

**CONDITION MONITORING OF
HYDRAULIC SYSTEMS USING NEURAL
NETWORK FOR DATA ANALYSIS**

Feng Yu

***A thesis submitted in partial fulfilment of the requirements of
Napier University for the degree of Doctor of Philosophy***

School of Engineering

September 2006

Feng Yu

Acknowledgements

I would like to express my deep gratitude to my director of studies and supervisor, Dr. Naren K. Gupta. Without his continued trust, support, understanding and encouragement, I would not have achieved what I have got. I also would like to thank my second supervisor, Mr. Jay Hoy, for his kind help.

I offer my sincere thanks to the following staff within the School of Engineering, Napier University, Prof. Jorge Kubie, Dr. John Sharp, Prof. Abid Almaini, Prof. Tariq Muneer, and Mr. Alan Davidson, who have given me generous help.

Big thanks go to my parents, two sisters and my beloved brother. I would also like to thank my daughter and my husband for their patience and encouragement throughout the duration of this project.

Finally, I would like to thank the School of Engineering for providing the funding for this research project.

Declaration

The work in this thesis is based on the research carried out at the School of Engineering, Napier University, Edinburgh UK. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Feng Yu

Abstract

Condition monitoring of engineering processes or equipment has become of paramount importance as there is a growing need for improved performance, reliability, safety and more efficient maintenance. Condition monitoring in railway industry as a whole covers a very wide field. To restrict the field we have confined ourselves to the non-intrusive monitoring of hydraulic systems. This thesis is mainly concerned with the investigation of the non-intrusive method based on ultrasonic concepts and neural networks for rapid condition monitoring and/or fault diagnosis of the hydraulic systems.

A comparison between diagnosing hydraulic systems and electric systems is made. The location of faults in hydraulic systems is more difficult. The key to fault finding in hydraulic systems is the location of pressure. The development of pressure measurement instruments is reviewed. In case of trouble-shooting hydraulic systems, pressure readings are often required to be taken at several temporary locations. Since the hydraulic system is fully sealed, the direct measurement instruments can not be practically utilised for this purpose unless they are built-in during the production stage of the system. Instead, the indirect pressure measurement systems can be very helpful for rapid diagnosis of hydraulic systems. The new approach is a combination of the acoustic effect of the fully sealed oil inside the pipe and the penetrating capability of the ultrasonic waves. The ultrasonic wave energy enters the interior of the hydraulic piping and

passes through the contained fluid, of which the pressure is being measured.

Two modelling approaches for this non-intrusive pressure monitoring system have been presented based on FLNN and MLP respectively. They offer the ability to establish the direct and inverse models. For both methods the maximum relative error (%FS) achieved for either the direct model or the inverse model is well within 2 %FS in our case studies. However, compared to the MLP, the FLNN provides a reduced cost of computational complexity.

The novel non-intrusive measurement of hydraulic pressure based on ultrasonic concepts offers the capability of making pressure measurements for trouble-shooting without intruding into the pipe. It is specifically designed for rapid diagnosis of hydraulic equipment, where the conventional measurement instruments fail to make the necessary pressure readings within the sealed pipes. This has the advantage of not having an effect on the condition of the sealed hydraulic system and also of assisting rapid trouble-shooting to save time and cost. Testing the pipes with such a non-intrusive technique is of great interest to all metal pipe related industries for the provision of no disruption to pipe operations.

Contents

Title Page	i
Acknowledgements	ii
Declaration	iii
Abstract	iv
Contents	vi
List of Abbreviations	xi
List of Figures	xiii
List of Tables	xvii
1 Introduction	1
1.1 The need for condition monitoring.....	1
1.2 Monitoring techniques.....	4
1.3 Condition monitoring of railway equipment.....	8
1.3.1 Overview of maintenance in railways.....	8
1.3.2 Motivation for condition-based maintenance.....	9
1.3.3 Condition monitoring of railway equipment.....	10
1.4 Fault finding of hydraulic systems.....	12
1.4.1 Applications of hydraulic systems.....	12
1.4.2 Advantages of using hydraulic systems.....	14
1.4.3 Fault finding in hydraulic systems.....	14

1.5	Proposed research project.....	15
2	Existing Pressure Measuring Methods for Hydraulic Systems	19
2.1	Introduction.....	19
2.2	Existing pressure measurement techniques for hydraulic systems.....	20
2.2.1	The conventional measuring techniques.....	20
2.2.2	External pressure measurement techniques.....	22
2.3	Summary.....	26
3	Novel Non-Intrusive Pressure Measuring Method for Monitoring Hydraulic Systems	28
3.1	Introduction.....	28
3.2	Measurement principle of the proposed approach.....	30
3.3	Design of the monitoring system.....	33
3.4	Summary.....	35
4	FLNN-Based Modelling of Non-Intrusive Monitoring System	37
4.1	Statement of problem.....	37
4.2	Applications of ANN in measurement and instrumentation....	38

4.3	Proposed approach for modelling of non-intrusive monitoring system.....	43
4.3.1	Theoretical background.....	43
4.3.2	Proposed approach for neural modelling of monitoring system.....	51
4.3.3	Selection of ANNs.....	55
4.4	Principle of system modelling based on FLNN.....	56
4.4.1	General structure of FLNN.....	56
4.4.2	Mathematical algorithm for FLNN.....	58
4.4.3	Principle of system modelling based on FLNN.....	60
4.5	Direct modelling of monitoring system with FLNN and results.....	62
4.6	Inverse modelling of monitoring system with FLNN and results.....	65
4.7	Summary.....	69

5 MLP-Based Modelling of the Non-Intrusive Monitoring System —

Part I		70
5.1	Neuron model.....	70
5.1.1	Single-input neuron.....	70
5.1.2	Transfer functions.....	72

5.1.3	Multiple-input neuron.....	74
5.2	Network Architectures.....	77
5.2.1	A layer of neurons.....	77
5.2.2	Multiple layers of neurons.....	80
5.3	Back-propagation algorithm.....	82
5.3.1	Steepest descent algorithm.....	82
5.3.2	Newton's method.....	83
5.3.3	The back-propagation algorithm.....	85
5.3.4	Summary of the back-propagation algorithm.....	95
5.4	Summary.....	98

6 MLP-Based Modelling of the Non-Intrusive Monitoring System —

Part II 99

6.1	Principle of system modelling based on multilayer networks...	99
6.1.1	Three-layer feed-forward network.....	99
6.1.2	Levenberg-Marquardt algorithm.....	105
6.2	Implementation of direct modelling with MLP for non-intrusive pressure monitoring system.....	117
6.2.1	Flow chart of direct modelling with MLP.....	117
6.2.2	Results of the direct model with 1-5-1 structure.....	122

6.2.3	Results of the direct model with 1-6-1 structure.....	130
6.2.4	Results of the direct model with 1-7-1 structure.....	135
6.3	Implementation of Inverse Modelling with MLP for Non-Intrusive Pressure Monitoring System.....	140
6.3.1	Results of the inverse model with 1-5-1 structure.....	140
6.3.2	Results of the inverse model with 1-6-1 structure.....	145
6.3.3	Results of the inverse model with 1-7-1 structure.....	150
6.4	Discussion on Results of Modelling with MLP.....	155
6.5	Comparison between two modelling approaches.....	156
6.6	Summary.....	157
7	Conclusions and Future Work	160
7.1	Conclusions.....	160
7.2	Future Work.....	165
7.2.1	Further development of non-intrusive pressure monitoring system.....	165
7.2.2	Vibrating-wire sensing technique.....	166
8	References	167
9	Appendix-Publications	183

List of Abbreviations

ADC	analogue to digital converter
AE	acoustic emission
ANN	artificial neural network
B-P	back-propagation
CBM	condition based maintenance
CM	condition monitoring
DAC	digital to analogue converter
DC	direct current
Det	determinant
E ² PROM	electrically erasable programmable read only memory
EPMS	external pressure measurement system
EPROM	erasable & programmable read only memory
FLNN	functional link neural network
FS	full scale
GE	General Electric
GM	General Motors

IRT	infrared thermography
L-M	Levenberg-Marquard
L-M B-P	Levenberg-Marquard backpropagation
LMS	least-mean-square
LVDT	linear variable differential transformer
MLP	multilayer perceptron
OCM	on condition maintenance
PC	personal computer
PM	predictive maintenance
RAM	random access memory
rms	root mean square
SRAM	static random access memory
VA	vibration analysis

List of Figures

Fig.2.1 Schematic diagram of conventional methods for measurement of fluid pressure	21
Fig.3.1 Schematic diagram of the non-intrusive measuring method	32
Fig.3.2 Block diagram of the monitoring system.....	34
Fig.3.3 Flow chart of the main program	35
Fig.4.1 Pressure monitoring system under influence of temperature	38
Fig.4.2 Block diagram of supervised learning	49
Fig. 4.3 Neural network model	54
Fig. 4.4 The general structure of the FLNN.....	57
Fig.4.5 Schematic diagram of the direct modelling.	64
Fig.4.6 Relative error (%FS) of estimated change in velocity against the expected variation of velocity.....	64
Fig.4.7 Schematic diagram of the inverse modelling process	67
Fig.4.8 Error (%FS) of estimated change in pressure against the expected variation in pressure.	68
Fig.4.9 Implementation scheme of on-line measurement	68
Fig.5.1 Single-input neuron	71

Fig.5.2 Hard-limit transfer function.....	72
Fig.5.3 Linear transfer function	73
Fig.5.4 Log-sigmoid transfer function	74
Fig.5.5 Multiple-input neuron.....	75
Fig.5.6 Neuron with R inputs, abbreviated notation.....	77
Fig.5.7 Layer of S neurons.....	78
Fig.5.8 Neuron with R inputs, abbreviated notation	79
Fig. 5.9 Three-layer feed-forward neural network.....	81
Fig.5.10 Three-layer feed-forward network, abbreviated notation	82
Fig.5.11 M layer feed-forward network	86
Fig.5.12 Operation of the back-propagation algorithm.....	96
Fig.6.1 Diagram of a three-layer feed-forward network	102
Fig.6.2 The Flow chart of direct modelling with MLP	125
Fig.6.3 Training parameters in progress for the direct model with 1-5-1 structure.....	126
Fig.6.4 Training performance of the direct model with 1-5-1 structure.....	127
Fig.6.5 Relative error (%FS) of the direct model with 1-5-1 structure for the training data.....	128
Fig.6.6 Relative error (%FS) of the direct model with 1-5-1 structure for the test data.....	129

Fig.6.7 Training parameters in progress for the direct model with 1-6-1 structure.....	131
Fig.6.8 Training performance of the direct model with 1-6-1 structure.....	132
Fig.6.9 Relative error (%FS) of the direct model with 1-6-1 structure for the training data	133
Fig.6.10 Relative error (%FS) of the direct model with 1-6-1 structure for the test data.....	134
Fig.6.11 Training parameters in progress for the direct model with 1-7-1 structure.....	136
Fig.6.12 Training performance of the direct model with 1-7-1 structure....	137
Fig.6.13 Relative error (%FS) of the direct model with 1-7-1 structure for the training data	138
Fig.6.14 Relative error (%FS) of the direct model with 1-7-1 structure for the test data.....	139
Fig.6.15 Training parameters in progress for the inverse model with 1-5-1 structure.....	141
Fig.6.16 Training performance of the inverse model with 1-5-1 structure..	142
Fig.6.17 Relative error (%FS) of the inverse model with 1-5-1 structure for the training data	143
Fig.6.18 Relative error (%FS) of the inverse model with 1-5-1 structure for the test data	144

Fig.6.19 Training parameters in progress for the inverse model with 1-6-1 structure.....	146
Fig. 6.20 Training performance of the inverse model with 1-6-1 structure..	147
Fig.6.21 Relative error (%FS) of the inverse model with 1-6-1 structure for the training data	148
Fig.6.22 Relative error (%FS) of the inverse model with 1-6-1 structure for the test data	149
Fig.6.23 Training parameters in progress for the inverse model with 1-7-1 structure.....	151
Fig.6.24 Training performance of the inverse model with 1-7-1 structure..	152
Fig.6.25 Relative error (%FS) of the inverse model with 1-7-1 structure for the training data	153
Fig.6.26 Relative error (%FS) of the inverse model with 1-7-1 structure for the test data	154

List of Tables

Tab.6.1	The weights and biases of the direct model with 1-5-1 structure...	127
Tab.6.2	Computed results of the direct model with 1-5-1 structure for training data (Unit: m/s)	128
Tab.6.3	Computed results of the direct model with 1-5-1 structure for the test data (Unit: m/s)	129
Tab.6.4	The weights and biases of the direct model with 1-6-1 structure...	132
Tab.6.5	Computed results of the direct model with 1-6-1 structure for training data (Unit: m/s)	133
Tab.6.6	Computed results of the direct model with 1-6-1 structure for the test data (Unit: m/s)	134
Tab.6.7	The weights and biases of the direct model with 1-7-1 structure...	137
Tab.6.8	Results of the direct model with 1-7-1 structure for training data (Unit: m/s)	138
Tab.6.9	Computed results of the direct model with 1-7-1 structure for the test data (Unit: m/s)	139
Tab.6.10	The weights and biases of the inverse model with 1-5-1 structure.....	142

Tab.6.11 Results of the inverse model with 1-5-1 structure for training data (Unit: MPa).....	143
Tab.6.12 Computed results of the inverse model with 1-5-1 structure for test data (Unit: MPa)	144
Tab.6.13 The weights and bias of the inverse model with 1-6-1 structure..	147
Tab.6.14 Computed results of the inverse model with 1-6-1 structure for training data (Unit: MPa).....	148
Tab.6.15 Results of the inverse model with 1-6-1 structure for test data (Unit: MPa)	149
Tab.6.16 The weights and biases of the inverse model with 1-7-1 structure.....	152
Tab.6.17 Results of the inverse model with 1-7-1 structure for training data (Unit: MPa).....	153
Tab.6.18 Results of the inverse model with 1-7-1 structure for test data (Unit: MPa)	154

1 Introduction

1.1 The need for condition monitoring

Condition monitoring (CM) has developed into an exact science and is still advancing rapidly. This is largely due to the introduction of modern low cost electronics, intelligent (microprocessor-based) sensing devices and data capture equipment, and its successful application to many industries, including processing, services and manufacture [1]-[4].

Condition monitoring of engineering processes or equipment has become of paramount importance as there is a growing need for improved performance, reliability, safety and more efficient maintenance [5] [90]. Condition monitoring is becoming an essential element of any pro-active maintenance management strategy.

Three main types of condition monitoring are listed here: CM while assets operate; CM during non-intrusive shutdown inspections; CM during intrusive shutdown inspections.

The corresponding predictive maintenance is commonly called: on condition maintenance (OCM) or condition based maintenance (CBM), and strategies to predict functional failure based on condition deterioration. CM, OCM and CBM are usually considered as the same type of strategy. In contrast, preventive maintenance covers strategies where components are changed irrespective of their conditions, and decisions are made based purely on the age of the components, not on their conditions. The distinction between predictive maintenance and preventive maintenance often leads to confusion, when considering planned intrusive shutdowns. During such shutdowns, the tasks are usually a combination of predictive tasks (condition monitoring) and the change of components for overhauled or new components (preventive maintenance) irrespective of their conditions. The intention of an intrusive shutdown is normally the predominance of condition based tasks [6].

We will briefly examine some of the advantages to be gained from condition monitoring. Three different types of action are considered: (1) breakdown maintenance; (2) fixed time maintenance; (3) maintenance on the basis of condition.

Plan (1) demands no more than a 'run it until it breaks then replace it' strategy, while method (2) may or may not involve a limited degree of monitoring. The final plan requires a definite commitment to monitoring. It is apparent that careful thought must be given to the most appropriate form of maintenance planning. For example, breakdown maintenance can only be effective when there is a substantial amount of redundant capacity, and a single breakdown does not cause the failure of a complete system. Many sectors of industry, such as electricity supply, have adopted maintenance planning based on replacement and overhaul at fixed time intervals. Such scheduling is usually planned on a basis of a limited amount of condition monitoring, and the monitoring is typically not done on a continuous basis. Such a maintenance policy makes heavy demands on scarce, skilled manpower. Also, it is estimated that only 10% of the components replaced during fixed interval maintenance outages actually need to be replaced at that time. The obvious implication is that 90% of what is replaced need not be.

In the longer term, condition monitoring also allows the operator to build up a data base which can be used for trend analysis, so that further improvements can be made in the scheduling of maintenance. The

measurements can be made using the diagnostic technologies of vibration analysis (VA), acoustic emission (AE) analysis, infrared thermography (IRT) analysis or lubrication management analysis, etc.

1.2 Monitoring techniques

Techniques collectively referred to as CM have a common objective of indicating the early signs of deterioration or malfunction, and wear trending in plant and machinery. There are several different types of measurements commonly used to determine condition, and the technical requirements for each of these are very different. The background of the designers for these diverse condition monitoring tools is also very different, each with a specialisation in one particular area. The result of this is that although various technologies can be used as part of condition monitoring systems, they have been seen historically as competing with one another. What is actually required was an integrated approach, where more techniques rather than one are used together. It is recognised that there is a need to integrate the different technologies together, so that they complement each other, rather than competing.

The major measurement technologies that are used within condition monitoring are briefly described here. The most established technology is vibration analysis, and it is the most tangible. Almost all machines vibrate, and the link between these vibrations and the machine condition is both easily measured, and the results easily interpreted. Transducers can be easily attached on a temporary basis to a machine, most often with a strong magnet or quick fit connector, so that collection of the data is quick and efficient. A major benefit of vibration, however, is that different mechanical processes within the machine all produce energy at different frequencies. If these different frequencies are separated from one another through spectrum analysis, then a whole new level of detail may be seen, with more advanced warning of the development of faults, as well as diagnostic capabilities.

The second most common technique is the testing of lubricant samples. This can have major benefits as it can detect the root cause of a problem, rather than the onset of a problem itself. For example, if the presence of particles such as sand and grit often in the form of very small dust particles are detected in a lubricant, then they can be removed even before they cause any wear or damage to seals, through their abrasive effect. Viscosity checks, moisture content as well as detection of contaminants

all fall into this category of test. The technique can also look for the effects of wear through detection of particles such as ferrous material which are carried away from a wear site with the lubricant. Again the examination of this material often allows diagnosis to be performed, but this technique relies on samples being taken away from a machine to a laboratory for a full analysis to be performed.

Electrical departments were the first to benefit from the use of thermal imaging cameras to obtain temperature distribution maps across electrical panels, looking for hot spots from loose connections. The technique is now being used more widely to look at pipe work, vessels, as well as bearings and couplings. The cameras are getting smaller, lighter and the pictures of better quality all the time, and the interpretation of the data requires little training, relative to other techniques.

Various techniques, using simpler approaches to vibration analysis, are used to detect friction and the presence of bursts of energy resulting from defects in rolling element bearings, where a rolling element may be impacting a defect in a race creating shocks and spikes of energy. Whilst having merits of their own, the use of these techniques often uses similar transducers, mounted in the same locations as vibration analysis. The

combination of vibration with these techniques can therefore create economies in the time and manpower needed to collect the data.

In summary, various condition monitoring techniques have been developed over many decades in different industries. There are no golden rules as to which techniques are best and the user must tailor them to suit the particular process he needs to monitor. The importance of condition monitoring is becoming more and more apparent from a number of emerging conferences dedicated to this subject. In the Condition Monitoring 2003 Conference held at Oxford 2003 [1], it was shown that the thermography and AE are growing, vibration, wear debris and oil analysis are still commonly used methods. Ordinary acoustics is also there as well as the low-cost steady-state monitors for pressure, level, leakage and flow. Perhaps the greatest advances are where sensors could be fitted to the existing operational equipment and simply tested non-intrusively, or on-line, with the minimum of modifications. These include sensing position, noise, ultrasonics (for pressure), speed (for motor condition), etc.

1.3 Condition monitoring of railway equipment

1.3.1 Overview of maintenance in railways

Maintenance techniques have evolved as new generations of equipment were introduced. Up to the 1950's a 'fix it when it breaks' philosophy was mostly implemented due to lack of engineering knowledge in certain areas. Regular inspection regime or time-based maintenance was later introduced. In the time-based scheme, effective maintenance management requires a trade-off between the frequencies, and hence the cost, of maintenance and the resulting benefit mainly of improved reliability [7].

Most safety-critical equipment is subject to hard time maintenance, via either an overhaul programme or finite life replacement. In such cases, it is imperative to analyse defect histories and adjust their maintenance programmes accordingly. The time-based approach, however, can not eliminate unexpected failures due to different stresses being experienced by individual pieces of equipment. Also increased reliability can only be brought through increased maintenance frequency/cost.

Modern maintenance strategy integrates knowledge-based fault diagnosis with fault prediction facilities to form a powerful maintenance management tool [8]. This forms part of a management information system, which helps to focus maintenance activity on the required standards of service.

1.3.2 Motivation for condition-based maintenance

The motivation for modern condition monitoring stems from the requirements for improved reliability and the fact that maintenance operations are a large cost centre and drain on profits. Therefore, there has been a gradual shift from time-based to condition-based maintenance. The increased performance and cost-effectiveness are found in many industries by switching from routine (time-based) to condition-based maintenance. It is required to take best advantage of readily available and cheap electronics and computer technology to reduce possibilities of failure, limit its effects and predict developing faults and degradation in advance. The extra cost of sensory equipment and condition monitoring system may be weighed against passenger disbenefits associated with delays.

1.3.3 Condition monitoring of railway equipment

A number of techniques have been developed for condition monitoring of railway equipment. Such a condition monitoring scheme forms an integral part of condition-based maintenance management system [9]-[12]. The electro-pneumatically operated point machines and train stops are among the case studies. Then other case studies of railway power system equipment include DC circuit breakers and substations and so on.

Some rapid transit railway systems, such as London Underground Ltd., have tended to fit trip cock arms underneath trains, which allow the emergency brake system to be applied externally by a tripping mechanism called a train-stop. Its operation is electro-pneumatic. A PC-based data acquisition system was developed to activate the electro-magnetic valve and register data from sensors. Three AE sensors were attached to the train-stop. A standard potentiometer type rotary sensor was fitted to the shaft on the axis of rotation. The results showed a high degree of repeatability of both AE signatures and angular displacement profiles for the same equipment. Performance monitoring of train-stop operation led to a deeper understanding and accurate characterisation of the normal operation.

The type MM74 high-speed DC circuit breaker (800v, 4kA rating) is electrically operated with a mechanical latch-in feature. The most common failures experienced by circuit breakers are those due to mechanical wear, high temperature variations, dust and damp conditions. Circuit breaker failures are mostly connected to mechanical problems of some kind as a result of overheating. Whenever an increase in temperature occurs without any limit, it leads to failure. The heating of the contacts will destroy the equipment by moving the relative position of parts and by changing the contact shape. The fault situation develops further until the system suddenly disintegrates and complete failure occurs. Signatures of the following parameters are considered to be monitored during normal/healthy and induced fault conditions.

- Electrical characteristics: involves the measurement of variations of voltage and current from each operation;
- Timing trends for consecutive operations;
- Acoustic emission during closing and tripping of the DC circuit breaker;
- Angular displacement profile during closing and tripping.

It should be noted that in applications of single shot machines like power circuit breakers, it is advisable to carry out condition monitoring in time domain rather than in frequency domain. Because all information is used

in time domain, the phase information is discarded in condition monitoring based on frequency domain analysis.

1.4 Fault finding of hydraulic systems

1.4.1 Applications of hydraulic systems

Hydraulic systems are being used in all types of machinery and can be seen in all industries. Putting and squeezing on rolling mills in steel works; roof supports in coal mines; the muscle power of excavating; brakes in motor cars; pushing plastic into shape in moulding machines; controlling cutting rates and feeds in machine tools; winches and steering gear in ships; nozzle control in missiles; remote handling in atom plant. These are but a few hydraulics applications. Following is the introduction to two examples in railway [12].

Electro-hydraulic point machines are very popular on the British Railway network. They are also called clamp locks, although this is a description of their locking mechanism which can also be adapted to operate with electric or electro-pneumatic point machines. The reason for the name is that when the clamp lock device was first introduced, it was associated

with a hydraulic drive mechanism. Clamp locks operate by clamping the switch rail to the stock rail on the closed side ensuring that if the track spreads, as it usually does in new installations, the switch rails remain locked in place. The machines are powered by a power pack which consists of a reservoir, a motor-driven pump and normal and reverse control valves.

The diesel locomotive is by far the most common source of power on railways all over the world. The modern diesel locomotive is a self contained version of the electric locomotive. It differs from electric locomotive principally in that it carries its own generating station around with it, instead of being connected to a remote generating station through overhead wires or a third rail. The generating station consists of a large diesel engine coupled to an alternator producing the necessary electricity. There are two types of diesel engines, the two-stroke engine and the four-stroke engine. In the UK, both types of diesel engines were used but the four-stroke became the standard. In the US, the General Electric (GE) built locomotives have four-stroke engines whereas General Motors (GM) always used two-stroke engines until the introduction of their SD90MAC 6000 hp four-stroke engine.

1.4.2 Advantages of using hydraulic systems

The advantages of using hydraulic methods are summarised as follows:

- High gain both in force amplifying or power amplifying systems;
- The facility to divide and distribute power in any or all directions and the limiting and balancing of forces;
- Hydraulic systems are mechanically safe, compact and controllable and are well suited to the industrial environment while able to transmit large powers;
- Reciprocating or rotating outputs can be obtained and the use of flexible connections permits compound motions;
- The absence of external moving parts leads to safety and silence and inherent long life.

1.4.3 Fault finding in hydraulic systems

Unless some external mechanical failure has occurred, or unless there is some fairly obvious noise or temperature, the location of faults in hydraulic systems can be difficult. With electrics one can check the volts by using test probes — sometimes current checks are necessary too but these again are comparatively easily done with inductive devices. With hydraulics the system is sealed and silent without visual indication of what is occurring inside. The key to fault finding in hydraulic systems is the location of pressure and flow. Since, generally, a flow may occur even

when the system is not capable of generating pressure, but a pressure can usually only be given when a flow is available, flow sensing is not quite so valuable an aid as pressure sensing. Unfortunately, pressure is not so easy to sense.

It is true that some services can be dealt with on a trial-and error basis—replace this unit and see what happens. But this may be time-consuming and could be very expensive on components and even dangerous, if a hydraulic system were tampered with in this way. Every system will have to be treated individually when fault finding. Assuming that the system has been correctly designed, installed and maintained, and so is capable of functioning properly, fault finding will be concerned mainly with the malfunction of components.

1.5 Proposed research project

As the condition monitoring of railway transportation systems develops, there will be an increasing emphasis on the presentation of information from monitoring devices and aiding decision-making based on that information. Condition monitoring in railway industry as a whole covers a

very wide field. To restrict the field we have confined ourselves to the non-intrusive monitoring of hydraulic systems.

In case of trouble-shooting hydraulic systems, pressure readings are often required to be taken at several temporary locations. Since the hydraulic system is fully sealed, the conventional instruments cannot be practically utilised for this purpose unless they are built-in during the production stage of the system. Instead, the indirect pressure measurement systems can be very helpful for rapid diagnosis of hydraulic systems because of their non-intrusiveness, easy installation, cost effectiveness and time-saving. Taking the acoustic characteristics of oil into consideration, a novel method for the non-intrusive measurement of pressure has been investigated. However, without an appropriate measuring scheme involved, the new method can not be applied to make accurate measurements for condition monitoring.

For a given fluid, its acoustic speed usually can not be measured directly. The common method is to measure the travelling time for a fixed length path in the fluid through which the ultrasound travels, then to calculate the speed. Either the pulse echo method or the pulse transmission method can be used for this purpose. The contact of ultrasonic probe with the fluid

may be required to ensure the high accuracy of the speed measurement, otherwise the error resulting from the thickness influence of the pipe or vessel must be corrected. For hydraulic systems, different pipes often have different thicknesses so that the determination of acoustic speed outside the pipe becomes even more complicated. To get rid of the thickness influence of the pipes, a special measuring scheme has been studied, that is, the combination of pulse echo and transmission.

The experiment has been done at the Research Centre for Instrumentation and Measurement, Shandong University of Science and Technology, China. The calibration process is designed for a measuring range of 0-20 MPa with a hydraulic oil of viscosity grade 10 inside the pipe. At the initial calibration point, two readings from the non-intrusive measurement system and the standard pressure meter respectively are gathered as references for calculating the changes in velocity and pressure correspondingly. Therefore, after the completion of each of the rest calibration points, a pair of calibration data is obtained.

The primary studies have shown that the relationship between the change in pressure and the variation in acoustic velocity is non-linear in nature. Evaluating the performance of such a new system is the objective of the

PhD project carried at Napier University. It includes the direct modelling process dealing with the estimation of the system non-linearity and inverse modelling process for direct digital reading of the measurand. This research work would provide a better understanding of the input-output static characteristic of the novel non-intrusive pressure measurement system and lay a firm foundation for its practical application to quick diagnosis with sufficient accuracy.

