0.0415
1_2x2dc	2, 2	0.0401	0.0421	0.0421	0.0401

Three Time Steps Input for One Step Prediction

1_3x1aa	18	0.0303	0.0298	0.0297	0.0297
1_3x1ab	16	0.0295	0.0291	0.0295	0.0290
1_3x1ac	14	0.0293	0.0296	0.0292	0.0287
1_3x1ad	12	0.0280	0.0286	0.0279	0.0285
1_3x1ae	10	0.0278	0.0283	0.0283	0.0281
1_3x1af	8	0.0276	0.0274	0.0277	0.0279
1_3x1ag	6	0.0278	0.0272	0.0276	0.0278
1_3x1ah	4	0.0289	0.0263	0.0261	0.0264
1_3x1ai	2	0.0363	0.0366	0.0360	0.0359
1_3x2aa	18, 18	0.0406	0.0411	0.0436	0.0429

Filename	Size of ANN	RMS Error			
1_3x2ab	18, 16	0.0406	0.0401	0.0421	0.0427
1_3x2ac	18, 14	0.0425	0.0386	0.0420	0.0433
1_3x2ad	18, 12	0.0399	0.0408	0.0420	0.0419
1_3x2ae	18, 10	0.0388	0.0390	0.0416	0.0417
1_3x2af	18, 8	0.0381	0.0390	0.0421	0.0419
1_3x2ag	18, 6	0.0387	0.0389	0.0412	0.0407
1_3x2ah	18, 4	0.0395	0.0374	0.0403	0.0412
1_3x2ai	18, 2	0.0386	0.0384	0.0420	0.0414
1_3x2aj	16, 18	0.0403	0.0416	0.0431	0.0434
1_3x2ak	16, 16	0.0394	0.0413	0.0438	0.0426
1_3x2al	16, 14	0.0392	0.0405	0.0415	0.0423
1_3x2am	16, 12	0.0394	0.0396	0.0420	0.0419
1_3x2an	16, 10	0.0386	0.0394	0.0418	0.0409
1_3x2ao	16, 8	0.0385	0.0385	0.0423	0.0419
1_3x2ap	16, 6	0.0375	0.0387	0.0412	0.0409
1_3x2aq	16, 4	0.0366	0.0370	0.0406	0.0409
1_3x2ar	16, 2	0.0381	0.0378	0.0411	0.0412
1_3x2as	14, 18	0.0428	0.0398	0.0424	0.0425
1_3x2at	14, 16	0.0401	0.0426	0.0419	0.0416
1_3x2au	14, 14	0.0385	0.0393	0.0420	0.0423
1_3x2av	14, 12	0.0386	0.0385	0.0418	0.0414
1_3x2aw	14, 10	0.0383	0.0381	0.0413	0.0411
1_3x2ax	14, 8	0.0380	0.0382	0.0411	0.0409
1_3x2ay	14, 6	0.0393	0.0374	0.0399	0.0410
1_3x2az	14, 4	0.0381	0.0378	0.0408	0.0400
1_3x2ba	14, 2	0.0383	0.0385	0.0408	0.0408
1_3x2bb	12, 18	0.0397	0.0392	0.0420	0.0423
1_3x2bc	12, 16	0.0398	0.0415	0.0416	0.0411
1_3x2bd	12, 14	0.0391	0.0393	0.0419	0.0414

Filename	Size of ANN	RMS Error			
1_3x2be	12, 12	0.0388	0.0379	0.0416	0.0415
1_3x2bf	12, 10	0.0386	0.0384	0.0413	0.0413
1_3x2bg	12, 8	0.0376	0.0374	0.0408	0.0411
1_3x2bh	12, 6	0.0377	0.0396	0.0404	0.0408
1_3x2bi	12, 4	0.0375	0.0384	0.0405	0.0410
1_3x2bj	12, 2	0.0379	0.0376	0.0413	0.0409
1_3x2bk	10, 18	0.0391	0.0391	0.0413	0.0410
1_3x2bl	10, 16	0.0394	0.0384	0.0413	0.0434
1_3x2bm	10, 14	0.0385	0.0386	0.0410	0.0411
1_3x2bn	10, 12	0.0378	0.0397	0.0409	0.0407
1_3x2bo	10, 10	0.0389	0.0393	0.0406	0.0406
1_3x2bp	10, 8	0.0372	0.0364	0.0406	0.0416
1_3x2bq	10, 6	0.0373	0.0366	0.0420	0.0406
1_3x2br	10, 4	0.0370	0.0371	0.0393	0.0401
1_3x2bs	10, 2	0.0380	0.0379	0.0395	0.0413
1_3x2bt	8, 18	0.0382	0.0388	0.0410	0.0424
1_3x2bu	8, 16	0.0385	0.0378	0.0412	0.0414
1_3x2bv	8, 14	0.0384	0.0396	0.0409	0.0412
1_3x2bw	8, 12	0.0381	0.0384	0.0404	0.0404
1_3x2bx	8, 10	0.0377	0.0383	0.0400	0.0411
1_3x2by	8, 8	0.0370	0.0372	0.0407	0.0411
1_3x2bz	8, 6	0.0382	0.0363	0.0404	0.0410
1_3x2ca	8, 4	0.0364	0.0373	0.0403	0.0406
1_3x2cb	8, 2	0.0373	0.0376	0.0408	0.0411
1_3x2cc	6, 18	0.0376	0.0386	0.0421	0.0408
1_3x2cd	6, 16	0.0383	0.0378	0.0405	0.0402
1_3x2ce	6, 14	0.0379	0.0394	0.0414	0.0399
1_3x2cf	6, 12	0.0373	0.0370	0.0399	0.0408
1_3x2cg	6, 10	0.0373	0.0365	0.0402	0.0401

Filename	Size of ANN	RMS Error			
1_3x2ch	6, 8	0.0378	0.0372	0.0392	0.0401
1_3x2ci	6, 6	0.0364	0.0365	0.0393	0.0390
1_3x2cj	6, 4	0.0380	0.0359	0.0402	0.0399
1_3x2ck	6, 2	0.0378	0.0374	0.0408	0.0411
1_3x2cl	4, 18	0.0406	0.0385	0.0401	0.0407
1_3x2cm	4, 16	0.0376	0.0370	0.0407	0.0415
1_3x2cn	4, 14	0.0378	0.0379	0.0407	0.0402
1_3x2co	4, 12	0.0377	0.0359	0.0405	0.0401
1_3x2cp	4, 10	0.0368	0.0349	0.0450	0.0402
1_3x2cq	4, 8	0.0372	0.0357	0.0404	0.0400
1_3x2cr	4, 6	0.0370	0.0367	0.0384	0.0382
1_3x2cs	4, 4	0.0341	0.0351	0.0375	0.0394
1_3x2ct	4, 2	0.0383	0.0378	0.0769	0.0399
1_3x2cu	2, 18	0.0776	0.0786	0.0763	0.0782
1_3x2cv	2, 16	0.0554	0.0756	0.0580	0.0870
1_3x2cw	2, 14	0.0846	0.0775	0.0781	0.0863
1_3x2cx	2, 12	0.0519	0.0690	0.0777	0.0745
1_3x2cy	2, 10	0.0842	0.0997	0.0804	0.0748
1_3x2cz	2, 8	0.0830	0.0840	0.0428	0.0841
1_3x2da	2, 6	0.0411	0.0482	0.0546	0.0413
1_3x2db	2, 4	0.0864	0.0740	0.0372	0.0401
1_3x2dc	2, 2	0.0710	0.0391	0.0402	0.0402

Four Time Steps Input for One Step Prediction

1_4x1aa	18	0.0388	0.0388	0.0390	0.0395
1_4x1ab	16	0.0388	0.0377	0.0383	0.0373
1_4x1ac	14	0.0382	0.0390	0.0383	0.0388
1_4x1ad	12	0.0384	0.0387	0.0377	0.0381

Filename	Size of ANN	RMS Error			
1_4x1ae	10	0.0378	0.0387	0.0366	0.0371
1_4x1af	8	0.0368	0.0369	0.0388	0.0374
1_4x1ag	6	0.0383	0.0362	0.0370	0.0364
1_4x1ah	4	0.0351	0.0380	0.0381	0.0356
1_4x1ai	2	0.0391	0.0390	0.0390	0.0397
1_4x2aa	18, 18	0.0483	0.0446	0.0470	0.0453
1_4x2ab	18, 16	0.0454	0.0457	0.0449	0.0457
1_4x2ac	18, 14	0.0452	0.0454	0.0452	0.0451
1_4x2ad	18, 12	0.0447	0.0441	0.0448	0.0445
1_4x2ae	18, 10	0.0441	0.0449	0.0440	0.0438
1_4x2af	18, 8	0.0439	0.0447	0.0445	0.0443
1_4x2ag	18, 6	0.0433	0.0433	0.0437	0.0434
1_4x2ah	18, 4	0.0436	0.0427	0.0434	0.0435
1_4x2ai	18, 2	0.0442	0.0442	0.0442	0.0441
1_4x2aj	16, 18	0.0464	0.0475	0.0446	0.0457
1_4x2ak	16, 16	0.0449	0.0458	0.0458	0.0450
1_4x2al	16, 14	0.0451	0.0443	0.0445	0.0444
1_4x2am	16, 12	0.0470	0.0445	0.0447	0.0442
1_4x2an	16, 10	0.0440	0.0438	0.0433	0.0443
1_4x2ao	16, 8	0.0441	0.0441	0.0432	0.0443
1_4x2ap	16, 6	0.0441	0.0439	0.0436	0.0445
1_4x2aq	16, 4	0.0436	0.0431	0.0428	0.0436
1_4x2ar	16, 2	0.0442	0.0447	0.0437	0.0440
1_4x2as	14, 18	0.0466	0.0443	0.0449	0.0447
1_4x2at	14, 16	0.0457	0.0454	0.0457	0.0462
1_4x2au	14, 14	0.0446	0.0454	0.0447	0.0452
1_4x2av	14, 12	0.0444	0.0437	0.0437	0.0441
1_4x2aw	14, 10	0.0434	0.0436	0.0437	0.0439
1_4x2ax	14, 8	0.0437	0.0430	0.0434	0.0431

Filename	Size of ANN	RMS Error			
1_4x2ay	14, 6	0.0431	0.0432	0.0434	0.0440
1_4x2az	14, 4	0.0438	0.0424	0.0424	0.0434
1_4x2ba	14, 2	0.0440	0.0443	0.0449	0.0439
1_4x2bb	12, 18	0.0450	0.0441	0.0446	0.0442
1_4x2bc	12, 16	0.0441	0.0440	0.0445	0.0459
1_4x2bd	12, 14	0.0448	0.0434	0.0440	0.0444
1_4x2be	12, 12	0.0443	0.0439	0.0438	0.0433
1_4x2bf	12, 10	0.0439	0.0446	0.0439	0.0438
1_4x2bg	12, 8	0.0440	0.0425	0.0430	0.0425
1_4x2bh	12, 6	0.0436	0.0419	0.0430	0.0435
1_4x2bi	12, 4	0.0422	0.0432	0.0422	0.0427
1_4x2bj	12, 2	0.0439	0.0442	0.0536	0.0442
1_4x2bk	10, 18	0.0444	0.0457	0.0439	0.0436
1_4x2bl	10, 16	0.0443	0.0440	0.0439	0.0437
1_4x2bm	10, 14	0.0431	0.0436	0.0435	0.0438
1_4x2bn	10, 12	0.0439	0.0439	0.0441	0.0436
1_4x2bo	10, 10	0.0435	0.0435	0.0435	0.0433
1_4x2bp	10, 8	0.0444	0.0435	0.0437	0.0433
1_4x2bq	10, 6	0.0435	0.0436	0.0418	0.0428
1_4x2br	10, 4	0.0437	0.0431	0.0423	0.0419
1_4x2bs	10, 2	0.0441	0.0435	0.0439	0.0439
1_4x2bt	8, 18	0.0441	0.0439	0.0442	0.0422
1_4x2bu	8, 16	0.0432	0.0436	0.0439	0.0438
1_4x2bv	8, 14	0.0437	0.0430	0.0434	0.0431
1_4x2bw	8, 12	0.0432	0.0435	0.0427	0.0435
1_4x2bx	8, 10	0.0431	0.0433	0.0433	0.0422
1_4x2by	8, 8	0.0434	0.0420	0.0431	0.0434
1_4x2bz	8, 6	0.0431	0.0421	0.0434	0.0426
1_4x2ca	8, 4	0.0417	0.0429	0.0426	0.0439

Filename	Size of ANN	RMS Error			
1_4x2cb	8, 2	0.0435	0.0438	0.0436	0.0440
1_4x2cc	6, 18	0.0435	0.0432	0.0446	0.0436
1_4x2cd	6, 16	0.0424	0.0427	0.0435	0.0448
1_4x2ce	6, 14	0.0437	0.0432	0.0435	0.0430
1_4x2cf	6, 12	0.0435	0.0429	0.0428	0.0434
1_4x2cg	6, 10	0.0430	0.0434	0.0425	0.0421
1_4x2ch	6, 8	0.0425	0.0429	0.0412	0.0414
1_4x2ci	6, 6	0.0425	0.0424	0.0426	0.0434
1_4x2cj	6, 4	0.0436	0.0435	0.0433	0.0424
1_4x2ck	6, 2	0.0436	0.0429	0.0436	0.0430
1_4x2cl	4, 18	0.0433	0.0432	0.0430	0.0498
1_4x2cm	4, 16	0.0430	0.0432	0.0427	0.0425
1_4x2cn	4, 14	0.0429	0.0426	0.0436	0.0439
1_4x2co	4, 12	0.0408	0.0425	0.0428	0.0505
1_4x2cp	4, 10	0.0413	0.0417	0.0421	0.0429
1_4x2cq	4, 8	0.0401	0.0403	0.0416	0.0426
1_4x2cr	4, 6	0.0413	0.0427	0.0423	0.0420
1_4x2cs	4, 4	0.0440	0.0427	0.0419	0.0416
1_4x2ct	4, 2	0.0437	0.0438	0.0424	0.0434
1_4x2cu	2, 18	0.0598	0.0900	0.0817	0.0882
1_4x2cv	2, 16	0.0730	0.0862	0.0853	0.0877
1_4x2cw	2, 14	0.0889	0.0509	0.0524	0.0594
1_4x2cx	2, 12	0.0937	0.0862	0.0873	0.0769
1_4x2cy	2, 10	0.0744	0.0764	0.0761	0.0884
1_4x2cz	2, 8	0.0771	0.0485	0.0825	0.0828
1_4x2da	2, 6	0.0465	0.0689	0.0467	0.0665
1_4x2db	2, 4	0.0692	0.0711	0.0743	0.0580
1_4x2dc	2, 2	0.0568	0.0437	0.0945	0.0751

Appendix

D

Appendix D

This appendix contains the full results of all the ANNs trained to predict Transient 2 in Section 5.2. The ANNs were all designed to predict the values of PWR variables one step ahead. Four sets of recent PWR variables were used as inputs the ANNs. These were one to four time steps. For each input set a range of one and two hidden layer ANNs were developed. The number of nodes in the hidden was varied from a maximum of 18 to a minimum of 2. The values in between were in steps of two nodes. In the case of two hidden layers both layers were varied between these limits. All the ANN training was repeated four times to avoid possible local minima. The value recorded in each case is the lowest RMS error for the training. As explained in Chapter 5 all ANNs were trained for 120,000 cycles of presentation of the training set and then a further 40,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing. In the following tables the best RMS error result for each ANN architecture is underlined while the best overall ANN for each hidden layer configuration is in bold.

One Time Step Input for One Step Prediction

Filename	Size of ANN	RMS Error			
2_1x1aa	18	0.0329	0.0323	0.0321	0.0322
2_1x1ab	16	0.0324	0.0321	0.0325	0.0321
2_1x1ac	14	0.0321	0.0317	0.0326	0.0317
2_1x1ad	12	0.0319	0.0324	0.0318	0.0318
2_1x1ae	10	0.0314	0.0318	0.0319	0.0325
2_1x1af	8	0.0332	0.0317	0.0311	0.0321
2_1x1ag	6	0.0307	0.0325	0.0312	0.0323
2_1x1ah	4	0.0325	0.0343	0.0319	0.0306
2_1x1ai	2	0.0386	0.0386	0.0381	0.0386
2_1x2aa	18, 18	0.0395	0.0399	0.0398	0.0406
2_1x2ab	18, 16	0.0397	0.0419	0.0390	0.0393
2_1x2ac	18, 14	0.0393	0.0399	0.0397	0.0392
2_1x2ad	18, 12	0.0392	0.0392	0.0391	0.0396
2_1x2ae	18, 10	0.0390	0.0390	0.0390	0.0390
2_1x2af	18, 8	0.0390	0.0389	0.0389	0.0389
2_1x2ag	18, 6	0.0386	0.0387	0.0386	0.0387
2_1x2ah	18, 4	0.0384	0.0387	0.0385	0.0385
2_1x2ai	18, 2	0.0386	0.0383	0.0384	0.0386
2_1x2aj	16, 18	0.0393	0.0392	0.0395	0.0393
2_1x2ak	16, 16	0.0392	0.0394	0.0396	0.0391
2_1x2al	16, 14	0.0394	0.0391	0.0390	0.0393
2_1x2am	16, 12	0.0396	0.0397	0.0390	0.0388
2_1x2an	16, 10	0.0390	0.0393	0.0389	0.0390
2_1x2ao	16, 8	0.0388	0.0388	0.0390	0.0386
2_1x2ap	16, 6	0.0391	0.0386	0.0385	0.0388
2_1x2aq	16, 4	0.0384	0.0383	0.0385	0.0381
2_1x2ar	16, 2	0.0386	0.0385	0.0384	0.0387

Filename	Size of ANN	RMS Error			
2_1x2as	14, 18	0.0395	0.0391	0.0392	0.0394
2_1x2at	14, 16	0.0394	0.0392	0.0389	0.0389
2_1x2au	14, 14	0.0388	0.0392	0.0390	0.0387
2_1x2av	14, 12	0.0388	0.0388	0.0389	0.0385
2_1x2aw	14, 10	0.0388	0.0387	0.0389	0.0390
2_1x2ax	14, 8	0.0388	0.0390	0.0389	0.0386
2_1x2ay	14, 6	0.0386	0.0387	0.0387	0.0385
2_1x2az	14, 4	0.0384	0.0383	0.0385	0.0381
2_1x2ba	14, 2	0.0385	0.0385	0.0384	0.0389
2_1x2bb	12, 18	0.0388	0.0389	0.0393	0.0391
2_1x2bc	12, 16	0.0391	0.0391	0.0391	0.0392
2_1x2bd	12, 14	0.0386	0.0387	0.0392	0.0390
2_1x2be	12, 12	0.0390	0.0387	0.0388	0.0388
2_1x2bf	12, 10	0.0385	0.0386	0.0388	0.0388
2_1x2bg	12, 8	0.0383	0.0389	0.0387	0.0384
2_1x2bh	12, 6	0.0383	0.0381	0.0384	0.0383
2_1x2bi	12, 4	0.0385	0.0382	0.0380	0.0378
2_1x2bj	12, 2	0.0388	0.0385	0.0386	0.0383
2_1x2bk	10, 18	0.0388	0.0387	0.0388	0.0392
2_1x2bl	10, 16	0.0386	0.0389	0.0387	0.0389
2_1x2bm	10, 14	0.0386	0.0386	0.0391	0.0384
2_1x2bn	10, 12	0.0384	0.0392	0.0389	0.0389
2_1x2bo	10, 10	0.0387	0.0385	0.0388	0.0385
2_1x2bp	10, 8	0.0385	0.0386	0.0383	0.0385
2_1x2bq	10, 6	0.0376	0.0378	0.0379	0.0379
2_1x2br	10, 4	0.0384	0.0377	0.0377	0.0387
2_1x2bs	10, 2	0.0383	0.0384	0.0385	0.0388

Filename	Size of ANN	RMS Error			
2_1x2bt	8, 18	0.0389	0.0388	<u>0.0386</u>	0.0387
2_1x2bu	8, 16	<u>0.0386</u>	0.0389	0.0390	0.0389
2_1x2bv	8, 14	0.0387	0.0388	<u>0.0385</u>	<u>0.0385</u>
2_1x2bw	8, 12	<u>0.0382</u>	0.0387	0.0383	0.0384
2_1x2bx	8, 10	0.0386	0.0388	<u>0.0383</u>	0.0385
2_1x2by	8, 8	0.0388	0.0386	<u>0.0383</u>	0.0385
2_1x2bz	8, 6	0.0386	0.0382	0.0385	<u>0.0378</u>
2_1x2ca	8, 4	0.0380	0.0382	0.0385	<u>0.0377</u>
2_1x2cb	8, 2	0.0385	0.0384	<u>0.0381</u>	0.0383
2_1x2cc	6, 18	0.0390	0.0389	<u>0.0385</u>	<u>0.0385</u>
2_1x2cd	6, 16	0.0389	0.0387	<u>0.0384</u>	0.0385
2_1x2ce	6, 14	0.0390	0.0389	0.0389	<u>0.0381</u>
2_1x2cf	6, 12	0.0383	0.0383	<u>0.0381</u>	0.0387
2_1x2cg	6, 10	<u>0.0382</u>	<u>0.0382</u>	0.0384	0.0385
2_1x2ch	6, 8	<u>0.0381</u>	0.0387	0.0384	0.0391
2_1x2ci	6, 6	0.0384	<u>0.0377</u>	0.0380	0.0387
2_1x2cj	6, 4	0.0381	0.0385	<u>0.0380</u>	0.0384
2_1x2ck	6, 2	0.0387	0.0384	0.0384	<u>0.0383</u>
2_1x2cl	4, 18	0.0390	0.0599	<u>0.0389</u>	0.0391
2_1x2cm	4, 16	<u>0.0378</u>	0.0387	0.0393	0.0386
2_1x2cn	4, 14	<u>0.0383</u>	0.0384	0.0393	0.0390
2_1x2co	4, 12	0.0391	0.0388	0.0382	<u>0.0380</u>
2_1x2cp	4, 10	0.0380	0.0386	<u>0.0377</u>	0.0385
2_1x2cq	4, 8	<u>0.0383</u>	0.0387	0.0384	<u>0.0383</u>
2_1x2cr	4, 6	0.0380	<u>0.0378</u>	0.0380	0.0384
2_1x2cs	4, 4	0.0383	<u>0.0372</u>	0.0376	0.0374
2_1x2ct	4, 2	0.0387	0.0386	0.0386	<u>0.0384</u>

Filename	Size of ANN	RMS Error			
2_1x2cv	2, 18	0.0583	0.0568	0.0556	0.0569
2_1x2cw	2, 16	0.0438	0.0545	0.0767	0.0581
2_1x2cx	2, 14	0.0614	0.0574	0.0576	0.0577
2_1x2cy	2, 12	0.0573	0.0551	0.0587	0.0422
2_1x2cz	2, 10	0.0473	0.0405	0.0581	0.0567
2_1x2da	2, 8	0.0558	0.0465	0.0413	0.0434
2_1x2db	2, 6	0.0388	0.0394	0.0430	0.0564
2_1x2dc	2, 4	0.0384	0.0416	0.0386	0.0461
2_1x2dd	2, 2	0.0433	0.0387	0.0781	0.0716

Two Time Steps Input for One Step Prediction

2_2x1aa	18	0.0323	0.0313	0.0324	0.0327
2_2x1ab	16	0.0318	0.0314	0.0314	0.0323
2_2x1ac	14	0.0302	0.0312	0.0306	0.0311
2_2x1ad	12	0.0308	0.0309	0.0308	0.0308
2_2x1ae	10	0.0297	0.0300	0.0308	0.0308
2_2x1af	8	0.0284	0.0294	0.0293	0.0302
2_2x1ag	6	0.0308	0.0299	0.0283	0.0318
2_2x1ah	4	0.0303	0.0286	0.0288	0.0299
2_2x1ai	2	0.0390	0.0390	0.0391	0.0390
2_2x2aa	18, 18	0.0411	0.0401	0.0403	0.0414
2_2x2ab	18, 16	0.0400	0.0404	0.0409	0.0402
2_2x2ac	18, 14	0.0399	0.0399	0.0404	0.0405
2_2x2ad	18, 12	0.0406	0.0393	0.0398	0.0399
2_2x2ae	18, 10	0.0396	0.0395	0.0399	0.0399
2_2x2af	18, 8	0.0394	0.0397	0.0391	0.0400
2_2x2ag	18, 6	0.0392	0.0391	0.0388	0.0391
2_2x2ah	18, 4	0.0392	0.0393	0.0390	0.0393

Filename	Size of ANN	RMS Error			
2_2x2ai	18, 2	0.0388	0.0389	0.0388	0.0391
2_2x2aj	16, 18	0.0407	0.0400	0.0404	0.0401
2_2x2ak	16, 16	0.0401	0.0396	0.0400	0.0400
2_2x2al	16, 14	0.0395	0.0397	0.0401	0.0395
2_2x2am	16, 12	0.0399	0.0393	0.0394	0.0397
2_2x2an	16, 10	0.0404	0.0396	0.0396	0.0397
2_2x2ao	16, 8	0.0393	0.0392	0.0397	0.0394
2_2x2ap	16, 6	0.0399	0.0394	0.0395	0.0390
2_2x2aq	16, 4	0.0389	0.0390	0.0387	0.0387
2_2x2ar	16, 2	0.0389	0.0390	0.0390	0.0390
2_2x2as	14, 18	0.0410	0.0400	0.0402	0.0402
2_2x2at	14, 16	0.0398	0.0404	0.0398	0.0400
2_2x2au	14, 14	0.0395	0.0397	0.0401	0.0397
2_2x2av	14, 12	0.0396	0.0395	0.0395	0.0399
2_2x2aw	14, 10	0.0392	0.0394	0.0395	0.0396
2_2x2ax	14, 8	0.0393	0.0395	0.0391	0.0392
2_2x2ay	14, 6	0.0388	0.0389	0.0388	0.0391
2_2x2az	14, 4	0.0388	0.0384	0.0389	0.0394
2_2x2ba	14, 2	0.0390	0.0393	0.0392	0.0392
2_2x2bb	12, 18	0.0398	0.0402	0.0400	0.0398
2_2x2bc	12, 16	0.0402	0.0400	0.0396	0.0396
2_2x2bd	12, 14	0.0395	0.0397	0.0394	0.0398
2_2x2be	12, 12	0.0402	0.0395	0.0394	0.0394
2_2x2bf	12, 10	0.0393	0.0391	0.0392	0.0395
2_2x2bg	12, 8	0.0391	0.0388	0.0391	0.0392
2_2x2bh	12, 6	0.0386	0.0391	0.0390	0.0397
2_2x2bi	12, 4	0.0389	0.0391	0.0392	0.0387
2_2x2bj	12, 2	0.0391	0.0392	0.0391	0.0391
2_2x2bk	10, 18	0.0398	0.0396	0.0396	0.0399

Filename	Size of ANN	RMS Error			
2_2x2bl	10, 16	0.0398	<u>0.0395</u>	0.0398	0.0397
2_2x2bm	10, 14	0.0395	0.0395	0.0395	<u>0.0390</u>
2_2x2bn	10, 12	<u>0.0393</u>	0.0394	<u>0.0393</u>	0.0395
2_2x2bo	10, 10	<u>0.0390</u>	0.0395	0.0394	0.0396
2_2x2bp	10, 8	0.0392	0.0392	<u>0.0391</u>	<u>0.0391</u>
2_2x2bq	10, 6	0.0387	<u>0.0386</u>	0.0388	0.0389
2_2x2br	10, 4	0.0389	0.0391	<u>0.0387</u>	<u>0.0387</u>
2_2x2bs	10, 2	0.0393	<u>0.0388</u>	0.0389	0.0390
2_2x2bt	8, 18	<u>0.0396</u>	0.0399	0.0398	0.0397
2_2x2bu	8, 16	0.0397	0.0397	<u>0.0394</u>	0.0396
2_2x2bv	8, 14	0.0396	0.0394	<u>0.0393</u>	0.0399
2_2x2bw	8, 12	0.0397	0.0392	0.0393	<u>0.0389</u>
2_2x2bx	8, 10	0.0393	0.0394	0.0396	<u>0.0391</u>
2_2x2by	8, 8	0.0392	0.0389	<u>0.0388</u>	0.0389
2_2x2bz	8, 6	0.0389	<u>0.0388</u>	0.0389	0.0396
2_2x2ca	8, 4	0.0395	0.0390	<u>0.0384</u>	0.0385
2_2x2cb	8, 2	0.0391	<u>0.0388</u>	0.0389	0.0391
2_2x2cc	6, 18	0.0407	0.0397	<u>0.0395</u>	<u>0.0395</u>
2_2x2cd	6, 16	<u>0.0392</u>	0.0394	0.0397	0.0395
2_2x2ce	6, 14	0.0395	<u>0.0394</u>	0.0396	0.0395
2_2x2cf	6, 12	0.0395	<u>0.0393</u>	0.0394	0.0394
2_2x2cg	6, 10	0.0392	0.0394	0.0394	<u>0.0391</u>
2_2x2ch	6, 8	0.0391	0.0391	<u>0.0389</u>	0.0391
2_2x2ci	6, 6	0.0388	0.0390	<u>0.0386</u>	0.0388
2_2x2cj	6, 4	<u>0.0382</u>	0.0384	0.0389	0.0386
2_2x2ck	6, 2	0.0394	<u>0.0391</u>	0.0392	<u>0.0391</u>
2_2x2cl	4, 18	0.0401	0.0400	0.0402	<u>0.0397</u>
2_2x2cm	4, 16	<u>0.0395</u>	0.0397	0.0400	<u>0.0395</u>
2_2x2cn	4, 14	0.0394	0.0393	<u>0.0389</u>	0.0391

Filename	Size of ANN	RMS Error			
2_2x2co	4, 12	0.0389	0.0401	0.0393	0.0394
2_2x2cp	4, 10	0.0393	0.0391	0.0389	0.0392
2_2x2cq	4, 8	0.0388	0.0388	0.0391	0.0383
2_2x2cr	4, 6	0.0389	0.0391	0.0387	0.0390
2_2x2cs	4, 4	0.0389	0.0383	0.0393	0.0389
2_2x2ct	4, 2	0.0390	0.0392	0.0392	0.0391
2_2x2cu	2, 18	0.0945	0.0517	0.0517	0.0518
2_2x2cv	2, 16	0.0512	0.0516	0.0466	0.0504
2_2x2cw	2, 14	0.0518	0.0947	0.0491	0.0485
2_2x2cx	2, 12	0.0943	0.0508	0.0943	0.0945
2_2x2cy	2, 10	0.0944	0.0449	0.0439	0.0462
2_2x2cz	2, 8	0.0415	0.0400	0.0429	0.0429
2_2x2da	2, 6	0.0454	0.0404	0.0497	0.0929
2_2x2db	2, 4	0.0450	0.0442	0.0394	0.0463
2_2x2dc	2, 2	0.0501	0.0399	0.0393	0.0396

Three Time Steps Input for One Step Prediction

2_3x1aa	18	0.0385	0.0384	0.0386	0.0387
2_3x1ab	16	0.0391	0.0382	0.0384	0.0387
2_3x1ac	14	0.0386	0.0384	0.0389	0.0390
2_3x1ad	12	0.0386	0.0388	0.0384	0.0386
2_3x1ae	10	0.0383	0.0388	0.0375	0.0381
2_3x1af	8	0.0376	0.0372	0.0388	0.0377
2_3x1ag	6	0.0382	0.0374	0.0373	0.0376
2_3x1ah	4	0.0367	0.0360	0.0363	0.0366
2_3x1ai	2	0.0368	0.0383	0.0381	0.0378
2_3x2aa	18, 18	0.0420	0.0416	0.0413	0.0410
2_3x2ab	18, 16	0.0441	0.0415	0.0417	0.0406

Filename	Size of ANN	RMS Error			
2_3x2ac	18, 14	0.041	0.0404	0.041	0.041
2_3x2ad	18, 12	0.0407	0.042	0.042	0.041
2_3x2ae	18, 10	0.041	0.04	0.0403	0.041
2_3x2af	18, 8	0.041	0.0399	0.04	0.04
2_3x2ag	18, 6	0.04	0.04	0.0398	0.04
2_3x2ah	18, 4	0.0391	0.039	0.039	0.04
2_3x2ai	18, 2	0.04	0.042	0.0392	0.039
2_3x2aj	16, 18	0.043	0.0410	0.041	0.042
2_3x2ak	16, 16	0.041	0.041	0.042	0.0409
2_3x2al	16, 14	0.0407	0.042	0.041	0.041
2_3x2am	16, 12	0.041	0.0404	0.041	0.041
2_3x2an	16, 10	0.0402	0.0402	0.04	0.0402
2_3x2ao	16, 8	0.0396	0.04	0.04	0.04
2_3x2ap	16, 6	0.0395	0.04	0.04	0.04
2_3x2aq	16, 4	0.0393	0.039	0.04	0.04
2_3x2ar	16, 2	0.0393	0.0393	0.042	0.0393
2_3x2as	14, 18	0.0407	0.042	0.042	0.042
2_3x2at	14, 16	0.041	0.041	0.043	0.0402
2_3x2au	14, 14	0.0403	0.041	0.041	0.041
2_3x2av	14, 12	0.04	0.04	0.0402	0.041
2_3x2aw	14, 10	0.04	0.04	0.0399	0.04
2_3x2ax	14, 8	0.04	0.04	0.0395	0.04
2_3x2ay	14, 6	0.04	0.0395	0.0395	0.04
2_3x2az	14, 4	0.04	0.0392	0.039	0.04
2_3x2ba	14, 2	0.04	0.04	0.0394	0.0394
2_3x2bb	12, 18	0.041	0.0405	0.041	0.041
2_3x2bc	12, 16	0.041	0.041	0.042	0.0403
2_3x2bd	12, 14	0.04	0.04	0.0400	0.041
2_3x2be	12, 12	0.04	0.042	0.04	0.0399

Filename	Size of ANN	RMS Error			
2_3x2bf	12, 10	0.04	0.04	0.0396	0.0396
2_3x2bg	12, 8	0.04	0.04	0.0393	0.04
2_3x2bh	12, 6	0.04	0.039	0.04	0.0390
2_3x2bi	12, 4	0.04	0.039	0.0388	0.039
2_3x2bj	12, 2	0.0394	0.04	0.0394	0.04
2_3x2bk	10, 18	0.041	0.04	0.0401	0.041
2_3x2bl	10, 16	0.0400	0.041	0.04	0.04
2_3x2bm	10, 14	0.041	0.04	0.04	0.0402
2_3x2bn	10, 12	0.0397	0.04	0.04	0.04
2_3x2bo	10, 10	0.04	0.0394	0.04	0.04
2_3x2bp	10, 8	0.04	0.0395	0.04	0.04
2_3x2bq	10, 6	0.04	0.0389	0.039	0.039
2_3x2br	10, 4	0.039	0.0389	0.039	0.04
2_3x2bs	10, 2	0.039	0.04	0.0393	0.0393
2_3x2bt	8, 18	0.0401	0.0404	0.0406	0.0416
2_3x2bu	8, 16	0.0401	0.0404	0.0400	0.0402
2_3x2bv	8, 14	0.0402	0.0402	0.0400	0.0402
2_3x2bw	8, 12	0.0402	0.0400	0.0398	0.0395
2_3x2bx	8, 10	0.0393	0.0395	0.0398	0.0396
2_3x2by	8, 8	0.0396	0.0391	0.0393	0.0395
2_3x2bz	8, 6	0.0395	0.0394	0.0397	0.0394
2_3x2ca	8, 4	0.0387	0.0392	0.0390	0.0390
2_3x2cb	8, 2	0.0393	0.0392	0.0397	0.0392
2_3x2cc	6, 18	0.0400	0.0402	0.0403	0.0400
2_3x2cd	6, 16	0.0401	0.0399	0.0400	0.0398
2_3x2ce	6, 14	0.0404	0.0398	0.0400	0.0397
2_3x2cf	6, 12	0.0396	0.0398	0.0402	0.0394
2_3x2cg	6, 10	0.0394	0.0395	0.0391	0.0393
2_3x2ch	6, 8	0.0393	0.0396	0.0394	0.0396

Filename	Size of ANN	RMS Error			
2_3x2ci	6, 6	0.0393	0.0390	0.0393	0.0394
2_3x2cj	6, 4	0.0391	0.0387	0.0393	0.0384
2_3x2ck	6, 2	0.0395	0.0391	0.0395	0.0393
2_3x2cl	4, 18	0.0401	0.0410	0.0401	0.0399
2_3x2cm	4, 16	0.0407	0.0426	0.0401	0.0400
2_3x2cn	4, 14	0.0398	0.0396	0.0408	0.0396
2_3x2co	4, 12	0.0395	0.0405	0.0400	0.0411
2_3x2cp	4, 10	0.0392	0.0401	0.0393	0.0404
2_3x2cq	4, 8	0.0403	0.0406	0.0396	0.0393
2_3x2cr	4, 6	0.0389	0.0390	0.0391	0.0401
2_3x2cs	4, 4	0.0390	0.0389	0.0390	0.0385
2_3x2ct	4, 2	0.0396	0.0391	0.0397	0.0405
2_3x2cu	2, 18	0.0962	0.0543	0.0488	0.0965
2_3x2cv	2, 16	0.0486	0.0966	0.0487	0.0474
2_3x2cw	2, 14	0.0487	0.0474	0.0964	0.0575
2_3x2cx	2, 12	0.0963	0.0815	0.0462	0.0558
2_3x2cy	2, 10	0.0473	0.0492	0.0493	0.0494
2_3x2cz	2, 8	0.0450	0.0453	0.0484	0.0405
2_3x2da	2, 6	0.0468	0.0477	0.0446	0.0444
2_3x2db	2, 4	0.0433	0.0401	0.0422	0.0401
2_3x2dc	2, 2	0.0458	0.0415	0.0441	0.1052

Four Time Steps Input for One Step Prediction

2_4x1aa	18	0.0388	0.0388	0.0390	0.0395
2_4x1ab	16	0.0388	0.0377	0.0383	0.0373
2_4x1ac	14	0.0382	0.0390	0.0383	0.0388
2_4x1ad	12	0.0384	0.0387	0.0377	0.0381
2_4x1ae	10	0.0378	0.0387	0.0366	0.0371
2_4x1af	8	0.0368	0.0369	0.0388	0.0374

Filename	Size of ANN	RMS Error			
2_4x1ag	6	0.0383	0.0362	0.0370	0.0364
2_4x1ah	4	0.0351	0.0380	0.0381	0.0356
2_4x1ai	2	0.0391	0.0390	0.0390	0.0397
2_4x2aa	18, 18	0.0427	0.0428	0.0441	0.0438
2_4x2ab	18, 16	0.0430	0.0431	0.0434	0.0444
2_4x2ac	18, 14	0.0436	0.0431	0.0426	0.0429
2_4x2ad	18, 12	0.0438	0.0423	0.0424	0.0420
2_4x2ae	18, 10	0.0422	0.0424	0.0422	0.0431
2_4x2af	18, 8	0.0420	0.0417	0.0436	0.0420
2_4x2ag	18, 6	0.0409	0.0419	0.0419	0.0411
2_4x2ah	18, 4	0.0408	0.0416	0.0419	0.0411
2_4x2ai	18, 2	0.0407	0.0408	0.0412	0.0405
2_4x2aj	16, 18	0.0426	0.0444	0.0442	0.0445
2_4x2ak	16, 16	0.0438	0.0421	0.0437	0.0421
2_4x2al	16, 14	0.0428	0.0417	0.0422	0.0425
2_4x2am	16, 12	0.0422	0.0420	0.0428	0.0436
2_4x2an	16, 10	0.0425	0.0419	0.0413	0.0413
2_4x2ao	16, 8	0.0428	0.0413	0.0417	0.0414
2_4x2ap	16, 6	0.0412	0.0416	0.0411	0.0407
2_4x2aq	16, 4	0.0407	0.0405	0.0414	0.0413
2_4x2ar	16, 2	0.0404	0.0406	0.0408	0.0406
2_4x2as	14, 18	0.0426	0.0424	0.0428	0.0439
2_4x2at	14, 16	0.0432	0.0421	0.0415	0.0419
2_4x2au	14, 14	0.0431	0.0426	0.0413	0.0436
2_4x2av	14, 12	0.0420	0.0417	0.0420	0.0424
2_4x2aw	14, 10	0.0422	0.0425	0.0418	0.0412
2_4x2ax	14, 8	0.0413	0.0412	0.0410	0.0413
2_4x2ay	14, 6	0.0409	0.0411	0.0409	0.0405
2_4x2az	14, 4	0.0413	0.0406	0.0414	0.0407

Filename	Size of ANN	RMS Error			
2_4x2ba	14, 2	0.0405	0.0404	0.0406	0.0402
2_4x2bb	12, 18	0.0414	0.0414	0.0431	0.0415
2_4x2bc	12, 16	0.0422	0.0426	0.0419	0.0420
2_4x2bd	12, 14	0.0415	0.0424	0.0418	0.0417
2_4x2be	12, 12	0.0409	0.0413	0.0412	0.0413
2_4x2bf	12, 10	0.0417	0.0409	0.0414	0.0420
2_4x2bg	12, 8	0.0411	0.0407	0.0419	0.0412
2_4x2bh	12, 6	0.0415	0.0402	0.0413	0.0411
2_4x2bi	12, 4	0.0407	0.0403	0.0404	0.0402
2_4x2bj	12, 2	0.0408	0.0400	0.0400	0.0404
2_4x2bk	10, 18	0.0413	0.0412	0.0420	0.0416
2_4x2bl	10, 16	0.0411	0.0417	0.0417	0.0417
2_4x2bm	10, 14	0.0411	0.0408	0.0412	0.0410
2_4x2bn	10, 12	0.0416	0.0415	0.0412	0.0415
2_4x2bo	10, 10	0.0409	0.0406	0.0412	0.0407
2_4x2bp	10, 8	0.0412	0.0399	0.0407	0.0413
2_4x2bq	10, 6	0.0406	0.0400	0.0408	0.0409
2_4x2br	10, 4	0.0408	0.0402	0.0404	0.0404
2_4x2bs	10, 2	0.0400	0.0401	0.0404	0.0403
2_4x2bt	8, 18	0.0413	0.0409	0.0410	0.0412
2_4x2bu	8, 16	0.0418	0.0408	0.0415	0.0415
2_4x2bv	8, 14	0.0414	0.0411	0.0405	0.0410
2_4x2bw	8, 12	0.0409	0.0406	0.0408	0.0409
2_4x2bx	8, 10	0.0404	0.0403	0.0407	0.0409
2_4x2by	8, 8	0.0403	0.0403	0.0403	0.0403
2_4x2bz	8, 6	0.0395	0.0402	0.0407	0.0407
2_4x2ca	8, 4	0.0394	0.0402	0.0399	0.0399
2_4x2cb	8, 2	0.0399	0.0406	0.0403	0.0401
2_4x2cc	6, 18	0.0403	0.0405	0.0406	0.0411

Filename	Size of ANN	RMS Error			
2_4x2cd	6, 16	0.0405	0.0403	0.0408	0.0406
2_4x2ce	6, 14	0.0401	0.0408	0.0406	0.0402
2_4x2cf	6, 12	0.0402	0.0406	0.0398	0.0412
2_4x2cg	6, 10	0.0407	0.0408	0.0396	0.0396
2_4x2ch	6, 8	0.0397	0.0407	0.0392	0.0396
2_4x2ci	6, 6	0.0391	0.0395	0.0400	0.0400
2_4x2cj	6, 4	0.0390	0.0401	0.0401	0.0397
2_4x2ck	6, 2	0.0406	0.0400	0.0402	0.0397
2_4x2cl	4, 18	0.0415	0.0408	0.0400	0.0397
2_4x2cm	4, 16	0.0400	0.0399	0.0400	0.0401
2_4x2cn	4, 14	0.0404	0.0394	0.0404	0.0408
2_4x2co	4, 12	0.0397	0.0409	0.0396	0.0399
2_4x2cp	4, 10	0.0394	0.0389	0.0397	0.0393
2_4x2cq	4, 8	0.0391	0.0403	0.0400	0.0393
2_4x2cr	4, 6	0.0402	0.0401	0.0401	0.0392
2_4x2cs	4, 4	0.0393	0.0395	0.0387	0.0392
2_4x2ct	4, 2	0.0397	0.0394	0.0394	0.0402
2_4x2cu	2, 18	0.0975	0.0526	0.0828	0.0543
2_4x2cv	2, 16	0.1004	0.0973	0.0551	0.0975
2_4x2cw	2, 14	0.0974	0.0611	0.1016	0.0542
2_4x2cx	2, 12	0.0583	0.0578	0.0454	0.0924
2_4x2cy	2, 10	0.0468	0.0470	0.0930	0.0494
2_4x2cz	2, 8	0.0465	0.0428	0.0502	0.0483
2_4x2da	2, 6	0.0393	0.0386	0.0389	0.0412
2_4x2db	2, 4	0.0492	0.0389	0.0390	0.0401
2_4x2dc	2, 2	0.0399	0.0397	0.0392	0.0510

Appendix

E Results of One Step Prediction Tests for Transient 3

This appendix contains the full results of all the ANNs trained to predict Transient 3 in Section 5.2. The ANNs were all designed to predict the values of PWR variables one step ahead. Four sets of recent PWR variables were used as inputs the ANNs. These were one to four time steps. For each input set a range of one and two hidden layer ANNs were developed. The number of nodes in the hidden was varied from a maximum of 18 to a minimum of 2. The values in between were in steps of two nodes. In the case of two hidden layers both layers were varied between these limits. All the ANN training was repeated four times to avoid possible local minima. The value recorded in each case is the lowest RMS error for the training. As explained in Chapter 5 all ANNs were trained for 120,000 cycles of presentation of the training set and then a further 40,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing. In the following tables the best RMS error result for each ANN architecture is underlined while the best overall ANN for each hidden layer configuration is in bold.

One Time Step Input for One Step Prediction

Filename	Size of ANN	RMS Error			
3_1x1aa	18	0.0344	0.0345	0.0354	0.0347
3_1x1ab	16	0.0399	0.0361	0.0351	0.0342
3_1x1ac	14	0.0336	0.0349	0.0348	0.0341
3_1x1ad	12	0.0335	0.0336	0.0348	0.0340
3_1x1ae	10	0.0340	0.0342	0.0345	0.0353
3_1x1af	8	0.0346	0.0341	0.0359	0.0334
3_1x1ag	6	0.0336	0.0326	0.0343	0.0325
3_1x1ah	4	0.0326	0.0337	0.0334	0.0333
3_1x1ai	2	0.0381	0.0389	0.0381	0.0380
3_1x2aa	18, 18	0.0395	0.0399	0.0398	0.0406
3_1x2ab	18, 16	0.0397	0.0419	0.0390	0.0393
3_1x2ac	18, 14	0.0393	0.0399	0.0397	0.0392
3_1x2ad	18, 12	0.0392	0.0392	0.0391	0.0396
3_1x2ae	18, 10	0.0390	0.0390	0.0390	0.0390
3_1x2af	18, 8	0.0390	0.0389	0.0389	0.0389
3_1x2ag	18, 6	0.0386	0.0387	0.0386	0.0387
3_1x2ah	18, 4	0.0384	0.0387	0.0385	0.0385
3_1x2ai	18, 2	0.0386	0.0383	0.0384	0.0386
3_1x2aj	16, 18	0.0393	0.0392	0.0395	0.0393
3_1x2ak	16, 16	0.0392	0.0394	0.0396	0.0391
3_1x2al	16, 14	0.0394	0.0391	0.0390	0.0393
3_1x2am	16, 12	0.0396	0.0397	0.0390	0.0388
3_1x2an	16, 10	0.0390	0.0393	0.0389	0.0390
3_1x2ao	16, 8	0.0388	0.0388	0.0390	0.0386
3_1x2ap	16, 6	0.0391	0.0386	0.0385	0.0388
3_1x2aq	16, 4	0.0384	0.0383	0.0385	0.0381
3_1x2ar	16, 2	0.0386	0.0385	0.0384	0.0387

Filename	Size of ANN	RMS Error			
3_1x2as	14, 18	0.0395	0.0391	0.0392	0.0394
3_1x2at	14, 16	0.0394	0.0392	0.0389	0.0389
3_1x2au	14, 14	0.0388	0.0392	0.0390	0.0387
3_1x2av	14, 12	0.0388	0.0388	0.0389	0.0385
3_1x2aw	14, 10	0.0388	0.0387	0.0389	0.0390
3_1x2ax	14, 8	0.0388	0.0390	0.0389	0.0386
3_1x2ay	14, 6	0.0386	0.0387	0.0387	0.0385
3_1x2az	14, 4	0.0384	0.0383	0.0385	0.0381
3_1x2ba	14, 2	0.0385	0.0385	0.0384	0.0389
3_1x2bb	12, 18	0.0388	0.0389	0.0393	0.0391
3_1x2bc	12, 16	0.0391	0.0391	0.0391	0.0392
3_1x2bd	12, 14	0.0386	0.0387	0.0392	0.0390
3_1x2be	12, 12	0.0390	0.0387	0.0388	0.0388
3_1x2bf	12, 10	0.0385	0.0386	0.0388	0.0388
3_1x2bg	12, 8	0.0383	0.0389	0.0387	0.0384
3_1x2bh	12, 6	0.0383	0.0381	0.0384	0.0383
3_1x2bi	12, 4	0.0385	0.0382	0.0380	0.0378
3_1x2bj	12, 2	0.0388	0.0385	0.0386	0.0383
3_1x2bk	10, 18	0.0388	0.0387	0.0388	0.0392
3_1x2bl	10, 16	0.0386	0.0389	0.0387	0.0389
3_1x2bm	10, 14	0.0386	0.0386	0.0391	0.0384
3_1x2bn	10, 12	0.0384	0.0392	0.0389	0.0389
3_1x2bo	10, 10	0.0387	0.0385	0.0388	0.0385
3_1x2bp	10, 8	0.0385	0.0386	0.0383	0.0385
3_1x2bq	10, 6	0.0376	0.0378	0.0379	0.0379
3_1x2br	10, 4	0.0384	0.0377	0.0377	0.0387
3_1x2bs	10, 2	0.0383	0.0384	0.0385	0.0388