To reveal the input-output characteristics of a measuring system or sensor behind the calibration data, several structures of neural networks, such as the multilayer perceptron (MLP), the functional link neural network (FLNN), could be considered depending on certain circumstances. The neurons of one layer in a MLP are connected to all the neurons of the following layer via unidirectional links by connection weights. The determination of the activation functions, the thresholds of neurons and the weights are required in utilising the MLP. The FLNN has only one single neuron with an increased input space provided by the functional expansion of its initial inputs. Both the MLP and the FLNN are chosen for the direct modelling and the inverse modelling of the non-intrusive pressure monitoring system with the help of MATLAB. A comparison needs to be made between these two modelling approaches.

2 Existing Pressure Measuring Methods for Hydraulic Systems

2.1 Introduction

With the development of hydraulic techniques, the hydraulically operated systems are becoming more and more precise and complicated. The economic loss due to the failure is increasing rapidly. Therefore, it is very important to properly diagnose the defective components as soon as possible. When problems arise with the hydraulic systems, trouble shooting often requires pressure readings to be taken at a number of locations [13] [14]. Conventional technology needs the insertion of pressure gauges or transducers into the piping system to make these measurements. This is a costly and time-consuming procedure. It is even worse that at some locations the insertion of a pressure gauge is impossible or impractical. Because of the difficulty of making the necessary pressure measurements, it is hard to carry out the diagnosis rapidly and economically. However, non-conventional methods can offer

the advantage of measuring the pressure outside the oil pipes, that is, the non-intrusive pressure measurement methods.

2.2 Existing pressure measurement techniques for hydraulic systems

2.2.1 The conventional measuring techniques

Although there have existed many conventional methods to measure hydraulic pressure [15] [16], their measuring diagrams can be shown in a common figure (Fig.2.1). The instrument consists of three main components [17], that is, a pressure sensing element, a conversion element and a display element. In Fig.2.1, mechanical elements are generally used to first convert the applied pressure into a displacement and then transduce the displacement into an electrical signal, which is finally indicated either in an analogue form or in a digital form. To meet the need of practical measurement, various kinds of pressure transducers have been developed, such as capacitance-type pressure transducers, inductance-type pressure transducers, piezoelectric pressure transducers, potentiometer-type pressure transducers, strain-gauge types of pressure transducers, semiconductor strain-gauge pressure transducers, vibrating-wire pressure transducers, optical-fibre pressure transducers and so on [18]-[22].

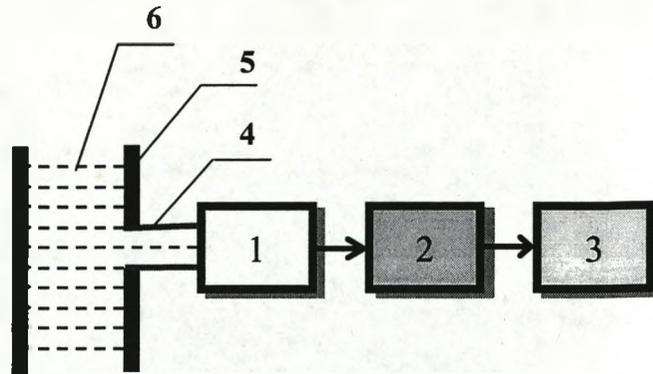


Fig.2.1 Schematic diagram of conventional methods for measurement of fluid pressure

1. Pressure sensing element, 2. Conversion element, 3. Display element,
4. Fitting, 5. Steel tube, 6. Hydraulic media.

These methods have been widely used in many fields, but face difficulties for the diagnosis of hydraulic systems. The reason is that, to make the measurement of fluid pressure, all the pressure-sensing elements must get in touch with the hydraulic media. Installing pressure gauges or transducers may require the manufacture of special fittings, which is costly and time consuming. It is even worse that at some locations the insertion of a pressure gauge is impossible or impractical. Because of the difficulty of making necessary measurement, trouble shooting often deteriorates into a process of replacing components until the problem disappears.

It is clear that substantial benefits would result if there were a method of determining pressure quickly, without the need for installation of a pressure gauge. This leads to the investigation of non-intrusive pressure measurement for effective condition monitoring and rapid fault diagnosis of hydraulic systems.

2.2.2 External pressure measurement techniques

The measuring gauges or devices of hydraulic pressure could be classified into two main categories, intrusive and non-intrusive based on whether the sensing element has to make contact with the measured liquid. It is evident that non-intrusive measuring techniques are not as developed as the intrusive techniques of measurement, although there are urgent needs in many applications. The non-intrusive measurement devices of hydraulic pressure fall chiefly into the following subcategories. In the first subcategory, the change in outer diameter of the pipe in response to the internal pressure is measured by means of micro-deformation as an indication of the pressure. The techniques for the implementation of such micro-deformation measurements can be a LVDT (Linear Variable Differential Transformer) or a capacitive sensor. This existing technique is actually an extension of the traditional measuring instruments from the

pressure sensing element point of view, that is, the expansion of tubing served as the role of converting pressure into a micro-displacement. However, the hydraulic tube is not a good sensing element; this could limit the performance of such an instrument in one way or another.

In the second subcategory, a piezoelectric transmitting transducer mounted on the outside of a fluid-filled pipe is used to excite vibration in the pipe. A receiving transducer, also attached to the exterior surface of the pipe, is used to measure the resonance frequency of the selected vibration mode. The pressure of the fluid is computed by the use of the correlation between the fluid pressure and the resonance frequency.

In our investigation of measuring pressure non-intrusively, the third subcategory, the ultrasonic wave energy enters the interior of the pipe and passes through the contained medium, of which the pressure is being measured. Broadly speaking, effects as a result of the waves passing through the hydraulic medium are dependant on the pressure of the medium and these effects could be measured as an indication of the pressure. In our design, the change in acoustic velocity of the hydraulic medium is used as a measure for the internal pressure. Acoustic velocity in oil increases with pressure and the relative change in acoustic velocity

over change in pressure is approximately 0.3% per MPa. The novel non-intrusive technique for the measurement of hydraulic pressure based on ultrasonics offers the capability of making pressure measurements for trouble-shooting without intruding into the pipeline. This has the advantage of not having an effect on the condition of the sealed hydraulic system and also of assisting rapid trouble-shooting to save time and cost.

Let us take the external pressure measurement system (EPMS) as an example of the existing unconventional measuring methods. EPMS is based on the fact that the expansion of tubing with pressure follows thick-wall pressure vessel theory. For the case of loading by internal pressure only, the approximated expansion equation is as follows:

$$\Delta D = \frac{4a^2b}{E(b^2 - a^2)} \Delta p, \quad (2.1)$$

where

ΔD : Change in diameter,

Δp : Change in pressure,

a : Inner radius,

b : Outer radius,

E : Modulus of elasticity of tube material.

Eq.(2.1) shows that the change in diameter is directly proportional to the change in pressure. The expansion is dependent on the inner and outer radii of the tube, and the modulus of elasticity of the tube material. The expansion is quite small. For a nominal 1-inch steel tube with the following characteristics: $b=0.501$ inches, $a=0.449$ inches, $E=28.5 \times 10^6$ PSI, then the theoretical calculation of $\Delta D/\Delta p$ is 287×10^{-9} inches/PSI.

According to the above theory, Jon K. Chandler and Don P. Foler designed an external pressure measurement system (EPMS) [23]. The tubing expansion is measured using a LVDT (Linear Variable Differential Transformer). EPMS was used in field for aircraft on several occasions. As an example, let us take the case where the nose gear of an aircraft was raising too slowly. Two readings by EPMS showed that the ground-cart hydraulic supply was not putting out the proper 3000 PSI supply pressure. After the ground-cart pressure was adjusted to the correct value, the nose gear retracted properly. Thus external pressure readings can be taken using EPMS, resulting in avoiding unnecessary component removal and time delay. The system was designed for use on aircraft hydraulic systems, but has wide-ranging applications.

In fact, the expansion of the tube with pressure can be measured by any approaches appropriate to the measurement of micro-displacement and the above LVDT used in EPMS is just one of them. For example, the capacitive displacement transducer can also be chosen for this purpose.

However, generally speaking, the above external pressure measurement is actually an extension of the traditional measuring instruments from the pressure sensing element point of view, that is, the expansion of tubing served as the role of converting pressure into a displacement. Its inherent disadvantage lies in the fact that the hydraulic pipeline is not a good pressure sensing element.

2.3 Summary

When problems arise with the hydraulic systems, trouble shooting often requires pressure readings to be taken at a number of locations. Conventional technology needs the insertion of pressure gauges or transducers into the piping system to make these measurements. This is a costly and time-consuming procedure, and there are locations where the conventional instruments fail to make the required measurements.

The external measurement of pressure can bring the following benefits: (1) the ability of making pressure measurements for trouble-shooting without intruding into the pipeline; (2) no effect on the sealed hydraulic system; (3) offering rapid trouble-shooting to save time and cost.

In the existing external pressure measurement system (EPMS), the tubing expansion with pressure is directly proportional to the change in pressure. EPMS is actually an extension of the traditional measuring instruments from the pressure sensing element point of view, that is, the expansion of tubing served as the role of converting pressure into a displacement. Its inherent disadvantage lie in the fact that the hydraulic pipeline is not designed for the purpose of being a sensing element.

3 Novel Non-Intrusive Pressure Measuring Method for Monitoring Hydraulic Systems

3.1 Introduction

Various ultrasonic testing techniques have been steadily developed since the beginning of an era using ultrasound for non-destructive testing in 1935. Here we would like to mention some of the great pioneers in this field [24]. The testing defect of metal objects with ultrasonic waves was first proposed by Sokolov in 1929. Sokolov published his results on the actual testing of objects with through-transmission method in 1935. Then an instrument developed on the basis of the Sokolov's experimental system was found for practical use after World War II. At almost the same time Firestone in America and Spoule in Britain first introduced a pulse-echo flaw-detector [25] [26]. Based on Erwin's patent [27], the first resonance instrument was built in 1947. Together with the efforts by other researchers the ultrasonic testing began to gain popularity. Industry started

to demand more and more testing, which became the driving force for the development of ultrasonic instruments. It has been shown that ultrasonic waves can be used for the measurement of following quantities: position, distance, thickness, level, flow, temperature, density, composition and so on [28]-[32].

The present trend in standard ultrasonic testing is to refine existing methods and to further develop the various automatic testing techniques. For unusual testing problems, where conventional methods fail, special techniques are developed to meet the market needs.

Now let us consider the problem of our interest. For a fully sealed hydraulic system, the action of the internal pressure can produce several phenomena. One of these is obviously the expansion of the tubing. There are also other promising effects that may be utilised to determine the pressure in non-intrusive ways with the aid of advanced sensing techniques, signal processing techniques and computer techniques. Considering the acoustic characteristics of the oil and the features of ultrasonic waves, the new idea of non-intrusive pressure measurement can be formed. The pressure readings can be taken by the determination of the ultrasonic velocity travelling through the hydraulic oil.

For a given fluid, its acoustic speed usually can not be measured directly. The common method is to measure the travelling time for a fixed length path in the fluid through which the ultrasound travels, then to calculate the speed [33] [34]. Either the pulse echo method or the pulse transmission method can be used for this purpose [35] [36]. The contact of ultrasonic probe with the fluid may be required to ensure the high accuracy of the speed measurement, otherwise the error resulting from the thickness influence of the pipe or vessel must be corrected. For hydraulic systems, different pipes often have different thicknesses so that the determination of acoustic speed outside the pipe becomes even more complicated. To get rid of the thickness influence of the pipes, a special measuring scheme is needed, that is, the combination of pulse echo and transmission, which will be described below.

3.2 Measurement principle of the proposed approach

The proposed measurement strategy is basically a combination of acoustic effect of the fully sealed oil under pressure and the penetrating capability of the ultrasonic waves. With the aid of precision measurement technique and signal processing, the pressure readings can be taken non-intrusively

by the determination of the acoustic velocity travelling through the oil. The schematic diagram of measuring the acoustic velocity in oil is shown in Fig.3.1. Two ultrasonic probes working at 2.5 MHz are mounted outside the oil pipe in an arrangement used in the pulse transmission mode. However, the working mode of the two probes here is different from the conventional transmission mode in that probe 1 serves first as a transmitter, then as a receiver. The purpose of such a design is to eliminate the influence of pipe wall thicknesses and effectively remove the unexpected time delays.

Suppose that the inner radius of the oil pipe is a , the outer radius is b . The following equation can be obtained from Fig.3.1.

$$t_1 = 2t + 2t_s + \tau, \quad (3.1)$$

$$t_2 = t + 2t_s + \tau, \quad (3.2)$$

where

t_1 : time taken from emission to reception of echo received by probe 1.

t_2 : time taken from emission to reception of penetration wave received by probe 2.

t : time taken travelling through a length of $2a$ in oil.

t_s : time taken travelling through the wall thickness of the pipe ($b-a$).

τ : overall acoustic and electric delays, including the unexpected delays of probes, coupling material, connection cables, receiving circuitry and so on.

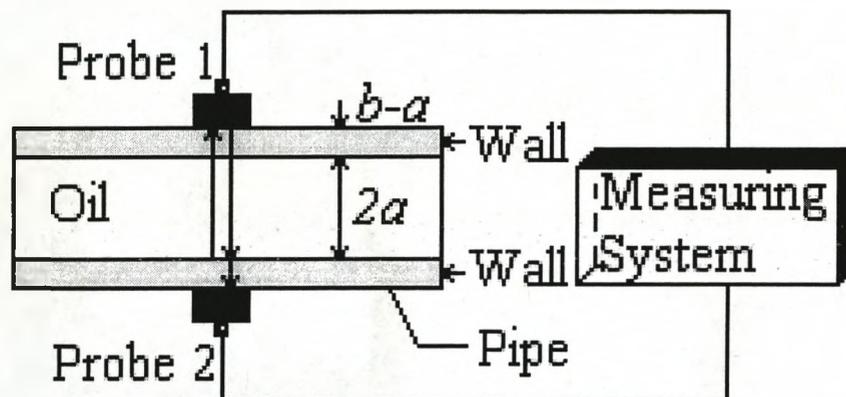


Fig.3.1 Schematic diagram of the non-intrusive measuring method

a =radius of the inner wall, b = radius of the outer wall.

From Eq.(3.1) and Eq.(3.2), we obtain:

$$t = t_1 - t_2. \quad (3.3)$$

The ultrasonic wave speed in oil c can be expressed as:

$$c = 2a / (t_1 - t_2). \quad (3.4)$$

Therefore, c can be accurately calculated after measuring t_1 and t_2 . This measuring scheme has two advantages. One is that the thickness influence of the pipe is eliminated; the other is that the overall time delay can be effectively removed.

3.3 Design of the monitoring system

The block diagram of the computer-controlled system designed for this research is shown in Fig.3.2. The single-chip microcomputer unit is composed of an 80C196, an EPROM, an E²PROM, a SRAM, a keyboard and a display module. This unit has five functions: pre-setting parameters, generating main control pulse, detecting signal, filtering data, calculating pressure and displaying pressure. The transmitting unit consists of an amplifier to amplify the main control pulse, a circuit for the generation of narrow pulse and a driving circuit for the ultrasonic probe. The receiving unit is made up of an attenuation circuit, a frequency selective

amplification circuit, a noise elimination circuit, a video amplification circuit and a time expansion circuit.

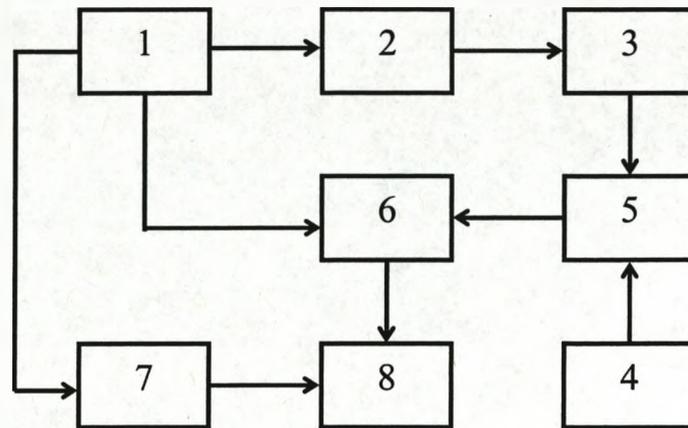


Fig.3.2 Block diagram of the monitoring system

1. Single-chip microcomputer unit, 2. Transmitting unit, 3. Both transmitting and receiving probe, 4. Receiving probe, 5. Switch, 6. Receiving unit, 7. Scanning unit, 8. Oscilloscope tube unit.

The whole system is controlled by a 16-bit single chip microcomputer. Besides having the function of digital display for the measured parameter directly, the system also provides a real-time analogue display function for the measuring waveforms concerned, which has been proved to be very useful for the effective adjustment of the system. After t_1 and t_2 are measured separately, the ultrasonic wave speed in the oil can be calculated by Eq. (3.4).

The software of the system consists of a main program and a lot of subroutines. The flow chart of the main program is shown in Fig.3.3.

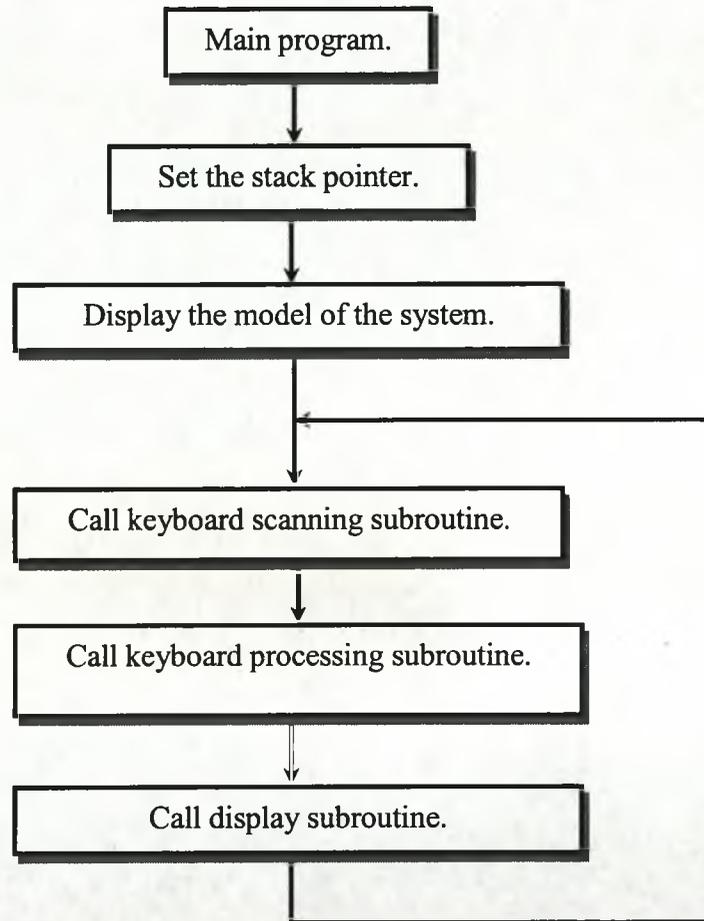


Fig.3.3 Flow chart of the main program

3.4 Summary

This research project is focused on the investigation of determining the hydraulic pressure using the ultrasonic techniques without the need for the insertion of pressure gauges or sensors into the oil pipes. The pressure

readings can be taken by the determination of the ultrasonic velocity travelling through the oil. The measuring scheme for the velocity which is a combination of the pulse echo and the pulse transmission methods could eliminate the thickness influence of the pipe on the measurement of velocity and effectively remove the unexpected acoustic and electric time delay.

4 FLNN-Based Modelling of Non-Intrusive Monitoring System

4.1 Statement of problem

In chapter 3, a novel method for the non-intrusive measurement of pressure has been proposed using ultrasonic techniques. For the given hydraulic oil, the change in pressure is not only a function of the variation in acoustic velocity, but also influenced by the temperature, as shown in Fig.4.1. Evaluating the performance of such a new non-intrusive monitoring system is one of our main objectives. It includes the direct modelling process dealing with the estimation of the system non-linearity and inverse modelling process for direct digital reading of the measurand. The proposed research work based on artificial neural network (ANN) method would provide a better understanding of the input-output static characteristics of the novel non-intrusive pressure measurement system and lay a firm foundation for its practical application to quick diagnosis.

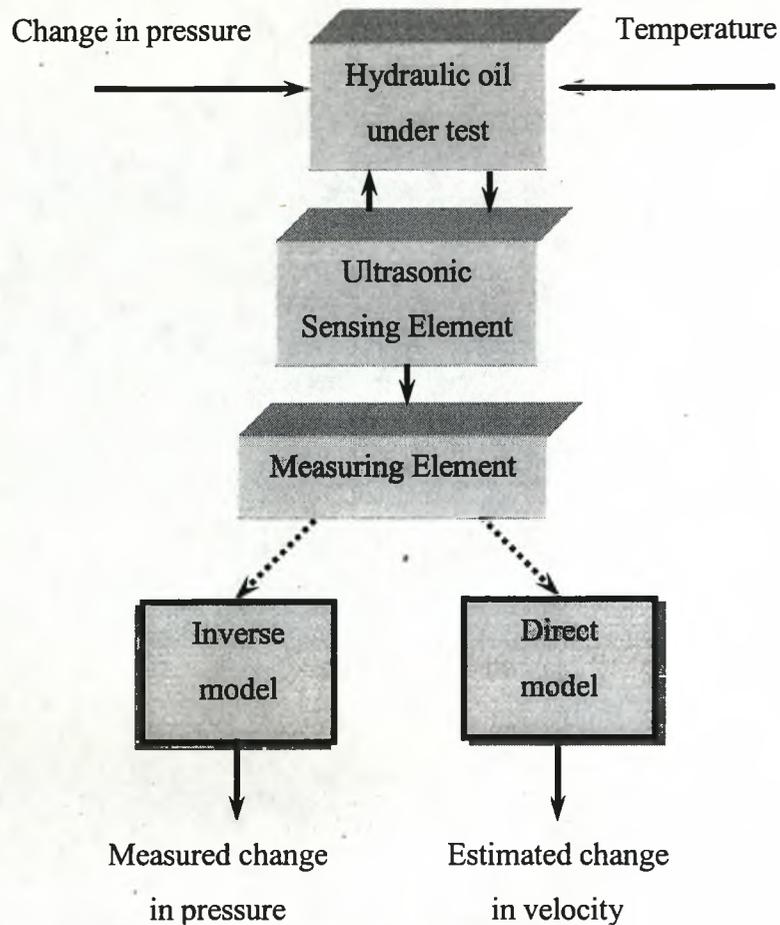


Fig.4.1 Pressure monitoring system under influence of temperature

4.2 Applications of ANN in measurement and instrumentation

To begin with, we briefly present the development of ANN and the review of their applications in measurement [59]. Artificial Neural Networks

(ANNs) are input information processing structures which imitate human brain behaviour. Formal realisation that the brain in some way performs information processing tasks was first spelt out by McCulloch and Pitts in 1943. This pioneering work is said to be the start of the modern era of neural networks. They represented the activity of individual neurons using simple threshold logic elements, and showed how networks made out of many of these units interconnected could perform arbitrary logical operations.

Some 15 years after the publication of McCulloch and Pitt's classic paper, a new approach to the pattern-recognition problem was introduced by Rosenblatt (1958) in his work on the perceptron. There followed a period of research into the behaviour of randomly connected systems and the network started to acquire a practical role. At the same time, as understanding about mechanisms of learning in biological systems grew, the study of neuro-dynamics intensified. The learning capabilities of one scheme in particular, Rosenblatt's perceptron, were reported: Minsky and Papert showed that there were classes of simple tasks which the perceptron could not perform.

The development of new learning algorithms continued, resulting in training algorithms capable of overcoming the limitations of the early perceptron. Hopfield network, a historical algorithm, is typical of recursive algorithms, in which all of the nodes are connected to one another. It has been applied successfully to a number of optimisation problems.

The multi-layered perceptron is arguably the most popular neural network architecture, and certainly the trigger of the widespread explosion of activity in this area. It is also known by the name of the algorithm used to train it, back error propagation. The training method can trap the network configuration in local minima of the error function, which halts the training process. However, this turns out not to be as much of a problem in practice as it might appear. The most important aspect of the application of these networks is to ensure that once trained, the classification ability extends, or generalises, to fresh data. Unlike the previous examples, the connectivities in Kohonen network are neither set by a predetermined set of states to recall, nor by attempting to produce correct classifications externally supplied by a tutor. The Kohonen network simply inspects the data for regularities, and organises itself in such a way to form an ordered description of the data. One of a few neural

network algorithms which are especially suited to modelling time-dependent processes is the temporal differences.

ANN is an expanding subject area in terms of financial investment, research and development [37]-[43]. The main selling points of this technology are its ability to adapt to the characteristics of example data, and classify them correctly among noise, distortions and non-linearity; the provision of simple tools for automatic feature selection; ease of integration with existing domain knowledge; and flexibility of use, with the ability to handle data from multiple sources and decision systems.

The benefits of the ANN are advantageously utilised in many different applications in the field of measurement and instrumentation. These are concerned with the calibration of instrument and sensor, the development of new measurement methodologies, the identification of complex systems, the processing of signals and the recognition of patterns. The effectiveness of ANN in this field may be attributed to the fact that they are a powerful tool for non-linear approximations and may be used when all other methods have failed [44]-[50].

ANN has been used for monitoring of transformer heating; on-line phasor detection and identification of the power system; historical data analysis for short time forecasting of electrical load; instrument fault detection; real-time monitoring of distribution systems. They have also been utilised for ADC (analogue to digital converter) dynamic error compensation. In this case, the generalisation capability of the neural networks permits reduction of the huge number of experiments which would otherwise be required. The capacity of ANN to model complex non-linear systems has been utilised in the modelling of measurement devices. In particular, two categories of components have been considered: analogue measurement devices, such as sensors, transducers, filters and amplifiers; and the mixed analogue/digital measurement devices, such as ADC and DAC (digital to analogue converter). By using a suitable ANN structure, and a proper organisation of the training set, it is possible, after a proper set-up phase, to obtain a neural model furnishing an output which corresponds to that which can be obtained from the real device to be modelled. The model's accuracy depends on the choice of both the ANN structure and the training set. In this way, the difficulties arising from the traditional method for modelling devices characterised by transfer functions, which can be complex and/or hardly non-linear or piece-wise linear, are overcome. ANN has recently gained more attention with regard to

instrument and sensor calibration. By incorporating the ANN in the calibration process, accuracy has been increased and interference from other variations has easily been compensated. ANN has been used for the improvement of both ultrasonic distance sensors and optical fibre sensors for displacement and surface reflectivity measurements; the sensor compensation of full bridge strain gauges; the calibration of sensor arrays, robot arms, artificial noses and industrial microwave six-port instruments [51]-[63].

The results achieved in theories and applications to date indicate that ANN will remain firmly rooted as useful new signal processing tools.

4.3 Proposed approach for modelling of non-intrusive monitoring system

4.3.1 Theoretical background

A major task for a neural network is to learn a model of the environment [59]. Our interest is focused on an important class of neural networks that perform modelling through a process of learning and generalisation.

A neural network has the ability to learn from its environment, and to improve its performance through learning [59] [64]. The network learns about its environment through an iterative process of adjusting its weights and thresholds. To be specific about the learning process, consider a pair of node signals x_j and v_k connected by a weight w_{kj} . Let $w_{kj}(n)$ denote the value of the weight w_{kj} at time n . At time n an adjustment $\Delta w_{kj}(n)$ is applied to the weight $w_{kj}(n)$, yielding the updated value $w_{kj}(n+1)$. Then, we may write:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n), \quad (4.1)$$

where $w_{kj}(n)$ and $w_{kj}(n+1)$ may be viewed as the old and new values of the weight w_{kj} , respectively. The adjustment $\Delta w_{kj}(n)$ is computed as a result of stimulation by the environment, and the updated value $w_{kj}(n+1)$ defines the change made in the network as a result of this stimulation. The neural network responds in a new way to the environment with the updated set of parameters $\{w_{kj}(n+1)\}$.

A learning algorithm is the rule for the solution of a learning problem. Basically, learning algorithms differ from each other in the way in which the adjustment Δw_{kj} to the weight w_{kj} is formulated. Another factor to be

considered is the manner in which a neural network relates to its environment, a learning paradigm.

Error-Correction Learning:

Let $d_k(n)$ denote some desired response or target response for neuron k at time n . Let the corresponding value of the actual response of this neuron be denoted by $y_k(n)$. The response $y_k(n)$ is produced by a stimulus (vector) $\mathbf{x}(n)$ applied to the input of the network in which neuron k is embedded. The input vector $\mathbf{x}(n)$ and desired response $d_k(n)$ for neuron k constitute a particular example presented to the network at time n .

Typically, the actual response $y_k(n)$ of neuron k is different from the desired response $d_k(n)$. Hence, an error signal is defined as the difference between the target response $d_k(n)$ and the actual response $y_k(n)$, as shown by

$$e_k(n) = d_k(n) - y_k(n). \quad (4.2)$$

The ultimate purpose of error-correction learning is to minimise a cost function based on the error signal $e_k(n)$, such that the actual response of each output neuron approaches the target response for that neuron in some

statistical sense. A criterion commonly used for the cost function is the mean-square-error criterion, defined as the mean-square value of the sum of squared errors [59]:

$$J = E\left[\frac{1}{2} \sum_k e_k^2(n)\right], \quad (4.3)$$

where E is the statistical expectation operator, and the summation is over all the neurons in the output layer of the network. The factor $\frac{1}{2}$ is used in Eq.(4.3) so as to simplify subsequent derivations resulting from the minimisation of J with respect to free parameters of the network. Minimisation of the cost function J with respect to the network parameters leads to the so-called method of gradient descent [65]. However, the difficulty with this optimisation procedure is that it requires knowledge of the statistical characteristics of the underlying processes. To overcome this practical difficulty, an approximate solution to the optimisation problem is considered. Specifically, the instantaneous value of the sum of squared errors is used as the criterion of interest [59]:

$$E(n) = \frac{1}{2} \sum_k e_k^2(n). \quad (4.4)$$

The network is then optimised by minimising $E(n)$ with respect to the weights of the network. Thus, according to the error-correction learning rule or delta rule, the adjustment $\Delta w_{kj}(n)$ made to the weight w_{kj} at time n is given by [66]:

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n), \quad (4.5)$$

where η is a positive constant that determines the rate of learning.

Error-correction learning relies on the error signal $e_k(n)$ to compute the correction $\Delta w_{kj}(n)$ applied to the weight $w_{kj}(n)$ of neuron k in accordance with Eq.(4.5). The error signal $e_k(n)$ is itself computed from Eq.(4.2). Finally, Eq.(4.1) is used to compute the updated (new) value $w_{kj}(n+1)$ of the weight.

The learning-rate parameter η has a profound impact on the performance of error-correction learning in that it affects not only the rate of convergence, but also the convergence itself. If η is small, the learning process proceeds smoothly, but it may take a long time for the system to converge to a stable solution. If, on the other hand, η is large, the rate of

learning is accelerated, but now there is a danger that the learning process may diverge and the system therefore becomes unstable.

Supervised Learning:

An essential ingredient of supervised or active learning is the availability of an external teacher, as indicated in the arrangement of Fig.4.2 [59]. The teacher may be considered as having the knowledge of the environment that is represented by a set of input-output examples. The environment is, however, unknown to the neural network of interest. Suppose that the teacher and the neural network are both exposed to a training vector drawn from the environment. Then the teacher is able to provide the neural network with a desired or target response for that training vector. The desired response represents the optimum action to be performed by the neural network. The network parameters are adjusted under the combined influence of the training vector and the error signal; the error signal is defined as the difference between the actual response of the network and the desired response. This adjustment is carried out iteratively in a step-by-step fashion with the aim of eventually making the neural network emulate the teacher; the emulation is presumed to be optimum in some statistical sense. In other words, knowledge of the environment available to the teacher is transferred to the neural network

as fully as possible. When this condition is reached, we may then dispense with the teacher and let the neural network deal with the environment completely by itself (i.e., in an unsupervised fashion).

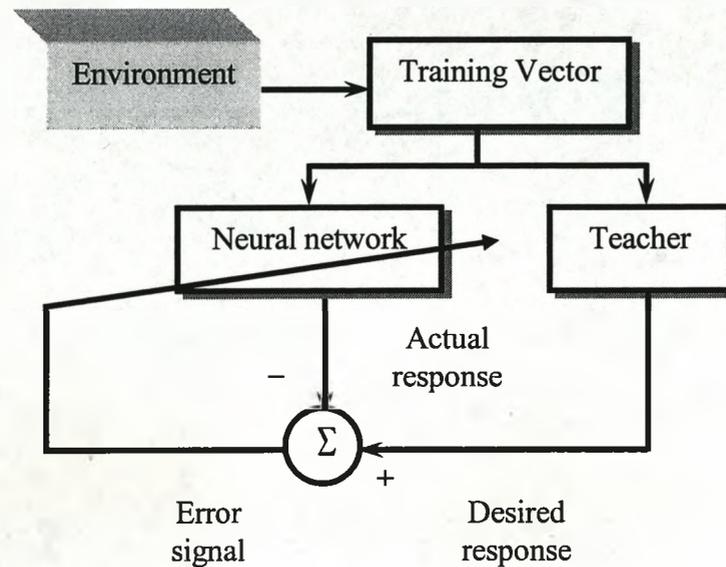


Fig.4.2 Block diagram of supervised learning

The form of supervised learning above is indeed the error-correction learning discussed previously. Given an algorithm designed to minimise the cost function of interest, and given an adequate set of input-output examples and enough time permitted to do the training, a supervised learning system is usually able to perform such tasks as pattern classification and function approximation satisfactorily.

Examples of supervised learning algorithms include the least-mean-square (LMS) algorithm [66] and its generalisation known as the back-propagation (BP) algorithm [67]. The LMS algorithm involves a single neuron, whereas the BP algorithm involves a multilayered interconnection of neurons. The back-propagation algorithm derives its name from the fact that error terms in the algorithm are back-propagated through the network, on a layer-by-layer basis. The BP algorithm includes the LMS algorithm as a special case.

Supervised learning can be performed in an off-line or on-line manner. In the off-line case, a separate computational facility is used to design the supervised learning system. Once the desired performance is accomplished, the design is "frozen," which means that the neural network operates in a static manner. On the other hand, in on-line learning, the learning procedure is implemented solely within the system itself, not requiring a separate computational facility. In other words, learning is accomplished in real time, with the result that the neural network is dynamic. Naturally, the requirement of on-line learning places a more severe requirement on a supervised learning procedure than off-line learning.

In this section, we have discussed important issues relating to the learning process in the context of neural networks. The learning rule, namely, error-correction learning, is indeed basic to the design of supervised neural networks for the problem of interest. In the study of supervised learning, a key provision is a “teacher” capable of applying exact corrections to the network outputs when an error occurs in error-correction learning. Supervised learning has established itself as a powerful paradigm for the design of artificial neural networks.

4.3.2 Proposed approach for neural modelling of monitoring system

The model of supervised learning consists of three interrelated components [68], which are described as follows for multi-input one-output monitoring system:

- An environment, which supplies a vector describing the state of the environment.
- A teacher, which provides a desired response (target output) for every input vector.
- A neural network (Learning machine), which is capable of implementing a set of input-output mapping functions.

We note that a neural network is one form in which empirical knowledge about a physical phenomenon (environment) of interest may be encoded. By “empirical” knowledge we mean a set of measurements that characterises the phenomenon. To be more specific, consider the example of a phenomenon described by a vector \mathbf{x} representing a set of independent variables, and a scalar representing a dependent variable. Suppose also that we have N measurements or observations of \mathbf{x} , denoted by $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$, and a corresponding set of observations of d , denoted by d_1, d_2, \dots, d_N .

Ordinarily, the vector \mathbf{x} and d are related by an unknown function, as follows:

$$d = g(\mathbf{x}), \quad (4.6)$$

where $g(\mathbf{x})$ is some function of the argument vector \mathbf{x} . The purpose of this model is to use \mathbf{x} in order to explain or predict d . Now consider using a neural network to implement this objective. It does so by encoding the empirical knowledge represented by the training data set $\{ \mathbf{x}_i, d_i \mid i = 1, 2, \dots, N \}$ into a set of weights. In this context, \mathbf{x}_i represents the input vector

and d_i represents the corresponding value of the desired response (target output). Let \mathbf{w} denote the weight vector of the neural network. Let the actual response of the network be defined by

$$y = F(\mathbf{x}, \mathbf{w}), \quad (4.7)$$

where y is the actual output produced by the network in response to the input vector \mathbf{x} , and \mathbf{w} is a set of free parameters (weights).

The weight vector \mathbf{w} of the network is adjusted in an iterative fashion in response to the error signal e , defined as the difference between the desired response d and the actual response y of the network; that is,

$$e = d - y. \quad (4.8)$$

Fig.4.3 shows the idea behind the adaptive procedure used to optimise the selection of the weight vector \mathbf{w} . The criterion for optimisation is the minimisation of the mean-square value of the error signal, as shown by

$$J(\mathbf{w}) = \frac{1}{2} E[e^2] = \frac{1}{2} E[(d-y)^2] = \frac{1}{2} E[(d-F(\mathbf{x}, \mathbf{w}))^2]. \quad (4.9)$$

The cost function defined here is based on ensemble averaging. The network is optimised by minimising the cost function $J(\mathbf{w})$ with respect to \mathbf{w} .

In a neural network that learns under the supervision of a teacher, information contained in the training set:

$$\{ (\mathbf{x}_1, d_1), (\mathbf{x}_2, d_2), \dots, (\mathbf{x}_N, d_N) \}, \quad (4.10)$$

is transferred to a set of weights represented by the weight vector \mathbf{w} , as indicated by:

$$\{ (\mathbf{x}_1, d_1), (\mathbf{x}_2, d_2), \dots, (\mathbf{x}_N, d_N) \} \rightarrow \mathbf{w}. \quad (4.11)$$

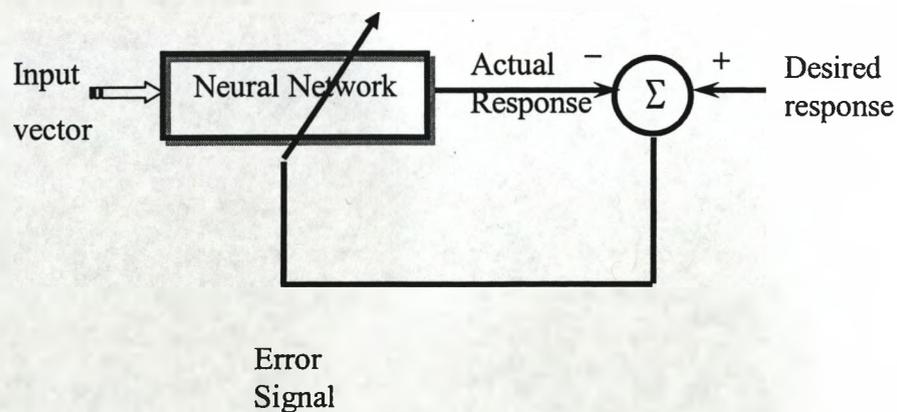


Fig. 4.3 Neural network model

4.3.3 Selection of ANNs

Eq.(4.6) is the learning task which a neural network is required to perform, where the vector \mathbf{x} is the input and the scalar d is the output. The function $g(\bullet)$ is assumed to be unknown. The requirement is to design a neural network that approximates the function $g(\bullet)$, given a set of examples denoted by the input-output pairs (\mathbf{x}_1, d_1) , (\mathbf{x}_2, d_2) , ..., (\mathbf{x}_N, d_N) . The approximation problem described here is a perfect candidate for supervised learning with \mathbf{x}_i serving as the input vector and d_i serving as the role of desired response, where $i = 1, 2, \dots, N$.

The supervised learning involves the modification of the weights of a neural network by applying a set of training samples or task examples. Each example consists of an input-output pair: an input signal and the corresponding desired response for the neural network. Observations (measurements) of the environment, obtained by means of sensors designed to probe the environment, provide the examples used to train the neural network. Typically, the set of examples used to train the network are representative of a real-world situation.

Given such a set of examples, the first step for the design of a neural network is to select an appropriate architecture for the neural network. To reveal the input-output characteristics of a measuring system or sensor behind the calibration data, several structures of neural networks, such as the multilayer perceptron (MLP), the functional link neural network (FLNN), could be considered. The neurons of one layer in a MLP are connected to all the neurons of the following layer via unidirectional links by connection weights. The determination of the activation functions, the thresholds of neurons and the weights are required in utilising the MLP. The FLNN has only one single neuron with an increased input space provided by the functional expansion of its initial inputs [69]-[71].

To implement the proposed ANN-based modelling of non-intrusive pressure measurement, two ANN-based methods are to be investigated. In the rest of this chapter, the FLNN-based modelling process is presented [89]. In chapters 5 and 6, MLP-based modelling process is described.

4.4 Principle of system modelling based on FLNN

4.4.1 General structure of FLNN

The general structure of the FLNN is shown in Fig.4.4. The FLNN is basically a flat single layer network, in which the need of hidden layers has been removed by incorporating functional expansion of the input pattern. The functional expansion effectively increases the dimensionality of the input vector, and hence the hyperplanes generated by the FLNN provide greater discrimination capability in the input pattern space.

The functional-link neural network (FLNN) offers an NN architecture which is quick and easy to train. The FLNN uses extra functional dimensions adjoined to the input space. The training method is greatly simplified and is much faster than conventional multilayer NNs [72].

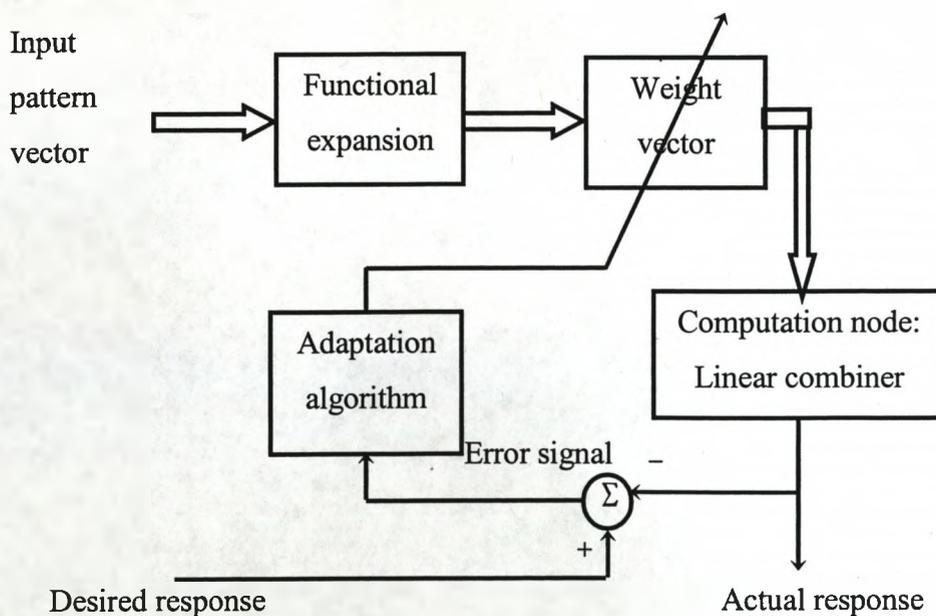


Fig. 4.4 The general structure of the FLNN

4.4.2 Mathematical algorithm for FLNN

Let there be K input patterns to train FLNN, each with n elements. For pattern k , its components are $x_i^{(k)}$, the output of FLNN is $\hat{y}_i^{(k)}$, $i = 1, 2, \dots, n$, $k = 1, 2, \dots, K$. The dimension of each input pattern is increased to N ($N \geq n$) after the action of functional expansion. If the k th input pattern vector is represented by $X^k = [x_1^{(k)}, \dots, x_i^{(k)}, \dots, x_n^{(k)}]$, the expanded vector is $F^k = [f_1(X^{(k)}), \dots, f_j(X^{(k)}), \dots, f_N(X^{(k)})]$, where f_j represents a set of basic functions, $j = 1, 2, \dots, N$. Let the weight vector of the FLNN be $W = [w_1, \dots, w_j, \dots, w_N]$, then we have:

$$\hat{y}^{(k)} = F^{(k)}W^T \quad (4.12)$$

For all K input patterns, Eq.(4.12) can be expressed as:

$$FW^T = \hat{Y}, \quad (4.13)$$

where F is a $K \times N$ matrix given by $F = [F^{(1)}, \dots, F^{(i)}, \dots, F^{(K)}]^T$, and

$$\hat{Y} = [\hat{y}^{(1)}, \dots, \hat{y}^{(i)}, \dots, \hat{y}^{(K)}]^T, i=1, 2, \dots, K.$$

If $K = N$ and the determinant of F is not zero, that is, $\text{Det } F \neq 0$, then:

$$W^T = F^{-1} \hat{Y}. \quad (4.14)$$

If $K < N$, then we can partition F to obtain a matrix F_F of dimension $K \times K$.

Set $W_m = 0$ for $m > K$, then W is modified to W_F , and if $\text{Det } F_F \neq 0$, then:

$$W_F^T = F_F^{-1} \hat{Y}. \quad (4.15)$$

If $K > N$, using conventional pseudo-inversion approach, we obtain:

$$W^T = (F^T F)^{-1} F^T \hat{Y}. \quad (4.16)$$

The above analysis indicates that the functional expansion approach always yields a flat solution [73]. The solution for W in FLNN is achieved iteratively using supervised learning algorithm [74].

4.4.3 Principle of system modelling based on FLNN

For the non-intrusive measuring system of hydraulic pressure, the input signal is the change in applied pressure, and the change of acoustic velocity in oil is the output signal. The direct modelling is concerned with building up the input-output static characteristics of the system, which is generally expressed as:

$$\Delta c = f(\Delta p), \quad (4.17)$$

where Δp stands for the change in measurand, Δc is the change in velocity.

Eq.(4.17) could be given by three terms of the power series as:

$$\Delta \hat{c} = \sum_{j=1}^3 K_j (\Delta p)^j, \quad (4.18)$$

where $\Delta \hat{c}$ represents the estimate of Δc , K_j ($j=1, 2, 3$) are the coefficients under determination.

The inverse modelling of the system is related to the establishment of the measurand reconstruction model for direct digital reading of the applied pressure. Similarly, this inverse model can be represented as

$$\Delta p = f(\Delta c), \quad (4.19)$$

and could be written as the following polynomial:

$$\Delta \hat{p} = \sum_{j=1}^3 L_j (\Delta c)^j, \quad (4.20)$$

where $\Delta \hat{p}$ is an estimate of Δp , and $L_j (j= 1, 2, 3)$ are coefficients to be determined.

To complete the system modelling, a functional link neural network is utilised. Suppose that the initial input is x_i , the expanded inputs x_i^2, x_i^3 , corresponding to an output pattern $y_i i=1, 2, \dots, N$. Under the condition of the FLNN not containing any nonlinearity, the learning algorithm is expressed as:

$$\hat{y}_i(k) = \sum_{j=1}^3 x_i^j \cdot w_j(k), \quad (4.21)$$

$$e_i(k) = y_i(k) - \hat{y}_i(k), \quad (4.22)$$

$$w_j(k+1) = w_j(k) + \eta \cdot e_i(k) \cdot x_i^j, \quad (4.23)$$

where $y(k)$, $\hat{y}(k)$, $e(k)$, $w_j(k)$ stand for the desired output, estimated output, error, and the j^{th} ($j = 1, 2, 3$) connection weight of FLNN at the k^{th} time step, respectively. η represents the learning constant, which governs the stability and the rate of convergence.

The idea of using the above FLNN for modelling the non-intrusive pressure measuring system is dependent on the estimation of the unknown coefficients K_j and L_j ($j = 1, 2, 3$). If the initial input of the FLNN is Δp , its output is Δc , and vice versa. The learning process is carried out based on the calibration data.

4.5 Direct modelling of monitoring system with FLNN and results

The schematic diagram of the direct modelling is shown in Fig.4.5. In equations (4.21) - (4.23), y , x are replaced with Δc , Δp , respectively. The initial values of weights W_j ($j = 1, 2, 3$) are selected. Each of the selected pairs of Δp , Δc is applied sequentially to the FLNN, and each time the weights are updated by the neural algorithm. Applying all pairs of data and adjusting the weights constitute a learning iteration. The learning process continues for as many iterations as needed for a minimum of the average mean square error (MSE). Then the weights W_j ($j=1, 2, 3$) represent the coefficients K_j ($j=1, 2, 3$).

The relative error (%FS) of estimated change in velocity against the expected variation of velocity is defined as:

$$\text{Relative error (\%FS)} = \frac{\Delta c - \hat{\Delta c}}{\Delta c_{\text{FS}}} \times 100\%, \quad (4.24)$$

where Δc_{FS} represents the full scale (FS) of the change in velocity for the measurement system.

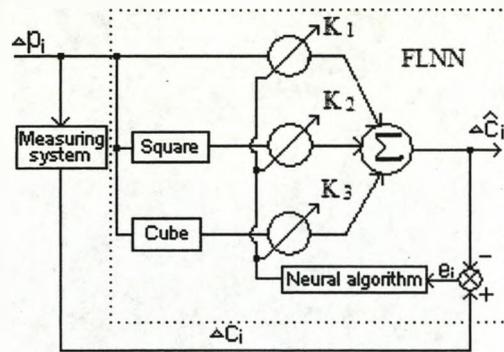


Fig.4.5 Schematic diagram of the direct modelling.

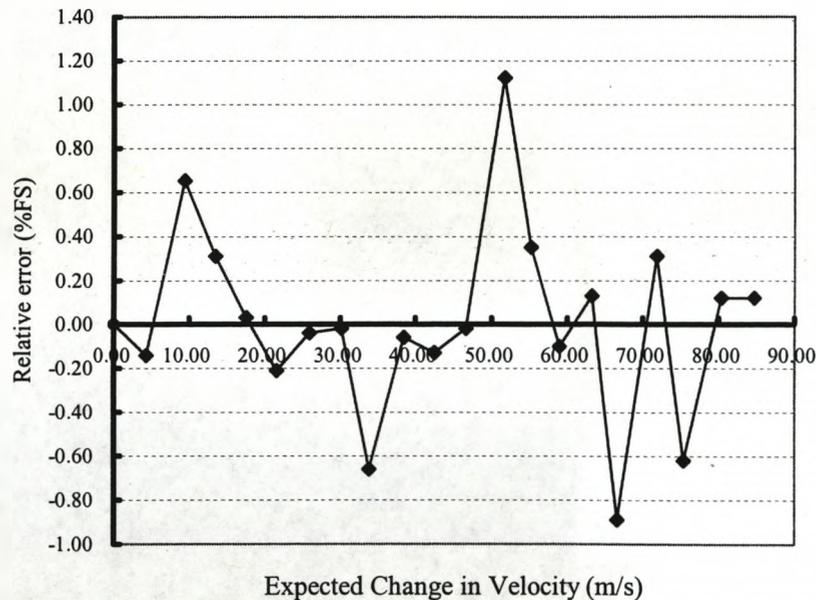


Fig.4.6 Relative error (%FS) of estimated change in velocity against the expected variation of velocity.

Fig.4.6 shows the curve of relative error (%FS) against the expected variation of velocity. These results have demonstrated that the estimated

outputs of the measuring system closely agree with the expected responses. The maximum relative error (%FS) of estimated change in acoustic velocity against the expected variation of velocity is 1.12 %FS. When the term of maximum is related to the error in the thesis, it always refers to the maximum of the absolute values of the errors involved. The effectiveness of the proposed direct modelling has been proven with a low cost of computational complexity.

4.6 Inverse modelling of monitoring system with FLNN and results

The schematic diagram of the inverse modelling is shown in Fig.4.7. In Equations (4.21) - (4.23), y , x are replaced with Δp , Δc respectively. Initially the weights of the FLNN are set at some random values. The input Δp_i is applied simultaneously to the FLNN and the ultrasonic sensing element. The sensing element produces an output Δc_i . Inside the FLNN, $\{ \Delta c_i, \Delta c_i^2, \Delta c_i^3 \}$ constitutes the expanded input set, and it produces an output $\hat{\Delta p}_i$. The error e_i to be used in updating the weights is obtained from the comparison of Δp_i with $\hat{\Delta p}_i$. Each pair of the Δc_i and Δp_i is applied sequentially to the FLNN, and each time the connection weights are updated by the neural algorithm. Application of all the training data

and adjusting the weights constitute one learning iteration. The learning process continues for 215 iterations for the average mean square error (MSE) to attain a predefined minimum value. At this stage, the training is completed; the weights reach their steady-state values and represent the estimated coefficients L_j ($j=1, 2, 3$). The validation study is carried out using the validation data set to get the response of the established model. The relative error (%FS) of estimated change in pressure against the expected variation of pressure is defined as:

$$\text{Relative error (\%FS)} = \frac{\Delta p - \hat{\Delta p}}{\Delta p_{\text{FS}}} \times 100\%, \quad (4.25)$$

Where Δp_{FS} represents the full scale of the measurable change in pressure for the measurement system.

The relative error (%FS) of estimated change in pressure against the expected variation in pressure is shown in Fig.4.8 for all the data set. It shows that the maximum relative error (%FS) is 0.98 %FS. Therefore, the direct digital reading of the measured pressure can be accurately given using the proposed inverse model, which is easily implemented in a microcontroller.

After the modelling process for estimation of pressure has been finished with a desktop or laptop PC (personal computer), the established model can be used in an embedded instrument system for non-intrusive monitoring of hydraulic systems.

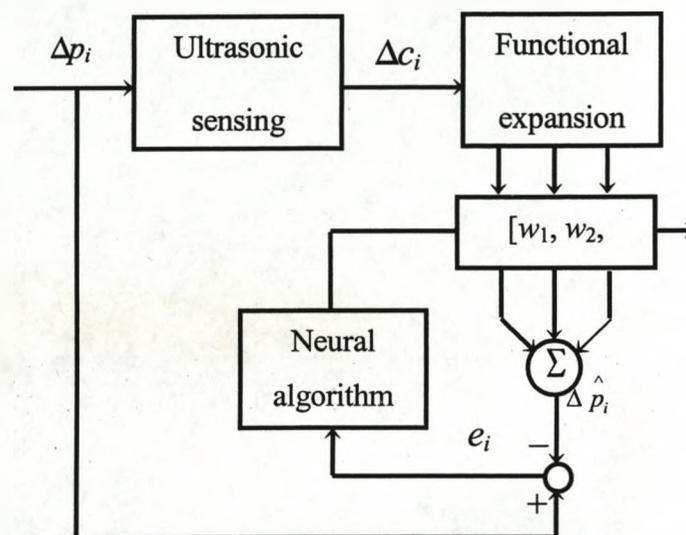


Fig.4.7 Schematic diagram of the inverse modelling process

The scheme of implementing such a function is shown in Fig.4.9. Either an 8 bit microcontroller 80C31 from Intel MCS-51 family or a 16 bit 80C196 from Intel MCS-96 family could be chosen for on-line signal processing. The purpose of using a keyboard and an E²PROM is to input the coefficients of the model and store them securely. It is recommended

that the coefficients are determined and entered after in-situ calibration to ensure that accurate measurements are taken.

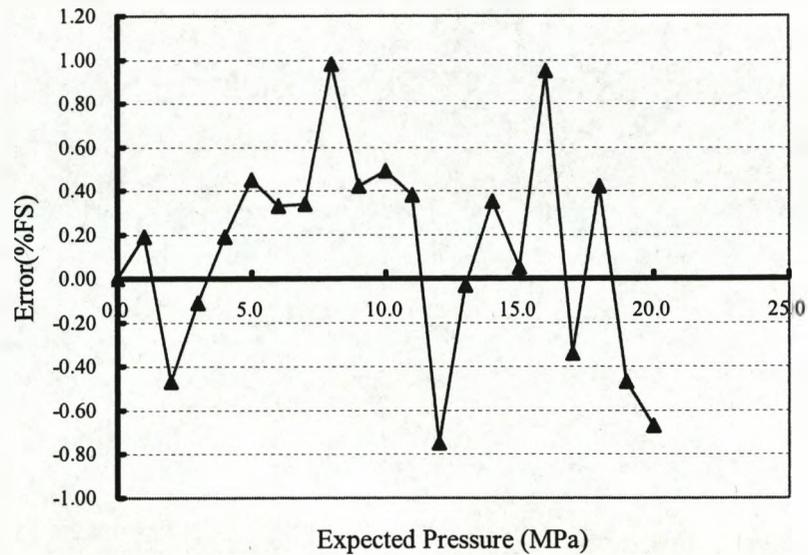


Fig.4.8 Relative error (%FS) of estimated change in pressure against the expected variation in pressure.

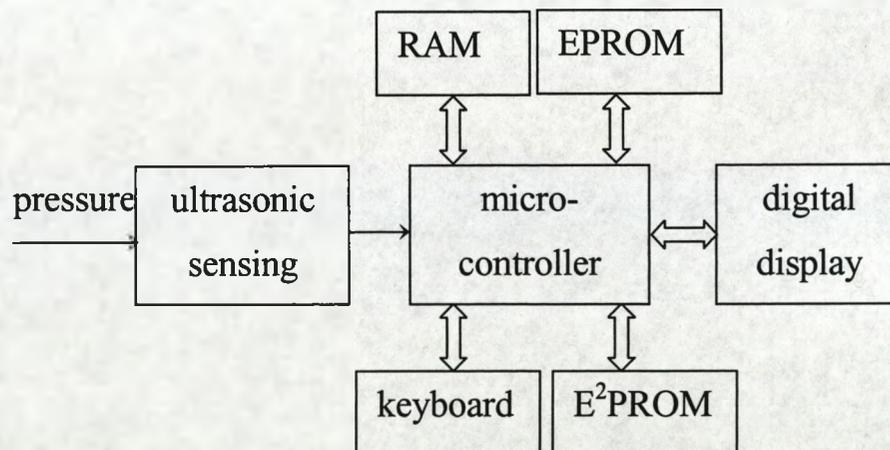


Fig.4.9 Implementation scheme of on-line measurement

4.7 Summary

The principle of modelling the newly developed non-intrusive pressure measuring system is described. Its direct model and the inverse model are constructed based on a functional-link neural network. They can be utilised to analyse the input-output characteristics of the system and provide direct digital readings of the measurand. The results have shown the effectiveness of the proposed method with a low cost of computational complexity. The maximum error (%FS) of estimated change in acoustic velocity against the expected variation of velocity is 1.12 %FS. The maximum error (%FS) of estimated change in pressure against the expected variation of pressure is 0.98 %FS. The inverse model can be easily implemented in a microcontroller to accurately display the measurand digitally.