Filename	Size of ANN	RMS Error			
3_1x2bt	8, 18	0.0389	0.0388	0.0386	0.0387
3_1x2bu	8, 16	0.0386	0.0389	0.0390	0.0389
3_1x2bv	8, 14	0.0387	0.0388	0.0385	0.0385
3_1x2bw	8, 12	0.0382	0.0387	0.0383	0.0384
3_1x2bx	8, 10	0.0386	0.0388	0.0383	0.0385
3_1x2by	8, 8	0.0388	0.0386	0.0383	0.0385
3_1x2bz	8, 6	0.0386	0.0382	0.0385	0.0378
3_1x2ca	8, 4	0.0380	0.0382	0.0385	0.0377
3_1x2cb	8, 2	0.0385	0.0384	0.0381	0.0383
3_1x2cc	6, 18	0.0390	0.0389	0.0385	0.0385
3_1x2cd	6, 16	0.0389	0.0387	0.0384	0.0385
3_1x2ce	6, 14	0.0390	0.0389	0.0389	0.0381
3_1x2cf	6, 12	0.0383	0.0383	0.0381	0.0387
3_1x2cg	6, 10	0.0382	0.0382	0.0384	0.0385
3_1x2ch	6, 8	0.0381	0.0387	0.0384	0.0391
3_1x2ci	6, 6	0.0384	0.0377	0.0380	0.0387
3_1x2cj	6, 4	0.0381	0.0385	0.0380	0.0384
3_1x2ck	6, 2	0.0387	0.0384	0.0384	0.0383
3_1x2cl	4, 18	0.0390	0.0599	0.0389	0.0391
3_1x2cm	4, 16	0.0378	0.0387	0.0393	0.0386
3_1x2cn	4, 14	0.0383	0.0384	0.0393	0.0390
3_1x2co	4, 12	0.0391	0.0388	0.0382	0.0380
3_1x2cp	4, 10	0.0380	0.0386	0.0377	0.0385
3_1x2cq	4, 8	0.0383	0.0387	0.0384	0.0383
3_1x2cr	4, 6	0.0380	0.0378	0.0380	0.0384
3_1x2cs	4, 4	0.0383	**0.0372**	0.0376	0.0374
3_1x2ct	4, 2	0.0387	0.0386	0.0386	0.0384

Filename	Size of ANN	RMS Error			
3_1x2cu	2, 18	0.0583	0.0568	0.0556	0.0569
3_1x2cv	2, 16	0.0438	0.0545	0.0767	0.0581
3_1x2cw	2, 14	0.0614	0.0574	0.0576	0.0577
3_1x2cx	2, 12	0.0573	0.0551	0.0587	0.0422
3_1x2cy	2, 10	0.0473	0.0405	0.0581	0.0567
3_1x2cz	2, 8	0.0558	0.0465	0.0413	0.0434
3_1x2da	2, 6	0.0388	0.0394	0.0430	0.0564
3_1x2db	2, 4	0.0384	0.0416	0.0386	0.0461
3_1x2dc	2, 2	0.0433	0.0387	0.0781	0.0716

Two Time Steps Input for One Step Prediction

3_2x1aa	18	0.0359	0.0336	0.0348	0.0356
3_2x1ab	16	0.0359	0.0361	0.0378	0.0363
3_2x1ac	14	0.0364	0.0363	0.0346	0.0355
3_2x1ad	12	0.0345	0.0346	0.0363	0.0352
3_2x1ae	10	0.0363	0.0358	0.0349	0.0364
3_2x1af	8	0.0360	0.0347	**0.0328**	0.0360
3_2x1ag	6	0.0354	0.0373	0.0340	0.0354
3_2x1ah	4	0.0381	0.0356	0.0360	0.0351
3_2x1ai	2	0.0392	0.0391	0.0391	0.0395
3_2x2aa	18, 18	0.0403	0.0403	0.0403	0.0402
3_2x2ab	18, 16	0.0417	0.0398	0.0398	0.0402
3_2x2ac	18, 14	0.0397	0.0402	0.0402	0.0407
3_2x2ad	18, 12	0.0395	0.0397	0.0397	0.0400
3_2x2ae	18, 10	0.0396	0.0395	0.0395	0.0393
3_2x2af	18, 8	0.0392	0.0395	0.0395	0.0395
3_2x2ag	18, 6	0.0390	0.0397	0.0397	0.0393
3_2x2ah	18, 4	0.0391	0.0385	0.0385	0.0389

Filename	Size of ANN	RMS Error			
3_2x2ai	18, 2	0.0394	0.0340	0.0389	0.0390
3_2x2aj	16, 18	0.0401	0.0402	0.0402	0.0400
3_2x2ak	16, 16	0.0408	0.0399	0.0400	0.0402
3_2x2al	16, 14	0.0402	0.0398	0.0398	0.0396
3_2x2am	16, 12	0.0398	0.0395	0.0394	0.0395
3_2x2an	16, 10	0.0395	0.0397	0.0396	0.0393
3_2x2ao	16, 8	0.0393	0.0392	0.0393	0.0394
3_2x2ap	16, 6	0.0391	0.0392	0.0392	0.0392
3_2x2aq	16, 4	0.0392	0.0394	0.0394	0.0392
3_2x2ar	16, 2	0.0406	0.0471	0.0471	0.0391
3_2x2as	14, 18	0.0399	0.0403	0.0403	0.0400
3_2x2at	14, 16	0.0397	0.0398	0.0398	0.0400
3_2x2au	14, 14	0.0396	0.0398	0.0398	0.0396
3_2x2av	14, 12	0.0396	0.0396	0.0396	0.0397
3_2x2aw	14, 10	0.0395	0.0394	0.0394	0.0394
3_2x2ax	14, 8	0.0396	0.0393	0.0393	0.0394
3_2x2ay	14, 6	0.0389	0.0391	0.0391	0.0389
3_2x2az	14, 4	0.0389	0.0393	0.0393	0.0393
3_2x2ba	14, 2	0.0391	0.0393	0.0393	0.0390
3_2x2bb	12, 18	0.0396	0.0395	0.0395	0.0404
3_2x2bc	12, 16	0.0402	0.0396	0.0396	0.0398
3_2x2bd	12, 14	0.0395	0.0397	0.0397	0.0396
3_2x2be	12, 12	0.0394	0.0395	0.0395	0.0394
3_2x2bf	12, 10	0.0392	0.0394	0.0394	0.0396
3_2x2bg	12, 8	0.0391	0.0391	0.0391	0.0390
3_2x2bh	12, 6	0.0391	0.0392	0.0392	0.0390
3_2x2bi	12, 4	0.0385	0.0393	0.0393	0.0388
3_2x2bj	12, 2	0.0392	0.0390	0.0391	0.0393
3_2x2bk	10, 18	0.0394	0.0397	0.0398	0.0394

Filename	Size of ANN	RMS Error			
3_2x2bl	10, 16	0.0400	0.0396	0.0398	0.0400
3_2x2bm	10, 14	0.0396	0.0394	0.0395	0.0396
3_2x2bn	10, 12	0.0393	0.0398	0.0393	0.0393
3_2x2bo	10, 10	0.0397	0.0396	0.0391	0.0397
3_2x2bp	10, 8	0.0391	0.0391	0.0390	0.0391
3_2x2bq	10, 6	0.0392	0.0396	0.0392	0.0388
3_2x2br	10, 4	0.0391	0.0389	0.0390	0.0391
3_2x2bs	10, 2	0.0392	0.0389	0.0393	0.0391
3_2x2bt	8, 18	0.0398	0.0399	0.0397	0.0398
3_2x2bu	8, 16	0.0395	0.0395	0.0396	0.0397
3_2x2bv	8, 14	0.0398	0.0397	0.0396	0.0400
3_2x2bw	8, 12	0.0395	0.0392	0.0393	0.0394
3_2x2bx	8, 10	0.0395	0.0393	0.0393	0.0389
3_2x2by	8, 8	0.0393	0.0389	0.0389	0.0391
3_2x2bz	8, 6	0.0392	0.0389	0.0387	0.0384
3_2x2ca	8, 4	0.0391	0.0392	0.0387	0.0391
3_2x2cb	8, 2	0.0394	0.0395	0.0394	0.0393
3_2x2cc	6, 18	0.0396	0.0396	0.0400	0.0397
3_2x2cd	6, 16	0.0398	0.0400	0.0400	0.0395
3_2x2ce	6, 14	0.0395	0.0395	0.0395	0.0393
3_2x2cf	6, 12	0.0391	0.0391	0.0393	0.0394
3_2x2cg	6, 10	0.0393	0.0392	0.0393	0.0396
3_2x2ch	6, 8	0.0392	0.0393	0.0391	0.0387
3_2x2ci	6, 6	0.0386	0.0389	0.0391	0.0391
3_2x2cj	6, 4	0.0390	0.0391	0.0388	0.0394
3_2x2ck	6, 2	0.0392	0.0392	0.0393	0.0389
3_2x2cl	4, 18	0.0392	0.0396	0.0395	0.0397
3_2x2cm	4, 16	0.0400	0.0393	0.0396	0.0397
3_2x2cn	4, 14	0.0397	0.0387	0.0397	0.0399

Filename	Size of ANN	RMS Error			
3_2x2co	4, 12	0.0382	0.0396	0.0400	0.0389
3_2x2cp	4, 10	0.0393	0.0388	**0.0380**	0.0394
3_2x2cq	4, 8	0.0393	0.0393	0.0393	0.0394
3_2x2cr	4, 6	0.0388	0.0387	0.0398	0.0389
3_2x2cs	4, 4	0.0390	0.0393	0.0393	0.0393
3_2x2ct	4, 2	0.0395	0.0398	0.0392	0.0397
3_2x2cu	2, 18	0.0572	0.0533	0.0610	0.0577
3_2x2cv	2, 16	0.0574	0.0480	0.0554	0.0601
3_2x2cw	2, 14	0.0545	0.0562	0.0546	0.0505
3_2x2cx	2, 12	0.0514	0.0474	0.0470	0.0470
3_2x2cy	2, 10	0.0419	0.0734	0.0563	0.0460
3_2x2cz	2, 8	0.0459	0.0469	0.0486	0.0403
3_2x2da	2, 6	0.0566	0.0545	0.0471	0.0417
3_2x2db	2, 4	0.0389	0.0406	0.0455	0.0477
3_2x2dc	2, 2	0.0418	0.0394	0.0404	0.0395

Three Time Steps Input for One Step Prediction

3_3x1aa	18	0.0346	0.0347	0.0346	0.0351
3_3x1ab	16	0.0344	0.0342	0.0350	0.0343
3_3x1ac	14	0.0346	0.0341	0.0342	0.0345
3_3x1ad	12	0.0330	0.0339	0.0348	0.0334
3_3x1ae	10	0.0343	0.0332	0.0337	0.0337
3_3x1af	8	0.0333	0.0341	0.0333	**0.0324**
3_3x1ag	6	0.0329	0.0335	0.0350	0.0336
3_3x1ah	4	0.0353	0.0326	0.0331	0.0348
3_3x1ai	2	0.0400	0.0394	0.0394	0.0401
3_3x2aa	18, 18	0.0415	0.0423	0.0420	0.0416
3_3x2ab	18, 16	0.0418	0.0411	0.0419	0.0412

Filename	Size of ANN	RMS Error			
3_3x2ac	18, 14	0.0411	0.0413	0.0410	0.0413
3_3x2ad	18, 12	0.0414	0.0412	0.0416	0.0411
3_3x2ae	18, 10	0.0406	0.0408	0.0408	0.0409
3_3x2af	18, 8	0.0405	0.0408	0.0405	0.0406
3_3x2ag	18, 6	0.0399	0.0403	0.0405	0.0404
3_3x2ah	18, 4	0.0403	0.0399	0.0399	0.0402
3_3x2ai	18, 2	0.0400	0.0409	0.0405	0.0400
3_3x2aj	16, 18	0.0413	0.0415	0.0414	0.0410
3_3x2ak	16, 16	0.0413	0.0416	0.0413	0.0421
3_3x2al	16, 14	0.0414	0.0415	0.0414	0.0412
3_3x2am	16, 12	0.0409	0.0415	0.0408	0.0411
3_3x2an	16, 10	0.0405	0.0409	0.0405	0.0402
3_3x2ao	16, 8	0.0405	0.0406	0.0403	0.0408
3_3x2ap	16, 6	0.0402	0.0404	0.0402	0.0404
3_3x2aq	16, 4	0.0401	0.0399	0.0396	0.0401
3_3x2ar	16, 2	0.0403	0.0403	0.0402	0.0399
3_3x2as	14, 18	0.0426	0.0414	0.0408	0.0415
3_3x2at	14, 16	0.0413	0.0412	0.0410	0.0412
3_3x2au	14, 14	0.0408	0.0416	0.0416	0.0410
3_3x2av	14, 12	0.0407	0.0408	0.0409	0.0407
3_3x2aw	14, 10	0.0410	0.0405	0.0404	0.0415
3_3x2ax	14, 8	0.0407	0.0406	0.0402	0.0401
3_3x2ay	14, 6	0.0405	0.0401	0.0402	0.0400
3_3x2az	14, 4	0.0407	0.0400	0.0402	0.0400
3_3x2ba	14, 2	0.0398	0.0399	0.0401	0.0401
3_3x2bb	12, 18	0.0411	0.0414	0.0412	0.0413
3_3x2bc	12, 16	0.0412	0.0409	0.0414	0.0414
3_3x2bd	12, 14	0.0409	0.0413	0.0408	0.0413
3_3x2be	12, 12	0.0411	0.0410	0.0409	0.0404

Filename	Size of ANN	RMS Error			
3_3x2bf	12, 10	0.0407	0.0408	0.0403	0.0404
3_3x2bg	12, 8	0.0401	0.0406	0.0410	0.0401
3_3x2bh	12, 6	0.0401	0.0400	0.0400	0.0401
3_3x2bi	12, 4	0.0404	0.0399	0.0394	0.0399
3_3x2bj	12, 2	0.0401	0.0400	0.0399	0.0399
3_3x2bk	10, 18	0.0411	0.0411	0.0408	0.0410
3_3x2bl	10, 16	0.0417	0.0406	0.0406	0.0410
3_3x2bm	10, 14	0.0407	0.0408	0.0411	0.0407
3_3x2bn	10, 12	0.0405	0.0402	0.0404	0.0404
3_3x2bo	10, 10	0.0401	0.0402	0.0406	0.0404
3_3x2bp	10, 8	0.0401	0.0401	0.0402	0.0405
3_3x2bq	10, 6	0.0401	0.0404	0.0402	0.0401
3_3x2br	10, 4	0.0398	0.0395	0.0406	0.0404
3_3x2bs	10, 2	0.0400	0.0398	0.0399	0.0399
3_3x2bt	8, 18	0.0408	0.0407	0.0407	0.0417
3_3x2bu	8, 16	0.0410	0.0406	0.0407	0.0406
3_3x2bv	8, 14	0.0408	0.0406	0.0405	0.0406
3_3x2bw	8, 12	0.0401	0.0404	0.0407	0.0402
3_3x2bx	8, 10	0.0402	0.0403	0.0403	0.0400
3_3x2by	8, 8	0.0400	0.0403	0.0404	0.0403
3_3x2bz	8, 6	0.0398	0.0401	0.0402	0.0399
3_3x2ca	8, 4	0.0397	0.0398	0.0399	0.0395
3_3x2cb	8, 2	0.0400	0.0400	0.0400	0.0399
3_3x2cc	6, 18	0.0409	0.0415	0.0408	0.0413
3_3x2cd	6, 16	0.0406	0.0405	0.0404	0.0413
3_3x2ce	6, 14	0.0407	0.0414	0.0404	0.0405
3_3x2cf	6, 12	0.0402	0.0406	0.0405	0.0403
3_3x2cg	6, 10	0.0402	0.0403	0.0402	0.0409
3_3x2ch	6, 8	0.0400	0.0399	0.0401	0.0402

Filename	Size of ANN	RMS Error			
3_3x2ci	6, 6	0.0400	0.0393	0.0405	0.0405
3_3x2cj	6, 4	0.0400	0.0394	0.0397	0.0388
3_3x2ck	6, 2	0.0401	0.0398	0.0398	0.0402
3_3x2cl	4, 18	0.0406	0.0400	0.0409	0.0404
3_3x2cm	4, 16	0.0405	0.0399	0.0406	0.0404
3_3x2cn	4, 14	0.0404	0.0404	0.0401	0.0410
3_3x2co	4, 12	0.0405	0.0403	0.0399	0.0402
3_3x2cp	4, 10	0.0401	0.0399	0.0395	0.0400
3_3x2cq	4, 8	0.0405	0.0402	0.0397	0.0403
3_3x2cr	4, 6	0.0398	0.0393	0.0398	0.0403
3_3x2cs	4, 4	0.0398	0.0402	0.0396	0.0394
3_3x2ct	4, 2	0.0405	0.0455	0.0402	0.0406
3_3x2cu	2, 18	0.0726	0.0519	0.0811	0.0521
3_3x2cv	2, 16	0.0476	0.0527	0.0809	0.0584
3_3x2cw	2, 14	0.0519	0.0557	0.0503	0.0515
3_3x2cx	2, 12	0.0516	0.0515	0.0491	0.0795
3_3x2cy	2, 10	0.0466	0.0480	0.0763	0.0521
3_3x2cz	2, 8	0.0513	0.0432	0.0480	0.0407
3_3x2da	2, 6	0.0480	0.0406	0.0735	0.0441
3_3x2db	2, 4	0.0454	0.0419	0.0979	0.0413
3_3x2dc	2, 2	0.0401	0.0397	0.0401	0.0401

Four Time Steps Input for One Step Prediction

3_4x1aa	18	0.0388	0.0388	0.0390	0.0395
3_4x1ab	16	0.0388	0.0377	0.0383	0.0373
3_4x1ac	14	0.0382	0.0390	0.0383	0.0388
3_4x1ad	12	0.0384	0.0387	0.0377	0.0381
3_4x1ae	10	0.0378	0.0387	0.0366	0.0371
3_4x1af	8	0.0368	0.0369	0.0388	0.0374

Filename	Size of ANN	RMS Error			
3_4x1ag	6	0.0383	0.0362	0.0370	0.0364
3_4x1ah	4	**0.0351**	0.0380	0.0381	0.0356
3_4x1ai	2	0.0391	0.0390	0.0390	0.0397
3_4x2aa	18, 18	0.0415	0.0499	0.0419	0.0435
3_4x2ab	18, 16	0.0422	0.0424	0.0420	0.0422
3_4x2ac	18, 14	0.0425	0.0418	0.0414	0.0409
3_4x2ad	18, 12	0.0430	0.0409	0.0414	0.0413
3_4x2ae	18, 10	0.0419	0.0418	0.0413	0.0417
3_4x2af	18, 8	0.0410	0.0418	0.0405	0.0411
3_4x2ag	18, 6	0.0403	0.0396	0.0404	0.0407
3_4x2ah	18, 4	0.0404	0.0407	0.0407	0.0408
3_4x2ai	18, 2	0.0402	0.0399	0.0407	0.0399
3_4x2aj	16, 18	0.0417	0.0418	0.0420	0.0420
3_4x2ak	16, 16	0.0420	0.0423	0.0420	0.0414
3_4x2al	16, 14	0.0420	0.0419	0.0422	0.0414
3_4x2am	16, 12	0.0415	0.0416	0.0415	0.0412
3_4x2an	16, 10	0.0415	0.0409	0.0413	0.0422
3_4x2ao	16, 8	0.0411	0.0409	0.0401	0.0424
3_4x2ap	16, 6	0.0406	0.0402	0.0412	0.0406
3_4x2aq	16, 4	0.0401	0.0400	0.0400	0.0401
3_4x2ar	16, 2	0.0399	0.0398	0.0403	0.0398
3_4x2as	14, 18	0.0414	0.0420	0.0418	0.0415
3_4x2at	14, 16	0.0426	0.0413	0.0423	0.0415
3_4x2au	14, 14	0.0420	0.0418	0.0410	0.0420
3_4x2av	14, 12	0.0410	0.0413	0.0412	0.0414
3_4x2aw	14, 10	0.0411	0.0411	0.0407	0.0415
3_4x2ax	14, 8	0.0408	0.0403	0.0402	0.0406
3_4x2ay	14, 6	0.0404	0.0415	0.0408	0.0411
3_4x2az	14, 4	0.0404	0.0398	0.0402	0.0399

Filename	Size of ANN	RMS Error			
3_4x2ba	14, 2	0.0415	0.0398	0.0400	0.0400
3_4x2bb	12, 18	0.0409	0.0408	0.0412	0.0411
3_4x2bc	12, 16	0.0417	0.0412	0.0406	0.0418
3_4x2bd	12, 14	0.0407	0.0418	0.0413	0.0406
3_4x2be	12, 12	0.0408	0.0408	0.0403	0.0409
3_4x2bf	12, 10	0.0412	0.0407	0.0406	0.0405
3_4x2bg	12, 8	0.0401	0.0407	0.0411	0.0408
3_4x2bh	12, 6	0.0407	0.0399	0.0398	0.0402
3_4x2bi	12, 4	0.0395	0.0397	0.0402	0.0406
3_4x2bj	12, 2	0.0397	0.0402	0.0399	0.0396
3_4x2bk	10, 18	0.0411	0.0415	0.0406	0.0410
3_4x2bl	10, 16	0.0402	0.0406	0.0417	0.0409
3_4x2bm	10, 14	0.0405	0.0407	0.0409	0.0421
3_4x2bn	10, 12	0.0412	0.0401	0.0405	0.0405
3_4x2bo	10, 10	0.0402	0.0403	0.0408	0.0401
3_4x2bp	10, 8	0.0406	0.0403	0.0397	0.0398
3_4x2bq	10, 6	0.0403	0.0401	0.0398	0.0401
3_4x2br	10, 4	0.0401	0.0392	0.0396	0.0401
3_4x2bs	10, 2	0.0397	0.0399	0.0396	0.0402
3_4x2bt	8, 18	0.0407	0.0404	0.0408	0.0408
3_4x2bu	8, 16	0.0409	0.0403	0.0401	0.0403
3_4x2bv	8, 14	0.0406	0.0409	0.0404	0.0401
3_4x2bw	8, 12	0.0403	0.0401	0.0400	0.0401
3_4x2bx	8, 10	0.0396	0.0398	0.0398	0.0401
3_4x2by	8, 8	0.0403	0.0398	0.0402	0.0398
3_4x2bz	8, 6	0.0393	0.0392	0.0394	0.0396
3_4x2ca	8, 4	0.0387	0.0389	0.0397	0.0406
3_4x2cb	8, 2	0.0394	0.0400	0.0392	0.0395
3_4x2cc	6, 18	0.0401	0.0405	0.0405	0.0402

Filename	Size of ANN	RMS Error			
3_4x2cd	6, 16	0.0394	0.0398	0.0395	0.0404
3_4x2ce	6, 14	0.0397	0.0401	0.0399	0.0397
3_4x2cf	6, 12	0.0395	0.0394	0.0400	0.0402
3_4x2cg	6, 10	0.0398	0.0399	0.0385	0.0399
3_4x2ch	6, 8	0.0394	0.0393	0.0395	0.0391
3_4x2ci	6, 6	0.0391	0.0386	0.0395	0.0393
3_4x2cj	6, 4	0.0389	0.0394	0.0390	0.0393
3_4x2ck	6, 2	0.0392	0.0393	0.0396	0.0396
3_4x2cl	4, 18	0.0400	0.0397	0.0396	0.0402
3_4x2cm	4, 16	0.0402	0.0399	0.0397	0.0400
3_4x2cn	4, 14	0.0387	0.0401	0.0511	0.0396
3_4x2co	4, 12	0.0397	0.0393	0.0393	0.0391
3_4x2cp	4, 10	0.0398	0.0396	0.0395	0.0393
3_4x2cq	4, 8	0.0388	**0.0383**	0.0394	0.0389
3_4x2cr	4, 6	0.0394	0.0387	0.0387	0.0395
3_4x2cs	4, 4	0.0389	**0.0383**	0.0388	0.0391
3_4x2ct	4, 2	0.0394	0.0391	0.0393	0.0391
3_4x2cu	2, 18	0.0579	0.0579	0.0576	0.0563
3_4x2cv	2, 16	0.0744	0.0567	0.0560	0.0781
3_4x2cw	2, 14	0.0567	0.0560	0.0580	0.0458
3_4x2cx	2, 12	0.0970	0.0563	0.0586	0.0561
3_4x2cy	2, 10	0.0422	0.0594	0.0810	0.0561
3_4x2cz	2, 8	0.0571	0.0811	0.0569	0.0495
3_4x2da	2, 6	0.0533	0.0461	0.0456	0.0406
3_4x2db	2, 4	0.0570	0.0525	0.0389	0.0428
3_4x2dc	2, 2	0.0390	0.0561	0.0389	0.0556

Appendix

F Results of One Step Prediction Tests for Transient 4

This appendix contains the full results of all the ANNs trained to predict Transient 4 in Section 5.2. The ANNs were all designed to predict the values of PWR variables one step ahead. Four sets of recent PWR variables were used as inputs the ANNs. These were one to four time steps. For each input set a range of one and two hidden layer ANNs were developed. The number of nodes in the hidden was varied from a maximum of 18 to a minimum of 2. The values in between were in steps of two nodes. In the case of two hidden layers both layers were varied between these limits. All the ANN training was repeated four times to avoid possible local minima. The value recorded in each case is the lowest RMS error for the training. As explained in Chapter 5 all ANNs were trained for 120,000 cycles of presentation of the training set and then a further 40,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing. In the following tables the best RMS error result for each ANN architecture is underlined while the best overall ANN for each hidden layer configuration is in bold.

One Time Step for One Step Prediction

Filename	Size of ANN	RMS Error			
4_1x1aa	18	0.0347	0.0347	0.0359	0.0362
4_1x1ab	16	0.0360	0.0366	0.0352	0.0338
4_1x1ac	14	0.0352	0.0347	0.0359	0.0352
4_1x1ad	12	0.0348	0.0349	0.0331	0.0338
4_1x1ae	10	0.0341	0.0344	0.0332	0.0344
4_1x1af	8	0.0333	0.0337	0.0333	0.0333
4_1x1ag	6	0.0335	0.0345	0.0336	0.0321
4_1x1ah	4	0.0331	0.0343	0.0354	0.0377
4_1x1ai	2	0.0393	0.0423	0.0424	0.0422
4_1x2aa	18, 18	0.0433	0.0435	0.0435	0.0434
4_1x2ab	18, 16	0.0424	0.0463	0.0427	0.0431
4_1x2ac	18, 14	0.0432	0.0445	0.0426	0.0425
4_1x2ad	18, 12	0.0418	0.0427	0.0429	0.0428
4_1x2ae	18, 10	0.0424	0.0427	0.0426	0.0431
4_1x2af	18, 8	0.0424	0.0418	0.0425	0.0422
4_1x2ag	18, 6	0.0407	0.0424	0.0412	0.0427
4_1x2ah	18, 4	0.0426	0.0427	0.0415	0.0406
4_1x2ai	18, 2	0.0427	0.0426	0.0424	0.0426
4_1x2aj	16, 18	0.0453	0.0427	0.0432	0.0448
4_1x2ak	16, 16	0.0425	0.0425	0.0430	0.0436
4_1x2al	16, 14	0.0427	0.0431	0.0424	0.0433
4_1x2am	16, 12	0.0427	0.0427	0.0423	0.0427
4_1x2an	16, 10	0.0413	0.0422	0.0425	0.0428
4_1x2ao	16, 8	0.0422	0.0419	0.0419	0.0423
4_1x2ap	16, 6	0.0425	0.0419	0.0423	0.0426
4_1x2aq	16, 4	0.0425	0.0415	0.0419	0.0417
4_1x2ar	16, 2	0.0419	0.0424	0.0429	0.0421

Filename	Size of ANN	RMS Error			
4_1x2as	14, 18	0.0423	0.0431	0.0422	0.0433
4_1x2at	14, 16	0.0422	0.0425	0.0423	0.0422
4_1x2au	14, 14	0.0423	0.0422	0.0425	0.0429
4_1x2av	14, 12	0.0416	0.0416	0.0422	0.0422
4_1x2aw	14, 10	0.0422	0.0421	0.0420	0.0424
4_1x2ax	14, 8	0.0421	0.0424	0.0419	0.0426
4_1x2ay	14, 6	0.0431	0.0418	0.0417	0.0417
4_1x2az	14, 4	0.0416	0.0414	0.0401	0.0410
4_1x2ba	14, 2	0.0427	0.0425	0.0425	0.0425
4_1x2bb	12, 18	0.0419	0.0423	0.0426	0.0421
4_1x2bc	12, 16	0.0419	0.0426	0.0416	0.0422
4_1x2bd	12, 14	0.0426	0.0417	0.0423	0.0422
4_1x2be	12, 12	0.0415	0.0422	0.0425	0.0411
4_1x2bf	12, 10	0.0432	0.0417	0.0427	0.0418
4_1x2bg	12, 8	0.0419	0.0430	0.0421	0.0420
4_1x2bh	12, 6	0.0420	0.0429	0.0403	0.0411
4_1x2bi	12, 4	0.0418	0.0424	0.0422	0.0415
4_1x2bj	12, 2	0.0430	0.0427	0.0426	0.0426
4_1x2bk	10, 18	0.0424	0.0425	0.0425	0.0413
4_1x2bl	10, 16	0.0422	0.0421	0.0420	0.0417
4_1x2bm	10, 14	0.0418	0.0419	0.0424	0.0420
4_1x2bn	10, 12	0.0414	0.0415	0.0433	0.0425
4_1x2bo	10, 10	0.0419	0.0414	0.0413	0.0411
4_1x2bp	10, 8	0.0424	0.0416	0.0418	0.0418
4_1x2bq	10, 6	0.0413	0.0407	0.0415	0.0404
4_1x2br	10, 4	0.0425	0.0416	0.0420	0.0405
4_1x2bs	10, 2	0.0425	0.0431	0.0431	0.0415

Filename	Size of ANN	RMS Error			
4_1x2bt	8, 18	<u>0.0418</u>	<u>0.0418</u>	0.0423	0.0424
4_1x2bu	8, 16	0.0416	0.0417	0.0417	<u>0.0413</u>
4_1x2bv	8, 14	0.0418	0.0415	<u>0.0414</u>	0.0424
4_1x2bw	8, 12	<u>0.0406</u>	0.0413	0.0420	0.0426
4_1x2bx	8, 10	0.0409	0.0416	<u>0.0404</u>	0.0421
4_1x2by	8, 8	0.0418	<u>0.0409</u>	0.0415	<u>0.0409</u>
4_1x2bz	8, 6	<u>0.0411</u>	0.0427	0.0415	0.0413
4_1x2ca	8, 4	0.0421	0.0424	<u>0.0409</u>	0.0418
4_1x2cb	8, 2	0.0427	0.0431	0.0416	<u>0.0406</u>
4_1x2cc	6, 18	0.0414	<u>0.0405</u>	0.0416	0.0422
4_1x2cd	6, 16	0.0413	<u>0.0411</u>	0.0424	0.0420
4_1x2ce	6, 14	0.0417	<u>0.0409</u>	0.0415	0.0425
4_1x2cf	6, 12	0.0432	0.0430	<u>0.0408</u>	0.0421
4_1x2cg	6, 10	<u>0.0418</u>	0.0424	0.0427	0.0423
4_1x2ch	6, 8	<u>0.0416</u>	0.0418	0.0417	0.0426
4_1x2ci	6, 6	0.0424	0.0416	0.0420	<u>0.0411</u>
4_1x2cj	6, 4	0.0431	0.0411	<u>0.0405</u>	0.0397
4_1x2ck	6, 2	0.0423	0.0432	<u>0.0413</u>	0.0426
4_1x2cl	4, 18	<u>0.0415</u>	0.0418	0.0440	0.0434
4_1x2cm	4, 16	0.0429	0.0429	0.0429	<u>0.0427</u>
4_1x2cn	4, 14	0.0414	0.0419	0.0426	<u>0.0402</u>
4_1x2co	4, 12	0.0411	0.0402	<u>0.0401</u>	0.0409
4_1x2cp	4, 10	0.0407	0.0398	0.0403	<u>0.0396</u>
4_1x2cq	4, 8	0.0417	0.0425	<u>0.0406</u>	0.0408
4_1x2cr	4, 6	0.0414	0.0400	<u>0.0399</u>	0.0425
4_1x2cs	4, 4	<u>0.0397</u>	0.0431	0.0401	0.0410
4_1x2ct	4, 2	0.0428	0.0439	<u>0.0425</u>	0.0432

Filename	Size of ANN	RMS Error			
4_1x2cu	2, 18	0.0576	0.0655	0.0809	0.0584
4_1x2cv	2, 16	0.0659	0.0524	0.0764	0.0748
4_1x2cw	2, 14	0.0763	0.0819	0.0476	0.0679
4_1x2cx	2, 12	0.0817	0.0550	0.0671	0.0817
4_1x2cy	2, 10	0.0576	0.0466	0.0659	0.0544
4_1x2cz	2, 8	0.0522	0.0785	0.0486	0.0561
4_1x2da	2, 6	0.0724	0.0425	0.0544	0.0403
4_1x2db	2, 4	0.0418	0.0442	0.0449	0.0746
4_1x2dc	2, 2	0.0411	0.0434	0.0598	0.0412

Two Time Steps for One Step Prediction

4_2x1aa	18	0.0367	0.0347	0.0349	0.0345
4_2x1ab	16	0.0340	0.0360	0.0354	0.0340
4_2x1ac	14	0.0349	0.0336	0.0355	0.0367
4_2x1ad	12	0.0346	0.0367	0.0370	0.0348
4_2x1ae	10	0.0340	0.0343	0.0339	0.0348
4_2x1af	8	0.0351	0.0334	0.0363	0.0341
4_2x1ag	6	0.0339	0.0306	0.0349	0.0309
4_2x1ah	4	0.0295	0.0379	0.0329	0.0341
4_2x1ai	2	0.0389	0.0390	0.0389	0.0422
4_2x2aa	18, 18	0.0436	0.0435	0.0426	0.0413
4_2x2ab	18, 16	0.0424	0.0422	0.0419	0.0418
4_2x2ac	18, 14	0.0446	0.0409	0.0416	0.0407
4_2x2ad	18, 12	0.0406	0.0413	0.0405	0.0411
4_2x2ae	18, 10	0.0410	0.0417	0.0405	0.0412
4_2x2af	18, 8	0.0400	0.0411	0.0410	0.0408
4_2x2ag	18, 6	0.0400	0.0407	0.0402	0.0409
4_2x2ah	18, 4	0.0403	0.0408	0.0399	0.0406

Filename	Size of ANN	RMS Error			
4_2x2ai	18, 2	0.0411	<u>0.0399</u>	0.0407	0.0406
4_2x2aj	16, 18	<u>0.0412</u>	0.0420	0.0414	0.0421
4_2x2ak	16, 16	0.0417	<u>0.0414</u>	0.0417	0.0439
4_2x2al	16, 14	0.0417	<u>0.0414</u>	<u>0.0414</u>	0.0422
4_2x2am	16, 12	0.0410	0.0412	0.0414	<u>0.0402</u>
4_2x2an	16, 10	0.0414	0.0422	0.0406	<u>0.0404</u>
4_2x2ao	16, 8	0.0409	0.0424	0.0401	<u>0.0398</u>
4_2x2ap	16, 6	0.0407	0.0406	0.0408	<u>0.0398</u>
4_2x2aq	16, 4	0.0411	0.0401	<u>0.0395</u>	0.0399
4_2x2ar	16, 2	0.0411	<u>0.0398</u>	0.0411	0.0403
4_2x2as	14, 18	<u>0.0407</u>	0.0415	0.0414	0.0425
4_2x2at	14, 16	0.0413	0.0415	<u>0.0405</u>	0.0408
4_2x2au	14, 14	0.0413	0.0420	<u>0.0408</u>	0.0415
4_2x2av	14, 12	0.0409	0.0414	<u>0.0402</u>	0.0411
4_2x2aw	14, 10	0.0409	0.0415	0.0404	<u>0.0398</u>
4_2x2ax	14, 8	0.0403	0.0406	<u>0.0402</u>	0.0410
4_2x2ay	14, 6	0.0401	0.0411	<u>0.0395</u>	0.0409
4_2x2az	14, 4	0.0402	<u>0.0399</u>	0.0401	0.0404
4_2x2ba	14, 2	0.0407	<u>0.0400</u>	0.0411	0.0404
4_2x2bb	12, 18	0.0408	0.0411	0.0408	<u>0.0407</u>
4_2x2bc	12, 16	<u>0.0406</u>	0.0418	0.0411	<u>0.0406</u>
4_2x2bd	12, 14	0.0407	0.0406	0.0408	<u>0.0403</u>
4_2x2be	12, 12	0.0409	0.0409	<u>0.0406</u>	0.0409
4_2x2bf	12, 10	0.0400	0.0405	<u>0.0399</u>	0.0401
4_2x2bg	12, 8	<u>0.0395</u>	0.0408	0.0401	0.0405
4_2x2bh	12, 6	0.0404	<u>0.0402</u>	0.0405	0.0405
4_2x2bi	12, 4	0.0402	0.0406	<u>0.0391</u>	0.0392
4_2x2bj	12, 2	0.0404	<u>0.0396</u>	0.0407	0.0407
4_2x2bk	10, 18	0.0423	0.0410	<u>0.0409</u>	<u>0.0409</u>

Filename	Size of ANN	RMS Error			
4_2x2bl	10, 16	0.0407	0.0409	0.0409	0.0400
4_2x2bm	10, 14	0.0413	0.0421	0.0405	0.0401
4_2x2bn	10, 12	0.0404	0.0405	0.0402	0.0397
4_2x2bo	10, 10	0.0404	0.0401	0.0409	0.0402
4_2x2bp	10, 8	0.0391	0.0398	0.0386	0.0409
4_2x2bq	10, 6	0.0397	0.0401	0.0395	0.0393
4_2x2br	10, 4	0.0393	0.0401	0.0393	0.0395
4_2x2bs	10, 2	0.0411	0.0402	0.0404	0.0409
4_2x2bt	8, 18	0.0406	0.0408	0.0404	0.0411
4_2x2bu	8, 16	0.0406	0.0403	0.0407	0.0412
4_2x2bv	8, 14	0.0402	0.0401	0.0399	0.0397
4_2x2bw	8, 12	0.0402	0.0401	0.0400	0.0403
4_2x2bx	8, 10	0.0396	0.0401	0.0391	0.0403
4_2x2by	8, 8	0.0392	0.0398	0.0396	0.0403
4_2x2bz	8, 6	0.0398	0.0396	0.0398	0.0391
4_2x2ca	8, 4	0.0392	0.0406	0.0403	0.0388
4_2x2cb	8, 2	0.0392	0.0395	0.0389	0.0399
4_2x2cc	6, 18	0.0404	0.0402	0.0397	0.0396
4_2x2cd	6, 16	0.0399	0.0404	0.0398	0.0408
4_2x2ce	6, 14	0.0396	0.0397	0.0399	0.0400
4_2x2cf	6, 12	0.0399	0.0402	0.0395	0.0395
4_2x2cg	6, 10	0.0395	0.0399	0.0392	0.0395
4_2x2ch	6, 8	0.0393	0.0391	0.0397	0.0393
4_2x2ci	6, 6	0.0385	0.0393	0.0390	0.0402
4_2x2cj	6, 4	0.0393	0.0393	0.0381	0.0358
4_2x2ck	6, 2	0.0404	0.0396	0.0402	0.0410
4_2x2cl	4, 18	0.0395	0.0402	0.0387	0.0461
4_2x2cm	4, 16	0.0396	0.0400	0.0405	0.0399
4_2x2cn	4, 14	0.0393	0.0396	0.0402	0.0404

Filename	Size of ANN	RMS Error			
4_2x2co	4, 12	0.0406	0.0391	0.0408	0.0426
4_2x2cp	4, 10	0.0391	0.0393	0.0386	0.0397
4_2x2cq	4, 8	0.0392	0.0389	0.0390	0.0380
4_2x2cr	4, 6	0.0399	0.0395	0.0393	0.0390
4_2x2cs	4, 4	0.0384	0.0391	0.0406	0.0391
4_2x2ct	4, 2	0.0403	0.0391	0.0402	0.0389
4_2x2cu	2, 18	0.0736	0.0563	0.0825	0.0526
4_2x2cv	2, 16	0.0753	0.0781	0.0795	0.0765
4_2x2cw	2, 14	0.0830	0.0458	0.0822	0.0550
4_2x2cx	2, 12	0.0683	0.0561	0.0802	0.0732
4_2x2cy	2, 10	0.0741	0.0561	0.0678	0.0716
4_2x2cz	2, 8	0.0792	0.0495	0.0588	0.0487
4_2x2da	2, 6	0.0714	0.0406	0.0813	0.0388
4_2x2db	2, 4	0.0715	0.0428	0.0785	0.0484
4_2x2dc	2, 2	0.0401	0.0556	0.0421	0.0408

Three Time Steps for One Step Prediction

4_3x1aa	18	0.0301	0.0300	0.0298	0.0296
4_3x1ab	16	0.0296	0.0294	0.0295	0.0299
4_3x1ac	14	0.0291	0.0294	0.0292	0.0296
4_3x1ad	12	0.0286	0.0292	0.0296	0.0284
4_3x1ae	10	0.0282	0.0277	0.0281	0.0283
4_3x1af	8	0.0283	0.0275	0.0272	0.0275
4_3x1ag	6	0.0270	0.0280	0.0272	0.0266
4_3x1ah	4	0.0259	0.0264	0.0265	0.0279
4_3x1ai	2	0.0365	0.0362	0.0362	0.0362
4_3x2aa	18, 18	0.0392	0.0411	0.0404	0.0426
4_3x2ab	18, 16	0.0397	0.0393	0.0427	0.0417

Filename	Size of ANN	RMS Error			
4_3x2ac	18, 14	0.0397	0.0394	0.0400	0.0404
4_3x2ad	18, 12	0.0409	0.0393	0.0394	0.0395
4_3x2ae	18, 10	0.0394	0.0389	0.0386	0.0394
4_3x2af	18, 8	0.0403	0.0396	0.0384	0.0393
4_3x2ag	18, 6	0.0383	0.0384	0.0383	0.0379
4_3x2ah	18, 4	0.0379	0.0380	0.0380	0.0378
4_3x2ai	18, 2	0.0386	0.0391	0.0386	0.0386
4_3x2aj	16, 18	0.0404	0.0405	0.0413	0.0410
4_3x2ak	16, 16	0.0404	0.0414	0.0410	0.0400
4_3x2al	16, 14	0.0409	0.0389	0.0394	0.0394
4_3x2am	16, 12	0.0396	0.0393	0.0389	0.0391
4_3x2an	16, 10	0.0380	0.0391	0.0377	0.0398
4_3x2ao	16, 8	0.0385	0.0397	0.0386	0.0386
4_3x2ap	16, 6	0.0378	0.0384	0.0388	0.0377
4_3x2aq	16, 4	0.0380	0.0375	0.0387	0.0380
4_3x2ar	16, 2	0.0387	0.0382	0.0384	0.0382
4_3x2as	14, 18	0.0405	0.0397	0.0398	0.0402
4_3x2at	14, 16	0.0397	0.0393	0.0417	0.0403
4_3x2au	14, 14	0.0389	0.0394	0.0395	0.0387
4_3x2av	14, 12	0.0400	0.0391	0.0385	0.0428
4_3x2aw	14, 10	0.0382	0.0386	0.0387	0.0377
4_3x2ax	14, 8	0.0387	0.0383	0.0394	0.0383
4_3x2ay	14, 6	0.0377	0.0370	0.0384	0.0377
4_3x2az	14, 4	0.0378	0.0380	0.0374	0.0376
4_3x2ba	14, 2	0.0380	0.0379	0.0379	0.0385
4_3x2bb	12, 18	0.0391	0.0395	0.0394	0.0404
4_3x2bc	12, 16	0.0393	0.0390	0.0393	0.0387
4_3x2bd	12, 14	0.0385	0.0393	0.0387	0.0385
4_3x2be	12, 12	0.0387	0.0391	0.0378	0.0394

Filename	Size of ANN	RMS Error			
4_3x2bf	12, 10	0.0378	0.0386	0.0379	0.0381
4_3x2bg	12, 8	0.0380	0.0381	0.0373	0.0370
4_3x2bh	12, 6	0.0380	0.0374	0.0381	0.0377
4_3x2bi	12, 4	0.0374	0.0376	0.0377	0.0374
4_3x2bj	12, 2	0.0383	0.0382	0.0385	0.0387
4_3x2bk	10, 18	0.0382	0.0387	0.0398	0.0397
4_3x2bl	10, 16	0.0397	0.0386	0.0386	0.0380
4_3x2bm	10, 14	0.0380	0.0384	0.0398	0.0385
4_3x2bn	10, 12	0.0382	0.0381	0.0386	0.0378
4_3x2bo	10, 10	0.0386	0.0377	0.0390	0.0376
4_3x2bp	10, 8	0.0374	0.0377	0.0383	0.0378
4_3x2bq	10, 6	0.0376	0.0391	0.0376	0.0377
4_3x2br	10, 4	0.0346	0.0363	0.0375	0.0371
4_3x2bs	10, 2	0.0380	0.0370	0.0384	0.0383
4_3x2bt	8, 18	0.0387	0.0383	0.0380	0.0391
4_3x2bu	8, 16	0.0380	0.0383	0.0382	0.0379
4_3x2bv	8, 14	0.0384	0.0382	0.0375	0.0376
4_3x2bw	8, 12	0.0380	0.0375	0.0388	0.0374
4_3x2bx	8, 10	0.0373	0.0370	0.0378	0.0378
4_3x2by	8, 8	0.0369	0.0377	0.0373	0.0369
4_3x2bz	8, 6	0.0360	0.0373	0.0379	0.0373
4_3x2ca	8, 4	0.0349	0.0373	0.0368	0.0380
4_3x2cb	8, 2	0.0382	0.0379	0.0382	0.0376
4_3x2cc	6, 18	0.0381	0.0390	0.0385	0.0382
4_3x2cd	6, 16	0.0375	0.0376	0.0393	0.0381
4_3x2ce	6, 14	0.0373	0.0385	0.0382	0.0382
4_3x2cf	6, 12	0.0381	0.0372	0.0372	0.0374
4_3x2cg	6, 10	0.0369	0.0378	0.0378	0.0370
4_3x2ch	6, 8	0.0359	0.0361	0.0357	0.0359

Filename	Size of ANN	RMS Error			
4_3x2ci	6, 6	0.0362	0.0364	0.0346	0.0344
4_3x2cj	6, 4	0.0360	0.0372	0.0364	0.0339
4_3x2ck	6, 2	0.0370	0.0379	0.0382	0.0379
4_3x2cl	4, 18	0.0372	0.0369	0.0381	0.0376
4_3x2cm	4, 16	0.0382	0.0377	0.0393	0.0387
4_3x2cn	4, 14	0.0367	0.0378	0.0366	0.0386
4_3x2co	4, 12	0.0383	0.0377	0.0368	0.0409
4_3x2cp	4, 10	0.0378	0.0440	0.0369	0.0368
4_3x2cq	4, 8	0.0368	0.0371	0.0373	0.0377
4_3x2cr	4, 6	0.0368	0.0352	0.0361	0.0367
4_3x2cs	4, 4	0.0371	0.0344	0.0359	0.0389
4_3x2ct	4, 2	0.0383	0.0763	0.0381	0.0370
4_3x2cu	2, 18	0.0780	0.0750	0.0828	0.0774
4_3x2cv	2, 16	0.0580	0.0560	0.0557	0.0822
4_3x2cw	2, 14	0.0845	0.0764	0.0832	0.0855
4_3x2cx	2, 12	0.0878	0.0760	0.0849	0.0753
4_3x2cy	2, 10	0.0533	0.0528	0.0819	0.0543
4_3x2cz	2, 8	0.0755	0.0755	0.0417	0.0699
4_3x2da	2, 6	0.0846	0.0753	0.0687	0.0785
4_3x2db	2, 4	0.0383	0.0523	0.0700	0.0526
4_3x2dc	2, 2	0.0707	0.0801	0.0808	0.0856

Four Time Steps for One Step Prediction

4_4x1aa	18	0.0401	0.0402	0.0408	0.0399
4_4x1ab	16	0.0393	0.0393	0.0398	0.0411
4_4x1ac	14	0.0393	0.0402	0.0395	0.0398
4_4x1ad	12	0.0395	0.0390	0.0398	0.0405
4_4x1ae	10	0.0401	0.0407	0.0385	0.0397

Filename	Size of ANN	RMS Error			
4_4x1af	8	<u>0.0371</u>	0.0381	0.0380	0.0386
4_4x1ag	6	<u>0.0375</u>	0.0381	0.0413	0.0401
4_4x1ah	4	0.0409	0.0377	0.0362	<u>0.0359</u>
4_4x1ai	2	<u>0.0422</u>	0.0428	0.0423	0.0427
4_4x2aa	18, 18	0.0463	<u>0.0448</u>	0.0453	0.0451
4_4x2ab	18, 16	0.0458	0.0453	0.0446	<u>0.0445</u>
4_4x2ac	18, 14	0.0466	<u>0.0449</u>	0.0452	0.0466
4_4x2ad	18, 12	0.0447	<u>0.0445</u>	0.0450	0.0450
4_4x2ae	18, 10	<u>0.0435</u>	0.0442	0.0450	0.0436
4_4x2af	18, 8	<u>0.0439</u>	0.0447	0.0444	0.0447
4_4x2ag	18, 6	<u>0.0432</u>	0.0440	0.0449	0.0437
4_4x2ah	18, 4	0.0447	<u>0.0427</u>	0.0437	0.0431
4_4x2ai	18, 2	0.0441	<u>0.0438</u>	0.0440	0.0439
4_4x2aj	16, 18	0.0463	0.0454	0.0462	<u>0.0453</u>
4_4x2ak	16, 16	0.0450	0.0453	<u>0.0447</u>	0.0456
4_4x2al	16, 14	<u>0.0445</u>	0.0479	0.0461	0.0452
4_4x2am	16, 12	0.0447	0.0475	0.0444	<u>0.0443</u>
4_4x2an	16, 10	<u>0.0438</u>	0.0441	0.0446	0.0452
4_4x2ao	16, 8	0.0441	<u>0.0438</u>	0.0442	<u>0.0438</u>
4_4x2ap	16, 6	0.0440	<u>0.0432</u>	0.0438	0.0442
4_4x2aq	16, 4	<u>0.0424</u>	0.0438	0.0436	0.0444
4_4x2ar	16, 2	0.0439	0.0439	0.0441	<u>0.0438</u>
4_4x2as	14, 18	0.0448	<u>0.0443</u>	0.0450	0.0446
4_4x2at	14, 16	0.0466	0.0445	<u>0.0442</u>	0.0446
4_4x2au	14, 14	0.0444	0.0450	<u>0.0442</u>	0.0454
4_4x2av	14, 12	0.0455	<u>0.0436</u>	0.0437	0.0445
4_4x2aw	14, 10	0.0442	0.0441	<u>0.0439</u>	<u>0.0439</u>
4_4x2ax	14, 8	0.0443	0.0450	0.0431	<u>0.0427</u>
4_4x2ay	14, 6	0.0436	0.0445	<u>0.0426</u>	0.0439