5 MLP-Based Modelling of the Non-Intrusive Monitoring System — Part I

The following notation is used: (1) using small *italic* letters for scalars; (2) using **bold** non-italic letters for vectors; (3) using capital **BOLD** non-italic letters for matrices.

5.1 Neuron model

5.1.1 Single-input neuron

A single-input neuron is shown in Fig.5.1. The scalar input p is multiplied by the scalar weight w , to form the weighted input wp , one of the terms that is sent to the summer. The neuron has a scalar bias, b . The other input, 1, is multiplied by b and then passed to the summer. The summer output, often referred to as the net input, goes into a transfer function f , which produces the scalar output a .

The transfer function net input n , again a scalar, is the sum of the weighted input wp and the bias b . This sum is the argument of the transfer function f . Here f is a transfer function, which takes the argument n and produces the output a . The neuron output is calculated as:

$$a = f(wp + b). \quad (5.1)$$

Note that w and b are both adjustable scalar parameters of the neuron. The central idea of neural networks is that such parameters can be adjusted, so that the network exhibits some desired or interesting behaviour. Thus, we can train the network to do a particular job by adjusting the weight or bias parameters.

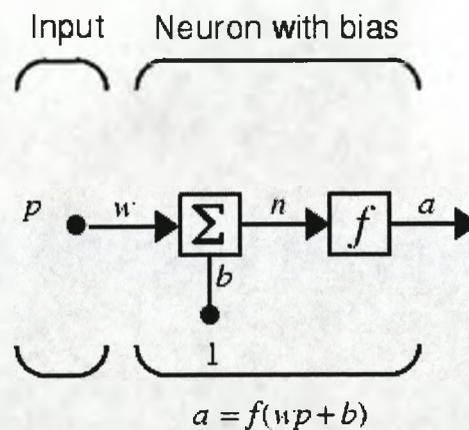


Fig.5.1 Single-input neuron

5.1.2 Transfer functions

Three of the most commonly used functions are shown below.

The hard-limit transfer function:

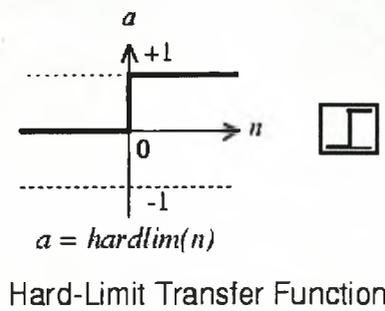


Fig.5.2 Hard-limit transfer function

The hard-limit transfer function, shown in Fig.5.2, limits the output of the neuron to 0 if the net input argument n is less than 0; or 1, if n is greater than or equal to 0. This function may be used to create neurons that make classification decisions.

The linear transfer function:

As illustrated in Fig.5.3, the output of a linear transfer function is equal to its input:

$$a = n. \quad (5.2)$$

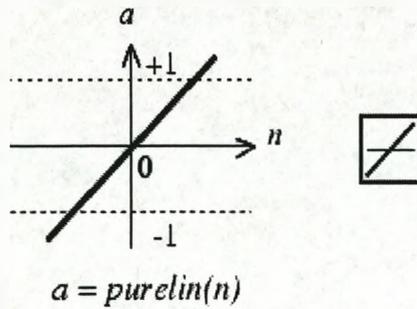


Fig.5.3 Linear transfer function

The log-sigmoid transfer function:

The log-sigmoid transfer function is shown in Fig.5.4. It takes the input, which may have any value between plus and minus infinity, and squashes the output into the range 0 to 1, according to the expression:

$$a = \frac{1}{1 + e^{-n}}. \quad (5.3)$$

This transfer function is commonly used in multilayer networks that are trained using the back-propagation algorithm, in part because it is differentiable [87].

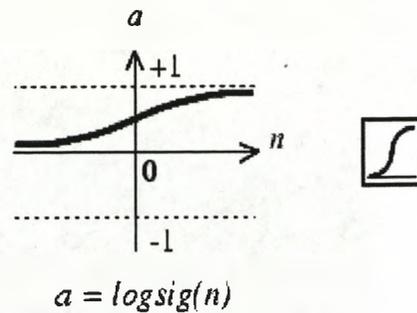


Fig.5.4 Log-sigmoid transfer function

The symbol in the square to the right of each transfer function graph represents the associated transfer function. These icons will replace the general f in the boxes of network diagrams to show the particular transfer function being used.

5.1.3 Multiple-input neuron

A neuron with R inputs is shown in Fig.5.5. Here the individual inputs p_1, p_2, \dots, p_R are each weighted by corresponding elements $w_{1,1}, w_{1,2}, \dots, w_{1,R}$ of the weight matrix \mathbf{W} . The weighted values are fed to the summing junction. Their sum is $\mathbf{W}\mathbf{p}$, the dot product of the (single row) matrix \mathbf{W} and the vector \mathbf{p} .

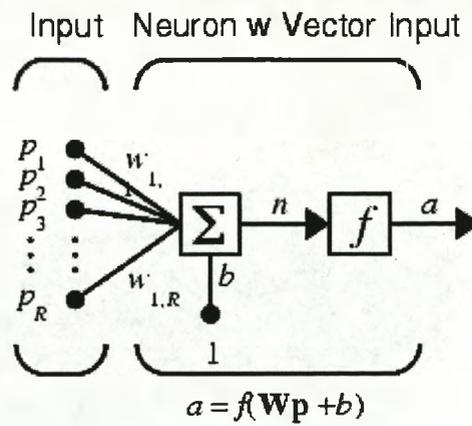


Fig.5.5 Multiple-input neuron

The neuron has a bias b , which is summed with the weighted inputs to form the net input n :

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b. \quad (5.4)$$

This expression can be written in matrix form:

$$n = \mathbf{Wp} + b. \quad (5.5)$$

Now the neuron output can be written as:

$$\begin{aligned} a &= f(n) \\ &= f(\mathbf{W}\mathbf{p} + b) \end{aligned} \tag{5.6}$$

In the indices of the elements of the weight matrix, the first index indicates the particular neuron destination for that weight. The second index indicates the source of the signal fed to the neuron. Thus, $w_{1,2}$ means that this weight represents the connection to the first (and only) neuron from the second source.

The figure of a single neuron shown in Fig.5.5 contains a lot of detail. When considering networks with many neurons and perhaps layers of many neurons, so much detail would cause the main thoughts to be lost. Thus, an abbreviated notation for an individual neuron is illustrated in Fig.5.6 [75].

In Fig.5.6, the input vector \mathbf{p} is represented by the solid dark vertical bar at the left. The dimensions of \mathbf{p} are shown below the symbol \mathbf{p} as $R \times 1$. \mathbf{p} is a input vector of R elements. These inputs go to the weight matrix \mathbf{W} , which has R columns but only one row for this single neuron case. A constant 1 enters the neuron as an input and is multiplied by a scalar bias b . The net input to the transfer function f is n , the sum of the bias b and the

product \mathbf{Wp} . This sum is passed to the transfer function f to get the neuron's output a , which in this case is a scalar. Note that the number of inputs is set by the external specifications of the problem.

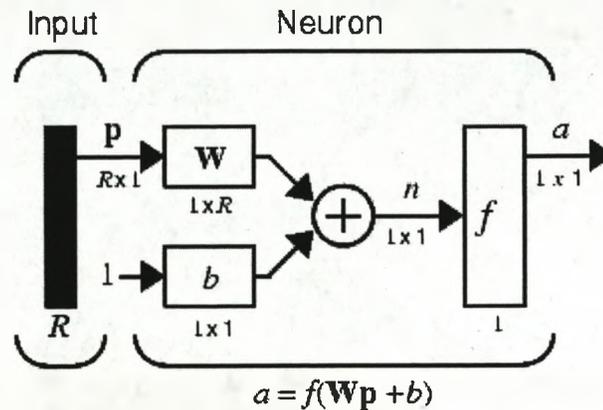


Fig.5.6 Neuron with R inputs, abbreviated notation

5.2 Network Architectures

5.2.1 A layer of neurons

A single-layer network with R input elements and S neurons is shown in Fig.5.7. The layer includes the weight matrix, the summers, the bias vector \mathbf{b} , the transfer function boxes and the output vector \mathbf{a} . The array of inputs, vector \mathbf{p} , is not called a layer here.

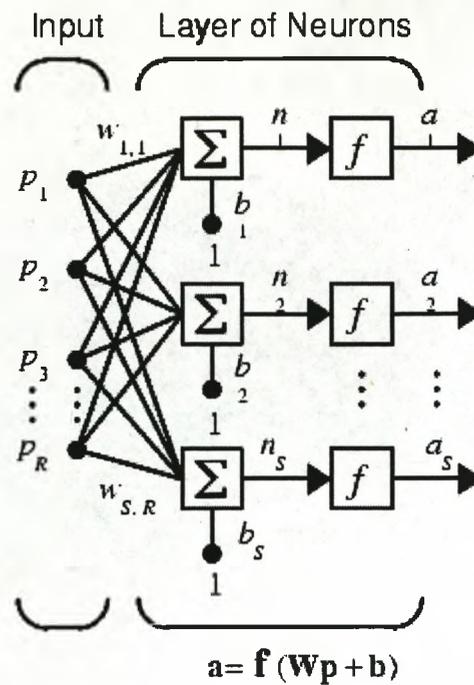


Fig.5.7 Layer of S neurons

The input vector elements enter the network through the weight matrix

\mathbf{W} :

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix} \quad (5.7)$$

Note that the row indices on the elements of matrix \mathbf{W} indicate the destination neuron of the weight, and the column indices indicate which source is the input for that weight.

The i th neuron has a summer that gathers its weighted inputs and bias to form its own scalar output $n(i)$. The various $n(i)$ taken together form an S -element net input vector \mathbf{n} . Finally, the neuron layer outputs form a column vector \mathbf{a} . The expression for \mathbf{a} is:

$$\mathbf{a} = \mathbf{f}(\mathbf{W}\mathbf{p} + \mathbf{b}). \quad (5.8)$$

This expression is also shown at the bottom of the Fig.5.7.

The S neuron R input one-layer network can also be drawn in abbreviated notation, as shown in Fig.5.8.

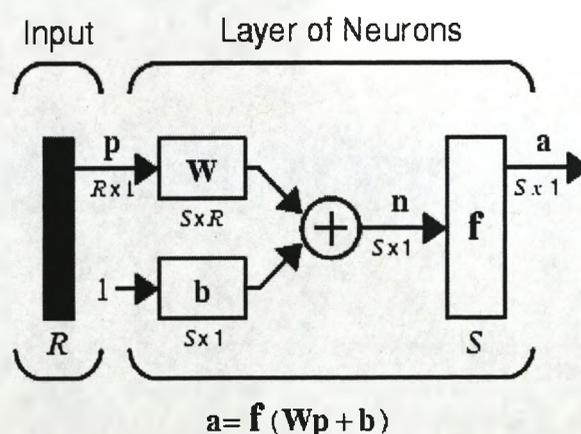


Fig.5.8 Neuron with R inputs, abbreviated notation

Here \mathbf{p} is an R length input vector, \mathbf{W} is an $S \times R$ matrix, \mathbf{a} and \mathbf{b} are S length vectors. The neuron layer includes the weight matrix, the multiplication operations, the bias vector, the summer, the transfer function boxes and the output vector.

5.2.2 Multiple layers of neurons

Now consider a network with several layers. Each layer has a weight matrix, a bias vector, a net input vector and an output vector. To distinguish between the weight matrices, output vectors, etc., for each of these layers, some additional notation is introduced. We append the number of the layer as a superscript to the variable of interest. For instance, the weight matrix for the first layer is written as \mathbf{W}^1 , and the weight matrix for the second layer is written as \mathbf{W}^2 . This notation is used in the three-layer network shown in Fig.5.9.

The network shown in Fig.5.9 has R inputs, S^1 neurons in the first layer, S^2 neurons in the second layer, etc. A constant input 1 is fed to the biases for each neuron. Layer 1 can be viewed as a one-layer network with R inputs, S^1 neurons, and an $S^1 \times R$ weight matrix \mathbf{W}^1 . The input to layer 1 is \mathbf{p} ; the output is \mathbf{a}^1 . The outputs of each intermediate layer are the inputs to the

following layer. Thus layer 2 can be viewed as a one-layer network with S^1 inputs, S^2 neurons, and an $S^2 \times S^1$ weight matrix \mathbf{W}^2 . The input to layer 2 is \mathbf{a}^1 ; the output is \mathbf{a}^2 . Now, all the vectors and matrices of layer 2 are identified. Layer 3 can be viewed as a one-layer network with S^2 inputs, S^3 neurons, and an $S^3 \times S^2$ weight matrix \mathbf{W}^3 . The input to layer 3 is \mathbf{a}^2 ; the output is \mathbf{a}^3 . Layer 3 is an output layer, which produces the network output. Layer 1 and layer 2 are hidden layers in this three-layer network.

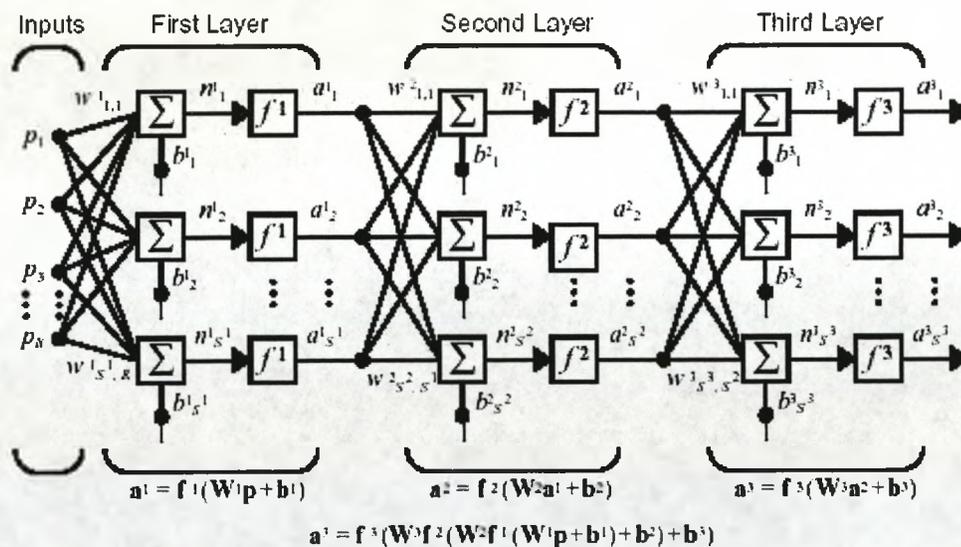


Fig. 5.9 Three-layer feed-forward neural network

The same three-layer network discussed previously also can be drawn using abbreviated notation, as shown in Fig.5.10.

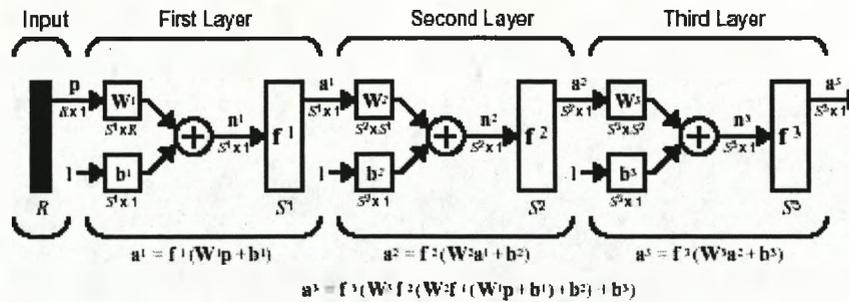


Fig.5.10 Three-layer feed-forward network, abbreviated notation

Here we assume that the output of the third layer, a^3 , is the network output of interest, and we have labelled this output as y . We will use this notation to specify the output of multilayer networks.

5.3 Back-propagation algorithm

5.3.1 Steepest descent algorithm

For our purpose, to optimise a performance index $F(\mathbf{x})$ means to find the value of \mathbf{x} that minimises $F(\mathbf{x})$. The optimisation algorithms discussed here are iterative. Starting from some initial guess and then updating our guess in stages according to the general minimisation algorithm [75]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k, \quad (5.9)$$

or

$$\Delta \mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k = \alpha_k \mathbf{p}_k, \quad (5.10)$$

where the vector \mathbf{p}_k represents a search direction, and the positive scalar α_k is the learning rate, which determines the length of the step.

The steepest descent algorithm:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{g}_k, \quad (5.11)$$

where \mathbf{g}_k is the gradient evaluated at the old guess \mathbf{x}_k :

$$\mathbf{g}_k \equiv \nabla F(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_k}. \quad (5.12)$$

5.3.2 Newton's method

The derivation of the steepest descent algorithm is based on the first-order Taylor series expansion of $F(\mathbf{x})$ about the old guess. Newton's method is based on the second-order Taylor series expansion [75]:

$$F(\mathbf{x}_{k+1}) = F(\mathbf{x}_k + \Delta\mathbf{x}_k) \approx F(\mathbf{x}_k) + \mathbf{g}_k^T \Delta\mathbf{x}_k + \frac{1}{2} \Delta\mathbf{x}_k^T \mathbf{A}_k \Delta\mathbf{x}_k, \quad (5.13)$$

where \mathbf{A}_k is the Hessian evaluated at the old guess \mathbf{x}_k :

$$\mathbf{A}_k \equiv \nabla^2 F(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_k}. \quad (5.14)$$

Newton's method is defined:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1} \mathbf{g}_k. \quad (5.15)$$

A brief comparison between some of the properties of Newton's method and those of steepest descent:

Newton's method usually produces faster convergence than steepest descent. However, the behavior of Newton's method can be quite complex. In addition to the problem of convergence to saddle points (which is very unlikely with steepest descent), it is possible for the algorithm to oscillate or diverge. Steepest descent is guaranteed to converge, if the learning rate is not too large.

We will discuss a variation of Newton's method in next chapter, i.e. L-M algorithm, which is well suited to the neural network training. It eliminates the divergence problem by using steepest descent steps whenever divergence begins to occur.

Another problem with Newton's method is that it requires the computation and storage of the Hessian matrix, as well as its inverse.

5.3.3 The back-propagation algorithm

Consider the M layer, $R - S^1 - S^2 - \dots - S^M$ network in abbreviated notation shown in Fig. 5.11. It has $M-1$ hidden layers and an output layer. The output of one layer becomes the input to the following layer. This operation is described by the following equations [75]:

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{w}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}) \text{ for } m = 0, 1, \dots, M-1, \quad (5.16)$$

where M is the number of layers in the network.

The neurons in the first layer receive the external inputs:

$$\mathbf{a}^0 = \mathbf{p}, \quad (5.17)$$

which provides the starting point for Eq.(5.16).

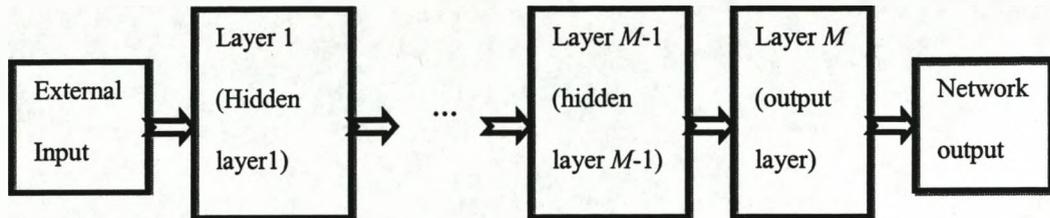


Fig.5.11 M layer feed-forward network

The outputs of the neurons in the third layer are the network outputs:

$$\mathbf{a} = \mathbf{a}^M. \quad (5.18)$$

Performance Index [75]:

The back-propagation algorithm for the multilayer networks is a generalisation of the LMS algorithm, and both algorithms use the same performance index: mean square error. The algorithm is provided with a set of examples of proper network behavior:

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\}, \quad (5.19)$$

where \mathbf{p}_q is an input to the network, and \mathbf{t}_q is the corresponding target output ($q = 1, 2, \dots, Q$).

As each input is applied to the network, the network output is compared with the target. The algorithm should adjust the network parameters in order to minimise the mean square error:

$$F(\mathbf{x}) = E(\mathbf{e}^T \mathbf{e}) = E[(\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a})] \quad (5.20)$$

where \mathbf{x} is the vector of network weights and biases.

As with the LMS algorithm, we will approximate the mean square error by:

$$\hat{F}(\mathbf{x}) = (\mathbf{t}(k) - \mathbf{a}(k))^T (\mathbf{t}(k) - \mathbf{a}(k)) = \mathbf{e}^T(k) \mathbf{e}(k), \quad (5.21)$$

where the expectation of the squared error is replaced by the squared error at iteration k .

The steepest decent algorithm for the approximate mean square error is

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha \frac{\partial \hat{F}}{\partial w_{i,j}^m}, \quad (5.22)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial \hat{F}}{\partial b_i^m}, \quad (5.23)$$

where α is the learning rate.

Computation of the partial derivatives [75]:

Because the error is an indirect function of the weights in the hidden layers, the chain rule of calculus is used to calculate the derivatives in Eq.(5.22) and Eq.(5.23):

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial w_{i,j}^m}, \quad (5.24)$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial b_i^m}. \quad (5.25)$$

Note that the net input to layer m is an explicit function of the weights and bias in that layer:

$$n_i^m = \sum_{j=1}^{S^{m-1}} w_{i,j}^m a_j^{m-1} + b_i^m. \quad (5.26)$$

Therefore, the second term in each of Eq.(5.24) and Eq.(5.25) can be computed:

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1}, \quad (5.27)$$

$$\frac{\partial n_i^m}{\partial b_i^m} = 1. \quad (5.28)$$

If we define:

$$s_i^m \equiv \frac{\partial \hat{F}}{\partial n_i^m}, \quad (5.29)$$

(the sensitivity of \hat{F} to changes in the i th element of the net input at layer m), then Eq.(5.24) and Eq.(5.25) can be simplified to:

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = s_i^m a_j^{m-1}, \quad (5.30)$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = s_i^m. \quad (5.31)$$

We can now express the approximate steepest decent algorithm as:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1}, \quad (5.32)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m. \quad (5.33)$$

In matrix form it becomes:

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T, \quad (5.34)$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m. \quad (5.35)$$

where

$$\mathbf{s}^m \equiv \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \left[\frac{\partial \hat{F}}{\partial n_1^m}, \frac{\partial \hat{F}}{\partial n_2^m}, \dots, \frac{\partial \hat{F}}{\partial n_{s^m}^m} \right]^T. \quad (5.36)$$

Backpropagating the sensitivities [75]:

It now remains for us to compute the sensitivities \mathbf{s}^m . The process of backpropagation describes a recurrence relationship in which the sensitivity at layer m is computed from the sensitivity at layer $m+1$.

To derive the recurrence relationship for the sensitivities, we use the following Jacobian matrix:

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \equiv \begin{bmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \frac{\partial n_1^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_1^{m+1}}{\partial n_{s^m}^m} \\ \frac{\partial n_2^{m+1}}{\partial n_1^m} & \frac{\partial n_2^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_2^{m+1}}{\partial n_{s^m}^m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_1^m} & \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_{s^m}^m} \end{bmatrix}. \quad (5.37)$$

Consider the i, j element of the matrix:

$$\begin{aligned}
\frac{\partial n_i^{m+1}}{\partial n_j^m} &= \frac{\partial \left(\sum_{l=1}^{S^m} w_{i,l}^{m+1} a_l^m + b_i^{m+1} \right)}{\partial n_j^m} \\
&= w_{i,j}^{m+1} \frac{\partial a_j^m}{\partial n_j^m} \\
&= w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} \\
&= w_{i,j}^{m+1} \dot{f}^m(n_j^m),
\end{aligned} \tag{5.38}$$

where

$$\dot{f}^m(n_j^m) = \frac{\partial f^m(n_j^m)}{\partial n_j^m}. \tag{5.39}$$

Therefore the Jacobian matrix can be written as:

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \equiv \mathbf{W}^{m+1} \dot{\mathbf{F}}^m(\mathbf{n}^m), \tag{5.40}$$

where

$$\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \dots & 0 \\ 0 & \dot{f}^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & \dots & & \dot{f}^m(n_{S^m}^m) \end{bmatrix} \quad (5.41)$$

We can now write out the recurrence relation for the sensitivity in matrix form:

$$\begin{aligned} \mathbf{s}^m &= \frac{\partial \hat{F}}{\partial \mathbf{n}^m} \\ &= \left(\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \right) \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} \\ &= \dot{\mathbf{F}}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} \\ &= \dot{\mathbf{F}}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \mathbf{s}^{m+1} \end{aligned} \quad (5.42)$$

We can see that the sensitivities are propagated backward through the network from the last layer to the first layer:

$$\mathbf{s}^1 \leftarrow \mathbf{s}^2 \leftarrow \dots \leftarrow \mathbf{s}^{M-1} \leftarrow \mathbf{s}^M \quad (5.43)$$

The starting point, s^M , is needed for the recurrence relation of Eq.(5.42).

This is obtained at the final layer:

$$\begin{aligned}
 s_i^M &\equiv \frac{\partial \hat{F}}{\partial n_i^M} \\
 &= \frac{\partial (\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a})}{\partial n_i^M} \\
 &= \frac{\partial \sum_{j=1}^{s^M} (t_j - a_j)^2}{\partial n_i^M} \\
 &= -2(t_i - a_i) \frac{\partial a_i}{\partial n_i^M}.
 \end{aligned} \tag{5.44}$$

Now, since:

$$\begin{aligned}
 \frac{\partial a_i}{\partial n_i^M} &= \frac{\partial a_i^M}{\partial n_i^M} \\
 &= \frac{\partial f^M(n_j^M)}{\partial n_i^M} \\
 &= \dot{f}^M(n_j^M),
 \end{aligned} \tag{5.45}$$

we can write

$$s_i^M = -2(t_i - a_i) \dot{f}^M(n_j^M). \tag{5.46}$$

This can be expressed in matrix form as:

$$\mathbf{s}^M = -\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}). \quad (5.47)$$

5.3.4 Summary of the back-propagation algorithm

Definition of the performance index:

$$\begin{aligned} F(\mathbf{x}) &= E(\mathbf{e}^T \mathbf{e}) \\ &= E[(\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a})] \end{aligned} \quad (5.48)$$

Definition of the approximate performance index:

$$\begin{aligned} \hat{F}(\mathbf{x}) &= \mathbf{e}^T(k) \mathbf{e}(k) \\ &= (\mathbf{t}(k) - \mathbf{a}(k))^T (\mathbf{t}(k) - \mathbf{a}(k)). \end{aligned} \quad (5.49)$$

Definition of the sensitivity:

$$\mathbf{s}^m \equiv \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial \hat{F}}{\partial n_1^m} \\ \frac{\partial \hat{F}}{\partial n_2^m} \\ \vdots \\ \frac{\partial \hat{F}}{\partial n_{s^m}^m} \end{bmatrix} \quad (5.50)$$

The whole operation process of the back-propagation algorithm can be divided into three steps: forward propagation, backward propagation and weight update, as summarised below.

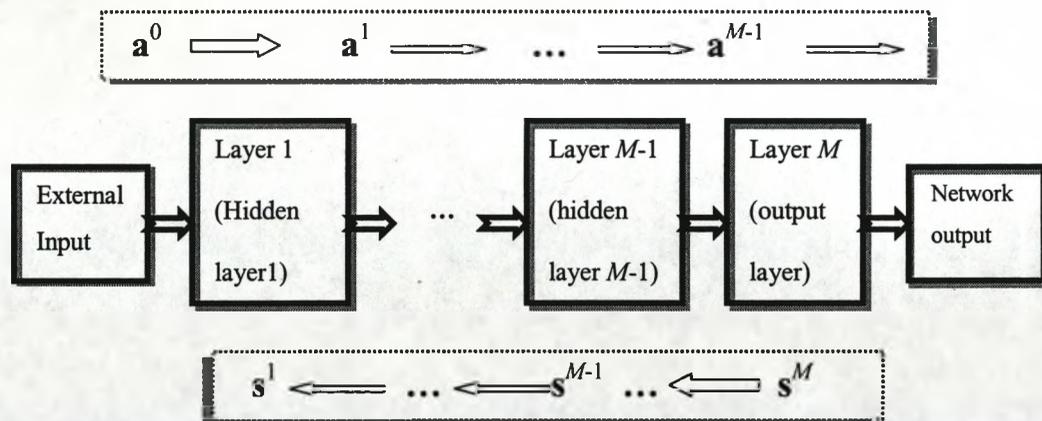


Fig.5.12 Operation of the back-propagation algorithm

The first step, i.e. forward propagation as shown in Fig.5.12, is to propagate the input forward through the network:

$$\mathbf{a}^0 = \mathbf{p}, \quad (5.51)$$

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{w}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}) \text{ for } m = 0, 1, \dots, M-1, \quad (5.52)$$

$$\mathbf{a} = \mathbf{a}^M. \quad (5.53)$$

The next step, i.e. backward propagation, is to propagate the sensitivities backward through the network:

$$\mathbf{s}^M = -\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}). \quad (5.54)$$

$$\mathbf{s}^m = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}, \text{ for } m = M-1, \dots, 2, 1. \quad (5.55)$$

Finally, the weights and biases are updated using the approximate steepest descent rule:

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T, \quad (5.56)$$

$$b^m(k+1) = b^m(k) - \alpha s^m. \quad (5.57)$$

5.4 Summary

The mathematical model of the neuron is introduced. How these neurons can be interconnected to form a variety of network architectures is explained.

The back-propagation algorithm can be used to train multilayer networks. It is an approximate steepest decent algorithm, in which the performance index is mean square error.

The architecture of a multilayer network is not completely constrained by the problem to be solved. The number of inputs to the network is constrained by the problem, and the number of neurons in the output layer is constrained by the number of outputs required by the problem. However, the number of layers between network inputs and the output layer and the sizes of the layers are up to the designer.

6 MLP-Based Modelling of the Non-Intrusive Monitoring System — Part II

6.1 Principle of system modelling based on multilayer networks

6.1.1 Three-layer feed-forward network

This section presents the architecture of the network that is used with the L-M back-propagation algorithm to build the direct model and inverse model for the non-intrusive pressure monitoring system.

This feed-forward network is created with three layers as shown in Fig.6.1, that is, an output layer (layer 3) producing the network output and two hidden layers (layers 1 and 2). Here the array of inputs is not included in or called a layer. Each layer has its own a weight matrix, its own bias

vector, and an output vector. A constant input 1 is fed to the biases for each neuron.

To distinguish between the weight matrices, output vectors, etc., for each of these layers, we append the number of the layer as a superscript to the variable of interest. For example, the weight matrix for the first layer is written as \mathbf{W}^1 , and the weight matrix for the second layer is written as \mathbf{W}^2 . The network has R inputs, S^1 neurons in the first layer, S^2 neurons in the second layer, S^3 neurons in the third layer.

To identify the structure of a multilayer network, the following shorthand notation is used, where the number of inputs is followed by the number of neurons in each layer:

$$R - S^1 - S^2 - S^3. \quad (6.1)$$

The output of the first layer is the input to the second layer, and the output of the second layer is the input to the third layer. Layer 1 can be viewed as a one-layer network with R inputs, S^1 neurons, and a weight matrix \mathbf{W}^1 , the bias vector \mathbf{b}^1 , S^1 summers, S^1 transfer function boxes and the output vector \mathbf{a}^1 . The input to layer 1 is \mathbf{p} , the net input vector is \mathbf{n}^1 and the output

is \mathbf{a}^1 . Here the input vector \mathbf{p} is represented by the solid dark vertical bar at the left. Here \mathbf{p} is an R length input vector, \mathbf{W}^1 is an $S^1 \times R$ matrix, \mathbf{a}^1 and \mathbf{b}^1 are S^1 length vectors.

$$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{n}^1) = \mathbf{f}^1(\mathbf{w}^1\mathbf{p} + \mathbf{b}^1), \quad (6.2)$$

$$\mathbf{a}^1 = [a^1_1, a^1_2, \dots, a^1_{S^1}]^T, \quad (6.3)$$

$$\mathbf{f}^1(\mathbf{n}^1) = [f^1_1(n^1_1), f^1_2(n^1_2), \dots, f^1_{S^1}(n^1_{S^1})]^T, \quad (6.4)$$

$$\mathbf{W}^1 = \begin{bmatrix} w^1_{1,1} & w^1_{1,2} & \dots & w^1_{1,R} \\ w^1_{2,1} & w^1_{2,2} & \dots & w^1_{2,R} \\ \vdots & \vdots & & \vdots \\ w^1_{S^1,1} & w^1_{S^1,2} & \dots & w^1_{S^1,R} \end{bmatrix}, \quad (6.5)$$

$$\mathbf{p} = [p_1, p_2, \dots, p_R]^T, \quad (6.6)$$

$$\mathbf{b}^1 = [b^1_1, b^1_2, \dots, b^1_{S^1}]^T. \quad (6.7)$$

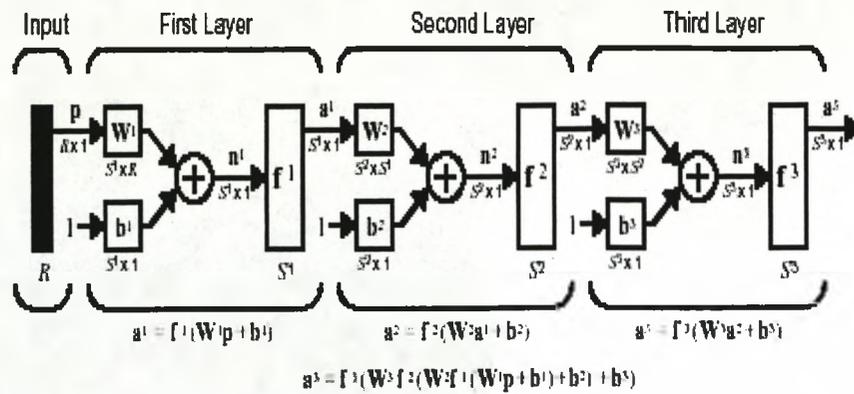


Fig.6.1 Diagram of a three-layer feed-forward network

Layer 2 can be viewed as a one-layer network with S^1 inputs, S^2 neurons, and an $S^2 \times S^1$ weight matrix \mathbf{W}^2 . The input to layer 2 is \mathbf{a}^1 , the net input vector is \mathbf{n}^1 and the output is \mathbf{a}^2 .

$$\mathbf{a}^2 = \mathbf{f}^2(\mathbf{n}^2) = \mathbf{f}^2(\mathbf{w}^2 \mathbf{a}^1 + \mathbf{b}^2), \quad (6.8)$$

$$\mathbf{a}^2 = [a^2_1 \ a^2_2 \ \dots \ a^2_{S^2}]^T, \quad (6.9)$$

$$\mathbf{f}^2(\mathbf{n}^2) = [f^2_1(n^2_1), f^2_2(n^2_2), \dots, f^2_{S^2}(n^2_{S^2})]^T, \quad (6.10)$$

$$\mathbf{W}^2 = \begin{bmatrix} w^2_{1,1} & w^2_{1,2} & \dots & w^2_{1,S^1} \\ w^2_{2,1} & w^2_{2,2} & \dots & w^2_{2,S^1} \\ \vdots & \vdots & & \vdots \\ w^2_{S^2,1} & w^2_{S^2,2} & \dots & w^2_{S^2,S^1} \end{bmatrix}, \quad (6.11)$$

$$\mathbf{b}^2 = [b^2_1, b^2_2, \dots, b^2_{S^2}]^T \quad (6.12)$$

Layer 3 can be viewed as a one-layer network with S^2 inputs, S^3 neurons, and an $S^3 \times S^2$ weight matrix \mathbf{W}^3 . The input to layer 3 is \mathbf{a}^2 , the net input vector is \mathbf{n}^3 and the output is \mathbf{a}^3 .

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{n}^3) = \mathbf{f}^3(\mathbf{W}^3 \mathbf{a}^2 + \mathbf{b}^3), \quad (6.13)$$

$$\mathbf{a}^3 = [a^3_1, a^3_2, \dots, a^3_{S^3}]^T, \quad (6.14)$$

$$\mathbf{f}^3(\mathbf{n}^3) = [f^3_1(n^3_1), f^3_2(n^3_2), \dots, f^3_{S^3}(n^3_{S^3})]^T, \quad (6.15)$$

$$\mathbf{W}^3 = \begin{bmatrix} w^3_{1,1} & w^3_{1,2} & \dots & w^3_{1,S^2} \\ w^3_{2,1} & w^3_{2,2} & \dots & w^3_{2,S^2} \\ \vdots & \vdots & & \vdots \\ w^3_{S^3,1} & w^3_{S^3,2} & \dots & w^3_{S^3,S^2} \end{bmatrix}, \quad (6.16)$$

$$\mathbf{b}^3 = [b^3_1, b^3_2, \dots, b^3_{S^3}]^T \quad (6.17)$$

The transfer functions for all the neurons in the same layer are chosen to be the same:

$$\begin{aligned}
 f^1_1(n^1_1) &= f^1(n^1_1), \\
 f^1_2(n^1_2) &= f^1(n^1_2), \\
 &\dots, \\
 f^1_{s^1}(n^1_{s^1}) &= f^1(n^1_{s^1}),
 \end{aligned}
 \tag{6.18}$$

$$\begin{aligned}
 f^2_1(n^2_1) &= f^2(n^2_1), \\
 f^2_2(n^2_2) &= f^2(n^2_2), \\
 &\dots, \\
 f^2_{s^2}(n^2_{s^2}) &= f^2(n^2_{s^2}),
 \end{aligned}
 \tag{6.19}$$

$$\begin{aligned}
 f^3_1(n^3_1) &= f^3(n^3_1), \\
 f^3_2(n^3_2) &= f^3(n^3_2), \\
 &\dots, \\
 f^3_{s^3}(n^3_{s^3}) &= f^3(n^3_{s^3}).
 \end{aligned}
 \tag{6.20}$$

In summary, the network output vector can be expressed as the following compact form:

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{w}^3 \mathbf{f}^2(\mathbf{w}^2 \mathbf{f}^1(\mathbf{w}^1 \mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3).
 \tag{6.21}$$

6.1.2 Levenberg-Marquardt algorithm

The development of the back-propagation learning algorithm was regarded as a significant milestone in the area of neural networks. However, considerable research on methods to accelerate the convergence of the algorithm has been carried out since then. This research falls roughly into two categories: heuristic approaches [76]-[79] and numerical optimisation techniques. The approaches in the first category include such ideas as using momentum [80], varying the learning rate [81] and stochastic learning [82]. In the second category there are conjugate gradient method and Levenberg-Marquardt (L-M) technique [83]-[85]. Among these mentioned methods, L-M algorithm is widely regarded as the most efficient in the sense of realisation accuracy [85].

Basic algorithm [75][85]:

The L-M (Levenberg-Marquardt) algorithm is a variation of Newton's method designed for minimising functions that are sums of squares of other non-linear functions.

Newton's method for optimising a performance index, $F(\mathbf{x})$, i.e. a sum of squares, is:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1} \mathbf{g}_k, \quad (6.22)$$

where

$$\mathbf{A}_k \equiv \nabla^2 F(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_k}, \quad (6.23)$$

and

$$\mathbf{g}_k \equiv \nabla F(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_k}. \quad (6.24)$$

First, we want to find $\nabla F(\mathbf{x})$. Assume that $F(\mathbf{x})$, a sum of squares function, takes the following form:

$$F(\mathbf{x}) = \sum_{i=1}^N v_i^2(\mathbf{x}) = \mathbf{v}^T(\mathbf{x}) \mathbf{v}(\mathbf{x}), \quad (6.25)$$

then, the j th element of the gradient would be:

$$\begin{aligned} [\nabla F(\mathbf{x})]_j &= \frac{\partial F(\mathbf{x})}{\partial x_j} \\ &= 2 \sum_{i=1}^N v_i(\mathbf{x}) \frac{\partial v_i(\mathbf{x})}{\partial x_j}. \end{aligned} \quad (6.26)$$

Therefore, the gradient can be written in matrix form:

$$\nabla F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{v}(\mathbf{x}), \quad (6.27)$$

where

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial v_1(\mathbf{x})}{\partial x_1} & \frac{\partial v_1(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial v_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial v_2(\mathbf{x})}{\partial x_1} & \frac{\partial v_2(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial v_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial v_N(\mathbf{x})}{\partial x_1} & \frac{\partial v_N(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial v_N(\mathbf{x})}{\partial x_n} \end{bmatrix}. \quad (6.28)$$

is the Jacobian matrix.

Next, we want to find $\nabla^2 F(\mathbf{x})$, the Hessian matrix. The k, j element of the

Hessian matrix would be:

$$\begin{aligned} [\nabla^2 F(\mathbf{x})]_{k,j} &= \frac{\partial^2 F(\mathbf{x})}{\partial x_k \partial x_j} \\ &= 2 \sum_{i=1}^N \left\{ \frac{\partial v_i(\mathbf{x})}{\partial x_k} \frac{\partial v_i(\mathbf{x})}{\partial x_j} + v_i(\mathbf{x}) \frac{\partial^2 v_i(\mathbf{x})}{\partial x_k \partial x_j} \right\}. \end{aligned} \quad (6.29)$$

The Hessian matrix can then be expressed in matrix form:

$$\nabla^2 F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x}), \quad (6.30)$$

where

$$\mathbf{S}(\mathbf{x}) = \sum_{i=1}^N v_i(\mathbf{x}) \nabla^2 v_i(\mathbf{x}). \quad (6.31)$$

Assume that $\mathbf{S}(\mathbf{x})$ is small; the Hessian matrix can be approximated as:

$$\nabla^2 F(\mathbf{x}) \cong 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}). \quad (6.32)$$

Substituting Eq.(5.89) and Eq.(5.84) into Eq.(5.79), we obtain the Gauss-Newton method:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k)]^{-1} \mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k). \quad (6.33)$$

The advantage of Gauss-Newton method over the standard Newton's method is that it does not require calculation of second derivatives. To

overcome the problem that the matrix $\mathbf{H} = \mathbf{J}^T \mathbf{J}$ may not be invertible, the following modification is used to the approximate Hessian matrix:

$$\mathbf{G} = \mathbf{H} + \mu \mathbf{I} . \quad (6.34)$$

This leads to the Levenberg-Marquardt algorithm:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x}_k) \mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I}]^{-1} \mathbf{J}^T(\mathbf{x}_k) \mathbf{v}(\mathbf{x}_k) . \quad (6.35)$$

or

$$\Delta \mathbf{x}_k = -[\mathbf{J}^T(\mathbf{x}_k) \mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I}]^{-1} \mathbf{J}^T(\mathbf{x}_k) \mathbf{v}(\mathbf{x}_k) . \quad (6.36)$$

The very useful feature of this algorithm is that, as μ_k is increased, it approaches the steepest descent algorithm with small learning rate:

$$\begin{aligned} \mathbf{x}_{k+1} &\cong \mathbf{x}_k - \frac{1}{\mu_k} \mathbf{J}^T(\mathbf{x}_k) \mathbf{v}(\mathbf{x}_k) \\ &= \mathbf{x}_k - \frac{1}{2\mu_k} F(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_k}, \text{ for large } \mu_k . \end{aligned} \quad (6.37)$$

while as μ_k is decreased to zero, it becomes Gauss-Newton method [86].

The algorithm begins with μ_k set to some small value (e.g. $\mu_k = 0.01$). If a step does not yield a smaller value for $F(\mathbf{x})$, then the step is repeated with μ_k multiplied by some factor $\nu > 1$ (e.g. $\nu = 10$). Eventually $F(\mathbf{x})$ should decrease, since we could be taking a small step in the direction of steepest descent. If a step does produce a smaller value for $F(\mathbf{x})$, then μ_k is divided by ν for the next step, so that the algorithm will approach Gauss-Newton, which should provide faster convergence. The algorithm provides a nice compromise between the speed of Newton's method and the guaranteed convergence of steepest descent.

Application of L-M algorithm to multilayer network training problem [75]:

The performance index for multilayer network training is the mean squared error. Here we calculate the sum of squared errors over all inputs as an approximate mean square error:

$$\begin{aligned}
 F(\mathbf{x}) &= \sum_{q=1}^Q (\mathbf{t}_q - \mathbf{a}_q)^T (\mathbf{t}_q - \mathbf{a}_q) \\
 &= \sum_{q=1}^Q \mathbf{e}_q^T \mathbf{e}_q \\
 &= \sum_{q=1}^Q \sum_{j=1}^{S^H} e_{j,q}^2 \\
 &= \sum_{i=1}^N v_i^2,
 \end{aligned} \tag{6.38}$$

where $e_{j,q}$ is the j th element of the error for the q th input/target pair.

Comparing Eq.(6.38) and Eq.(6.25), it can see that it should be a straightforward matter to adapt the algorithm for network training. It is true in concept, but it does require some care in working out the details.

The key step in the L-M algorithm is the computation of the Jacobian matrix. To perform this computation, we will use a variation of the back-propagation algorithm.

As said earlier in the standard back-propagation procedure, the derivatives of the squared errors are computed with respect to the weights and biases of the network. To create the Jacobian matrix, there is a need to compute the derivatives of the errors, in stead of the derivatives of the squared errors.

Now let us modify the back-propagation algorithm to compute the elements of the Jacobian matrix. The error vector is:

$$\begin{aligned} \mathbf{v}^T &= [v_1, v_2, \dots, v_N] \\ &= [e_{1,1}, e_{2,1}, \dots, e_{s^M,1}, \dots, e_{1,2}, \dots, e_{s^M,2}], \end{aligned} \tag{6.39}$$

Where

$$N = Q \times S^M.$$

The parameter vector is:

$$\begin{aligned} \mathbf{x}^T &= [x_1 \ x_2 \ \dots \ x_n] \\ &= [w^{1,1,1} \ w^{1,1,2} \ \dots \ w^{1,s^1,R} \ b^1 \ \dots \ b^{s^1} \ w^{2,1,1} \ \dots \ b^M \ b^M] \end{aligned} \quad (6.40)$$

Where

$$n = S^1(R+1) + S^2(S^1+1) + \dots + S^M(S^{M-1}+1).$$

The Jacobian matrix for multilayer network training is:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w^{1,1,1}} & \frac{\partial e_{1,1}}{\partial w^{1,1,2}} & \dots & \frac{\partial e_{1,1}}{\partial w^{1,s^1,R}} & \frac{\partial e_{1,1}}{\partial b^1} & \dots & \frac{\partial e_{1,1}}{\partial b^{s^1}} \\ \frac{\partial e_{2,1}}{\partial w^{1,1,1}} & \frac{\partial e_{2,1}}{\partial w^{1,1,2}} & \dots & \frac{\partial e_{2,1}}{\partial w^{1,s^1,R}} & \frac{\partial e_{2,1}}{\partial b^1} & \dots & \frac{\partial e_{2,1}}{\partial b^{s^1}} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial e_{s^M,1}}{\partial w^{1,1,1}} & \frac{\partial e_{s^M,1}}{\partial w^{1,1,2}} & \dots & \frac{\partial e_{s^M,1}}{\partial w^{1,s^1,R}} & \frac{\partial e_{s^M,1}}{\partial b^1} & \dots & \frac{\partial e_{s^M,1}}{\partial b^{s^1}} \\ \frac{\partial e_{1,2}}{\partial w^{1,1,1}} & \frac{\partial e_{1,2}}{\partial w^{1,1,2}} & \dots & \frac{\partial e_{1,2}}{\partial w^{1,s^1,R}} & \frac{\partial e_{1,2}}{\partial b^1} & \dots & \frac{\partial e_{1,2}}{\partial b^{s^1}} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \end{bmatrix} \quad (6.41)$$

The elements of the Jacobian can now be computed. The Marquardt sensitivity is defined as:

$$\begin{aligned} \bar{s}^{m_{i,h}} &\equiv \frac{\partial v_h}{\partial n^{m_{i,q}}} \\ &= \frac{\partial e_{k,q}}{\partial n^{m_{i,q}}}, \end{aligned} \quad (6.42)$$

where

$$h = (q-1)s^M + k.$$

If x_l is a weight, then:

$$\begin{aligned} [\mathbf{J}]_{h,l} &= \frac{\partial v_h}{\partial x_l} \\ &= \frac{\partial e_{k,q}}{\partial w^{m_{i,j}}} \\ &= \frac{\partial e_{k,q}}{\partial n^{m_{i,q}}} \times \frac{\partial n^{m_{i,q}}}{\partial w^{m_{i,j}}} \\ &= \bar{s}^{m_{i,h}} \times \frac{\partial n^{m_{i,q}}}{\partial w^{m_{i,j}}} \\ &= \bar{s}^{m_{i,h}} \times a^{m-1}_{j,q}. \end{aligned} \quad (6.43)$$

If x_l is a bias, then:

$$\begin{aligned}
[\mathbf{J}]_{h,l} &= \frac{\partial v_h}{\partial x_l} \\
&= \frac{\partial e_{k,q}}{\partial b^{m_i}} \\
&= \frac{\partial e_{k,q}}{\partial n^{m_{i,q}}} \times \frac{\partial n^{m_{i,q}}}{\partial b^{m_i}} \\
&= \tilde{s}^{m_{i,h}} \times \frac{\partial n^{m_{i,q}}}{\partial b^{m_i}} \\
&= \tilde{s}^{m_{i,h}}
\end{aligned} \tag{6.44}$$

The Levenberg-Marquardt back-propagation is initialised with:

$$\tilde{\mathbf{S}}^M_q = -\dot{\mathbf{F}}^M(\mathbf{n}^M_q), \tag{6.45}$$

where

$$\mathbf{F}^m(\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \dots & 0 \\ 0 & \dot{f}^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \dot{f}^m(n_{s^m}^m) \end{bmatrix}$$

Each column of the matrix $\tilde{\mathbf{S}}^M_q$ must be back-propagated through the network using Eq.(5.36) to produce one row of the Jacobian matrix.

$$\begin{aligned}
\mathbf{s}^m &= \frac{\partial \hat{F}}{\partial \mathbf{n}^m} \\
&= \left(\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \right) \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} \\
&= \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} \\
&= \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1} .
\end{aligned} \tag{6.46}$$

The columns can also be back-propagated using:

$$\tilde{\mathbf{S}}^m_q = \dot{\mathbf{F}}^m(\mathbf{n}^m_q)(\mathbf{W}^{m+1})^T \tilde{\mathbf{S}}^{m+1}_q \tag{6.47}$$

The total Marquardt sensitivity matrices for each layer are then created by augmenting the matrices computed for each input:

$$\tilde{\mathbf{S}}^m = [\tilde{\mathbf{S}}^{m_1} | \tilde{\mathbf{S}}^{m_2} | \dots | \tilde{\mathbf{S}}^{m_\varrho}] \tag{6.48}$$

For each input presented to the network, S^M sensitivity vectors are back-propagated. This is because the derivatives of each individual error, rather than the derivative of the sum of squared errors, are computed. There are S^M errors (one for each element of the network output) for every input supplied to the network. For each error there is one row of the Jacobian

matrix. After the sensitivities have been back-propagated, the Jacobian matrix is computed using Eq.(6.43) and Eq.(6.44).

The iterations of the Levenberg-Marquardt back-propagation algorithm (L-M BP) can be summarised as follows:

Present all inputs to the network and compute the corresponding network outputs (using Eq.(5.51) and Eq.(5.52)) and the errors $\mathbf{e}_q = \mathbf{t}_q - \mathbf{a}_q^M$.

Compute the sum of squared errors over all inputs, $F(\mathbf{x})$, using Eq.(6.38).

Compute the Jacobian matrix, Eq.(6.41). Calculate the sensitivities with the recurrence relations Eq.(6.47), after initialising with Eq.(6.45).

Augment the individual matrices into the Marquard sensitivities using Eq.(6.48). Compute the elements of the Jacobian matrix with Eq.(6.43) and Eq.(6.44).

Solve Eq. (6.36) to obtain $\Delta\mathbf{x}_k$.

Recompute the sum of squared errors using $\mathbf{x}_k + \Delta\mathbf{x}_k$. If this new sum of squares is smaller than that computed in step 1, then divide μ by ν , let $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$ and go back to step 1. If the sum of squares is not reduced, then multiply μ by ν and go back to step 3.

The algorithm is able to reduce the sum of squares at each iteration. It requires more computation than any of the other algorithms, since it

involves a matrix inversion. Even given the large number of computations, however, the L-M BP algorithm appears to be the fastest neural network training algorithm for moderate numbers of network parameters.

6.2 Implementation of direct modelling with MLP for non-intrusive pressure monitoring system

6.2.1 Flow chart of direct modelling with MLP

Now we present the use of the above three-layer feed-forward network for the direct modelling of the non-intrusive pressure monitoring system. There are four steps in the training process: assemble the training data, create the network object, train the network, simulate the network response to training inputs and test inputs. The flow chart of direct modelling with MLP is shown in Fig.6.2.

The input vector covers the entire range of the input variable, and contains twenty elements as [1 2 4 5 7 8 9 11 13 14 16 17 19 20 3 6 10 12 15 18]. The first fourteen elements of the input vector form the training input vector, and the last six elements of the input vector form the test input vector. Correspondingly, the target vector with twenty elements

is [4.35 9.39 17.53 21.59 30.14 33.79 38.41 46.69 55.24 59.02 66.74 71.92 80.33 84.68 13.46 25.94 42.48 51.73 63.37 75.43]. The target vector is split into two vectors: the training target vector with the first fourteen elements of the target vector, and the test target vector with the rest of the target vector. Therefore, a training set of inputs and targets and a test set of inputs and targets are created.

Creating the network object is the first step in training a feed-forward back-propagation network. The function used to create the required feed-forward network needs four inputs and returns the network object. The first input is an R by 2 matrix of minimum and maximum values for each of the R elements of the training input vector. The second input is an array containing the sizes of each layer. The third input is a cell array containing the names of the transfer functions to be used in each layer. The final input contains the name of the training function to be used. The function *minmax* is used to determine the range of the inputs to be used in creating the network. For our application, the created network has an input ranging from 1 to 20, followed by a layer of one *logsig* neuron, followed by a layer of several *logsig* neurons, followed by a layer with one *purelin* neuron. In back-propagation it is important to be able to calculate the derivatives of any transfer functions used. The *trainlm* network training

function is used. Like the quasi-Newton methods, the Levenberg-Marquardt algorithm has the ability of approaching second-order training speed without having to compute the Hessian matrix. The performance function has the form of a sum of squares, which is typical in training feed-forward networks. The parameter μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm.

Before training a feed-forward network, the weights and biases must be initialised. The layer's weights and biases are initialised according to the Nguyen-Widrow initialization algorithm. This algorithm chooses values in order to distribute the active region of each neuron in the layer approximately evenly across the layer's input space. It makes the training work faster since each area of the input space has neurons.

The next task is to prepare the training parameters. The training parameters for `trainlm` are *epochs*, *show*, *goal*, *time*, *min_grad*, *max_fail*, *mu*, *mu_dec*, *mu_inc*, *mu_max*, *mem_reduc*. The parameter *mu* is the initial value for the scalar μ . μ is decreased after each successful step

(reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm. This value is multiplied by *mu_dec* whenever the performance function is reduced by a step. It is multiplied by *mu_inc* whenever a step would increase the performance function. If *mu* becomes larger than *mu_max*, the algorithm is stopped. The parameters *mu*, *mu_dec*, and *mu_inc* are set to 0.2, 0.7, 10 respectively. The parameters *epochs* and *goal* are set to 10000 and 0.01 respectively. Default values for the rest of the training parameters: *max_fail* is 5, *mem_reduc* is 1, *min_grad* is 1e-10, *mu_max* is 1e+10, *show* is 25, and *time* is Inf. The parameter *mem_reduc* is used to control the amount of memory used by the algorithm. If there is enough memory available, then it is better to set *mem_reduc* to 1 and no memory reduction is achieved. If there is a large training set, and the memory is running out, then *mem_reduc* should be set to 2, and try again. If there is still running out of memory, continue to increase *mem_reduc*.

Once the network weights/biases are initialized and the training parameters are set, the network is ready for training. The training process requires a set of examples of proper network behaviour – network training inputs and training target outputs. During training the weights and biases

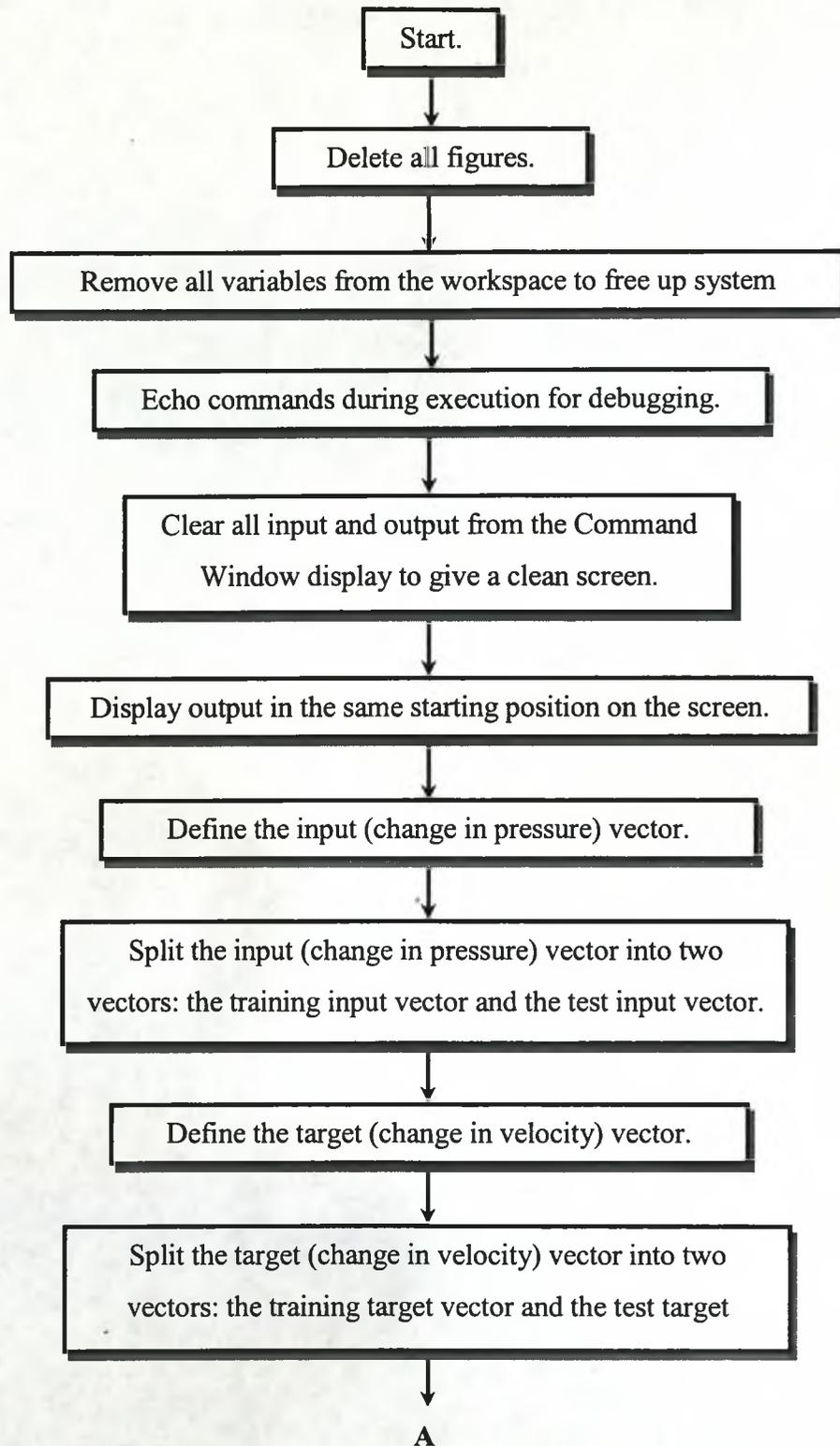
of the network are iteratively adjusted to minimise the network performance function *net.performFcn*. The default performance function for feed-forward networks is mean square error *mse* - the average squared error between the network outputs and the target outputs. The training status is displayed for every *show* iteration of the algorithm. The training stops if the number of iterations exceeds *epochs*, or if the performance function drops below *goal*, or if the training time is longer than *time* seconds.