Filename	Size of ANN	RMS Error			
4_4x2az	14, 4	0.0422	0.0438	0.0441	0.0428
4_4x2ba	14, 2	0.0439	0.0443	0.0442	0.0439
4_4x2bb	12, 18	0.0443	0.0439	0.0444	0.0450
4_4x2bc	12, 16	0.0442	0.0446	0.0440	0.0444
4_4x2bd	12, 14	0.0444	0.0437	0.0441	0.0444
4_4x2be	12, 12	0.0442	0.0444	0.0437	0.0433
4_4x2bf	12, 10	0.0436	0.0442	0.0431	0.0432
4_4x2bg	12, 8	0.0434	0.0433	0.0430	0.0438
4_4x2bh	12, 6	0.0424	0.0433	0.0420	0.0414
4_4x2bi	12, 4	0.0408	0.0412	0.0417	0.0433
4_4x2bj	12, 2	0.0438	0.0435	0.0438	0.0441
4_4x2bk	10, 18	0.0442	0.0438	0.0439	0.0447
4_4x2bl	10, 16	0.0435	0.0442	0.0441	0.0450
4_4x2bm	10, 14	0.0437	0.0438	0.0438	0.0438
4_4x2bn	10, 12	0.0433	0.0436	0.0435	0.0432
4_4x2bo	10, 10	0.0435	0.0441	0.0427	0.0434
4_4x2bp	10, 8	0.0437	0.0432	0.0439	0.0429
4_4x2bq	10, 6	0.0431	0.0434	0.0422	0.0430
4_4x2br	10, 4	0.0437	0.0432	0.0430	0.0418
4_4x2bs	10, 2	0.0439	0.0439	0.0438	0.0440
4_4x2bt	8, 18	0.0433	0.0441	0.0438	0.0446
4_4x2bu	8, 16	0.0440	0.0434	0.0429	0.0444
4_4x2bv	8, 14	0.0438	0.0431	0.0435	0.0434
4_4x2bw	8, 12	0.0431	0.0433	0.0428	0.0430
4_4x2bx	8, 10	0.0419	0.0433	0.0443	0.0435
4_4x2by	8, 8	0.0430	0.0421	0.0431	0.0426
4_4x2bz	8, 6	0.0430	0.0413	0.0429	0.0412
4_4x2ca	8, 4	0.0431	0.0416	0.0433	0.0428
4_4x2cb	8, 2	0.0435	0.0436	0.0438	0.0442

Filename	Size of ANN	RMS Error			
4_4x2cc	6, 18	0.0437	0.0431	0.0439	0.0439
4_4x2cd	6, 16	0.0435	0.0432	0.0491	0.0439
4_4x2ce	6, 14	0.0421	0.0430	0.0434	0.0437
4_4x2cf	6, 12	0.0421	0.0419	0.0430	0.0427
4_4x2cg	6, 10	0.0429	0.0427	0.0416	0.0432
4_4x2ch	6, 8	0.0393	0.0433	0.0426	0.0434
4_4x2ci	6, 6	0.0388	0.0408	0.0414	0.0434
4_4x2cj	6, 4	0.0421	0.0408	0.0406	0.0432
4_4x2ck	6, 2	0.0441	0.0435	0.0439	0.0437
4_4x2cl	4, 18	0.0437	0.0430	0.0434	0.0433
4_4x2cm	4, 16	0.0414	0.0458	0.0435	0.0428
4_4x2cn	4, 14	0.0443	0.0413	0.0429	0.0427
4_4x2co	4, 12	0.0425	0.0435	0.0421	0.0413
4_4x2cp	4, 10	0.0432	0.0416	0.0419	0.0414
4_4x2cq	4, 8	0.0433	0.0435	0.0431	0.0431
4_4x2cr	4, 6	0.0412	0.0417	0.0410	0.0436
4_4x2cs	4, 4	0.0391	0.0435	0.0574	0.0412
4_4x2ct	4, 2	0.0437	0.0435	0.0429	0.0435
4_4x2cu	2, 18	0.0922	0.0915	0.0814	0.0795
4_4x2cv	2, 16	0.0834	0.0885	0.0779	0.0798
4_4x2cw	2, 14	0.0873	0.0674	0.0760	0.0885
4_4x2cx	2, 12	0.0742	0.0704	0.0792	0.0887
4_4x2cy	2, 10	0.0797	0.0849	0.0587	0.0494
4_4x2cz	2, 8	0.0562	0.0489	0.0856	0.0836
4_4x2da	2, 6	0.0492	0.0738	0.0472	0.0411
4_4x2db	2, 4	0.0544	0.0751	0.0434	0.0459
4_4x2dc	2, 2	0.0428	0.0914	0.0861	0.0470

Appendix

G Results of One Step Prediction Tests for Transient 5

This appendix contains the full results of all the ANNs trained to predict Transient 5 in Section 5.2. The ANNs were all designed to predict the values of PWR variables one step ahead. Four sets of recent PWR variables were used as inputs the ANNs. These were one to four time steps. For each input set a range of one and two hidden layer ANNs were developed. The number of nodes in the hidden was varied from a maximum of 18 to a minimum of 2. The values in between were in steps of two nodes. In the case of two hidden layers both layers were varied between these limits. All the ANN training was repeated four times to avoid possible local minima. The value recorded in each case is the lowest RMS error for the training. As explained in Chapter 5 all ANNs were trained for 120,000 cycles of presentation of the training set and then a further 40,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing. In the following tables the best RMS error result for each ANN architecture is underlined while the best overall ANN for each hidden layer configuration is in bold.

One Time Step Input for One Step Prediction

Filename	Size of ANN	RMS Error			
5_1x1aa	18	0.0344	0.0345	0.0354	0.0347
5_1x1ab	16	0.0399	0.0361	0.0351	0.0342
5_1x1ac	14	0.0336	0.0349	0.0348	0.0341
5_1x1ad	12	0.0335	0.0336	0.0348	0.0340
5_1x1ae	10	0.0340	0.0342	0.0345	0.0353
5_1x1af	8	0.0346	0.0341	0.0359	0.0334
5_1x1ag	6	0.0336	0.0326	0.0343	0.0325
5_1x1ah	4	0.0326	0.0337	0.0334	0.0333
5_1x1ai	2	0.0381	0.0389	0.0381	0.0380
5_1x2aa	18, 18	0.0395	0.0399	0.0398	0.0406
5_1x2ab	18, 16	0.0397	0.0419	0.0390	0.0393
5_1x2ac	18, 14	0.0393	0.0399	0.0397	0.0392
5_1x2ad	18, 12	0.0392	0.0392	0.0391	0.0396
5_1x2ae	18, 10	0.0390	0.0390	0.0390	0.0390
5_1x2af	18, 8	0.0390	0.0389	0.0389	0.0389
5_1x2ag	18, 6	0.0386	0.0387	0.0386	0.0387
5_1x2ah	18, 4	0.0384	0.0387	0.0385	0.0385
5_1x2ai	18, 2	0.0386	0.0383	0.0384	0.0386
5_1x2aj	16, 18	0.0393	0.0392	0.0395	0.0393
5_1x2ak	16, 16	0.0392	0.0394	0.0396	0.0391
5_1x2al	16, 14	0.0394	0.0391	0.0390	0.0393
5_1x2am	16, 12	0.0396	0.0397	0.0390	0.0388
5_1x2an	16, 10	0.0390	0.0393	0.0389	0.0390
5_1x2ao	16, 8	0.0388	0.0388	0.0390	0.0386
5_1x2ap	16, 6	0.0391	0.0386	0.0385	0.0388
5_1x2aq	16, 4	0.0384	0.0383	0.0385	0.0381
5_1x2ar	16, 2	0.0386	0.0385	0.0384	0.0387

Filename	Size of ANN	RMS Error			
5_1x2as	14, 18	0.0395	0.0391	0.0392	0.0394
5_1x2at	14, 16	0.0394	0.0392	0.0389	0.0389
5_1x2au	14, 14	0.0388	0.0392	0.0390	0.0387
5_1x2av	14, 12	0.0388	0.0388	0.0389	0.0385
5_1x2aw	14, 10	0.0388	0.0387	0.0389	0.0390
5_1x2ax	14, 8	0.0388	0.0390	0.0389	0.0386
5_1x2ay	14, 6	0.0386	0.0387	0.0387	0.0385
5_1x2az	14, 4	0.0384	0.0383	0.0385	0.0381
5_1x2ba	14, 2	0.0385	0.0385	0.0384	0.0389
5_1x2bb	12, 18	0.0388	0.0389	0.0393	0.0391
5_1x2bc	12, 16	0.0391	0.0391	0.0391	0.0392
5_1x2bd	12, 14	0.0386	0.0387	0.0392	0.0390
5_1x2be	12, 12	0.0390	0.0387	0.0388	0.0388
5_1x2bf	12, 10	0.0385	0.0386	0.0388	0.0388
5_1x2bg	12, 8	0.0383	0.0389	0.0387	0.0384
5_1x2bh	12, 6	0.0383	0.0381	0.0384	0.0383
5_1x2bi	12, 4	0.0385	0.0382	0.0380	0.0378
5_1x2bj	12, 2	0.0388	0.0385	0.0386	0.0383
5_1x2bk	10, 18	0.0388	0.0387	0.0388	0.0392
5_1x2bl	10, 16	0.0386	0.0389	0.0387	0.0389
5_1x2bm	10, 14	0.0386	0.0386	0.0391	0.0384
5_1x2bn	10, 12	0.0384	0.0392	0.0389	0.0389
5_1x2bo	10, 10	0.0387	0.0385	0.0388	0.0385
5_1x2bp	10, 8	0.0385	0.0386	0.0383	0.0385
5_1x2bq	10, 6	0.0376	0.0378	0.0379	0.0379
5_1x2br	10, 4	0.0384	0.0377	0.0377	0.0387
5_1x2bs	10, 2	0.0383	0.0384	0.0385	0.0388

Filename	Size of ANN	RMS Error			
5_1x2bt	8, 18	0.0389	0.0388	0.0386	0.0387
5_1x2bu	8, 16	0.0386	0.0389	0.0390	0.0389
5_1x2bv	8, 14	0.0387	0.0388	0.0385	0.0385
5_1x2bw	8, 12	0.0382	0.0387	0.0383	0.0384
5_1x2bx	8, 10	0.0386	0.0388	0.0383	0.0385
5_1x2by	8, 8	0.0388	0.0386	0.0383	0.0385
5_1x2bz	8, 6	0.0386	0.0382	0.0385	0.0378
5_1x2ca	8, 4	0.0380	0.0382	0.0385	0.0377
5_1x2cb	8, 2	0.0385	0.0384	0.0381	0.0383
5_1x2cc	6, 18	0.0390	0.0389	0.0385	0.0385
5_1x2cd	6, 16	0.0389	0.0387	0.0384	0.0385
5_1x2ce	6, 14	0.0390	0.0389	0.0389	0.0381
5_1x2cf	6, 12	0.0383	0.0383	0.0381	0.0387
5_1x2cg	6, 10	0.0382	0.0382	0.0384	0.0385
5_1x2ch	6, 8	0.0381	0.0387	0.0384	0.0391
5_1x2ci	6, 6	0.0384	0.0377	0.0380	0.0387
5_1x2cj	6, 4	0.0381	0.0385	0.0380	0.0384
5_1x2ck	6, 2	0.0387	0.0384	0.0384	0.0383
5_1x2cl	4, 18	0.0390	0.0599	0.0389	0.0391
5_1x2cm	4, 16	0.0378	0.0387	0.0393	0.0386
5_1x2cn	4, 14	0.0383	0.0384	0.0393	0.0390
5_1x2co	4, 12	0.0391	0.0388	0.0382	0.0380
5_1x2cp	4, 10	0.0380	0.0386	0.0377	0.0385
5_1x2cq	4, 8	0.0383	0.0387	0.0384	0.0383
5_1x2cr	4, 6	0.0380	0.0378	0.0380	0.0384
5_1x2cs	4, 4	0.0383	0.0372	0.0376	0.0374
5_1x2ct	4, 2	0.0387	0.0386	0.0386	0.0384

Filename	Size of ANN	RMS Error			
5_1x2cu	2, 18	0.0583	0.0568	0.0556	0.0569
5_1x2cv	2, 16	0.0438	0.0545	0.0767	0.0581
5_1x2cw	2, 14	0.0614	0.0574	0.0576	0.0577
5_1x2cx	2, 12	0.0573	0.0551	0.0587	0.0422
5_1x2cy	2, 10	0.0473	0.0405	0.0581	0.0567
5_1x2cz	2, 8	0.0558	0.0465	0.0413	0.0434
5_1x2da	2, 6	0.0388	0.0394	0.0430	0.0564
5_1x2db	2, 4	0.0384	0.0416	0.0386	0.0461
5_1x2dc	2, 2	0.0433	0.0387	0.0781	0.0716

Two Time Steps Input for One Step Prediction

5_2x1aa	18	0.0367	0.0362	0.0343	0.0357
5_2x1ab	16	0.0355	0.0364	0.0339	0.0376
5_2x1ac	14	0.0356	0.0344	0.0344	0.0350
5_2x1ad	12	0.0318	0.0345	0.0345	0.0350
5_2x1ae	10	0.0325	0.0346	0.0339	0.0347
5_2x1af	8	0.0363	0.0331	0.0363	0.0364
5_2x1ag	6	0.0328	0.0369	0.0333	0.0315
5_2x1ah	4	0.0379	0.0313	**0.0303**	0.0325
5_2x1ai	2	0.0390	0.0391	0.0387	0.0389
5_2x2aa	18, 18	0.0414	0.0417	0.0422	0.0415
5_2x2ab	18, 16	0.0417	0.0423	0.0415	0.0425
5_2x2ac	18, 14	0.0405	0.0418	0.0438	0.0414
5_2x2ad	18, 12	0.0412	0.0409	0.0427	0.0411
5_2x2ae	18, 10	0.0412	0.0414	0.0411	0.0414
5_2x2af	18, 8	0.0411	0.0406	0.0411	0.0405
5_2x2ag	18, 6	0.0399	0.0404	0.0407	0.0401
5_2x2ah	18, 4	0.0394	0.0398	0.0408	0.0399

Filename	Size of ANN	RMS Error			
5_2x2ai	18, 2	0.0402	0.0407	0.0419	0.0411
5_2x2aj	16, 18	0.0419	0.0461	0.0423	0.0415
5_2x2ak	16, 16	0.0424	0.0416	0.0421	0.0414
5_2x2al	16, 14	0.0410	0.0418	0.0415	0.0410
5_2x2am	16, 12	0.0409	0.0415	0.0414	0.0410
5_2x2an	16, 10	0.0407	0.0409	0.0405	0.0409
5_2x2ao	16, 8	0.0420	0.0407	0.0414	0.0408
5_2x2ap	16, 6	0.0400	0.0399	0.0395	0.0401
5_2x2aq	16, 4	0.0403	0.0401	0.0395	0.0406
5_2x2ar	16, 2	0.0402	0.0403	0.0413	0.0411
5_2x2as	14, 18	0.0433	0.0418	0.0428	0.0427
5_2x2at	14, 16	0.0423	0.0419	0.0414	0.0415
5_2x2au	14, 14	0.0415	0.0406	0.0408	0.0427
5_2x2av	14, 12	0.0413	0.0407	0.0409	0.0406
5_2x2aw	14, 10	0.0408	0.0408	0.0410	0.0406
5_2x2ax	14, 8	0.0411	0.0406	0.0408	0.0410
5_2x2ay	14, 6	0.0408	0.0401	0.0397	0.0410
5_2x2az	14, 4	0.0395	0.0390	0.0406	0.0403
5_2x2ba	14, 2	0.0406	0.0410	0.0402	0.0411
5_2x2bb	12, 18	0.0410	0.0416	0.0407	0.0405
5_2x2bc	12, 16	0.0405	0.0422	0.0413	0.0409
5_2x2bd	12, 14	0.0408	0.0413	0.0407	0.0400
5_2x2be	12, 12	0.0408	0.0401	0.0409	0.0404
5_2x2bf	12, 10	0.0406	0.0395	0.0399	0.0408
5_2x2bg	12, 8	0.0405	0.0410	0.0397	0.0411
5_2x2bh	12, 6	0.0400	0.0397	0.0393	0.0404
5_2x2bi	12, 4	0.0387	0.0396	0.0400	0.0400
5_2x2bj	12, 2	0.0393	0.0404	0.0401	0.0411
5_2x2bk	10, 18	0.0407	0.0406	0.0414	0.0401

Filename	Size of ANN	RMS Error			
5_2x2bl	10, 16	0.0409	0.0399	0.0409	0.0404
5_2x2bm	10, 14	0.0403	0.0396	0.0411	0.0405
5_2x2bn	10, 12	0.0406	0.0401	0.0399	0.0401
5_2x2bo	10, 10	0.0403	0.0411	0.0394	0.0396
5_2x2bp	10, 8	0.0395	0.0399	0.0400	0.0403
5_2x2bq	10, 6	0.0398	0.0404	0.0392	0.0394
5_2x2br	10, 4	0.0400	0.0409	0.0399	0.0402
5_2x2bs	10, 2	0.0393	0.0391	0.0407	0.0396
5_2x2bt	8, 18	0.0405	0.0401	0.0404	0.0400
5_2x2bu	8, 16	0.0403	0.0403	0.0406	0.0399
5_2x2bv	8, 14	0.0388	0.0400	0.0405	0.0405
5_2x2bw	8, 12	0.0396	0.0400	0.0398	0.0397
5_2x2bx	8, 10	0.0394	0.0399	0.0399	0.0398
5_2x2by	8, 8	0.0399	0.0407	0.0400	0.0399
5_2x2bz	8, 6	0.0398	0.0395	0.0403	0.0406
5_2x2ca	8, 4	0.0400	0.0391	0.0397	0.0406
5_2x2cb	8, 2	0.0400	0.0407	0.0404	0.0411
5_2x2cc	6, 18	0.0394	0.0402	0.0402	0.0399
5_2x2cd	6, 16	0.0400	0.0405	0.0399	0.0405
5_2x2ce	6, 14	0.0387	0.0395	0.0397	0.0389
5_2x2cf	6, 12	0.0403	0.0396	0.0400	0.0405
5_2x2cg	6, 10	0.0392	0.0398	0.0399	0.0400
5_2x2ch	6, 8	0.0386	0.0403	0.0394	0.0396
5_2x2ci	6, 6	0.0395	0.0400	0.0376	0.0382
5_2x2cj	6, 4	0.0388	0.0380	0.0395	0.0379
5_2x2ck	6, 2	0.0409	0.0402	0.0401	0.0409
5_2x2cl	4, 18	0.0423	0.0399	0.0391	0.0401
5_2x2cm	4, 16	0.0403	0.0396	0.0399	0.0402
5_2x2cn	4, 14	0.0393	0.0394	0.0394	0.0403

Filename	Size of ANN	RMS Error			
5_2x2co	4, 12	0.0401	0.0399	0.0389	0.0400
5_2x2cp	4, 10	0.0392	0.0400	0.0384	0.0390
5_2x2cq	4, 8	0.0401	0.0388	0.0383	0.0400
5_2x2cr	4, 6	0.0396	0.0383	0.0390	**0.0355**
5_2x2cs	4, 4	0.0371	0.0382	0.0390	0.0379
5_2x2ct	4, 2	0.0409	0.0402	0.0394	0.0406
5_2x2cu	2, 18	0.0764	0.0822	0.0587	0.0837
5_2x2cv	2, 16	0.0719	0.0769	0.0542	0.0748
5_2x2cw	2, 14	0.0532	0.0744	0.0714	0.0750
5_2x2cx	2, 12	0.0820	0.0732	0.0810	0.0530
5_2x2cy	2, 10	0.0641	0.0595	0.0672	0.0515
5_2x2cz	2, 8	0.0821	0.0519	0.0411	0.0706
5_2x2da	2, 6	0.0754	0.0683	0.0400	0.0739
5_2x2db	2, 4	0.0720	0.0382	0.0401	0.0688
5_2x2dc	2, 2	0.0416	0.0402	0.0401	0.0392

Three Time Steps for One Step Prediction

5_3x1aa	18	0.0342	0.0341	0.0341	0.0327
5_3x1ab	16	0.0353	0.0359	0.0354	0.0348
5_3x1ac	14	0.0359	0.0331	0.0337	0.0351
5_3x1ad	12	0.0328	0.0329	0.0321	0.0347
5_3x1ae	10	0.0309	0.0303	0.0314	0.0328
5_3x1af	8	0.0344	0.0325	0.0344	0.0306
5_3x1ag	6	0.0293	0.0292	0.0327	**0.0291**
5_3x1ah	4	0.0314	0.0307	0.0314	0.0345
5_3x1ai	2	0.0503	0.0366	0.0367	0.0422
5_3x2aa	18, 18	0.0411	0.0398	0.0404	0.0404
5_3x2ab	18, 16	0.0401	0.0402	0.0438	0.0409

Filename	Size of ANN	RMS Error			
5_3x2ac	18, 14	0.0415	0.0401	0.0411	0.0403
5_3x2ad	18, 12	0.0388	0.0403	0.0399	0.0404
5_3x2ae	18, 10	0.0398	0.0396	0.0393	0.0424
5_3x2af	18, 8	0.0385	0.0387	0.0386	0.0388
5_3x2ag	18, 6	0.0380	0.0383	0.0382	0.0376
5_3x2ah	18, 4	0.0375	0.0372	0.0377	0.0388
5_3x2ai	18, 2	0.0385	0.0384	0.0382	0.0381
5_3x2aj	16, 18	0.0412	0.0403	0.0402	0.0393
5_3x2ak	16, 16	0.0429	0.0400	0.0400	0.0411
5_3x2al	16, 14	0.0405	0.0394	0.0394	0.0398
5_3x2am	16, 12	0.0390	0.0401	0.0400	0.0435
5_3x2an	16, 10	0.0383	0.0394	0.0386	0.0393
5_3x2ao	16, 8	0.0386	0.0383	0.0385	0.0407
5_3x2ap	16, 6	0.0383	0.0382	0.0370	0.0385
5_3x2aq	16, 4	0.0384	0.0371	0.0383	0.0385
5_3x2ar	16, 2	0.0380	0.0380	0.0380	0.0382
5_3x2as	14, 18	0.0395	0.0412	0.0388	0.0395
5_3x2at	14, 16	0.0382	0.0404	0.0394	0.0385
5_3x2au	14, 14	0.0401	0.0400	0.0399	0.0397
5_3x2av	14, 12	0.0385	0.0388	0.0379	0.0384
5_3x2aw	14, 10	0.0381	0.0386	0.0389	0.0381
5_3x2ax	14, 8	0.0379	0.0400	0.0399	0.0384
5_3x2ay	14, 6	0.0382	0.0378	0.0370	0.0381
5_3x2az	14, 4	0.0390	0.0382	0.0380	0.0374
5_3x2ba	14, 2	0.0383	0.0384	0.0382	0.0384
5_3x2bb	12, 18	0.0398	0.0391	0.0396	0.0392
5_3x2bc	12, 16	0.0390	0.0389	0.0391	0.0393
5_3x2bd	12, 14	0.0392	0.0388	0.0385	0.0409
5_3x2be	12, 12	0.0389	0.0383	0.0392	0.0398

Filename	Size of ANN	RMS Error			
5_3x2bf	12, 10	0.0382	<u>0.0369</u>	0.0381	0.0388
5_3x2bg	12, 8	<u>0.0376</u>	0.0378	0.0377	0.0377
5_3x2bh	12, 6	<u>0.0376</u>	0.0383	0.0378	0.0377
5_3x2bi	12, 4	0.0381	0.0370	0.0371	<u>0.0367</u>
5_3x2bj	12, 2	0.0382	0.0376	<u>0.0375</u>	0.0384
5_3x2bk	10, 18	0.0390	0.0403	0.0390	<u>0.0389</u>
5_3x2bl	10, 16	0.0388	<u>0.0383</u>	0.0388	0.0385
5_3x2bm	10, 14	0.0384	0.0387	<u>0.0383</u>	0.0385
5_3x2bn	10, 12	0.0386	0.0386	<u>0.0381</u>	<u>0.0381</u>
5_3x2bo	10, 10	0.0379	<u>0.0374</u>	0.0385	0.0375
5_3x2bp	10, 8	0.0382	<u>0.0375</u>	0.0376	0.0376
5_3x2bq	10, 6	0.0371	0.0368	<u>0.0362</u>	0.0377
5_3x2br	10, 4	0.0374	<u>0.0368</u>	0.0373	0.0374
5_3x2bs	10, 2	<u>0.0380</u>	0.0394	<u>0.0380</u>	0.0382
5_3x2bt	8, 18	0.0385	0.0386	<u>0.0384</u>	0.0403
5_3x2bu	8, 16	<u>0.0384</u>	<u>0.0384</u>	0.0386	<u>0.0384</u>
5_3x2bv	8, 14	0.0382	0.0385	0.0389	<u>0.0381</u>
5_3x2bw	8, 12	0.0381	0.0380	0.0389	<u>0.0372</u>
5_3x2bx	8, 10	0.0382	<u>0.0362</u>	0.0377	0.0373
5_3x2by	8, 8	0.0374	0.0377	<u>0.0373</u>	0.0376
5_3x2bz	8, 6	<u>0.0362</u>	0.0368	0.0370	0.0372
5_3x2ca	8, 4	0.0371	0.0368	<u>0.0358</u>	0.0379
5_3x2cb	8, 2	0.0377	0.0381	0.0379	<u>0.0373</u>
5_3x2cc	6, 18	0.0386	0.0402	<u>0.0378</u>	0.0384
5_3x2cd	6, 16	0.0377	<u>0.0373</u>	0.0399	0.0387
5_3x2ce	6, 14	<u>0.0373</u>	0.0374	0.0382	0.0377
5_3x2cf	6, 12	0.0378	0.0373	<u>0.0372</u>	0.0380
5_3x2cg	6, 10	0.0370	<u>0.0347</u>	0.0372	0.0382
5_3x2ch	6, 8	0.0370	<u>0.0363</u>	0.0373	0.0370