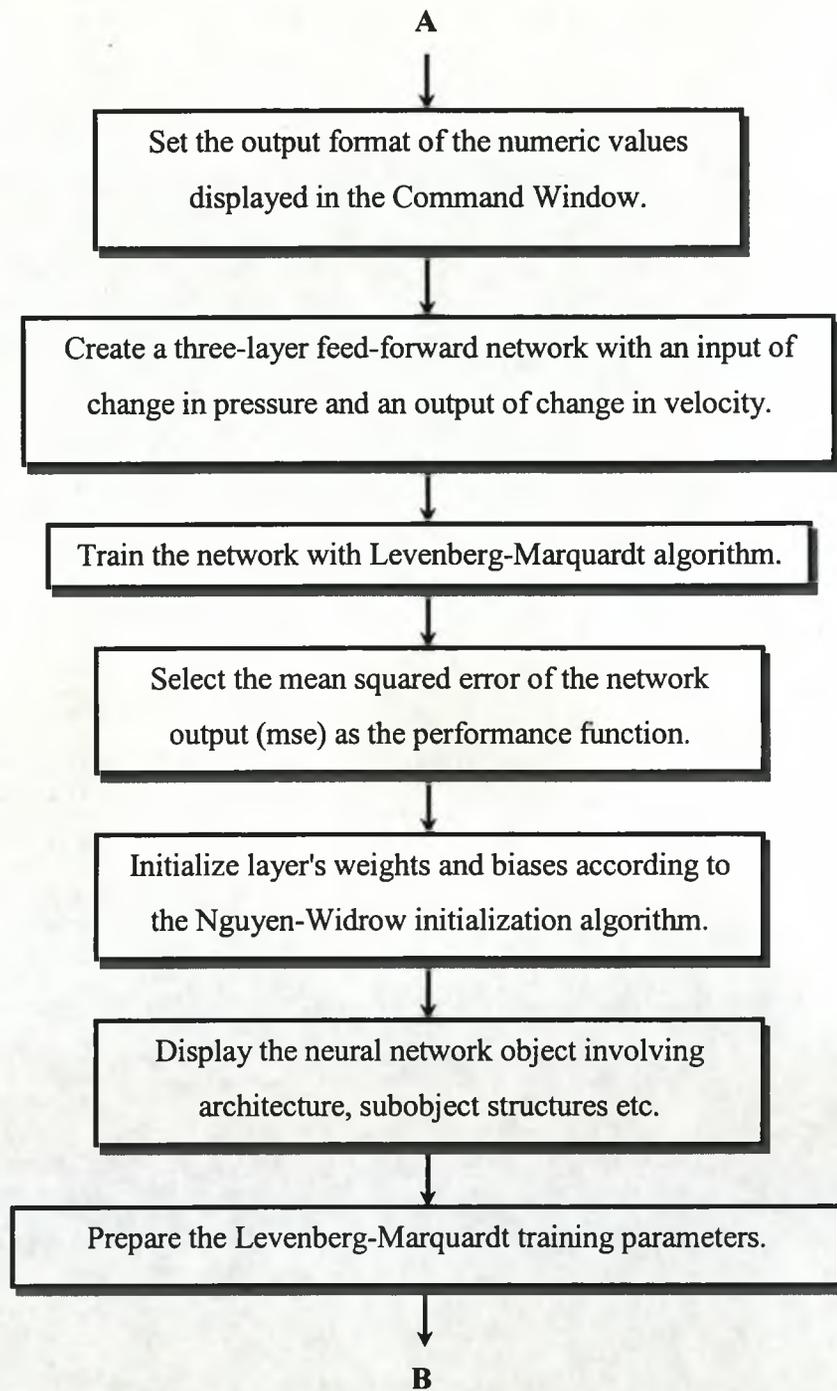
After training the network can be simulated to see if it has learned to respond correctly. The simulation function takes the network input and the network object, and returns the network output. First the trained network model is simulated to obtain its response to the inputs in the training set. Then the inputs in the test set are presented, and the corresponding outputs are calculated with the trained network model.

The direct models with 1-5-1 structure, 1-6-1 structure and 1-7-1 structure have been developed. The results of these models are presented in 6.2.2, 6.2.3 and 6.2.4 respectively.

6.2.2 Results of the direct model with 1-5-1 structure

The direct model with 1-5-1 structure has been developed. Fig.6.3 shows the progress of training the parameters. The training performance of this model shown in Fig.6.4 indicates that the performance goal is met at 346 epochs. Tab.6.1 includes all the weights and biases of the network after training. Tab.6.2 shows the computed results of this direct model for training data. Δc_{es} represents the estimate of Δc , and the absolute error between them is e_{ab} . It has been found that maximum of e_{ab} in terms of absolute value is 0.245 m/s for the training data. Fig.6.5 shows the relative error (%FS) of the model for the training data. Relative error (%FS) is the percentage representation of e_{ab} divided by the full scale of Δc . Here the maximum of relative error (%FS) is found to be 0.289 %FS for the training data. Tab.6.3 shows the computed results of the model for the test data. It can be found that the maximum of e_{ab} is 1.002 m/s. Fig.6.6 shows the relative error (%FS) of the model for the test data, and the maximum is 1.184 %FS.





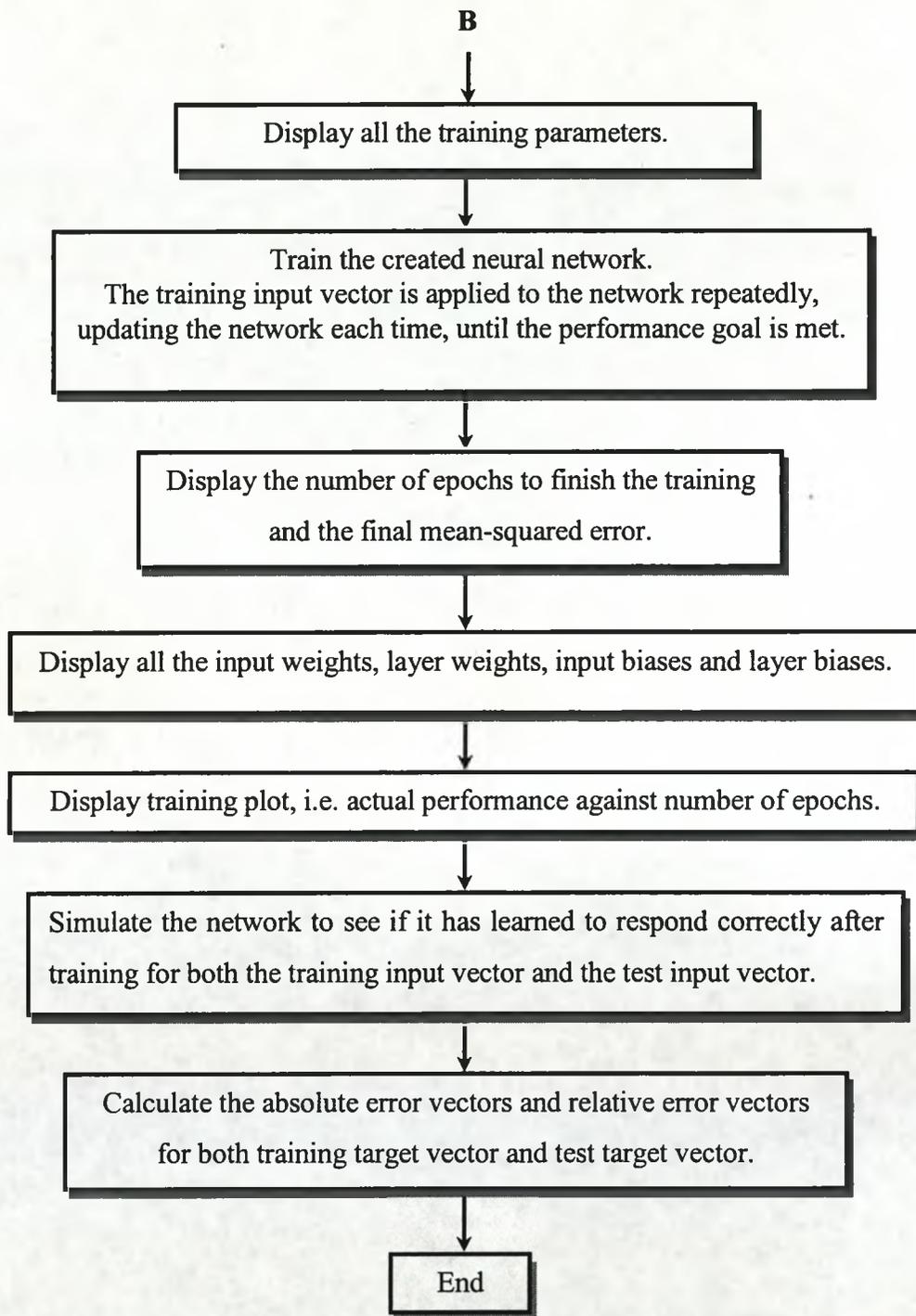


Fig.6.2 The Flow chart of direct modelling with MLP

Epoch 0/10000, MSE 2557.84/0.01, Gradient 2638.41/1e-010,
Epoch 25/10000, MSE 0.0372422/0.01, Gradient 88.392/1e-010,
Epoch 50/10000, MSE 0.0130995/0.01, Gradient 67.3048/1e-010,
Epoch 75/10000, MSE 0.0117962/0.01, Gradient 9.55408/1e-010,
Epoch 100/10000, MSE 0.0115326/0.01, Gradient 0.763015/1e-010,
Epoch 125/10000, MSE 0.011464/0.01, Gradient 4.15869/1e-010,
Epoch 150/10000, MSE 0.0114439/0.01, Gradient 5.41684/1e-010,
Epoch 175/10000, MSE 0.0114223/0.01, Gradient 0.339251/1e-010,
Epoch 200/10000, MSE 0.0113981/0.01, Gradient 2.05481/1e-010,
Epoch 225/10000, MSE 0.0113656/0.01, Gradient 0.0233301/1e-010,
Epoch 250/10000, MSE 0.0113211/0.01, Gradient 0.0290305/1e-010,
Epoch 275/10000, MSE 0.0112452/0.01, Gradient 0.0447535/1e-010,
Epoch 300/10000, MSE 0.0111002/0.01, Gradient 5.10928/1e-010,
Epoch 325/10000, MSE 0.0105573/0.01, Gradient 33.8699/1e-010,
Epoch 346/10000, MSE 0.00993843/0.01, Gradient 0.856465/1e-010,
Performance goal met.

Fig.6.3 Training parameters in progress for the direct model
with 1-5-1 structure

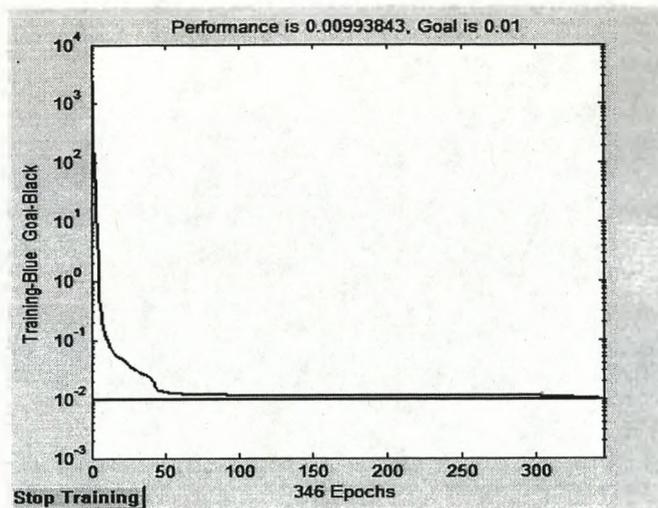


Fig.6.4 Training performance of the direct model with 1-5-1 structure

Tab.6.1 The weights and biases of the direct model with 1-5-1 structure

weight		bias	
$w_{1,1}^1$	0.158734	b_1^1	-1.791086
$w_{1,1}^2$	32.326660	b_1^2	-26.718741
$w_{2,1}^2$	55.701649	b_2^2	-39.187323
$w_{3,1}^2$	-4.690077	b_3^2	2.092633
$w_{4,1}^2$	-55.835277	b_4^2	15.546586
$w_{5,1}^2$	25.975304	b_5^2	-2.554822
$w_{1,1}^3$	18.070949	b_1^3	36.893451
$w_{1,2}^3$	5.261220		
$w_{1,3}^3$	-94.108551		
$w_{1,4}^3$	-2.347519		
$w_{1,5}^3$	52.286925		

Tab.6.2 Computed results of the direct model with 1-5-1 structure for training data (Unit: m/s)

Δc	4.35	9.39	17.53	21.59	30.14	33.79	38.41
Δc_{es}	4.349	9.393	17.523	21.600	30.062	34.029	38.165
e_{ab}	0.001	-0.003	0.007	-0.010	0.078	-0.239	0.245
Δc	46.69	55.24	59.02	66.74	71.92	80.33	84.68
Δc_{es}	46.806	55.189	59.032	66.741	71.918	80.331	84.679
e_{ab}	-0.116	0.051	-0.012	0.001	0.002	-0.001	0.001

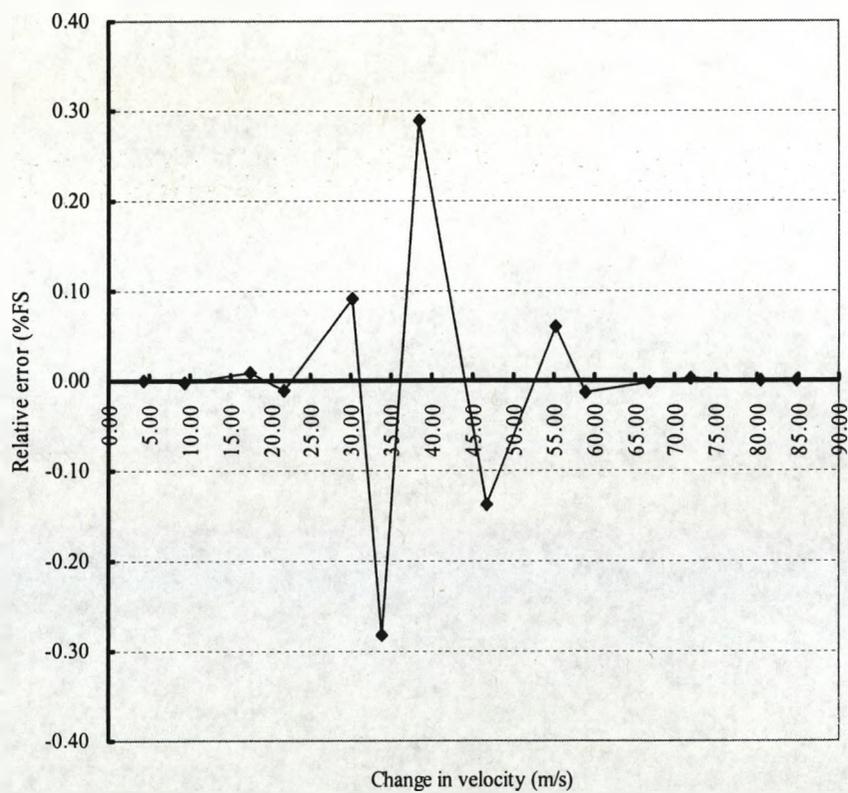


Fig.6.5 Relative error (%FS) of the direct model with 1-5-1 structure for the training data

Tab.6.3 Computed results of the direct model with 1-5-1 structure for the test data (Unit: m/s)

Δc	13.46	25.94	42.48	51.73	63.37	75.43
Δc_{es}	13.667	26.013	42.457	51.088	62.680	76.432
e_{ab}	-0.207	-0.073	0.023	0.642	0.690	-1.002

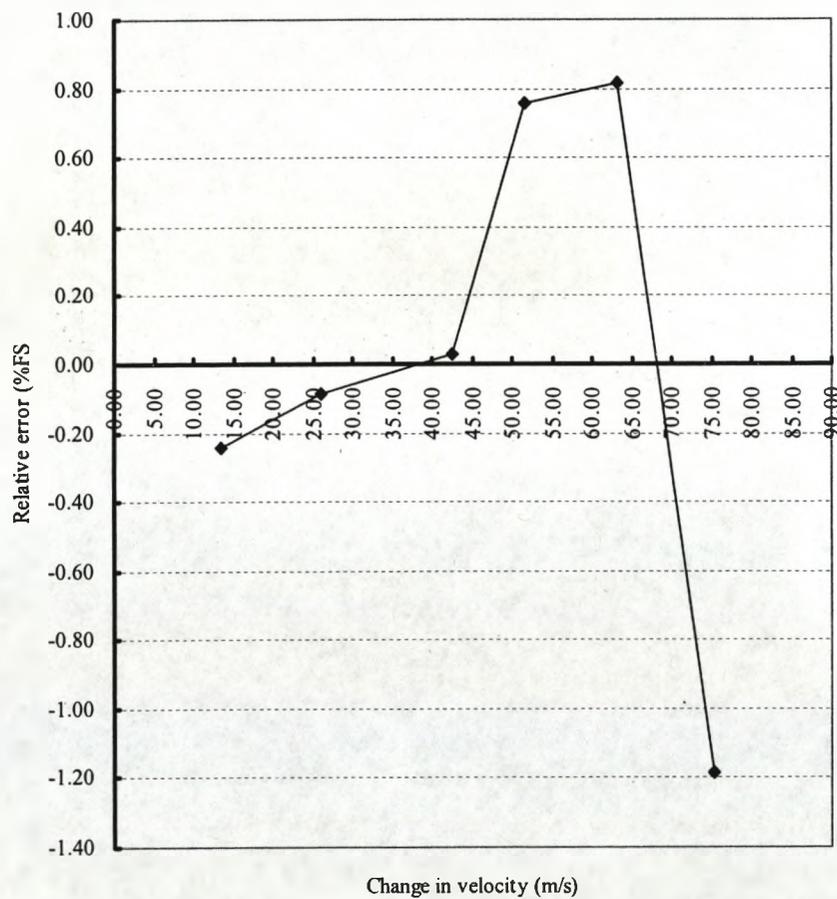


Fig.6.6 Relative error (%FS) of the direct model with 1-5-1 structure for the test data

6.2.3 Results of the direct model with 1-6-1 structure

The direct model with 1-6-1 structure has been built up. Fig.6.7 shows the progress of training the parameters. The training performance of this model shown in Fig.6.8 indicates that the performance goal is met at 543 epochs. Tab.6.4 includes all the weights and biases of the network after training. Tab.6.5 shows the computed results of this direct model for training data. It has been found that maximum of e_{ab} is 0.250 m/s for the training data. Fig.6.9 shows the relative error (%FS) of the model for the training data. Here the maximum of relative error (%FS) is found to be 0.300 %FS for the training data. Tab.6.6 shows the computed results of the model for the test data. It can be found that the maximum of e_{ab} is 1.339 m/s. Fig.6.10 shows the relative error (%FS) of the model for the test data, and the maximum is 1.580 %FS.

Epoch 0/10000, MSE 2521.27/0.01, Gradient 1404.38/1e-010,
Epoch 25/10000, MSE 0.0382448/0.01, Gradient 130.505/1e-010,
Epoch 50/10000, MSE 0.0367092/0.01, Gradient 12.8023/1e-010,
Epoch 75/10000, MSE 0.0366347/0.01, Gradient 7.26666/1e-010,
Epoch 100/10000, MSE 0.0365235/0.01, Gradient 4.72629/1e-010,
Epoch 125/10000, MSE 0.0363383/0.01, Gradient 3.15306/1e-010,
Epoch 150/10000, MSE 0.0360766/0.01, Gradient 1.90856/1e-010,
Epoch 175/10000, MSE 0.0356821/0.01, Gradient 0.90528/1e-010,
Epoch 200/10000, MSE 0.0352344/0.01, Gradient 0.249361/1e-010,
Epoch 225/10000, MSE 0.0348245/0.01, Gradient 3.97921/1e-010,
Epoch 250/10000, MSE 0.0345172/0.01, Gradient 0.473806/1e-010,
Epoch 275/10000, MSE 0.0343094/0.01, Gradient 0.0622357/1e-010,
Epoch 300/10000, MSE 0.03417/0.01, Gradient 1.45813/1e-010,
Epoch 325/10000, MSE 0.0340743/0.01, Gradient 0.404629/1e-010,
Epoch 350/10000, MSE 0.0340064/0.01, Gradient 2.18055/1e-010,
Epoch 375/10000, MSE 0.0339503/0.01, Gradient 1.00906/1e-010,
Epoch 400/10000, MSE 0.0338624/0.01, Gradient 1.08881/1e-010,
Epoch 425/10000, MSE 0.0337478/0.01, Gradient 0.88734/1e-010,
Epoch 450/10000, MSE 0.0333875/0.01, Gradient 2.65441/1e-010,
Epoch 475/10000, MSE 0.0325292/0.01, Gradient 5.73942/1e-010,
Epoch 500/10000, MSE 0.0288684/0.01, Gradient 3.53001/1e-010,
Epoch 525/10000, MSE 0.0170407/0.01, Gradient 37.6339/1e-010,
Epoch 543/10000, MSE 0.0098187/0.01, Gradient 34.8195/1e-010,
Performance goal met.

Fig.6.7 Training parameters in progress for the direct model
with 1-6-1 structure

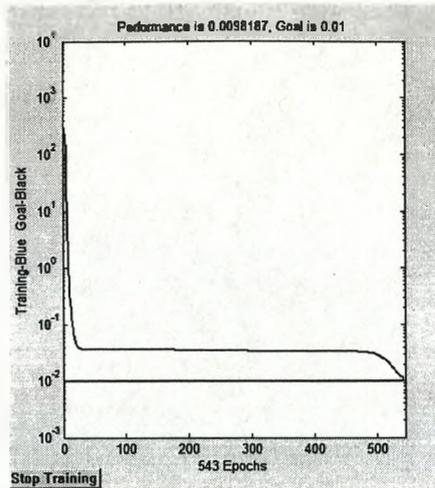


Fig.6.8 Training performance of the direct model with 1-6-1 structure

Tab.6.4 The weights and biases of the direct model with 1-6-1 structure

weight		bias	
$w_{1,1}^1$	0.180738	b_1^1	-2.390946
$w_{1,1}^2$	-74.331928	b_1^2	14.186691
$w_{2,1}^2$	-48.667492	b_2^2	39.904038
$w_{3,1}^2$	36.976936	b_3^2	-24.637058
$w_{4,1}^2$	3.766732	b_4^2	-0.831438
$w_{5,1}^2$	-43.353478	b_5^2	3.411045
$w_{6,1}^2$	33.914261	b_6^2	8.731531
$w_{1,1}^3$	-0.024650	b_1^3	-1.820209
$w_{1,2}^3$	-0.340450		
$w_{1,3}^3$	0.080967		
$w_{1,4}^3$	1.187302		
$w_{1,5}^3$	-0.245191		
$w_{1,6}^3$	0.039369		

Tab.6.5 Computed results of the direct model with 1-6-1 structure for training data (Unit: m/s)

Δc	4.35	9.39	17.53	21.59	30.14	33.79	38.41
Δc_{es}	4.361	9.400	17.533	21.600	30.080	34.016	38.160
e_{ab}	-0.011	-0.010	-0.003	-0.009	0.060	-0.226	0.250
Δc	46.69	55.24	59.02	66.74	71.92	80.33	84.68
Δc_{es}	46.815	55.185	59.020	66.736	71.895	80.317	84.660
e_{ab}	-0.125	0.055	0.000	0.004	0.025	0.013	0.020

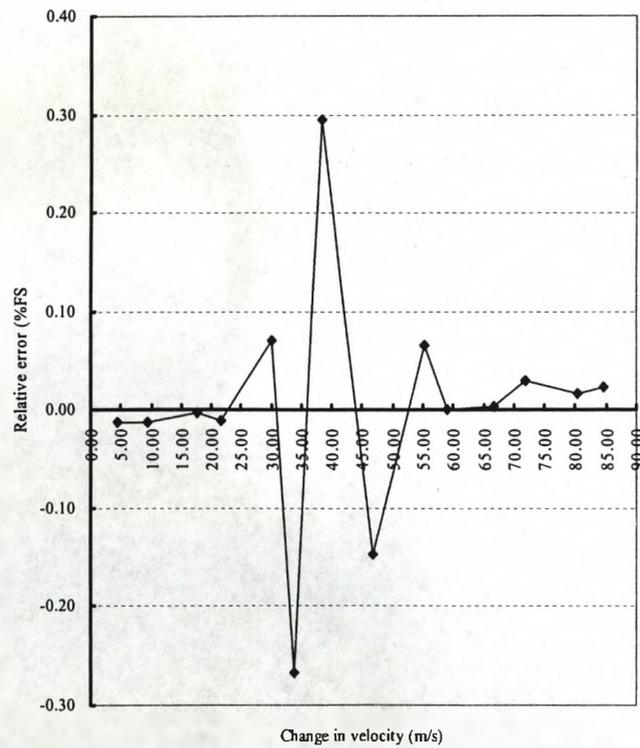


Fig.6.9 Relative error (%FS) of the direct model with 1-6-1 structure for the training data

Tab.6.6 Computed results of the direct model with 1-6-1 structure for the test data (Unit: m/s)

Δc	13.46	25.94	42.48	51.73	63.37	75.43
Δc_{es}	13.737	26.111	42.463	51.094	62.688	76.769
e_{ab}	-0.277	-0.171	0.017	0.636	0.682	-1.339

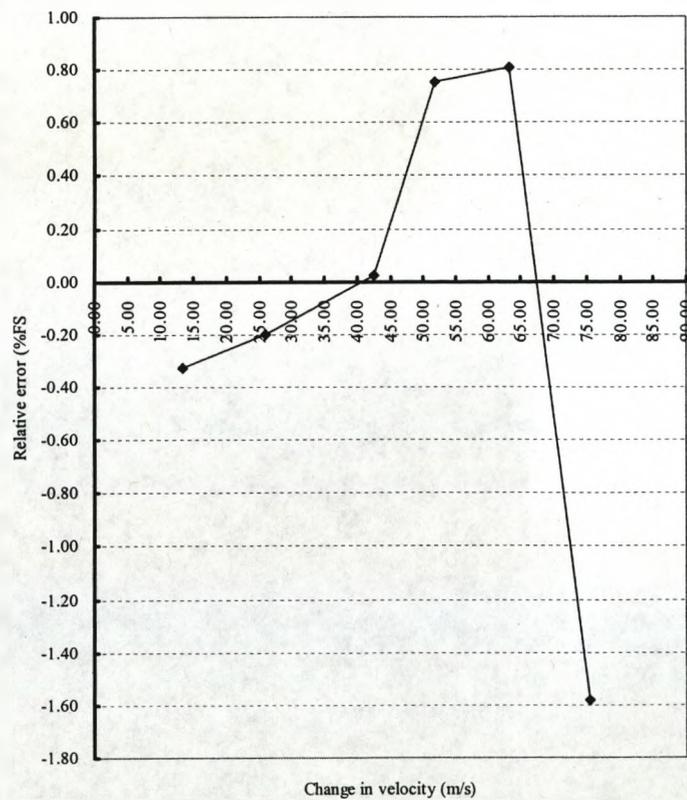


Fig.6.10 Relative error (%FS) of the direct model with 1-6-1 structure for the test data

6.2.4 Results of the direct model with 1-7-1 structure

The direct model with 1-7-1 structure has been investigated. Fig.6.11 shows the progress of training the parameters. The training performance of this model shown in Fig.6.12 indicates that the performance goal is met at 480 epochs. Tab.6.7 includes all the weights and biases of the network after training. Tab.6.8 shows the computed results of this direct model for training data. It has been found that maximum of e_{ab} is 0.275 m/s for the training data. Fig.6.13 shows the relative error (%FS) of the model for the training data. Here the maximum of relative error (%FS) is found to be 0.325 %FS for the training data. Tab.6.9 shows the computed results of the model for the test data. It can be found that the maximum of e_{ab} is 3.021 m/s. Fig.6.14 shows the relative error (%FS) of the model for the test data, and the maximum is 3.568 %FS.

Epoch 0/10000, MSE 2614.05/0.01, Gradient 1310.94/1e-010,
Epoch 25/10000, MSE 0.0308831/0.01, Gradient 311.241/1e-010,
Epoch 50/10000, MSE 0.0124457/0.01, Gradient 61.6244/1e-010,
Epoch 75/10000, MSE 0.0117149/0.01, Gradient 12.505/1e-010,
Epoch 100/10000, MSE 0.0115475/0.01, Gradient 3.90154/1e-010,
Epoch 125/10000, MSE 0.0114818/0.01, Gradient 0.668878/1e-010,
Epoch 150/10000, MSE 0.0114283/0.01, Gradient 34.0716/1e-010,
Epoch 175/10000, MSE 0.0113511/0.01, Gradient 8.52512/1e-010,
Epoch 200/10000, MSE 0.0113123/0.01, Gradient 2.58929/1e-010,
Epoch 225/10000, MSE 0.0112772/0.01, Gradient 0.792305/1e-010,
Epoch 250/10000, MSE 0.0112377/0.01, Gradient 0.244979/1e-010,
Epoch 275/10000, MSE 0.0111921/0.01, Gradient 11.3883/1e-010,
Epoch 300/10000, MSE 0.0111294/0.01, Gradient 0.226981/1e-010,
Epoch 325/10000, MSE 0.0110484/0.01, Gradient 0.743574/1e-010,
Epoch 350/10000, MSE 0.0109652/0.01, Gradient 1.0357/1e-010,
Epoch 375/10000, MSE 0.0108778/0.01, Gradient 0.732485/1e-010,
Epoch 400/10000, MSE 0.0107554/0.01, Gradient 0.272747/1e-010,
Epoch 425/10000, MSE 0.0105909/0.01, Gradient 0.0721624/1e-010,
Epoch 450/10000, MSE 0.0103638/0.01, Gradient 0.0407599/1e-010,
Epoch 475/10000, MSE 0.0100988/0.01, Gradient 1.12678/1e-010,
Epoch 480/10000, MSE 0.00999812/0.01, Gradient 0.352731/1e-010,
Performance goal met.

Fig.6.11 Training parameters in progress for the direct model
with 1-7-1 structure

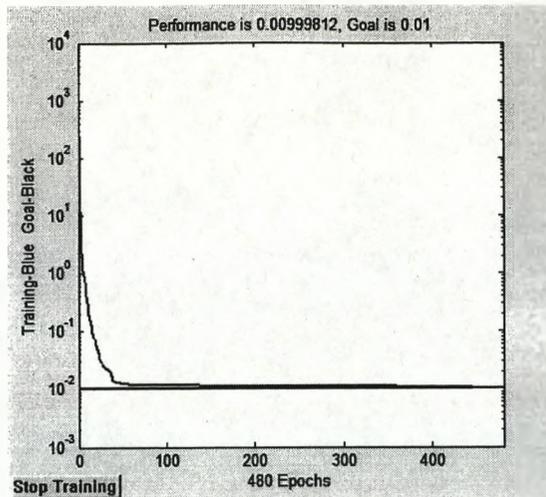


Fig.6.12 Training performance of the direct model with 1-7-1 structure

Tab.6.7 The weights and biases of the direct model with 1-7-1 structure

weight		bias	
$w_{1,1}^1$	0.277697	b_1^1	-2.703531
$w_{2,1}^2$	40.032027	b_1^2	-38.604620
$w_{2,1}^2$	37.761369	b_2^2	-36.508715
$w_{3,1}^2$	-60.705550	b_3^2	47.648306
$w_{4,1}^2$	29.318886	b_4^2	-23.714106
$w_{5,1}^2$	-3.576355	b_5^2	1.988984
$w_{6,1}^2$	-11.913638	b_6^2	0.921733
$w_{7,1}^2$	57.793011	b_7^2	-2.177788
$w_{1,1}^3$	4.270336	b_1^3	21.547230
$w_{1,2}^3$	25.670692		
$w_{1,3}^3$	14.619917		
$w_{1,4}^3$	25.482557		
$w_{1,5}^3$	-66.800695		
$w_{1,6}^3$	-29.358053		
$w_{1,7}^3$	42.148111		

Tab.6.8 Results of the direct model with 1-7-1 structure for training data (Unit: m/s)

Δc	4.35	9.39	17.53	21.59	30.14	33.79	38.41
Δc_{es}	4.355	9.400	17.494	21.665	29.962	34.065	38.254
e_{ab}	-0.005	-0.010	0.036	-0.075	0.178	-0.275	0.156
Δc	46.69	55.24	59.02	66.74	71.92	80.33	84.68
Δc_{es}	46.720	55.236	59.020	66.742	71.915	80.333	84.676
e_{ab}	-0.030	0.004	0.000	-0.002	0.005	-0.003	0.004

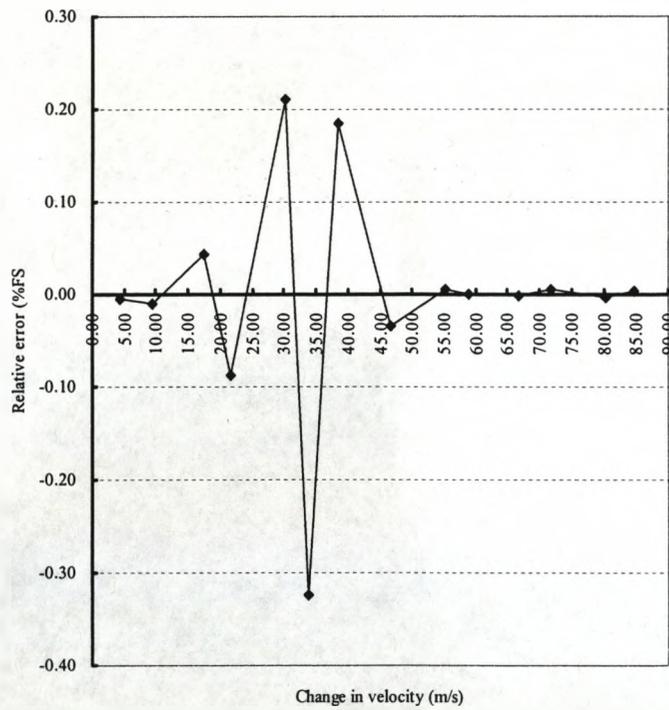


Fig.6.13 Relative error (%FS) of the direct model with 1-7-1 structure for the training data

Tab.6.9 Computed results of the direct model with 1-7-1 structure for the test data (Unit: m/s)

Δc	13.46	25.94	42.48	51.73	63.37	75.43
Δc_{es}	13.474	25.850	42.513	50.818	60.349	76.084
e_{ab}	-0.014	0.090	-0.033	0.912	3.021	-0.654

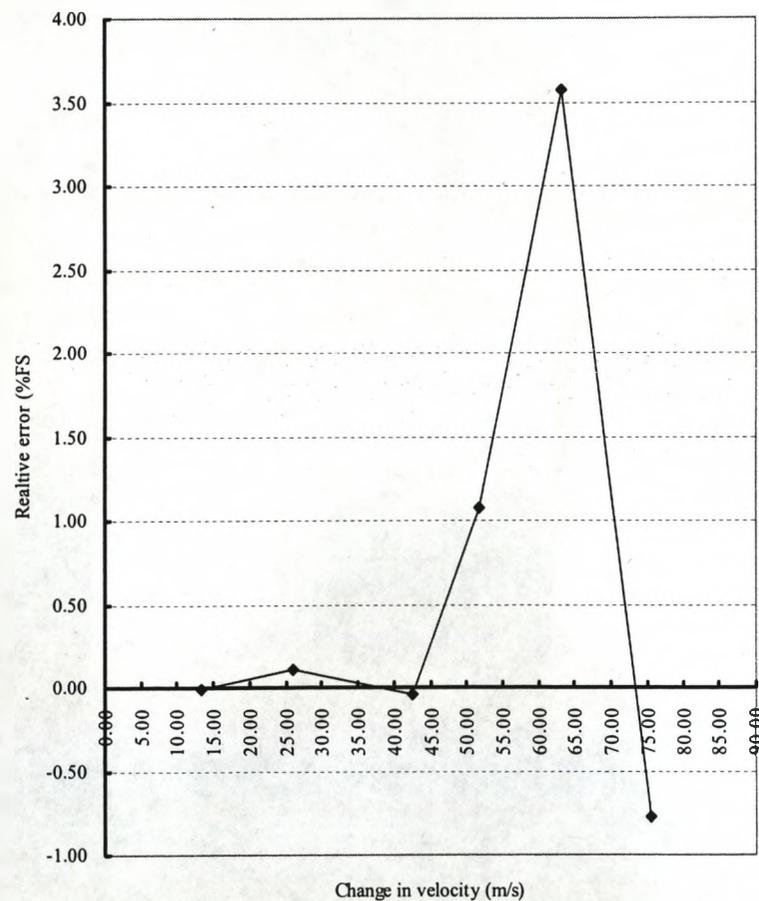


Fig.6.14 Relative error (%FS) of the direct model with 1-7-1 structure for the test data

6.3 Implementation of Inverse Modelling with MLP for Non-Intrusive Pressure Monitoring System

For the process of inverse modelling the input is the change in velocity, and the output is the change in pressure. The flow chart of inverse modelling with MLP is similar to that of direct modelling shown in Fig.6.2. The inverse models with 1-5-1 structure, 1-6-1 structure and 1-7-1 structure have been developed.

6.3.1 Results of the inverse model with 1-5-1 structure

Fig.6.15 shows the progress of training the parameters for the inverse model with 1-5-1 structure. The training performance of this model shown in Fig.6.16 indicates that the performance goal is met at 16 epochs. Tab.6.10 includes all the weights and biases of the network after training. Tab.6.11 shows the computed results of this direct model for training data. It has been found that maximum of e_{ab} is 0.053 MPa for the training data. Fig.6.17 shows the relative error (%FS) of the model for the training data. Here the maximum of relative error (%FS) is found to be 0.263 %FS for the training data. Tab.6.12 shows the computed results of the model for the test data. It can be found that the maximum of e_{ab} is 0.210 MPa.

Fig.6.18 shows the relative error (%FS) of the model for the test data, and the maximum is 1.050 %FS.

Epoch 0/15000, MSE 136.531/0.001, Gradient 2502.75/1e-010,
Epoch 1/15000, MSE 4.4104/0.001, Gradient 1262.16/1e-010,
Epoch 2/15000, MSE 0.72034/0.001, Gradient 1140.12/1e-010,
Epoch 3/15000, MSE 0.163134/0.001, Gradient 396.041/1e-010,
Epoch 4/15000, MSE 0.0187936/0.001, Gradient 90.3041/1e-010,
Epoch 5/15000, MSE 0.0128815/0.001, Gradient 27.657/1e-010,
Epoch 6/15000, MSE 0.0104405/0.001, Gradient 9.74583/1e-010,
Epoch 7/15000, MSE 0.00896942/0.001, Gradient 9.37776/1e-010,
Epoch 8/15000, MSE 0.00756937/0.001, Gradient 9.30889/1e-010,
Epoch 9/15000, MSE 0.00619825/0.001, Gradient 15.1661/1e-010,
Epoch 10/15000, MSE 0.00489473/0.001, Gradient 36.8931/1e-010,
Epoch 11/15000, MSE 0.0048362/0.001, Gradient 107.725/1e-010,
Epoch 12/15000, MSE 0.00216117/0.001, Gradient 33.2207/1e-010,
Epoch 13/15000, MSE 0.00172433/0.001, Gradient 33.8842/1e-010,
Epoch 14/15000, MSE 0.00136201/0.001, Gradient 39.4651/1e-010,
Epoch 15/15000, MSE 0.00104703/0.001, Gradient 20.3513/1e-010,
Epoch 16/15000, MSE 0.00091235/0.001, Gradient 5.49087/1e-010,
Performance goal met.

Fig.6.15 Training parameters in progress for the inverse model with 1-5-1 structure

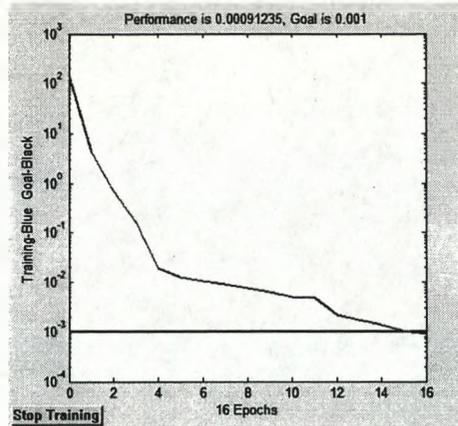


Fig.6.16 Training performance of the inverse model with 1-5-1 structure

Tab.6.10 The weights and biases of the inverse model with 1-5-1 structure

weight		bias	
$w_{1,1}^1$	0.046228	b_1^1	-1.629570
$w_{1,1}^2$	27.190229	b_1^2	-25.258665
$w_{2,1}^2$	23.500228	b_2^2	-17.849879
$w_{3,1}^2$	-26.208880	b_3^2	15.829413
$w_{4,1}^2$	-24.545468	b_4^2	10.884921
$w_{5,1}^2$	20.893992	b_5^2	-5.331058
$w_{1,1}^3$	9.535677	b_1^3	6.271448
$w_{1,2}^3$	4.939193		
$w_{1,3}^3$	-2.924478		
$w_{1,4}^3$	-3.528396		
$w_{1,5}^3$	5.540470		

Tab.6.11 Results of the inverse model with 1-5-1 structure for training data (Unit: MPa)

Δp	1.00	2.00	4.00	5.00	7.00	8.00	9.00
Δp_{es}	1.021	1.959	4.053	4.960	7.007	7.999	8.997
e_{ab}	-0.021	0.041	-0.053	0.040	-0.007	0.001	0.003
Δp	11.00	13.00	14.00	16.00	17.00	19.00	20.00
Δp_{es}	11.001	12.991	14.015	15.969	17.040	18.948	20.025
e_{ab}	-0.001	0.009	-0.015	0.031	-0.040	0.052	-0.025

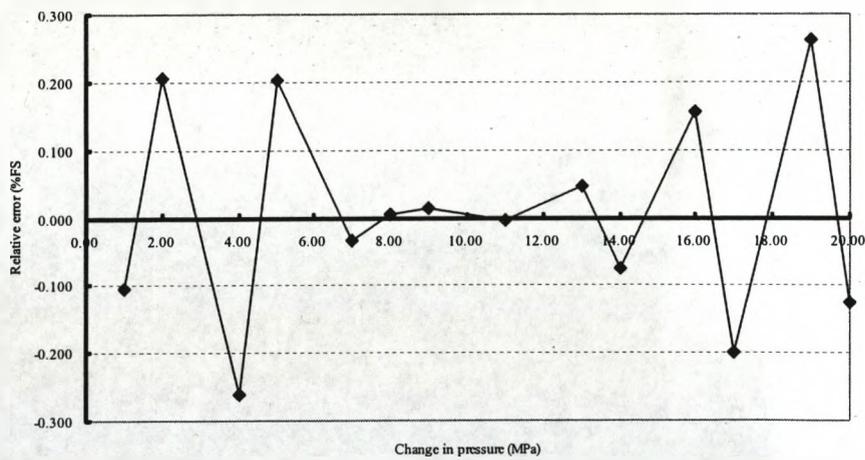


Fig.6.17 Relative error (%FS) of the inverse model with 1-5-1 structure for the training data

Tab.6.12 Computed results of the inverse model with 1-5-1 structure for test data (Unit: MPa)

Δp	3.00	6.00	10.00	12.00	15.00	18.00
Δp_{cs}	2.991	5.892	9.913	12.159	15.185	17.790
e_{ab}	0.009	0.108	0.087	-0.159	-0.185	0.210

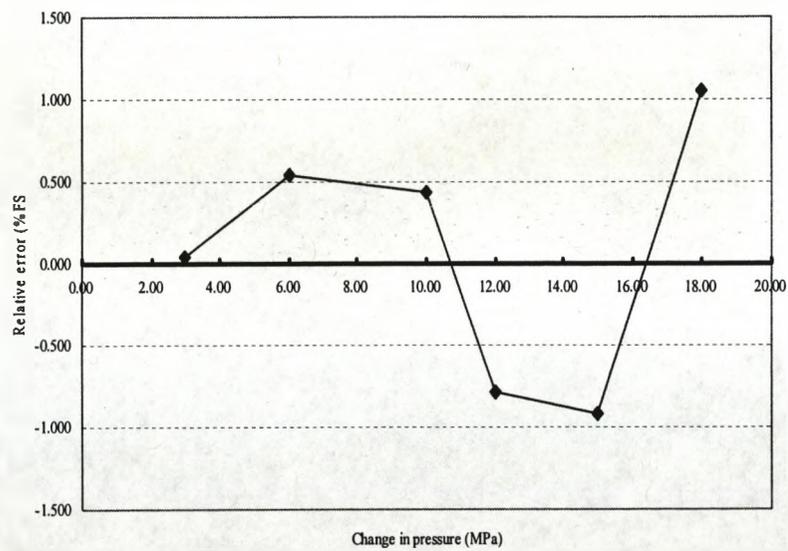


Fig.6.18 Relative error (%FS) of the inverse model with 1-5-1 structure for the test data

6.3.2 Results of the inverse model with 1-6-1 structure

The inverse model with 1-6-1 structure has been built up. Fig.6.19 shows the progress of training the parameters for the inverse model with 1-6-1 structure. The training performance of this model shown in Fig.6.20 indicates that the performance goal is met at 17 epochs. Tab.6.13 includes all the weights and biases of the network after training. Tab.6.14 shows the computed results of this direct model for training data. It has been found that maximum of e_{ab} is 0.080 MPa for the training data. Fig.6.21 shows the relative error (%FS) of the model for the training data. Here the maximum of relative error (%FS) is found to be 0.401 %FS for the training data. Tab.6.15 shows the computed results of the model for the test data. It can be found that the maximum of e_{ab} is 0.476 MPa. Fig.6.22 shows the relative error (%FS) of the model for the test data, and the maximum is 2.377 %FS.

Epoch 0/15000, MSE 128.001/0.001, Gradient 741.926/1e-010,
Epoch 1/15000, MSE 9.35907/0.001, Gradient 799.382/1e-010,
Epoch 2/15000, MSE 6.68133/0.001, Gradient 547.594/1e-010,
Epoch 3/15000, MSE 4.15664/0.001, Gradient 72.2077/1e-010,
Epoch 4/15000, MSE 0.91718/0.001, Gradient 689.245/1e-010,
Epoch 5/15000, MSE 0.06843/0.001, Gradient 189.121/1e-010,
Epoch 6/15000, MSE 0.0273029/0.001, Gradient 252.466/1e-010,
Epoch 7/15000, MSE 0.00707669/0.001, Gradient 65.2525/1e-010,
Epoch 8/15000, MSE 0.00641089/0.001, Gradient 42.5563/1e-010,
Epoch 9/15000, MSE 0.00448833/0.001, Gradient 58.0977/1e-010,
Epoch 10/15000, MSE 0.00323727/0.001, Gradient 19.9735/1e-010,
Epoch 11/15000, MSE 0.00272372/0.001, Gradient 9.53795/1e-010,
Epoch 12/15000, MSE 0.0017255/0.001, Gradient 11.3896/1e-010,
Epoch 13/15000, MSE 0.00142292/0.001, Gradient 16.0647/1e-010,
Epoch 14/15000, MSE 0.00122666/0.001, Gradient 15.9631/1e-010,
Epoch 15/15000, MSE 0.00109284/0.001, Gradient 14.1129/1e-010,
Epoch 16/15000, MSE 0.00100949/0.001, Gradient 11.8906/1e-010,
Epoch 17/15000, MSE 0.000964603/0.001, Gradient 9.95356/1e-010,
Performance goal met.

Fig.6.19 Training parameters in progress for the inverse model with 1-6-1 structure

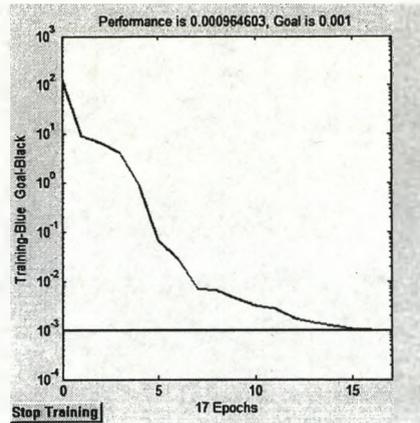


Fig. 6.20 Training performance of the inverse model with 1-6-1 structure

Tab.6.13 The weights and bias of the inverse model with 1-6-1 structure

weight		bias	
$w^1_{1,1}$	0.041359	b^1_1	-3.293526
$w^2_{1,1}$	-42.978477	b^2_1	21.439761
$w^2_{2,1}$	-32.593523	b^2_2	27.892932
$w^2_{3,1}$	36.887214	b^2_3	-13.326908
$w^2_{4,1}$	34.318336	b^2_4	-9.234716
$w^2_{5,1}$	-28.923911	b^2_5	4.535635
$w^2_{6,1}$	31.326405	b^2_6	-1.137321
$w^3_{1,1}$	-3.119306	b^3_1	6.463833
$w^3_{1,2}$	-5.561806		
$w^3_{1,3}$	2.562443		
$w^3_{1,4}$	2.888913		
$w^3_{1,5}$	-4.593667		
$w^3_{1,6}$	13.932302		

Tab.6.14 Computed results of the inverse model with 1-6-1 structure for training data (Unit: MPa)

Δp	1.00	2.00	4.00	5.00	7.00	8.00	9.00
Δp_{cs}	0.998	2.036	3.983	5.002	7.065	7.920	9.032
e_{ab}	0.002	-0.036	0.017	-0.002	-0.065	0.080	-0.032
Δp	11.00	13.00	14.00	16.00	17.00	19.00	20.00
Δp_{cs}	10.990	12.999	13.996	15.997	16.999	18.991	19.995
e_{ab}	0.010	0.001	0.004	0.003	0.001	0.009	0.005

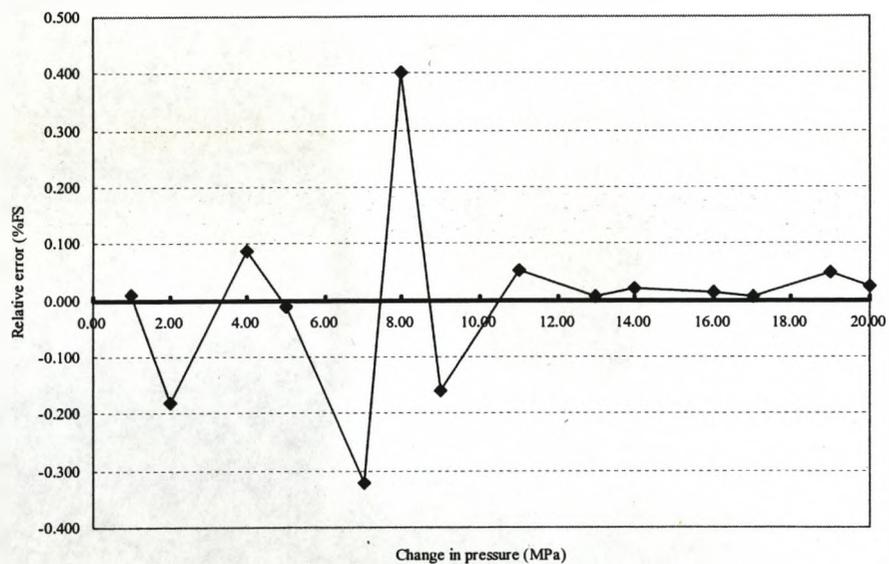


Fig.6.21 Relative error (%FS) of the inverse model with 1-6-1 structure for the training data

Tab.6.15 Results of the inverse model with 1-6-1 structure for test data (Unit: MPa)

Δp	3.00	6.00	10.00	12.00	15.00	18.00
Δp_{es}	2.979	6.071	10.024	12.111	15.094	17.525
e_{ab}	0.021	-0.071	-0.024	-0.111	-0.094	0.476

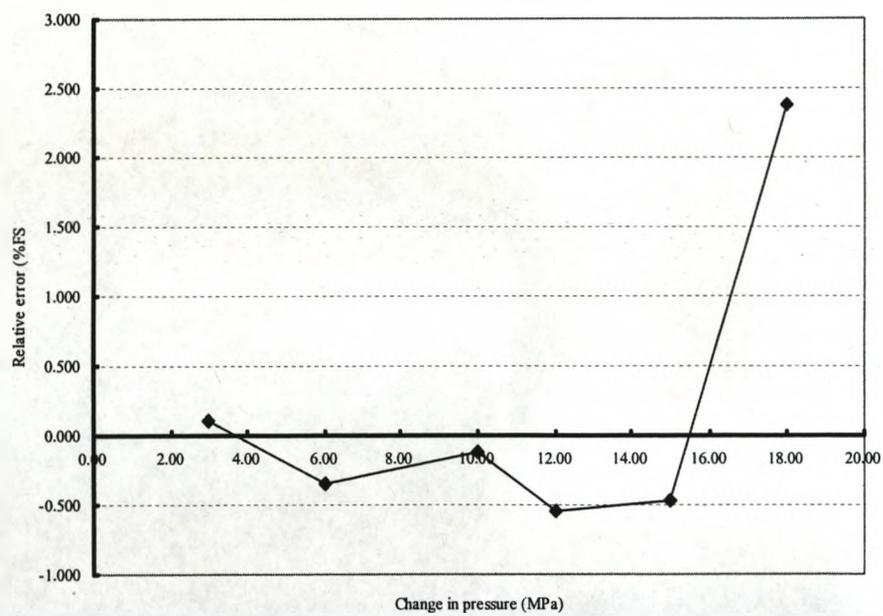


Fig.6.22 Relative error (%FS) of the inverse model with 1-6-1 structure for the test data

6.3.3 Results of the inverse model with 1-7-1 structure

The inverse model with 1-7-1 structure has been studied. Fig.6.23 shows the progress of training the parameters for the inverse model with 1-7-1 structure. The training performance of this model shown in Fig.6.24 indicates that the performance goal is met at 10 epochs. Tab.6.16 includes all the weights and biases of the network after training. Tab.6.17 shows the computed results of this direct model for training data. It has been found that maximum of e_{ab} is 0.032 MPa for the training data. Fig.6.25 shows the relative error (%FS) of the model for the training data. Here the maximum of relative error (%FS) is found to be 0.158 %FS for the training data. Tab.6.18 shows the computed results of the model for the test data. It can be found that the maximum of e_{ab} is 0.683 MPa. Fig.6.26 shows the relative error (%FS) of the model for the test data, and the maximum is 3.413 %FS.

Epoch 0/15000, MSE 149.143/0.001, Gradient 815.956/1e-010,
Epoch 1/15000, MSE 4.26445/0.001, Gradient 1156.29/1e-010,
Epoch 2/15000, MSE 1.65658/0.001, Gradient 801.397/1e-010,
Epoch 3/15000, MSE 0.198967/0.001, Gradient 113.6/1e-010,
Epoch 4/15000, MSE 0.151013/0.001, Gradient 308.279/1e-010,
Epoch 5/15000, MSE 0.0417167/0.001, Gradient 179.813/1e-010,
Epoch 6/15000, MSE 0.0175274/0.001, Gradient 78.1802/1e-010,
Epoch 7/15000, MSE 0.00979112/0.001, Gradient 32.8029/1e-010,
Epoch 8/15000, MSE 0.00295556/0.001, Gradient 19.3935/1e-010,
Epoch 9/15000, MSE 0.00108147/0.001, Gradient 4.8285/1e-010,
Epoch 10/15000, MSE 0.000182009/0.001, Gradient 0.91971/1e-010,
Performance goal met.