Filename	Size of ANN	RMS Error			
5_3x2ci	6, 6	0.0353	0.0364	0.0369	0.0365
5_3x2cj	6, 4	0.0340	0.0359	0.0364	0.0367
5_3x2ck	6, 2	0.0380	0.0373	0.0381	0.0382
5_3x2cl	4, 18	0.0376	0.0467	0.0410	0.0377
5_3x2cm	4, 16	0.0382	0.0363	0.0355	0.0382
5_3x2cn	4, 14	0.0369	0.0385	0.0374	0.0364
5_3x2co	4, 12	0.0371	0.0359	0.0370	0.0423
5_3x2cp	4, 10	0.0363	0.0359	0.0376	0.0365
5_3x2cq	4, 8	0.0372	0.0355	0.0370	0.0380
5_3x2cr	4, 6	0.0353	0.0357	0.0348	0.0366
5_3x2cs	4, 4	0.0352	0.0342	0.0356	0.0352
5_3x2ct	4, 2	0.0365	0.0376	0.0379	0.0378
5_3x2cu	2, 18	0.0848	0.0830	0.0760	0.0829
5_3x2cv	2, 16	0.0769	0.0769	0.0731	0.0854
5_3x2cw	2, 14	0.0754	0.0842	0.0761	0.0775
5_3x2cx	2, 12	0.0752	0.0540	0.0764	0.0687
5_3x2cy	2, 10	0.0766	0.0770	0.0827	0.0756
5_3x2cz	2, 8	0.0752	0.0514	0.0529	0.0752
5_3x2da	2, 6	0.0784	0.0769	0.0824	0.0415
5_3x2db	2, 4	0.0692	0.0825	0.0830	0.0396
5_3x2dc	2, 2	0.0864	0.0426	0.0378	0.0380

Four Time Steps for One Step Prediction

5_4x1aa	18	0.0388	0.0388	0.0390	0.0395
5_4x1ab	16	0.0388	0.0377	0.0383	0.0373
5_4x1ac	14	0.0382	0.0390	0.0383	0.0388
5_4x1ad	12	0.0384	0.0387	0.0377	0.0381
5_4x1ae	10	0.0378	0.0387	0.0366	0.0371

Filename	Size of ANN	RMS Error			
5_4x1af	8	0.0368	0.0369	0.0388	0.0374
5_4x1ag	6	0.0383	0.0362	0.0370	0.0364
5_4x1ah	4	**0.0351**	0.0380	0.0381	0.0356
5_4x1ai	2	0.0391	0.0390	0.0390	0.0397
5_4x2aa	18, 18	0.0458	0.0467	0.0457	0.0486
5_4x2ab	18, 16	0.0450	0.0454	0.0460	0.0462
5_4x2ac	18, 14	0.0455	0.0445	0.0459	0.0453
5_4x2ad	18, 12	0.0445	0.0442	0.0449	0.0443
5_4x2ae	18, 10	0.0457	0.0430	0.0448	0.0451
5_4x2af	18, 8	0.0436	0.0436	0.0442	0.0438
5_4x2ag	18, 6	0.0438	0.0434	0.0438	0.0436
5_4x2ah	18, 4	0.0429	0.0436	0.0435	0.0440
5_4x2ai	18, 2	0.0436	0.0438	0.0439	0.0437
5_4x2aj	16, 18	0.0452	0.0498	0.0457	0.0457
5_4x2ak	16, 16	0.0451	0.0447	0.0466	0.0444
5_4x2al	16, 14	0.0444	0.0447	0.0445	0.0446
5_4x2am	16, 12	0.0443	0.0445	0.0443	0.0450
5_4x2an	16, 10	0.0439	0.0436	0.0443	0.0439
5_4x2ao	16, 8	0.0440	0.0440	0.0436	0.0440
5_4x2ap	16, 6	0.0437	0.0438	0.0443	0.0446
5_4x2aq	16, 4	0.0437	0.0434	0.0440	0.0432
5_4x2ar	16, 2	0.0441	0.0440	0.0441	0.0442
5_4x2as	14, 18	0.0453	0.0449	0.0448	0.0479
5_4x2at	14, 16	0.0440	0.0444	0.0447	0.0443
5_4x2au	14, 14	0.0462	0.0455	0.0444	0.0441
5_4x2av	14, 12	0.0439	0.0437	0.0438	0.0440
5_4x2aw	14, 10	0.0437	0.0436	0.0441	0.0436
5_4x2ax	14, 8	0.0427	0.0442	0.0434	0.0443
5_4x2ay	14, 6	0.0436	0.0428	0.0438	0.0433

Filename	Size of ANN	RMS Error			
5_4x2az	14, 4	0.0415	0.0427	0.0412	0.0440
5_4x2ba	14, 2	0.0440	0.0434	0.0437	0.0440
5_4x2bb	12, 18	0.0441	0.0443	0.0448	0.0444
5_4x2bc	12, 16	0.0448	0.0439	0.0441	0.0451
5_4x2bd	12, 14	0.0440	0.0436	0.0443	0.0447
5_4x2be	12, 12	0.0438	0.0440	0.0432	0.0435
5_4x2bf	12, 10	0.0436	0.0435	0.0425	0.0436
5_4x2bg	12, 8	0.0441	0.0432	0.0441	0.0435
5_4x2bh	12, 6	0.0428	0.0435	0.0425	0.0429
5_4x2bi	12, 4	0.0434	0.0421	0.0435	0.0427
5_4x2bj	12, 2	0.0438	0.0439	0.0442	0.0439
5_4x2bk	10, 18	0.0443	0.0462	0.0436	0.0439
5_4x2bl	10, 16	0.0438	0.0447	0.0440	0.0434
5_4x2bm	10, 14	0.0433	0.0438	0.0440	0.0434
5_4x2bn	10, 12	0.0445	0.0440	0.0434	0.0437
5_4x2bo	10, 10	0.0436	0.0431	0.0431	0.0437
5_4x2bp	10, 8	0.0442	0.0441	0.0436	0.0422
5_4x2bq	10, 6	0.0434	0.0424	0.0420	0.0428
5_4x2br	10, 4	0.0427	0.0427	0.0433	0.0425
5_4x2bs	10, 2	0.0440	0.0439	0.0436	0.0435
5_4x2bt	8, 18	0.0435	0.0440	0.0440	0.0463
5_4x2bu	8, 16	0.0429	0.0442	0.0437	0.0433
5_4x2bv	8, 14	0.0433	0.0437	0.0425	0.0432
5_4x2bw	8, 12	0.0436	0.0438	0.0420	0.0432
5_4x2bx	8, 10	0.0430	0.0443	0.0439	0.0432
5_4x2by	8, 8	0.0424	0.0431	0.0424	0.0432
5_4x2bz	8, 6	0.0441	0.0435	0.0426	0.0432
5_4x2ca	8, 4	0.0417	0.0405	0.0423	0.0422
5_4x2cb	8, 2	0.0428	0.0434	0.0438	0.0437

Filename	Size of ANN	RMS Error			
5_4x2cc	6, 18	0.0432	0.0434	0.0436	0.0431
5_4x2cd	6, 16	0.0434	0.0436	0.0433	0.0429
5_4x2ce	6, 14	0.0430	0.0434	0.0425	0.0435
5_4x2cf	6, 12	0.0423	0.0414	0.0428	0.0429
5_4x2cg	6, 10	0.0408	0.0412	0.0418	0.0431
5_4x2ch	6, 8	0.0419	0.0408	0.0414	0.0419
5_4x2ci	6, 6	0.0419	0.0422	0.0421	0.0427
5_4x2cj	6, 4	**0.0385**	0.0426	0.0436	0.0420
5_4x2ck	6, 2	0.0435	0.0432	0.0437	0.0436
5_4x2cl	4, 18	0.0417	0.0403	0.0426	0.0457
5_4x2cm	4, 16	0.0417	0.0433	0.0429	0.0433
5_4x2cn	4, 14	0.0417	0.0428	0.0553	0.0430
5_4x2co	4, 12	0.0432	0.0430	0.0440	0.0436
5_4x2cp	4, 10	0.0423	0.0413	0.0430	0.0423
5_4x2cq	4, 8	0.0399	0.0417	0.0422	0.0434
5_4x2cr	4, 6	0.0433	0.0415	0.0432	0.0413
5_4x2cs	4, 4	0.0400	0.0404	0.0434	0.0424
5_4x2ct	4, 2	0.0437	0.0438	0.0436	0.0439
5_4x2cu	2, 18	0.0840	0.0856	0.0828	0.0885
5_4x2cv	2, 16	0.0608	0.0882	0.0864	0.0876
5_4x2cw	2, 14	0.0873	0.0707	0.0751	0.0817
5_4x2cx	2, 12	0.0507	0.0800	0.0808	0.0874
5_4x2cy	2, 10	0.0754	0.0894	0.0880	0.0887
5_4x2cz	2, 8	0.0584	0.0742	0.0838	0.0736
5_4x2da	2, 6	0.0817	0.0469	0.0768	0.0458
5_4x2db	2, 4	0.0913	0.0588	0.0746	0.0430
5_4x2dc	2, 2	0.0877	0.0827	0.0436	0.0424

Appendix

H Results of Training an ANN to Predict Different Transients

This appendix contains the full results of all the ANNs trained to predict combinations of the three Transients introduced in Section 5.5. The ANNs were all designed to predict the values of the PWR variables one step ahead. Two time steps of recent plant data was used as inputs the ANNs. For each combination of transient pairs a set of one hidden layer ANNs were developed. The number of nodes in the hidden layer was varied from a maximum of 20 to a minimum of 10. All the ANN training was repeated four times to avoid possible local minima. The value recorded in each case is the lowest RMS error for the training. All ANNs were trained for 60,000 cycles of presentation of the training set and then a further 20,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing.

Results of Training an ANN to Predict Different Transients

Filename	Size of ANN	Training Set	Test Set	RMS Error
a_bx.nnd	16-9-7	a_btra1.nna	a_btes1.nna	0.027
a_b1.nnd	15-20-7	a_btra.nna	a_btes.nna	0.0243
a_b1x.nnd	16-10-7	a_btra1.nna	a_btes1.nna	0.0255
a_b2.nnd	15-20-7	a_btra.nna	a_btes.nna	0.0263
a_b2x.nnd	16-11-7	a_btra1.nna	a_btes1.nna	0.0252
a_b3.nnd	15-10-7	a_btra.nna	a_btes.nna	0.0252
a_b3x.nnd	16-12-7	a_btra1.nnd	a_btes1.nna	0.0246
a_b4.nnd	15-11-7	a_btra.nna	a_btes.nna	0.0251
a_b4x.nnd	16-13-7	a_btra1.nnd	a_btes1.nnd	0.0254
a_b5x.nnd	16-14-7	a_btra1.nnd	a_btes1.nnd	0.0251
a_b6x.nnd	16-12-7	a_btra1.nnd	a_btes1.nnd	0.026
a_c1.nnd	15-20-7	a_ctra.nnd	a_ctes.nnd	0.0236
a_c2.nnd	15-20-7	a_ctra.nnd	a_ctes.nnd	0.0235
a_c3.nnd	15-10-7	a_ctra.nnd	a_ctes.nnd	0.0228
a_c4.nnd	15-12-7	a_ctra.nnd	a_ctes.nnd	0.0247
a_c5.nnd	15-11-7	a_ctra.nnd	a_ctes.nnd	0.0232
a_c9.nnd	15-10-7	a_ctra.nnd	a_ctes.nnd	0.023
b_c.nnd	15-1-7	b_ctra.nnd	b_ctes.nna	0.1332
b_c1.nnd	15-20-7	b_ctra.nnd	b_ctes.nna	0.0547
b_c9.nnd	15-10-7	b_ctra.nnd	b_ctes.nna	0.0528
a_b_c.nnd	15-20-7	a_b_ctra.nnd	a_b_ctes.nnd	0.0229
a_b_c1.nnd	15-20-7	a_b_ctra.nnd	a_b_ctes.nnd	0.0250
a_b_c2.nnd	15-10-7	a_b_ctra.nnd	a_b_ctes.nnd	0.0230
a_b_c3.nnd	15-11-7	a_b_ctra.nnd	a_b_ctes.nnd	0.0235
a_b_c4.nnd	15-12-7	a_b_ctra.nnd	a_b_ctes.nnd	0.0238
a_b_c5.nnd	15-13-7	a_b_ctra.nnd	a_b_ctes.nnd	0.0246
a_b_c9.nnd	15-10-7	a_b_ctra.nnd	a_b_ctes.nnd	0.0235

Appendix

I Results of Training an ANN to Predict Transients with different forms of Input

This appendix contains the full results of all the ANNs trained to predict Transients with different forms of inputs as introduced in Section 5.6. The ANNs were all designed to predict the values of the PWR variables one step ahead. Two time steps of recent plant data was used as inputs the ANNs. For each combination of transient pairs a set of one hidden layer ANNs were developed. The number of nodes in the hidden layer was varied between 5 and 7. All the ANN training was repeated four times to avoid possible local minima. The value recorded in each case is the lowest RMS error for the training. All ANNs were trained for 100,000 cycles of presentation of the training set and then a further 20,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing. The '*' in the training/test column indicates tra for the training file and tes for the test file, ie an ANN using squ_*4.nna has a training file named squ_tra4.nna and a test file named squ_tes4.nna.

Results of Training an ANN for different forms of input

Filename	Size of ANN	Train/Test	Transfer	RMS Error
tes1a.nnd	10-5-2	tes1*.nna	TanH	0.0674
tes1b.nnd	10-5-2	tes1*.nna	TanH	0.0348
tes1c.nnd	10-4-2	tes1*.nna	TanH	0.0296
tes1g.nnd	10-5-2	tes1*.nna	TanH	0.0465
tes1h.nnd	10-5-2	tes1*.nna	TanH	0.0419
tes1i.nnd	10-5-2	tes1*.nna	TanH	0.0286
tes1j.nnd	10-5-2	tes1*.nna	TanH	0.0481
tes1k.nnd	10-5-2	tes1*.nna	TanH	0.0565
tes1l.nnd	10-5-2	tes1*.nna	TanH	0.0406
tes1m.nnd	10-5-2	tes1*.nna	TanH	0.0496
tes1n.nnd	10-5-2	tes1*.nna	TanH	0.0470
tes1o.nnd	10-5-2	tes1*.nna	TanH	0.0565
tes2a.nnd	10-5-2	tes2*.nna	TanH	0.0256
tes2b.nnd	10-5-2	tes2*.nna	TanH	0.0512
tes2c.nnd	10-6-2	tes2*.nna	TanH	0.0336
tes2d.nnd	10-6-2	tes2*.nna	TanH	0.0165
tes2e.nnd	10-5-2	tes2*.nna	TanH	0.0093
tes3a.nnd	10-5-2	tes3*.nna	TanH	0.0373
tes3b.nnd	10-5-2	tes3*.nna	TanH	0.0349
tes3c.nnd	10-5-2	tes3*.nna	TanH	0.0130
tes4a.nnd	10-5-2	tes4*.nna	TanH	0.0191
tes4b.nnd	10-5-2	tes4*.nna	TanH	0.0342
tes5a.nnd	10-6-2	tes5*.nna	TanH	0.0512
tes5b.nnd	10-5-2	tes5*.nna	TanH	0.0304
tes5c.nnd	10-5-2	tes5*.nna	TanH	0.0296

Filename	Size of ANN	Train/Test	Transfer	RMS Error
tes5d.nnd	10-6-2	tes5*.nna	TanH	0.0286
tes5g.nnd	10-5-2	tes5*.nna	TanH	0.0559
tes5h.nnd	10-5-2	tes5*.nna	TanH	0.0317
tes5i.nnd	10-5-2	tes5*.nna	TanH	0.0492
tes5j.nnd	10-5-2	tes5*.nna	TanH	0.0305
tes5k.nnd	10-5-2	tes5*.nna	TanH	0.0320
tes5l.nnd	10-5-2	tes5*.nna	TanH	0.0446
tes5m.nnd	10-5-2	tes5*.nna	TanH	0.0443
tes5n.nnd	10-5-2	tes5*.nna	TanH	0.0469
tes5o.nna	10-5-2	tes5*.nna	TanH	0.0346
tesalla.nnd	10-5-2	allt*.nna	TanH	0.0549
tesallb.nnd	10-6-2	allt*.nna	TanH	0.0513
tesallc.nnd	10-7-2	allt*.nna	TanH	0.0603
tesallg.nnd	10-6-2	allt*.nna	TanH	0.0564
tesallh.nnd	10-6-2	allt*.nna	TanH	0.0448
tesalli.nnd	10-6-2	allt*.nna	TanH	0.0598
tesallj.nnd	10-6-2	allt*.nna	TanH	0.0554
tesallk.nnd	10-6-2	allt*.nna	TanH	0.0616
tesalll.nnd	10-6-2	allt*.nna	TanH	0.0535
tesallm.nnd	10-6-2	allt*.nna	TanH	0.0591
tesalln.nnd	10-6-2	allt*.nna	TanH	0.0489
tesallo.nnd	10-6-2	allt*.nna	TanH	0.0638

Appendix

J Results of Training an ANN to Predict Linearity

This appendix contains the full results of all the ANNs trained to predict linearity in a transient curve as discussed in Section 6.3.3.1. Each ANNs consisted of a single input, for the % linearity, and single output node, for predicted PWR variable value. For each percentage linearity a set of one hidden layer ANNs were developed. The number of nodes in the hidden layer was varied between 5 and 8. All the ANN training was repeated four times to avoid possible local minima. The value recorded in each case is the lowest RMS error for the training. All ANNs were trained for 40,000 cycles of presentation of the training set and then a further 10,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing.

Filename	Suffix	Transfer	HL Nodes	RMS Error
25aa	1	Sine	5	0.046304
	2		6	0.045032
	3		7	0.040742
	4		8	0.043617
25ab	1	Tanh	5	0.028471
	2		6	0.027906
	3		7	0.028692
	4		8	0.034959
25ac	1	Linear	5	0.284452
	2		6	0.284396
	3		7	0.284266
	4		8	0.284231
25ba	1	Sine	5	0.037717
	2		6	0.035634
	3		7	0.035739
	4		8	0.036242
25bb	1	Tanh	5	0.022907
	2		6	0.028809
	3		7	0.025364
	4		8	0.027814
25bc	1	Linear	5	0.370216
	2		6	0.370231
	3		7	0.370067
	4		8	0.369972
25ca	1	Sine	5	0.036780
	2		6	0.037091
	3		7	0.035745
	4		8	0.035910

Filename	Suffix	Transfer	HL Nodes	RMS Error
25cb	1	Tanh	5	0.027190
	2		6	0.029554
	3		7	0.026856
	4		8	0.027893
25cc	1	Linear	5	0.370012
	2		6	0.370223
	3		7	0.369993
	4		8	0.370032
25da	1	Sine	5	0.045498
	2		6	0.042752
	3		7	0.025228
	4		8	0.046095
25db	1	Tanh	5	0.032690
	2		6	0.028120
	3		7	0.027296
	4		8	0.030714
50aa	1	Sine	5	0.028009
	2		6	0.028800
	3		7	0.029482
	4		8	0.029525
50ab	1	Tanh	5	0.024770
	2		6	0.020590
	3		7	0.023012
	4		8	0.023115
50ba	1	Sine	5	0.102453
	2		6	0.101561
	3		7	0.100380
	4		8	0.128022

Filename	Suffix	Transfer	HL Nodes	RMS Error
50bb	1	Tanh	5	0.057344
	2		6	0.040284
	3		7	0.048163
	4		8	0.060863
50ca	1	Sine	5	0.029358
	2		6	0.033676
	3		7	0.029385
	4		8	0.030267
50cb	1	Tanh	5	0.025207
	2		6	0.023809
	3		7	0.023809
	4		8	0.027339
75aa	1	Sine	5	0.054545
	2		6	0.06463
	3		7	0.070238
	4		8	0.048667
75ab	1	Tanh	5	0.108147
	2		6	0.111393
	3		7	0.114466
	4		8	0.084984
75ba	1	Sine	5	0.065286
	2		6	0.056608
	3		7	0.067679
	4		8	0.060335
75bb	1	Tanh	5	0.082521
	2		6	0.097371
	3		7	0.115471
	4		8	0.122033

Filename	Suffix	Transfer	HL Nodes	RMS Error
100aa	1	Sine	5	0.005533
	2		6	0.007427
	3		7	0.004451
	4		8	0.004968
100ab	1	Tanh	5	0.009670
	2		6	0.012296
	3		7	0.007194
	4		8	0.007394
0aa	1	Sine	5	0.017273
	2		6	0.018206
	3		7	0.019386
	4		8	0.017758
0ab	1	Tanh	5	0.112501
	2		6	0.108689
	3		7	0.111768
	4		8	0.110860

Appendix

K Results of Training an ANN for Predicting Angularity

This appendix contains the full results of all the ANNs trained for predicting the included of a Transient curve as introduced in Section 6.3.3.2. The ANNs were all designed to predict the values of the PWR variables given the angle of change in the transient curve. For each included angle a set of one and two hidden layer ANNs were developed. The number of nodes in the hidden layers were varied from a maximum of 10 to a minimum of 3. All the ANN training was repeated four times to avoid possible local minima. The value recorded in each case is the lowest RMS error for the training. All ANNs were trained for 60,000 cycles of presentation of the training set and then a further 20,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing.

Results of Training an ANN to Predict Angularity

Filename	Size of ANN	Training Set	Test Set	RMS Error
t5a.nnd	1-5-1	t5tra.nna	t5tes.nna	0.0741
t5a1.nnd	1-5-1	t5tra.nna	t5tes.nna	0.0737
t5aa.nnd	1-8-4-1	t5tra.nna	t5tes.nna	0.0639
t5ab.nnd	1-10-5-1	t5tra.nna	t5tes.nna	0.069
t5ac.nnd	1-6-3-1	t5tra.nna	t5tes.nna	0.0788
t5ad.nnd	1-10-5-1	t5tra.nna	t5tes.nna	0.0624
t5b.nnd	1-6-1	t5tra.nna	t5tes.nna	0.0742
t5c.nnd	1-7-1	t5tra.nna	t5tes.nna	0.0732
t5d.nnd	1-8-1	t5tra.nna	t5tes.nna	0.0671
t15a.nnd	1-5-1	t15tra.nna	t15tes.nna	0.0734
t15aa.nnd	1-8-4-1	t15tra.nna	t15tes.nna	0.0653
t15b.nnd	1-6-1	t15tra.nna	t15tes.nna	0.0717
t15c.nnd	1-7-1	t15tra.nna	t15tes.nna	0.0683
t15d.nnd	1-8-1	t15tra.nna	t15tes.nna	0.0654
t25a.nnd	1-5-1	t25tra.nna	t25tes.nna	0.0778
t25b.nnd	1-6-1	t25tra.nna	t25tes.nna	0.0681
t25c.nnd	1-7-1	t25tra.nna	t25tes.nna	0.0703
t25d.nnd	1-8-1	t25tra.nna	t25tes.nna	0.0712
t35a.nnd	1-5-1	t35tra.nna	t35tes.nna	0.0736
t35b.nnd	1-6-1	t35tra.nna	t35tes.nna	0.0664
t35c.nnd	1-7-1	t35tra.nna	t35tes.nna	0.065
t35d.nnd	1-8-1	t35tra.nna	t35tes.nna	0.0671
t45a.nnd	1-5-1	t43tra.nna	t45tes.nna	0.0718
t45a1.nnd	1-5-1	t45tra.nna	t45tes.nna	0.0737
t45aa.nnd	1-10-5-1	t45tra.nna	t45tes.nna	0.0624
t45ab.nnd	1-10-5-1	t45tra.nna	t45tes.nna	0.0623

Filename	Size of ANN	Training Set	Test Set	RMS Error
t45b.nnd	1-6-1	t45tra.nna	t45tes.nna	0.0753
t45c.nnd	1-7-1	t45tra.nna	t45tes.nna	0.0682
t45d.nnd	1-8-1	t45tra.nna	t45tes.nna	0.0682
t55a.nnd	1-5-1	t55tra.nna	t55tes.nna	0.0696
t55aa.nnd	1-10-5-1	t55tra.nna	t55tes.nna	0.0624
t55b.nnd	1-6-1	t55tra.nna	t55tes.nna	0.0712
t55c.nnd	1-7-1	t55tra.nna	t55tes.nna	0.0641
t55d.nnd	1-8-1	t55tra.nna	t55tes.nna	0.065
t65a.nnd	1-5-1	t65tra.nna	t65tes.nna	0.0744
t65aa.nnd	1-10-5-1	t65tra.nna	t65tes.nna	0.0624
t65b.nnd	1-6-1	t65tra.nna	t65tes.nna	0.0738
t65c.nnd	1-7-1	t65tra.nna	t65tes.nna	0.0661
t65d.nnd	1-8-1	t65tra.nna	t65tes.nna	0.0679
t75a.nnd	1-5-1	t75tra.nna	t75tes.nna	0.0724
t75a1.nnd	1-5-1	t75tra.nna	t75tes.nna	0.0736
t75aa.nnd	1-10-5-1	t75tra.nna	t75tes.nna	0.0624
t75b.nnd	1-6-1	t75tra.nna	t75tes.nna	0.0745
t75c.nnd	1-7-1	t75tra.nna	t75tes.nna	0.067
t75d.nnd	1-8-1	t75tra.nna	t75tes.nna	0.0673
t85a.nnd	1-5-1	t85tra.nna	t85tes.nna	0.0743
t85a1.nnd	1-5-1	t85tra.nna	t85tes.nna	0.0742
t85aa.nnd	1-10-5-1	t85tra.nna	t85tes.nna	0.0626
t85ab.nnd	1-8-4-1	t85tra.nna	t85tes.nna	0.0627
t85ac.nnd	1-10-5-1	t85tra.nna	t85tes.nna	0.0624
t85b.nnd	1-6-1	t85tra.nna	t85tes.nna	0.0680
t85c.nnd	1-7-1	t85tra.nna	t85tes.nna	0.0655
t85d.nnd	1-8-1	t85tra.nna	t85tes.nna	0.0689

Appendix

L Code Listing for One Compartment Direct Equivalent ANN Program

The following program listing is the ANN for the one compartment model of a PWR discussed in Section 6.5. The program was written and complied using Borland Turbo C, Version 2. The program inputs are entered from the keyboard and the output is a ASCII text file of the series of model variables. Plotted results from the program are included in Chapter 6 as Figs 6.40 to 6.44.

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

        Program of coding of neural network molecule
        Author : P R Weller      Copyright (c) 1996
        26/04/96 - Version 0.1
        13/06/96 - Version 0.2 - Modified to produce NWorks *.nna files

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

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>

FILE *temp;

main()
{
 float q, t_1, t_1old, t_in, m_in, m, cp, t_1diff;
 float ip_1, ip_2, ip_3, op_1, op_2, op_3, ip_out, op_out;
 int time;

 time = 1;
 t_1old = 0;
 t_1diff = 100;
 ip_1 = ip_2 = ip_3 = op_1 = op_2 = op_3 = ip_out = op_out = 0;
 cp = 4900;
 if ((temp = fopen("temp.txt", "w")) == NULL)
  {
   printf("The temperature file could not be opened\n");
   return (1);
  }
 printf("\nInput initial temperature of PV : ");
 scanf("%f",&t_1);                /* Value of T1 */
```

```
printf("\nInput initial temperature of cold leg : ");
scanf("%f",&t_in);              /* Value of Tin */
printf("\nInput liquid mass in PV : ");
scanf("%f",&m);                 /* Value of M1 */
printf("\nInput rate of change of mass of cold leg : ");
scanf("%f",&m_in);              /* Value of Min */
printf("\nInput the heat rate : ");
scanf("%f",&q);                 /* Value of Q */
fprintf(temp," %d  %f", time, t_1);
time ++;
while (t_1diff > 0.0001)        /* Stopping Condition */
 {
  t_1old = t_1;

  /* Hidden Node 1 */
  ip_1 = t_in - t_1;
  op_1 = 0.5 * (pow(ip_1, 2));  /* Half Square Threshold */

  /* Hidden Node 2 */
  ip_2 = t_in + m_in - t_1;
  op_2 = 0.5 * (pow(ip_2, 2));  /* Half Square Threshold */

  /* Hidden Node 3 */
  ip_3 = m_in;
  op_3 = 0.5 * (pow(ip_3, 2));  /* Half Square Threshold */

  /* Output Node */
  ip_out = (op_2/m) - (op_1/m) - (op_3/m) + t_1 + (q/(cp*m));
  op_out = ip_out;              /* Linear Thrshold */

  t_1 = op_out;
  t_1diff = fmod(t_1, t_1old);

  fprintf(temp,"\n %f %f %f %f %f", t_1old, t_in, m_in, q, t_1);
```

```
    time ++;
  }
 fclose(temp);
}

/************************** End of Program ***********************/
```

Appendix

M Results of Training an ANN to Square a Number

This appendix contains the full results of all the ANNs trained to square a number of the as discussed in Section 6.5. The ANNs were all designed to take a single value as input and produce its square at the output. Two sets of one hidden layer ANNs were developed. The first had an input range of between 1 and 20 while the second range was -20 to 20. The number of nodes in the hidden layer was varied between 2 and 5. All the ANN training was repeated four times to avoid possible local minima. The value recorded in each case is the lowest RMS error for the training. All ANNs were trained for 40,000 cycles of presentation of the training set and then a further 10,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing.

Results of Training an ANN to Square a Number

Filename	Size of ANN	Train/Test	Threshold	RMS Error
square1a.nnd	1-2-1	squ_*.nna	Tanh	0.0224
square2a.nnd	1-2-1	squ_*.nna	Tanh	0.0153
square3a.nnd	1-3-1	squ_*.nna	Sigmoid	0.0313
square4a.nnd	1-3-1	squ_*.nna	Tanh	0.0175
square5a.nnd	1-4-1	squ_*1.nna	Tanh	0.0605
square6a.nnd	1-5-1	squ_*1.nna	Tanh	0.0554
square7a.nnd	1-3-1	squ_*1.nna	Sine	0.0296

The '*' in the training/test column indicates tra for the training file and tes for the test file, ie an ANN using squ_*1.nna has a training file named squ_tra1.nna and a test file named squ_tes1.nna. the presence of a '1' in the training set name signifies the input range of -20 → 20.

Appendix

N Results of Training an ANN to Multiply Two Numbers

This appendix contains the full results of all the ANNs trained to multiply two given numbers as discussed in Section 6.5. The ANNs were all designed to have an input of two numbers, between 0 and 10, and output the product. A set of one hidden layer ANNs were developed. The number of nodes in the hidden layer was varied between 3 and 7. The value recorded in each case is the lowest RMS error for the training. All ANNs were trained for 40,000 cycles of presentation of the training set and then a further 10,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing.

The '*' in the training/test column of the following table indicates tra for the training file and tes for the test file, ie an ANN using squ_*4.nna has a training file named squ_tra4.nna and a test file named squ_tes4.nna.

Results of Training an ANN to Multiply Two Numbers

Filename	Size of ANN	Train/Test	Threshold	RMS Error
test8a.nnd	2-3-1	squ_*2.nna	Sine	0.0200
test9a.nnd	2-3-1	squ_*3.nna	Sine	0.0298
test10a.nnd	2-4-1	squ_*3.nna	Sine	0.0304
testba.nnd	2-3-1	squ_*4.nna	Sine	0.0250
testbb.nnd	2-4-1	squ_*4.nna	Sine	0.0248
testbc.nnd	2-5-1	squ_*4.nna	Sine	0.0234
mult1.nnd	2-6-1	squ_*4.nna	Sine	0.0249
mult2.nnd	2-7-1	squ_*4.nna	Sine	0.0822
mult3.nnd	2-3-1	squ_*4.nna	Sine	0.0239
mult4.nnd	2-4-1	squ_*4.nna	Sine	0.0808
mult5.nnd	2-5-1	squ_*4.nna	Sine	0.0820
mult6.nnd	2-6-1	squ_*4.nna	Sine	0.0254
mult7.nnd	2-7-1	squ_*4.nna	Sine	0.0237
mult8.nnd	2-3-1	squ_*4.nna	Sine	0.0230
mult9.nnd	2-4-1	squ_*4.nna	Sine	0.0242
mult10.nnd	2-5-1	squ_*4.nna	Sine	0.0243
mult11.nnd	2-6-1	squ_*4.nna	Sine	0.0842
mult12.nnd	2-7-1	squ_*4.nna	Sine	0.0249
mult13.nnd	2-3-1	squ_*4.nna	Sine	0.0840
mult14.nnd	2-4-1	squ_*4.nna	Sine	0.0230
mult15.nnd	2-5-1	squ_*4.nna	Sine	0.0236
mult16.nnd	2-6-1	squ_*4.nna	Sine	0.0815
mult17.nnd	2-7-1	squ_*4.nna	Sine	0.0228
mult18.nnd	2-6-1	squ_*4.nna	Sine	0.0820
mult19.nnd	2-3-1	squ_*4.nna	Sine	0.0225
mult20.nnd	2-5-1	squ_*4.nna	Sine	0.0830

Appendix

O Results of Training an ANN to Model a One Compartment PWR

This appendix contains the full results of all the ANNs trained to model a one compartment PWR as introduced in Section 6.5. The ANNs were all designed to predict the temperature of the PWR variables one step ahead. The four variables from the energy equation on Page 127 were used as inputs the ANNs. A set of one and two hidden layer ANNs were developed. The number of nodes in the hidden layer was varied from a maximum of 10 to a minimum of 5, and several trial ANNs of a single hidden layer of 80, 1 and 0 nodes. All the ANN training was repeated four times to avoid possible local minima. The value recorded in each case is the lowest RMS error for the training. All ANNs were trained for 60,000 cycles of presentation of the training set and then a further 20,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing.

The `*' in the training/test column of the following table indicates tra for the training file and tes for the test file, ie an ANN using squ_*4.nna has a training file named squ_tra4.nna and a test file named squ_tes4.nna.

Results of Training an ANN to Model a One Compartment PWR

Filename	Size of ANN	Train/Test	Threshold	RMS Error
temp1.nnd	4-5-1	temp_*.nna	Sine	0.0191
temp2.nnd	4-5-1	temp_*.nna	Sine	0.0187
temp3.nnd	4-6-1	temp_*.nna	Sine	0.0188
temp4.nnd	4-6-1	temp_*.nna	Sine	0.0183
temp4a.nnd	4-80-1	temp_*.nna	GRNN	0.0708
temp5.nnd	4-7-1	temp_*.nna	Sine	0.0178
temp6.nnd	4-7-1	temp_*.nna	Sine	0.0177
temp7.nnd	4-4-1	temp_*.nna	Sine	0.0180
temp8.nnd	4-4-1	temp_*.nna	Sine	0.0186
temp9.nnd	4-6-3-1	temp_*.nna	Sine	0.0211
temp10.nnd	4-6-3-1	temp_*.nna	Sine	0.0203
temp11.nnd	4-7-1	temp11.nna	Sine	0.0123
temp12.nnd	4-7-1	temp12.nna	Sine	0.0145
temp13.nnd	4-7-1	temp13.nna	Sine	0.0225
temp13a.nnd	4-8-1	temp13.nna	Sine	0.0215
temp13b.nnd	4-8-1	temp13.nna	Sine	0.0218
temp13c.nnd	4-9-1	temp13.nna	Sine	0.0241
temp13d.nnd	4-8-1	temp13.nna	Sine	0.0209
temp13e.nnd	4-8-1	temp13.nna	Linear	0.0186
temp13f.nnd	4-10-1	temp13.nna	DNNA	0.0669
temp13g.nnd	4-5-1	temp13.nna	Linear	0.0174
temp13h.nnd	4-5-1	temp13.nna	Linear	0.0184
temp13i.nnd	4-2-1	temp13.nna	Linear	0.0183
temp13j.nnd	4-1-1	temp13.nna	Linear	0.0183
temp13k.nnd	4-0-1	temp13.nna	Linear	0.0258

Filename	Size of ANN	Train/Test	Threshold	RMS Error
tempa1.nnd	4-7-1	tempa.nna	Sine	0.0160
tempa2.nnd	4-6-1	temp1.nna	Sine	0.0188
tempa3.nnd	4-7-1	temp1.nna	Sine	0.0178
tempa4.nnd	4-8-1	temp1.nna	Tanh	0.0326
tempa5.nnd	4-6-1	temp1.nna	Sine	0.0190
tempb1.nnd	4-7-1	temp2*.nna	Sine	0.0195
tempb2.nnd	4-7-3-1	temp2*.nna	Sine	0.0193
tempc1.nnd	4-7-1	temp3*.nna	Sine	0.0196
tempd1.nnd	4-7-1	temp4*.nna	Sine	0.0194
tempd2.nnd	4-7-1	temp4*.nna	Sine	0.0205
tempe1.nnd	1-7-1	temp5*.nna	Sine	0.0214
tempf1.nnd	4-7-1	temp6*.nna	Sine	0.0254

The following table gives full details of the training data mentioned above.

No of Cases	T_{in} Start	T_{in} Finish	M_{in} Start	M_{in} Finish	Q Start	Q Finish
tempa.nna						
200	250	250	17400	17400	3.4×10^9	3.4×10^9
131	286	286	20000	20000	3.4×10^9	3.4×10^9
temp1.nna						
200	250	250	17400	17400	3.4×10^9	3.4×10^9
temp2.nna						
150	300	300	17400	17400	3.4×10^9	3.4×10^9
temp3.nna						
215	286	286	12500	12500	3.4×10^9	3.4×10^9
temp4.nna						
160	286	286	15000	15000	3.4×10^9	3.4×10^9
temp6.nna						
131	286	286	20000	20000	3.4×10^9	3.4×10^9

Appendix

P Code Listing for ANN based One Compartment Prediction Program

This appendix contains the full program listing for the ANN based one compartment prediction program introduced in Section 6.5, Page 139. The program was written and complied using Borland Turbo C, Version 2. The program are entered from the keyboard and the output is an ASCII text file of the compartment temperature for the conditions defined. Plotted results from the program are given as Figs 6.52 and 6.53.

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

        Program of coding of neural network molecule
        Author : P R Weller     Copyright (c) 1996
        26/04/96 - Version 0.1
        10/06/96 - Version 0.2 - NWorks trained square function added
        11/06/96 - Version 0.3 - Combined Square + NWorks trained NN
        11/06/96 - Version 0.4 - NWorks trained multiplying NN
        14/06/96 - Version 0.5 - Full NWorks 1CV model added
        21/06/96 - Version 0.6 - Ramping added for variables
        24/06/96 - Version 0.7 - Graphics added

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

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include <graphics.h>

FILE *temp1, *temp2, *temp3, *temp4;

float Yin[4];
float Yout[1];

float q, m_in, m, cp, t_in;
float q_start, q_end, m_in_start, m_in_end, t_in_start, t_in_end;
float t_1a, t_1aold, t_1adiff;
float t_1b, t_1bold, t_1bdiff;
float t_1c, t_1cold, t_1cdiff;
float t_1d, t_1dold, t_1ddiff;
float ip_1a, ip_2a, ip_3a, op_1a, op_2a, op_3a, ip_outa, op_outa;
float ip_1b, ip_2b, ip_3b, op_1b, op_2b, op_3b, ip_outb, op_outb;
float ip_1c, ip_2c, op_c, ip_outc, op_outc;
```

```
int time;

double  Xout[15], Xsum[15];  /* work arrays */

main()
{
 int driver, mode, font, direct;

 register int i;

 driver = VGA;
 mode = VGAMED;
 initgraph(&driver, &mode, "");

 ip_1a = ip_2a = ip_3a = op_1a = op_2a = op_3a = ip_outa = op_outa = 0;
 ip_1b = ip_2b = ip_3b = op_1b = op_2b = op_3b = ip_outb = op_outb = 0;
 ip_1c = ip_2c = op_c = ip_outc = op_outc = 0;

 cp = 4900;

 printf("\nInput initial temperature of PV : ");
 scanf("%f",&t_1a);
 t_1b = t_1c = t_1d = t_1a;
 printf("\nInput initial temperature of cold leg : ");
 scanf("%f",&t_in_start);
 printf("\nInput final temperature of cold leg : ");
 scanf("%f",&t_in_end);
 printf("\nInput liquid mass in PV : ");
 scanf("%f",&m);
 printf("\nInput initial rate of change of mass of cold leg : ");
 scanf("%f",&m_in_start);
 printf("\nInput final rate of change of mass of cold leg : ");
 scanf("%f",&m_in_end);
 printf("\nInput the initial heat rate : ");
```

```
scanf("%f",&q_start);
printf("\nInput the final heat rate : ");
scanf("%f",&q_end);

/*** Graphics Screen and Axis ***/

clrscr();
rectangle(0, 0, 639, 349);
setcolor(LIGHTCYAN);
setfillstyle(SOLID_FILL, LIGHTCYAN);
floodfill(1, 1, LIGHTCYAN);

setcolor(BLACK);
line(25,10,25,310);
line(25,310,625,310);

settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
outtextxy(2, 10, "375");
outtextxy(2, 85, "350");
outtextxy(2, 160, "325");
outtextxy(2, 235, "300");
outtextxy(2, 310, "275");

/*** System with square threshold ***/

for (time = 0; time <= 100; time ++)
 {
  q = q_start + ((q_end - q_start) * time / 100);
  t_in = t_in_start + ((t_in_end - t_in_start) * time / 100);
  m_in = m_in_start + ((m_in_end - m_in_start) * time / 100);

  if ((temp1 = fopen("temp1.txt", "a")) == NULL)
     {
      printf("The temperature file could not be opened\n");
```

```
        return (1);
      }
    fprintf(temp1,"\n %d  %f %f %f %f", time, t_1a, t_in, m_in, q);
    fclose(temp1);

    /* Hidden Node 1 */
    ip_1a = t_in - t_1a;
    op_1a = 0.5 * (pow(ip_1a, 2));

    /* Hidden Node 2 */
    ip_2a = t_in + m_in - t_1a;
    op_2a = 0.5 * (pow(ip_2a, 2));

    /* Hidden Node 3 */
    ip_3a = m_in;
    op_3a = 0.5 * (pow(ip_3a, 2));

    /* Output Node for A */
    ip_outa = (op_2a/m) - (op_1a/m) - (op_3a/m) + t_1a + (q/(cp*m));
    op_outa = ip_outa;
    t_1a = op_outa;

    setcolor(RED);
    outtextxy((25+6*time), (10+3*(375-t_1a)), "*");
  }

/*** System with NWorks trained square function ***/

for (time = 0; time <= 100; time ++)
  {
   q = q_start + ((q_end - q_start) * time / 100);
   t_in = t_in_start + ((t_in_end - t_in_start) * time / 100);
   m_in = m_in_start + ((m_in_end - m_in_start) * time / 100);
```

```
 if ((temp2 = fopen("temp2.txt", "a")) == NULL)
     {
      printf("The temperature file could not be opened\n");
      return (1);
     }
 fprintf(temp2,"\n %d  %f %f %f %f", time, t_1b, t_in, m_in, q);
 fclose(temp2);

 /* Hidden Node 1 */
 ip_1b = t_in - t_1b;
 Yin[0] = ip_1b / 6;
 square();
 op_1b = 0.5 * Yout[0] * 36;

 /* Hidden Node 2 */
 ip_2b = t_in + m_in - t_1b;
 Yin[0] = ip_2b / 1200;
 square();
 op_2b = 0.5 * Yout[0] * 1440000;

 /* Hidden Node 3 */
 ip_3b = m_in;
 Yin[0] = ip_3b / 1200;
 square();
 op_3b = 0.5 * Yout[0] * 1440000;

 /* Output Node for B */
 ip_outb = (op_2b/m) - (op_1b/m) - (op_3b/m) + t_1b + (q/(cp*m));
 op_outb = ip_outb;
 t_1b = op_outb;

 setcolor(GREEN);
 outtextxy((25+6*time), (10+3*(375-t_1b)), "*");
}
```

```
/*** System with NWorks multiplying function ***/

for (time = 0; time <= 100; time ++)
  {
   q = q_start + ((q_end - q_start) * time / 100);
   t_in = t_in_start + ((t_in_end - t_in_start) * time / 100);
   m_in = m_in_start + ((m_in_end - m_in_start) * time / 100);

   if ((temp3 = fopen("temp3.txt", "a")) == NULL)
       {
        printf("Temperature file 3 could not be opened\n");
        return (1);
       }
   fprintf(temp3,"\n %d  %f  %f  %f  %f", time, t_1c, t_in, m_in, q);
   fclose(temp3);

   ip_1c = t_in - t_1c;
   Yin[0] = ip_1c / -10;
   ip_2c = m_in;
   Yin[1] = ip_2c / 2000;
   mult();
   op_c = Yout[0] * -20000;

   /* Output Node for A */
   ip_outc = (op_c/m) + t_1c + (q/(cp*m));
   op_outc = ip_outc;
   t_1c = op_outc;

   setcolor(WHITE);
   outtextxy((25+6*time), (10+3*(375-t_1c)), "*");
 }

/*** System with NWorks full 1CV model ***/
```

```
for (time = 0; time <= 100; time ++)
  {
   q = q_start + ((q_end - q_start) * time / 100);
   t_in = t_in_start + ((t_in_end - t_in_start) * time / 100);
   m_in = m_in_start + ((m_in_end - m_in_start) * time / 100);

   if ((temp4 = fopen("temp4.txt", "a")) == NULL)
       {
        printf("Temperature file 4 could not be opened\n");
        return (1);
       }
   fprintf(temp4,"\n %d  %f %f %f %f", time, t_1d, t_in, m_in, q);
   fclose(temp4);

   Yin[0] = t_1d;
   Yin[1] = t_in;
   Yin[2] = m_in;
   Yin[3] = q;

   model();

   t_1d = t_1d + Yout[0];

   setcolor(YELLOW);
   outtextxy((25+6*time), (10+3*(375-t_1d)), "*");
  }

 getch();
 restorecrtmode();
}

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

square()
```

```
{
  /* Read and scale input into network */
  Xout[2] = Yin[0] * (0.05);
LAB107:

  /* Generating code for PE 0 in layer 3 */
  Xsum[3]  = 0.0377354249 + 0.0497449599 * Xout[2];

  /* Generating code for PE 1 in layer 3 */
  Xsum[4]  = (-0.0688849837) + (-0.0891667157) * Xout[2];

  /* Generating code for PE 2 in layer 3 */
  Xsum[5]  = (-1.5718521) + (-1.6629894) * Xout[2];

  /* Generating code for PE 0 in layer 3 */
  Xout[3] = sin( Xsum[3] );

  /* Generating code for PE 1 in layer 3 */
  Xout[4] = sin( Xsum[4] );

  /* Generating code for PE 2 in layer 3 */
  Xout[5] = sin( Xsum[5] );

  /* Generating code for PE 0 in layer 4 */
  Xsum[6]  = 0.629029393 + 0.0245846715 * Xout[3]
           + (-0.0388228707) * Xout[4]
           + 1.5541158 * Xout[5];
  Xout[6] = sin( Xsum[6] );

  /* De-scale and write output from network */
  Yout[0] = Xout[6] * 250 + 200;

  /* Generating code for PE 0 in layer 4 */
  return( 0 );
```

```
}

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

mult()
{

   /* Read and scale input into network */
   Xout[2] = Yin[0] * (0.2) + (-1);
   Xout[3] = Yin[1] * (0.2) + (-1);
LAB107:

   /* Generating code for PE 0 in layer 3 */
   Xsum[4]  = (-0.186730966) + (-0.478621244) * Xout[2] +
      0.451036364 * Xout[3];

   /* Generating code for PE 1 in layer 3 */
   Xsum[5]  = 0.831040561 + (-0.629294634) * Xout[2] + 0.601229131 * Xout[3];

   /* Generating code for PE 2 in layer 3 */
   Xsum[6]  = (-0.744259298) + 0.427741736 * Xout[2] + 0.440952927 * Xout[3];

   /* Generating code for PE 0 in layer 3 */
   Xout[4] = sin( Xsum[4] );

   /* Generating code for PE 1 in layer 3 */
   Xout[5] = sin( Xsum[5] );

   /* Generating code for PE 2 in layer 3 */
   Xout[6] = sin( Xsum[6] );

   /* Generating code for PE 0 in layer 4 */
   Xsum[7]  = (-0.143988386) + (-0.72245121) * Xout[4]
          + 0.794204772 * Xout[5] + 1.4647512 * Xout[6];
```

```
  Xout[7] = sin( Xsum[7] );

  /* De-scale and write output from network */
  Yout[0] = Xout[7] * (62.499999) + (50);

  /* Generating code for PE 0 in layer 4 */
  return( 0 );
}

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

model()
{
  /* Read and scale input into network */
  Xout[2] = Yin[0] * (0.0280331816) + (-8.7412243);
  Xout[3] = Yin[1] * (0.04) + (-11);
  Xout[4] = Yin[2] * (0.000266666667) + (-4.3333333);
  Xout[5] = Yin[3] * (1.1764706e-009) + (-3);
LAB107:

  /* Generating code for PE 0 in layer 3 */
  Xsum[6]  = 0.0432699621 + 0.173500374 * Xout[2] + (-0.25696522) * Xout[3]
     + (-0.0618465208) * Xout[4] + (-0.118797004) * Xout[5];

  /* Generating code for PE 1 in layer 3 */
  Xsum[7]  = (-0.167464197) + (-0.180572703) * Xout[2] +
    0.263423711 * Xout[3] + (-0.0826175511) * Xout[4] +
    0.0946433693 * Xout[5];

  /* Generating code for PE 2 in layer 3 */
  Xsum[8]  = (-0.143358409) + 0.515659809 * Xout[2] +
    (-0.159966484) * Xout[3] + (-0.087270394) * Xout[4] +
    (-0.0688505024) * Xout[5];
```

```
/* Generating code for PE 3 in layer 3 */
Xsum[9] = 0.0936020315 + (-0.26459676) * Xout[2] + 0.214385584 * Xout[3]
   + (-0.336718529) * Xout[4] + 0.132838547 * Xout[5];

/* Generating code for PE 4 in layer 3 */
Xsum[10] = 0.0503302142 + (-0.781614959) * Xout[2] + 0.623369515 * Xout[3]
   + (-0.371508271) * Xout[4] + 0.200226009 * Xout[5];

/* Generating code for PE 5 in layer 3 */
Xsum[11] = (-0.0832801238) + (-0.14036642) * Xout[2] +
   0.295333862 * Xout[3] + (-0.14117153) * Xout[4] +
   (-0.150969371) * Xout[5];

/* Generating code for PE 6 in layer 3 */
Xsum[12] = (-0.070078738) + (-0.542544127) * Xout[2] +
   0.145031795 * Xout[3] + (-0.0677929074) * Xout[4] +
   0.128840894 * Xout[5];

/* Generating code for PE 7 in layer 3 */
Xsum[13] = (-0.0182786733) + 0.735283971 * Xout[2] +
   (-0.372215182) * Xout[3] + 0.118579939 * Xout[4] +
   (-0.230088666) * Xout[5];

/* Generating code for PE 0 in layer 3 */
Xout[6] = Xsum[6];

/* Generating code for PE 1 in layer 3 */
Xout[7] = Xsum[7];

/* Generating code for PE 2 in layer 3 */
Xout[8] = Xsum[8];

/* Generating code for PE 3 in layer 3 */
Xout[9] = Xsum[9];
```

```
    /* Generating code for PE 4 in layer 3 */
    Xout[10] = Xsum[10];

    /* Generating code for PE 5 in layer 3 */
    Xout[11] = Xsum[11];

    /* Generating code for PE 6 in layer 3 */
    Xout[12] = Xsum[12];

    /* Generating code for PE 7 in layer 3 */
    Xout[13] = Xsum[13];

    /* Generating code for PE 0 in layer 4 */
    Xout[14]  = 0.133136854 + (-0.237154871) * Xout[6] + 0.120102882 * Xout[7]
       + (-0.314193368) * Xout[8] + 0.570697844 * Xout[9] +
      0.962584555 * Xout[10] + 0.0576984398 * Xout[11] +
      0.581108391 * Xout[12] + (-0.62760669) * Xout[13];

    /* De-scale and write output from network */
    Yout[0] = Xout[14] * (0.96921938) + (-0.365341514);

    /* Generating code for PE 0 in layer 4 */
    return( 0 );
}

/************************** End of Program ***********************/
```

Appendix

Q Code Listing for Full PWR Model using ANN Modules Program

This appendix contains the full program listing for full PWR model using ANN modules program as introduced in Section 6.6.1. The program was written and compiled using Borland Turbo C, Version 2. The program variables are entered from the keyboard and the output is both an ASCII text file and simple graph plot of regional temperature against time. Plotted results from the program are given in Figs 6.54 and 6.55.

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

            Program of coding of full PWR using
                  neural network molecules
          Author : P R Weller     Copyright (c) 1996
          04/07/96 - Version 0.1 - First Version
          10/07/96 - Version 0.2 - NWorks trained mole added

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


#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>

FILE *temp, *reactor;

int num_nodes, time, v, w, x, y, z;
float cp, init_temp, m, m_in, q, t_in;

double Yin[5], Yout[1];     /* Data */
double  Xout[23], Xsum[23]; /* work arrays */

float t_1[30];          /* Array for t1 values */
float t_1old[30];       /* Array for old t1 values */
float mass_in[10];      /* Array for flow rates */
float heat[8];          /* Array for heating rates */
float mass[12];         /* Array for masses */
float power[100][4];    /* Array for reactor power */

main()
{
 mass_in[1] = 0; mass_in[2] = 0; mass_in[3] = 295; mass_in[4] = 5;
 mass_in[5] = 0; mass_in[6] = 300; mass_in[7] = 300;
```

```
mass_in[8] = 0; mass_in[9] = 0;

heat[0] = 0; heat[1] = 5000000; heat[2] = 10000; heat[3] = 25000000;
heat[4] = 50000000;

mass[0] = 50;  mass[1] = 100; mass[2] = 250; mass[3] = 300;
mass[4] = 400; mass[5] = 440; mass[6] = 480; mass[7] = 500;
mass[8] = 1000; mass[9] = 1200; mass[10] = 3000; mass[11] = 10000;

if ((reactor = fopen("power.txt", "r")) == NULL)
  {
   printf("The reactor power file could not be opened\n");
   return (1);
  }
for (z = 0; z <= 99; z++)
  {
   for (y = 0; y <= 3; y++)
       fscanf(reactor," %f ", &power[z][y]);
  }
fclose(temp);

time = 1;
cp = 5000;
w = 10;

printf("\nInput initial temperature of PWR : ");
scanf("%f",&init_temp);
for (z = 0; z <= 29; z ++)
  {
   t_1[z] = init_temp;
   t_1old[z] = init_temp;
  }
for (time = 1; time <= 1000; time++)
  {
```

```
    x = 1;
    v = floor(time / 10);

    /* node 1 */
    m = mass[8]; t_in = t_1old[1]; m_in = -mass_in[1]; q = heat[0];
    x++;

    /* node 2 */
    m = mass[11]; t_in = t_1old[12]; m_in = mass_in[2]; q = heat[0];
/*    dir_equ();   */
    model();
    x++;

    /* node 3 */
    m = mass[2]; q = heat[0];
    m_in = mass_in[1] + mass_in[3] + mass_in[4] + mass_in[9];
    t_in = ((t_1old[1] * mass_in[1]) + (t_1old[7] * mass_in[3])
          + (t_1old[8] * mass_in[4]) + (t_1old[23] * mass_in[9])) / m_in;
/*    dir_equ();   */
    model();
    x++;

    /* node 4 */
    m = mass[7]; m_in = mass_in[5] + mass_in[6]; q = heat[0];
    t_in = ((t_1old[11] * mass_in[5]) + (t_1old[14] * mass_in[6])) / m_in;
/*    dir_equ();   */
    model();
    x++;

    /* node 5 */
    m = mass[1]; t_in = t_1old[4]; m_in = mass_in[3]; q = power[v][0];
/*    dir_equ();   */
    model();
    x++;
```

```
    /* node 6 */
    m = mass[4]; t_in = t_1old[5]; m_in = mass_in[3]; q = heat[0];
/*    dir_equ();   */
    model();
    x++;

    /* node 7 */
    m = mass[1]; t_in = t_1old[6]; m_in = mass_in[3]; q = power[v][1];
/*    dir_equ();   */
    model();
    x++;

    /* node 8 */
    m = mass[3]; m_in = 2 * mass_in[4]; q = heat[0];
    t_in = ((t_1old[4] * mass_in[4]) + (t_1old[24] * mass_in[4])) / m_in;
/*    dir_equ();   */
    model();
    x++;

    /* node 9 */
    m = mass[1]; t_in = t_1old[23]; m_in = mass_in[9]; q = heat[0];
/*    dir_equ();   */
    model();
    x++;

    /* node 10 */
    m = mass[4]; t_in = t_1old[9]; m_in = mass_in[9]; q = -heat[2];
/*    dir_equ();   */
    model();
    x++;

    /* node 11 */
    m = mass[1]; t_in = t_1old[10]; m_in = mass_in[9]; q = heat[0];
/*    dir_equ();   */
```

```
    model();
    x++;

    /* node 12 */
    m = mass[5]; t_in = t_1old[3]; m_in = mass_in[6]; q = heat[0];
/*    dir_equ();  */
    model();
    x++;

    /* node 13 */
    m = mass[6]; t_in = t_1old[26]; m_in = mass_in[6];
    q = -heat[1] * (t_1old[13] - t_1old[18]);
    /* q value depends on temp difference: UA.(t1-t2) */
/*    dir_equ();  */
    model();
    x++;

    /* node 14 */
    m = mass[9]; t_in = t_1old[13]; m_in = mass_in[6]; q = heat[0];
/*    dir_equ();  */
    model();
    x++;

    /* node 15 */
    m = mass[5]; t_in = t_1old[23]; m_in = mass_in[7]; q = heat[0];
/*    dir_equ();  */
    model();
    x++;

    /* node 16 */
    m = mass[6]; t_in = t_1old[27]; m_in = mass_in[7];
    q = -heat[1] * (t_1old[16] - t_1old[19]);
    /* q value depends on temp difference: UA.(t1-t2) */
/*    dir_equ();  */
```

```
     model();
     x++;

     /* node 17 */
     m = mass[9]; t_in = t_1old[16]; m_in = mass_in[7]; q = heat[0];
/*     dir_equ();  */
     model();
     x++;

     /* node 18 */
     m = mass[10]; t_in = t_1old[18]; m_in = 0;
     q = (heat[1] * (t_1old[13] - t_1old[18])) - heat[4];
     /* q value depends on temp difference: UA.(t1-t2) */
/*     dir_equ();  */
     model();
     x++;

     /* node 19 */
     m = mass[10]; t_in = t_1old[19]; m_in = 0;
     q = (heat[1] * (t_1old[16] - t_1old[19])) - heat[4];
     /* q value depends on temp difference: UA.(t1-t2) */
/*     dir_equ();  */
     model();
     x++;

     /* node 20 */
     m = mass[10]; t_in = t_1old[20]; m_in = 0; q = heat[2];
/*     dir_equ();  */
     model();
     x++;

     /* node 21 */
     m = mass[1]; t_in = t_1old[24]; m_in = mass_in[3]; q = power[v][2];
/*     dir_equ();  */
```

```
    model();
    x++;

    /* node 22 */
    m = mass[1]; t_in = t_1old[25]; m_in = mass_in[3]; q = power[v][3];
/*    dir_equ();   */
    model();
    x++;

    /* node 23 */
    m = mass[2]; m_in = mass_in[3] + mass_in[4]; q = heat[0];
    t_in = ((t_1old[8] * mass_in[4]) + (t_1old[22] * mass_in[3])) / m_in;
/*    dir_equ();   */
    model();
    x++;

    /* node 24 */
    m = mass[7]; m_in = mass_in[7] + mass_in[8]; q = heat[0];
    t_in = ((t_1old[14] * mass_in[8]) + (t_1old[17] * mass_in[7])) / m_in;
/*    dir_equ();   */
    model();
    x++;

    /* node 25 */
    m = mass[4]; t_in = t_1old[21]; m_in = mass_in[3]; q = heat[0];
/*    dir_equ();   */
    model();
    x++;

    /* node 26 */
    m = mass[1]; t_in = t_1old[12]; m_in = mass_in[6]; q = -heat[0];
/*    dir_equ();   */
    model();
    x++;
```

```
    /* node 27 */
    m = mass[1]; t_in = t_1old[15]; m_in = mass_in[7]; q = -heat[0];
/*    dir_equ();   */
    model();
    x++;

    /* node 28 */
    m = mass[0]; t_in = t_1old[28]; m_in = 0; q = heat[0];
/*    dir_equ();   */
    model();
    x++;

    /* node 29 */
    m = mass[0]; t_in = t_1old[29]; m_in = 0; q = heat[0];
/*    dir_equ();   */
    model();
    x++;

    if (w == 10)
       {
        if ((temp = fopen("temp2.txt", "a")) == NULL)
          {
           printf("The temperature file could not be opened\n");
           return (1);
          }
        fprintf(temp,"\n %d  ", time/10);
        for (z = 1; z <= 29; z++)
          fprintf(temp," %f ", t_1[z]);
        fclose(temp);
        w = 0;
       }
    w++;
    for (z = 1; z <= 29; z++)
       t_1old[z] = t_1[z];
```

```
   }
   for (z = 1; z <= 29; z++)
     printf(" %f ", t_1[z]);
}

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

dir_equ()
{
 float ip_1, ip_2, ip_3, op_1, op_2, op_3, ip_out, op_out;
 ip_1 = ip_2 = ip_3 = op_1 = op_2 = op_3 = ip_out = op_out = 0;

 /* Hidden Node 1 */
 ip_1 = t_in - t_1old[x];
 op_1 = 0.5 * (pow(ip_1, 2));

 /* Hidden Node 2 */
 ip_2 = t_in + m_in - t_1old[x];
 op_2 = 0.5 * (pow(ip_2, 2));

 /* Hidden Node 3 */
 ip_3 = m_in;
 op_3 = 0.5 * (pow(ip_3, 2));

 /* Output Node */
 ip_out = 0.1*((op_2/m) - (op_1/m) - (op_3/m)) + t_1old[x] + 0.1*(q/(cp*m));
 op_out = ip_out;

 t_1[x] = op_out;
}

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

model()
```

```
{
   Yin[0] = t_1old[x];
   Yin[1] = t_in;
   Yin[2] = m_in;
   Yin[3] = q;
   Yin[4] = m;

   Xout[2] = Yin[0] * (0.018946897) + (-4.8008711);
   Xout[3] = Yin[1] * (0.02) + (-5);
   Xout[4] = Yin[2] * (0.00677966102) + (-1.0338983);
   Xout[5] = Yin[3] * (4e-007) + (-1);
   Xout[6] = Yin[4] * (0.000689655172) + (-1.0689655);
LAB107:

   /* Generating code for PE 0 in layer 3 */
   Xsum[7]  = 0.0781149119 + (-0.0729916766) * Xout[2] +
      0.0706214681 * Xout[3] + (-0.0290338751) * Xout[4] +
      (-0.0163256004) * Xout[5] + 0.000542022928 * Xout[6];

   /* Generating code for PE 1 in layer 3 */
   Xsum[8]  = (-0.017377425) + 0.0524926223 * Xout[2] +
      (-0.0471051335) * Xout[3] + 0.01708184 * Xout[4] +
      (-0.0169406552) * Xout[5] + 0.00986460038 * Xout[6];

   /* Generating code for PE 2 in layer 3 */
   Xsum[9]  = 0.207093343 + (-0.202051669) * Xout[2] + 0.240440086 * Xout[3]
      + (-0.0796540454) * Xout[4] + (-0.0218407735) * Xout[5] +
      0.0485103354 * Xout[6];

   /* Generating code for PE 0 in layer 3 */
   Xout[7] = tanh( Xsum[7] );

   /* Generating code for PE 1 in layer 3 */
   Xout[8] = tanh( Xsum[8] );
```

```
    /* Generating code for PE 2 in layer 3 */
    Xout[9] = tanh( Xsum[9] );

    /* Generating code for PE 0 in layer 4 */
    Xsum[10]  = 0.718770266 + 0.133371726 * Xout[7] + (-0.0482432991) * Xout[8]
        + 0.388276219 * Xout[9];
    Xout[10] = tanh( Xsum[10] );

    /* De-scale and write output from network */
    Yout[0] = Xout[10] * (10.169) + (-6.8647995);

    t_1[x] = t_1old[x] + Yout[0];

    /* Generating code for PE 0 in layer 5 */
    return( 0 );
}

/************************** End of Program ***********************/
```

Appendix R

Results of Training an ANN for Temperature Prediction with Different Regional Masses

This appendix contains the full results of all the ANNs trained to predict PWR regional temperature for different regional masses as discussed in Section 6.6.1. The ANNs were all designed to predict the regional temperature one step ahead. The inputs for the ANNs are the five plant variable developed from the energy equation in Section 6.5. For various training data a set of one and two hidden layer ANNs were developed. The number of nodes in the hidden layer was varied from a maximum of 15 to a minimum of 2, with a test ANN consisting of a single node in the hidden layer. All the ANN training was repeated four times to avoid possible local minima. The value recorded in each case is the lowest RMS error for the training. All ANNs were trained for 100,000 cycles of presentation of the training set and then a further 20,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing.

In the following tables the '*' in the training/test column indicates tra for the training file and tes for the test file, ie an ANN using squ_*4.nna has a training file named squ_tra4.nna and a test file named squ_tes4.nna.

Results of Training an ANN to Model 1CV with Different Regional Mass

Filename	Size of ANN	Train/Test	Transfer	RMS Error
pwr1.nnd	5-9-1	pwr_cv*.nna	Sine	0.0926
pwr1a.nnd	5-15-1	pwr_cv*.nna	Linear	0.0918
pwr1b.nnd	5-15-1	pwr_cv*.nna	TanH	0.0611
pwr1c.nnd	5-15-8-1	pwr_cv*.nna	TanH	0.0980
pwr1d.nnd	5-15-8-1	pwr_cv*.nna	TanH	0.0974
pwr1e.nnd	5-10-1	pwr_cv1*.nna	TanH	0.0740
pwr1f.nnd	5-10-5-1	pwr_cv1*.nna	TanH	0.0794
pwr1g.nnd	5-15-8-1	pwr_cv1*.nna	TanH	0.0873
pwr1h.nnd	5-10-5-1	pwr_cv1*.nna	Sine	0.0753
pwr1i.nnd	5-7-3-1	pwr_cv1*.nna	Sine	0.0847
pwr3a.nnd	5-7-3-1	pwr_cv3*.nna	Sine	0.0883
pwr3b.nnd	5-10-5-1	pwr_cv3*.nna	Sine	0.0885
pwr3c.nnd	5-7-3-1	pwr_cv3*.nna	Sine	0.0881
pwr3d.nnd	5-7-3-1	pwr_cv3*.nna	Linear	0.0884
pwr3e.nnd	5-8-4-1	pwr_cv3*.nna	TanH	0.0879
pwr3f.nnd	5-3-1	pwr_cv3*.nna	TanH	0.0871
pwr3g.nnd	5-3-1	pwr_cv3*.nna	Sine	0.0876
pwr4a.nnd	5-3-1	pwr_cv3*.nna	Sine	0.1054
pwr4b.nnd	5-8-4-1	pwr_cv3*.nna	Sine	0.0725
pwr4c.nnd	5-6-1	pwr_cv3*.nna	TanH	0.0693
pwr4d.nnd	5-4-1	pwr_cv3*.nna	TanH	0.0689
pwr4e.nnd	5-3-1	pwr_cv3*.nna	Sine	0.0741
pwr4f.nnd	5-3-1	pwr_cv3*.nna	Sigmoid	0.0566
pwr4g.nnd	5-2-1	pwr_cv3*.nna	Sigmoid	0.0563
pwr4h.nnd	5-2-1	pwr_cv3*.nna	Sigmoid	0.0556

Filename	Size of ANN	Train/Test	Threshold	RMS Error
pwr4i.nnd	5-1-1	pwr_cv3*.nna	Sigmoid	0.0550
pwr4j.nnd	5-1-1	pwr_cv3*.nna	Sigmoid	0.0543
pwr4k.nnd	5-2-1	pwr_cv3*.nna	Sigmoid	0.0562
pwr4l.nnd	5-3-1	pwr_cv3*.nna	Sigmoid	0.0566
pwr4m.nnd	5-3-1	pwr_cv3*.nna	Sigmoid	0.0569
pwr4n.nnd	5-3-1	pwr_cv3*.nna	Linear	0.2964
pwr4z.nnd	5-5-1	pwr_cv4*.nna	Sigmoid	0.1071
pwr5a.nnd	4-3-1	pwr_cv5*.nna	Sigmoid	0.0307
pwr5b.nnd	4-4-1	pwr_cv5*.nna	Sigmoid	0.0300
pwr5c.nnd	4-5-1	pwr_cv5*.nna	Sigmoid	0.0308
pwr6a.nnd	10-4-1	pwr_cv6*.nna	Sigmoid	0.0372
pwr6b.nnd	10-5-1	pwr_cv6*.nna	Sigmoid	0.0298
pwr6b1.nnd	6-7-1	pwr_cv6b*.nna	Sigmoid	0.0629
pwr6b2.nnd	6-4-1	pwr_cv6b*.nna	Sigmoid	0.0595
pwr6b3.nnd	6-5-1	pwr_cv6b*.nna	Sigmoid	0.0604
pwr6b4.nnd	6-5-1	pwr_cv6b*.nna	Sigmoid	0.0604
pwr6c.nnd	10-6-1	pwr_cv6*.nna	Sigmoid	0.0278
pwr6d.nnd	10-7-1	pwr_cv6*.nna	Sigmoid	0.0252
pwr6d1.nnd	8-7-1	pwr_cv6a*.nna	Sigmoid	0.0399
pwr6d2.nnd	8-4-1	pwr_cv6a*.nna	Sigmoid	0.0421
pwr6d3.nnd	8-6-3-1	pwr_cv6a*.nna	Sigmoid	0.0436
pwr6d4.nnd	8-10-1	pwr_cv6a*.nna	Sigmoid	0.0401
pwr6e.nnd	10-8-1	pwr_cv6*.nna	Sigmoid	0.0300
pwr6f.nnd	10-7-1	pwr_cv6*.nna	Sigmoid	0.0268
pwr7a.nnd	5-5-1	pwr_cv7*.nna	Sigmoid	0.0448
pwr7b.nnd	5-3-1	pwr_cv7*.nna	Sigmoid	0.0469
pwr7c.nnd	5-3-1	pwr_cv7*.nna	Sigmoid	0.0471
pwr8a.nnd	4-4-1	pwr_cv8*.nna	Sigmoid	0.0472
pwr8b.nnd	4-4-1	pwr_cv8a*.nna	Sigmoid	0.0528

Filename	Size of ANN	Train/Test	Threshold	RMS Error
pwr8c.nnd	4-10-5-1	pwr_cv8a*.nna	Sigmoid	0.0474
pwr8d.nnd	4-4-1	pwr_cv8a*.nna	Sigmoid	0.0529
pwr8e.nnd	4-5-1	pwr_cv8a*.nna	Sigmoid	0.0530
pwr8f.nnd	4-3-1	pwr_cv8a*.nna	Sigmoid	0.0531
pwr8g.nnd	3-5-1	pwr_cv8b*.nna	Sigmoid	0.0461
pwr8h.nnd	3-5-1	pwr_cv8b*.nna	Sigmoid	0.0773

The following table gives the full details of the training data mentioned above.

No of Cases	T_{in} Start	T_{in} Finish	M_{in} Start	M_{in} Finish	Q Start	Q Finish
tempa.nna						
200	250	250	17400	17400	3.4×10^9	3.4×10^9
131	286	286	20000	20000	3.4×10^9	3.4×10^9
temp1.nna						
200	250	250	17400	17400	3.4×10^9	3.4×10^9
temp2.nna						
150	300	300	17400	17400	3.4×10^9	3.4×10^9
temp3.nna						
215	286	286	12500	12500	3.4×10^9	3.4×10^9
temp4.nna						
160	286	286	15000	15000	3.4×10^9	3.4×10^9
temp6.nna						
131	286	286	20000	20000	3.4×10^9	3.4×10^9
temp.nna						
200	250	250	17400	17400	3.4×10^9	3.4×10^9
151	300	300	17400	17400	3.4×10^9	3.4×10^9
216	286	286	12500	12500	3.4×10^9	3.4×10^9
161	286	286	15000	15000	3.4×10^9	3.4×10^9
132	286	286	20000	20000	3.4×10^9	3.4×10^9

No of Cases	T_{in} Start	T_{in} Finish	M_{in} Start	M_{in} Finish	Q Start	Q Finish
temp11.nna						
101	286	300	17400	12500	3.4×10^9	0
101	286	250	17400	20000	0	3.4×10^9
temp12.nna						
101	286	300	17400	15000	0	3.4×10^9
101	286	250	17400	20000	3.4×10^9	0
101	286	300	17400	12500	3.4×10^9	3.6×10^9
temp13.nna						
101	286	300	17400	15000	1.7×10^9	3.4×10^9
101	286	250	17400	20000	3.4×10^9	1.7×10^9
101	286	300	17400	12500	3.2×10^9	3.4×10^9

Appendix

S Results of Training a Diagnostic ANN

This appendix contains the full results of all the ANNs trained to diagnose the condition of a PWR as discussed in Section 73. The majority of the ANNs used a single step of values of PWR variables as inputs, but two time steps, with 134 inputs, were also considered. For each combination of transient pairs a set of both one and two hidden layer ANNs were developed. The number of nodes in the hidden layers varied from a maximum of 80 to a minimum of 10. All the ANN training was repeated four times to avoid possible local minima. The value recorded in each case is the lowest RMS error for the training. All ANNs were trained for 60,000 cycles of presentation of the training set and then a further 20,000 iterations with presentation of the test set every 100 cycles. The best performing ANN in terms of RMS error was saved for further testing.

The '*' in the Train/Test column of the following tables indicates tr for the training files and te for the test file, ie an ANN using the training data test3*.nna uses the training set test3tr.nna and the test set test3te.nna.

Filename	Size of ANN	Train/Test	Threshold	RMS Error
test11c.nnd	67-30-6	test11*.nna	TanH	0.1263
test11d.nnd	67-50-25-6	test11*.nna	TanH	0.1407
test11z.nnd	67-30-15-6	test11*.nna	TanH	0.0307
test12a.nnd	6-6-6	test12*.nna	TanH	0.2923
test12b.nnd	6-20-10-6	test12*.nna	TanH	0.2414
test12c.nnd	6-30-15-6	test12*.nna	TanH	0.2512
test12d.nnd	6-10-10-6	test12*.nna	TanH	0.2436
test13a.nnd	67-30-15-6	test13*.nna	TanH	0.1388
test13a1.nnd	67-30-15-6	test13*.nna	TanH	0.1431
test13aa.nnd	67-30-15-6	test13*.nna	TanH	0.0905
test13b.nnd	67-40-20-6	test13*.nna	TanH	0.1454
test13ba.nnd	67-30-15-6	test13*.nna	TanH	0.0909
test13bb.nnd	67-30-15-6	test13b*.nna	TanH	0.0883
test13c.nnd	67-50-25-6	test13*.nna	TanH	0.1514
test13ca.nnd	67-30-15-6	test13c*.nna	TanH	0.0868
test13d.nnd	67-25-12-6	test13*.nna	TanH	0.1386
test13da.nnd	67-30-15-6	test13d*.nna	TanH	0.0886
test13db.nnd	67-21-6	test13d*.nna	TanH	0.1667
test14a.nnd	134-30-6	test14*.nna	TanH	0.1489
test14b.nna	134-60-30-6	test14*.nna	TanH	0.1192
test14c.nna	134-80-40-6	test14*.nna	TanH	0.1226
test14d.nna	134-20-10-6	test14*.nna	TanH	0.1031
test14e.nna	134-20-10-6	test14*.nna	Sigmoid	0.1021
test14f.nna	134-30-6	test14*.nna	Sigmoid	0.1015
test14g.nna	134-40-6	test14*.nna	Sigmoid	0.1025