Fig.6.23 Training parameters in progress for the inverse model with 1-7-1 structure

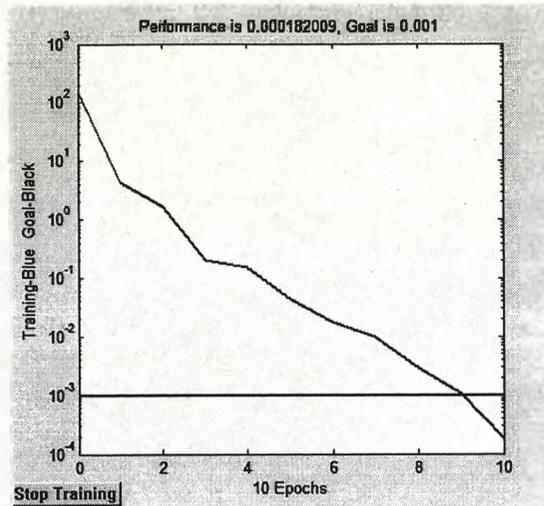


Fig.6.24 Training performance of the inverse model with 1-7-1 structure

Tab.6.16 The weights and biases of the inverse model with 1-7-1 structure

weight		bias	
$w_{1,1}^1$	0.086232	b_1^1	-4.283995
$w_{1,1}^2$	39.606115	b_1^2	-38.370283
$w_{2,1}^2$	37.448922	b_2^2	-32.799357
$w_{3,1}^2$	-37.712155	b_3^2	26.591823
$w_{4,1}^2$	39.067484	b_4^2	-19.167246
$w_{5,1}^2$	-39.506962	b_5^2	11.746101
$w_{6,1}^2$	-37.721908	b_6^2	5.745493
$w_{7,1}^2$	37.756895	b_7^2	-0.573708
$w_{1,1}^3$	5.747562	b_1^3	3.770094
$w_{1,2}^3$	2.250790		
$w_{1,3}^3$	-2.930996		
$w_{1,4}^3$	2.118515		
$w_{1,5}^3$	-2.481334		
$w_{1,6}^3$	-2.752829		
$w_{1,7}^3$	9.951923		

Tab.6.17 Results of the inverse model with 1-7-1 structure for training data (Unit: MPa)

Δp	1.00	2.00	4.00	5.00	7.00	8.00	9.00
Δp_{cs}	1.019	1.968	4.029	4.983	7.002	7.999	8.999
e_{ab}	-0.019	0.032	-0.029	0.017	-0.002	0.001	0.001
Δp	11.00	13.00	14.00	16.00	17.00	19.00	20.00
Δp_{cs}	11.004	12.999	13.998	16.001	16.999	19.000	19.997
e_{ab}	-0.004	0.001	0.002	-0.001	0.001	0.000	0.003

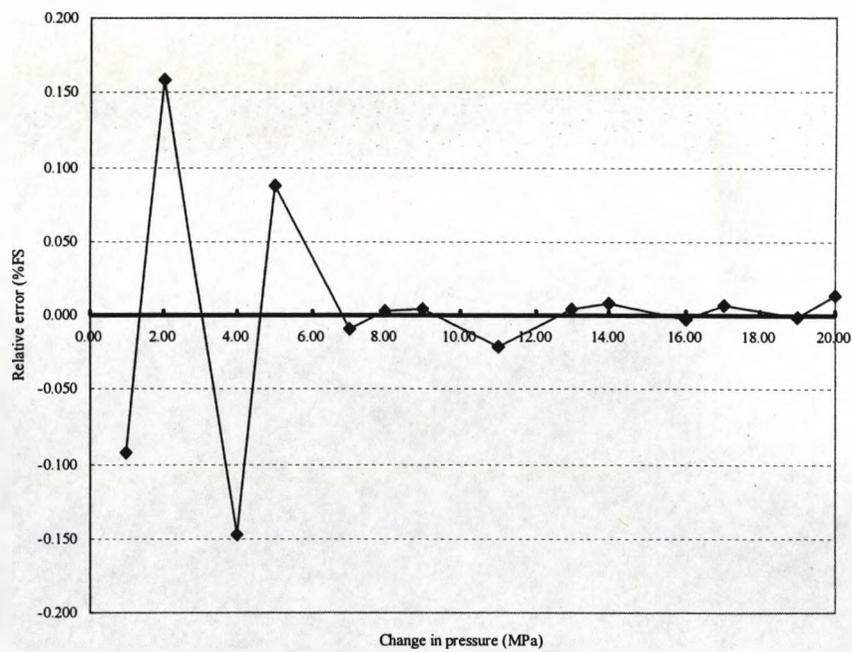


Fig.6.25 Relative error (%FS) of the inverse model with 1-7-1 structure for the training data

Tab.6.18 Results of the inverse model with 1-7-1 structure for test data (Unit: MPa)

Δp	3.00	6.00	10.00	12.00	15.00	18.00
Δp_{es}	2.958	5.859	10.519	12.683	15.600	17.858
e_{ab}	0.042	0.141	-0.519	-0.683	-0.600	0.142

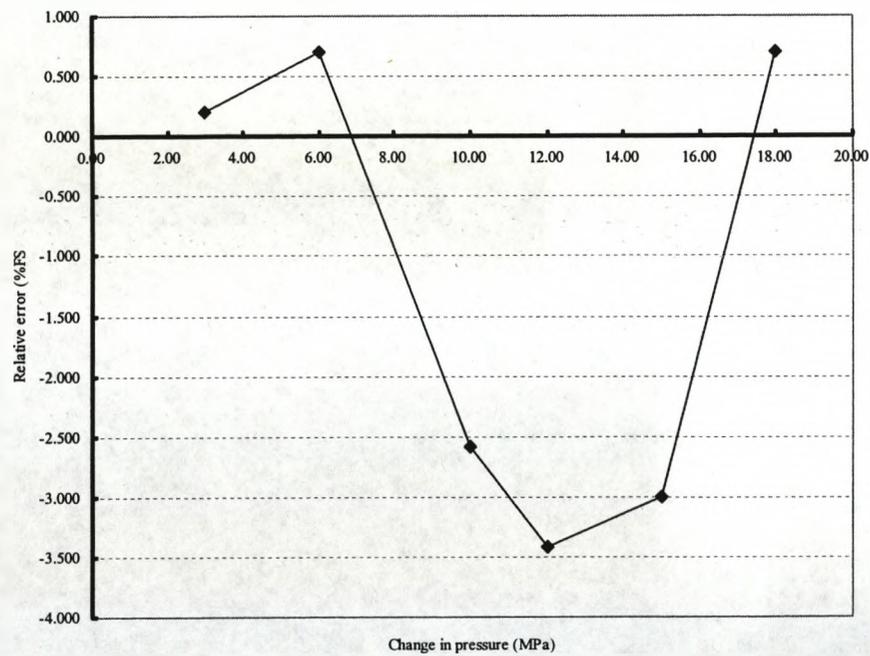


Fig.6.26 Relative error (%FS) of the inverse model with 1-7-1 structure for the test data

6.4 Discussion on Results of Modelling with MLP

In Section 6.2.2, the study of direct model with 1-5-1 structure has shown that the maximum relative error (%FS) of this model remains within 1.2%FS. In Section 6.2.3, the direct model with 1-6-1 structure is able to provide a maximum relative error (%FS) of 1.6%FS. The maximum relative error (%FS) of the direct model with 1-7-1 structure in Section 6.2.4 is 3.6%FS. It can be concluded that the maximum relative error (%FS) of the model with either 1-5-1 structure or 1-6-1 structure remains within 2%FS, while the 1-7-1 structure can only provide a maximum of 4%FS. Therefore, the first two models are acceptable for use to estimate the change in velocity from the change in pressure. Priority is given to the model with 1-5-1 structure for its less complexity of computation.

The inverse model with 1-5-1 structure in Section 6.3.1 can provide the maximum relative error (%FS) of 1.1%FS, while the models with 1-6-1 structure (in 6.3.2) and 1-7-1 structure (in 6.3.3) can only obtain those of 2.4%FS and 3.4%FS respectively. It is clear that the maximum relative error (%FS) of either 1-6-1 structure or 1-7-1 structure remains beyond

2%FS. The inverse model with 1-5-1 structure is preferred to calculate the change in pressure from the measurement of change in velocity.

6.5 Comparison between two modelling approaches

In Chapter 4, the principle of modelling the newly developed non-intrusive pressure measuring system is described using the functional-link neural network. The direct model and the inverse model of the system have been constructed. The results have shown that the maximum error (%FS) of estimated change in acoustic velocity against the expected variation of velocity is 1.12 %FS for the direct model. The maximum error (%FS) of estimated change in pressure against the expected variation of pressure is 0.98%FS for the inverse model.

In Chapter 6, the MLP is used with the L-M back-propagation algorithm to build the direct model and inverse model of the non-intrusive pressure monitoring system. The results have shown that the maximum relative error (%FS) of the direct model with either 1-5-1 structure or 1-6-1 structure remains within 2 %FS, while the 1-7-1 structure can only provide a maximum of 4 %FS. Therefore, the first two models are acceptable for use to estimate the change in velocity from the change in

pressure. Priority is given to the model with 1-5-1 structure for its less complexity of computation. The inverse model with 1-5-1 structure can provide the maximum relative error (%FS) of 1.1 %FS, while the inverse models with 1-6-1 structure and 1-7-1 structure can only obtain those of 2.4 %FS and 3.4 %FS respectively. The maximum relative error (%FS) of either 1-6-1 structure or 1-7-1 structure remains beyond 2 %FS. The inverse model with 1-5-1 structure is preferred to calculate the change in pressure from the measurement of change in velocity.

It can be concluded that the two modelling approach based on FLNN and MLP have the ability to establish the direct and inverse models for the non-intrusive pressure monitoring system. For both methods the maximum relative error (%FS) achieved for either the direct model or the inverse model is well within 2 %FS. However, compared to the MLP, the FLNN provides a reduced cost of computational complexity.

6.6 Summary

The architecture of the network is given, which is used with the L-M back-propagation algorithm to build the direct model and inverse model for the non-intrusive pressure monitoring system.

The L-M algorithm provides a nice compromise between the speed of Newton's method and the guaranteed convergence of steepest descent.

Implementation of direct modelling with MLP for non-intrusive pressure monitoring system is described. The maximum relative error (%FS) of the direct model with 1-7-1 structure is 3.6 %FS. The maximum relative error (%FS) of the model with either 1-5-1 structure or 1-6-1 structure remains within 2 %FS. Priority is given to the model with 1-5-1 structure for its less complexity of computation.

Implementation of inverse modelling with MLP for the non-intrusive monitoring system is stated. It is found that the maximum relative error (%FS) of either 1-6-1 structure or 1-7-1 structure remains beyond 2 %FS. The inverse model with 1-5-1 structure can provide the maximum relative error (%FS) of 1.1 %FS. The inverse model with 1-5-1 structure is preferred to calculate the change in pressure from the measurement of change in velocity.

Two modelling approach based on FLNN and MLP have the ability to establish the direct and inverse models for the non-intrusive pressure

monitoring system. For both methods the maximum relative error (%FS) achieved for either the direct model or the inverse model is well within 2 %FS. However, compared to the MLP, the FLNN provides a reduced cost of computational complexity.

7 Conclusions and Future Work

7.1 Conclusions

Condition monitoring of engineering processes or equipment has become of paramount importance as there is a growing need for improved performance, reliability, safety and more efficient maintenance. Condition monitoring in railway industry as a whole covers a very wide field. To restrict the field we have confined ourselves to the non-intrusive monitoring of hydraulic systems like diesel generators and Electro-hydraulic point machines. This thesis is mainly concerned with the investigation of the non-intrusive method based on ultrasonic concepts and neural networks for rapid condition monitoring and/or fault diagnosis of the hydraulic systems.

A comparison between diagnosing hydraulic systems and electric systems is made. Hydraulic systems are being used in all types of machinery and can be seen in all industries. However, unless some external mechanical failure has occurred, or unless there is some fairly obvious noise or

temperature, the location of faults in hydraulic systems can be difficult. With electrics one can check the volts by using test probes—sometimes current checks are necessary too but these again are comparatively easily done with inductive devices. With hydraulics the system is sealed and silent without visual indication of what is occurring inside. The key to fault finding in hydraulic systems is the location of pressure.

The development of pressure measurement instruments is reviewed. Conventional hydraulic pressure instruments are basically composed of pressure-sensing elements, conversion elements and display elements. The common point is that all these instruments are based on the principle of intrusive measurement, in which the pressure-sensing elements have to come in contact with the hydraulic media. In case of trouble-shooting hydraulic systems, pressure readings are often required to be taken at several temporary locations. Since the hydraulic system is fully sealed, these instruments can not be practically utilised for this purpose unless they are built-in during the production stage of the system. Instead, the indirect pressure measurement systems can be very helpful for rapid diagnosis of hydraulic systems because of their non-intrusiveness, easy installation, cost effectiveness and time-saving. The existing external pressure measurement is actually an extension of the traditional measuring

instruments from the pressure sensing element point of view, that is, the expansion of tubing served as the role of converting pressure into a displacement. Its inherent disadvantage lies in the fact that the hydraulic pipeline is not a good pressure sensing element.

A combined measuring scheme for non-intrusive pressure measurement is studied. Taking the acoustic characteristics of oil into consideration, a novel method for the non-intrusive measurement of pressure has been investigated. However, without an appropriate measuring scheme involved, the new method can not be applied to make meaningful measurements for condition monitoring. To get rid of the thickness influence of the pipes, a special measuring scheme, the combination of pulse echo and transmission is presented. The other advantage is that the overall time delay can be effectively removed.

Evaluating the performance of such a new non-intrusive monitoring system is our main objective. It includes the direct modelling process and the inverse modelling process.

Its direct model and the inverse model are constructed based on a functional-link neural network. The results have shown the effectiveness

of the proposed method with a low cost of computational complexity. The maximum error (%FS) of estimated change in acoustic velocity against the expected variation of velocity is 1.12%FS. The maximum error (%FS) of estimated change in pressure against the expected variation of pressure is 0.98%FS. The inverse model can be easily implemented in a microcontroller to accurately display the measurand digitally.

The architecture of the MLP network is also given, which is used with the L-M back-propagation algorithm to build the direct model and inverse model for the non-intrusive pressure monitoring system. The L-M algorithm provides a nice compromise between the speed of Newton's method and the guaranteed convergence of steepest descent.

Implementation of direct modelling with MLP for non-intrusive pressure monitoring system is described. The study has shown that the maximum relative error (%FS) of the direct model with 1-5-1 structure model remains within 1.2%FS. The direct model with 1-6-1 structure is able to provide a maximum relative error (%FS) of 1.6%FS. The maximum relative error (%FS) of the direct model with 1-7-1 structure is 3.6%FS. It can be concluded that the maximum relative error (%FS) of the model with either 1-5-1 structure or 1-6-1 structure remains within 2%FS, while

the 1-7-1 structure can only provide a maximum of nearly 4%FS. Therefore, the first two direct models are acceptable for use to estimate the change in velocity from the change in pressure. Priority is given to the model with 1-5-1 structure for its less complexity of computation.

Implementation of inverse modelling with MLP for the non-intrusive monitoring system is stated.

The inverse model with 1-5-1 structure can provide the maximum relative error (%FS) of 1.1%FS, while the models with 1-6-1 structure and 1-7-1 structure can only obtain that of 2.4%FS and 3.4%FS respectively. It is clear that the maximum relative error (%FS) of either 1-6-1 structure or 1-7-1 structure remains beyond 2.0%FS. The inverse model with 1-5-1 structure is preferred to calculate the change in pressure from the measurement of change in velocity.

Two modelling approach based on FLNN and MLP have the ability to establish the direct and inverse models for the non-intrusive pressure monitoring system. For both methods the maximum relative error (%FS) achieved for either the direct model or the inverse model is well within 2.0%FS. However, compared to the MLP, the FLNN provides a reduced cost of computational complexity.

7.2 Future Work

7.2.1 Further development of non-intrusive pressure monitoring system

The next move of this project is to develop the improved non-intrusive measurement of pressure in hydraulic systems using ultrasonic waves and apply it to engineering environments like the railway transportation for the quick diagnosis of the hydraulic devices.

An experimental apparatus will be developed to acquire the data set of velocity for various oils over a range of temperature and pressure. The ANN-based correction model will be established with the foundation laid by this thesis, which is associated with the simultaneous compensation of temperature and oil type on the acoustic measurements and the inverse modelling of the pressure measurement system.

The state of the art of hydraulic pressure measurement will be advanced by the development of such an improved non-intrusive measurement instrument.

7.2.2 Vibrating-wire sensing technique

This thesis has been concerned mainly with the non-intrusive techniques for the condition monitoring of hydraulic systems. However, the intention of the author is not to claim that the methods of this thesis will be appropriate in all systems. In some hydraulic systems, conventional methods, such as using vibrating-wire sensors [94]-[100], could find an ideal application. Unfortunately, detailed investigation into this technique is beyond the scope of this thesis. Nevertheless, it could make an appropriate research area for further investigation.

The author has considered this technique and done some limited work in this area. This has successfully led to the publication of a paper in the Measurement Journal of the International Confederation of Measurement, published by Elsevier Publications [101].

8 References

- [1] T. Hunt, "CM 2003 review: report on the condition monitoring 2003 conference, Oxford, 2-4 July 2003," *Insight Journal*, Vol.45, p636, September 2003.
- [2] J. Huggett, "The economics behind condition monitoring," *Proceedings of the International Conf. on Condition Monitoring*, Coxmoor Publishing Company, 2-4 July 2003, Oxford, UK, pp391-399, 2003.
- [3] R. A. Vingerhoeds, P. Janssens, B. D. Netten and M. A. Fernandez-Montesinos, "Enhancing off-line and on-line condition monitoring and fault diagnosis," *Control Eng. Practice*, Vol.3, No.11, pp1515-1528, 1995.
- [4] Cumming, "Condition monitoring today and tomorrow- an airline perspective," In: *Proc. COMADEM 90: 2nd International Congress on Condition Monitoring and Diagnostic Engineering Management*, pp1-7, 1990.
- [5] A. B. Sasi, A. Albarbar, P. Soobramaney, F. Gu and A. D. Baii, "Diesel engine condition monitoring using instantaneous angular

- speed extracted from the alternative voltage output," *Int. conf. on Condition Monitoring*, Oxford, Coxmoor Publishing Com., pp408-422, 2003.
- [6] R. Jones, "Computer-aided maintenance management systems," *Computing and Control Engineering Journal*, Vol.5, No.4, pp189-192, 1994.
- [7] S. Fararooy, J. Allan, "Condition-based maintenance of railway signalling equipment," *Conference Publication No. 405 IEE*, pp33-37, 1995.
- [8] F. Morris, "Bringing fourth generation maintenance to railways," *Railtech*, 24-26 May, Birmingham, IMechE: C478/23/012, 1994.
- [9] S. Fararooy and J. Allan, "Condition monitoring and fault diagnosis of railway equipment using acoustic emission sensors," *Insight Journal*, Vol.7, No.4, pp294-297, 1995.
- [10] S. Fararooy , J. Allan, and B. Mellitt, "Early failure warning system for railway signalling equipment:1-Train-stops," *Comprail'94*, Madrid, 7-9 Sept., Computational Mechanics Publication, pp135-142,1994.
- [11] Balgard, Iennart and Lundin, "Monitoring of primary circuit temperatures and breaker condition in a substation," *IEE Conf. No.373*, pp1-5, 1993.

- [12] B. Oyebande, "Condition monitoring of trackside signalling equipment," UMIST, *PhD thesis*, 1998.
- [13] G. Xian, *Diagnosis and trouble removal of hydraulic systems*, Ocean Press, 1998.
- [14] S. Ge, *Trouble shooting of hydraulic systems*, Xian Jiaotong University Press, 1992.
- [15] W. Bolton, *Instrument and Measurement*, Mewnes, 2000.
- [16] P. Harland, *Pressure Gauge Handbook*, Marcel Dekker Inc., 1988.
- [17] F. Oliver, *Practical instrumentation transducer*, Hayden Book Company, Inc., New York, 1971.
- [18] N. Tan, *Measurement Techniques and Hydraulically Operated Equipment*, Beijing University of Science and Technology Press, 1997.
- [19] G. Molinar, "Pressure transducer for pressure measurement in liquids up to 0.5 GPa," *Measurement*, **24**(3), pp161-172, 1998.
- [20] S. Knowles , "Multiple microbending optical-fibre sensors for measurement of fuel quantity in aircraft fuel tanks," *Sensors & Actuators*, , A68(1), pp320-323,1998.
- [21] W. Bock, "Temperature desensitization of a fiber-optic pressure sensor by simultaneous measurement of pressure and temperature," *Applied Optics*, Vol.37, No.18, pp3897-3901, 1998.

- [22] G. Brabander, "Integrated optical micromachined pressure sensor with spectrally encoded output and temperature compensation," *Applied Optics*, Vol.37, No.15, pp3264-3267, 1998.
- [23] J. K. Chandler and D. P. Foler, "External pressure measurement system," *ISA Transactions*, pp19-26, 1989.
- [24] Edited by J. Szilard, *Ultrasonic testing*, John Wiley & Sons Ltd., 1982.
- [25] F. Firestone, "Flaw detecting device and measuring instrument," US Patent No. 2 280 130, 1940.
- [26] C. Desch, D. Spoule, and W. Dawson, "The detection of cracks in steel by means of supersonic waves," *J. Iron Steel Inst.*, Vol.153, pp319-352, 1946.
- [27] W. Erwin, "Supersonic measuring means," US patent No. 2 431 233, 1944.
- [28] M. Greenwood and J. Bamberger, "Ultrasonic sensor to measure the density of a liquid or slurry during pipeline transport," *Ultrasonics*, Vol.40, pp413-417, 2002.
- [29] T. H. Gan, D. A. Hutchins, and D. R. Billson, "Preliminary studies of a novel air-coupled ultrasonic inspection system for food containers," *Journal of Food Engineering*, Vol.53, pp315-323, 2002.

- [30] J. S. Ridgway, K. S. Henthorn, and J. B. Hull, "Controlling of overfilling in food processing," *Journal of Materials Processing Technology*, Vol.92-93, pp360-367, 1999.
- [31] E. Vargas, R. Ceres, J. M. Martin, and L. Calderon, "Ultrasonic sensor for liquid-level inspection in bottles," *Sensors and Actuators*, A61, pp256-259, 1997.
- [32] Z. W. Huang, F. Y. Li, X. H. Wang, and S. Y. Wang, "Ultrasonic measuring method for oil density," *Journal of transducer technology*, Vol.19, No.4, pp47-48, 2000.
- [33] Z. Q. Liu, "On-line detection of H₂O₂ concentration with ultrasound," *Chinese Journal of Scientific Instrument*, Vol.15, No.1, pp49-54, 1994.
- [34] L. X. Zhang, "Measurement and analysis of density for liquids with big density," *Chinese Journal of Scientific Instrument*, Vol.18, No.3, pp314-317, 1997.
- [35] L. Filipczynski, Z. Pawlowski, and J. Wehr, *Ultrasonic methods of testing materials*, London: Butterworths, 1966.
- [36] S. Kocis and Z. Figura, *Ultrasonic measurements and technologies*, London: Chapman & Hall, 1996.
- [37] Edited by Paulo G. Lisboa, *Neural Networks Current applications*, London, UK: Chapman & Hall, 1992.

- [38] "Computing with neural networks," Technical comments, *Science*, **325**, pp1226-1229, 1987.
- [39] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagation errors," *Nature*, **323**, pp533-535, 1986.
- [40] J. Hopfield, "Artificial neural networks," *IEEE Circuits and Devices*, pp3-10, 1988.
- [41] R. Sutton, "Learning to predict by the methods of temporal differences," *Math. Learn.*, **3**, pp9-44, 1988.
- [42] P. Simpson, Foundation of neural networks. In: IEEE Technology Update Series Neural Networks Theory, Technology, and Applications, New York, pp1-22, 1996.
- [43] W. Hudson, "Introduction and overview of artificial neural networks," In: *IEEE Technology Update Series Neural Networks Theory, Technology, and Applications*, New York, pp746-749, 1996.
- [44] A. Armitage, "Neural networks in measurement and control," *Measurement and Control*, **28**, pp208-215, 1995.
- [45] M. Bernieri, L. Dapuzzo, M. Sansone, "A neural network approach for identification and diagnosis on dynamic systems,"

- IEEE Trans. on Instrumentation and Measurement*, **43**, pp867-873, 1994.
- [46] M. Attari, M. Heniche, F. Boudjema, "A two dimensional intelligent calibration of an ion sensor," *Proc. IEEE Instrumentation and Measurement and IMEKO Technical Conference*, Brussels, Belgium, June 1996, pp788-791, 1996.
- [47] M. Johnson, "Neural networks in advanced instrument design," *Measurement and Control*, **29**, pp101-105, 1996.
- [48] F. Carullo, S. Ferraris, U. Graziani, M. Grimaldi, "Ultrasonic distance sensor improvement using a two-level neural network," *IEEE Trans. on Instrumentation and Measurement*, pp677-682, April 1996.
- [49] K. Zhang, C. Butler, Q. Yang, Y. Lu, "A fiber optic sensor for the measurement of surface roughness and displacement using artificial neural networks," *Proc. IEEE Instrumentation and Measurement and IMEKO Technical Conference*, Brussels, Belgium, June 1996, pp917-920, 1996.
- [50] P. Daponte, D. Grimaldi, L. Michaeli, "Analog neural networks for pre-processing in measurement sensors," *XIV IMEKO World Congress*, Vol. IV, Finland, June 1997, pp56-61, 1997.

- [51] J. Gardner, E. Hines, C. Pang, "Detection of vapours and odours from a multisensor array using pattern recognition: self-organising adaptive resonance techniques," *Measurement and Control*, **29**, pp172-178, 1996.
- [52] Y. Liu, "Calibrating an industrial microwave six-port instrument using the artificial neural network technique," *IEEE Trans. on Instrumentation and Measurement*, pp651-656, April 1996.
- [53] H. Eren, C. Fung, and K. Wong, "An application of artificial neural network for prediction of densities and particle size distributions in mineral processing Industry," *Proc. IEEE Instrumentation and Measurement Technical Conference*, Ottawa, Canada, May 1997, pp1118-1121, 1997.
- [54] C. Liao, H. Lin, F. Chu, J. Shi, and C. Ho, "The feasibility of applying the neural network technology to the automatic time report system," *Proc. IEEE Instrumentation and Measurement Technical Conference*, Ottawa, Canada, May 1997, pp1122-1125, 1997.
- [55] P. Daponte, D. Grimaldi, A. Piccolo, and D. Villacci, "A neural diagnostic system for the monitoring of transformer heating," *Measurement*, **18**, pp 35-46, 1996.

- [56] Kamwa, R. Grondin, V. Sood, C. Gagnon, V. Nguyen, and J. Mereb, "Recurrent neural networks for phasor detection and adaptive identification in power system control and protection," *IEEE Trans. on Instrumentation and Measurement*, pp657-664, April 1996.
- [57] Bernieri, G. Betta, A. Pietrosanto, and C. Sansone, "A neural network approach to instrument fault detection and isolation," *IEEE Trans. on Instrumentation and Measurement*, **44**, pp747-750, 1995.
- [58] Bernieri, G. Betta, C. Liguori, and A. Losi, "Neural networks and pseudo-measurements for real-time monitoring of distribution systems," *IEEE Trans. on Instrumentation and Measurement*, pp645-650, April 1996.
- [59] S. Haykin, *Neural Networks: A comprehensive Foundation*, New York, USA: Macmillan College Publishing Company, 1994.
- [60] D. Barschdorff, "Neural networks and fuzzy logic: new concepts for technical failure diagnosis," *Proc. IMEKO Word Congress*, Torino, Italy, September 1994, pp130-137, 1994.
- [61] Y. Sukuki, "Self-organizing QRS-wave recognition in ECG using neural networks," *IEEE Trans. on Neural Networks*, **6**, pp1472-1478, 1995.

- [62] Baccigalupi, A. Bernieri, and C. Liguori, "Error compensation of A/D converters using neural networks," *IEEE Trans. on Instrumentation and Measurement*, pp640-644, April 1996.
- [63] P. Daponte, D. Grimaldi, and L. Michaeli, "A design method for signal processing in measurement instrumentation by neural networks," *Proc. IEEE Instrumentation and Measurement Technology Conference*, Brussels, Belgium, June 1996, pp1004-1009, 1996.
- [64] J. Mendel and R. McLaren, "Reinforcement-learning control and pattern systems," In *Adaptive, Learning, and Pattern Recognition Systems: Theory and Applications*, New York: Academic Press, pp287-318, 1970.
- [65] B. Widrow and S. Stearns, *Adaptive signal processing*, Englewood Cliffs, NJ: Prentice-Hall, 1985.
- [66] B. Widrow and M. Hoff, "Adaptive switching circuits," *IRE WESCON Convention Record*, pp96-104, 1960.
- [67] P. Werbos, "Back-propagation and neural control: A review and prospectus," *International Joint Conference on Neural Networks*, Washington DC, Vol.1, pp209-216, 1989.

- [68] V. Vapnik, "Principles of risk minimization for learning theory,"
In *Advances in Neural Information Processing systems*, San
Mateo, CA: Morgan Kaufmann, pp831-838, 1992.
- [69] Y. Pao, G. Park and D. Sobajic, "Learning and generalisation
characteristics of the random vector functional-link net,"
Neurocomputing, Vol.6, pp163-180, 1994.
- [70] C. Chen and J. Wan, "A rapid learning and dynamic stepwise
updating algorithm for flat neural networks and the application to
time-series prediction," *IEEE Trans. Systems, Man and
Cybernetics – Part B*, Vol.29, No.1, pp62-72, 1999.
- [71] J. Patra, R. Pal, B. Chatterji and G. Panda, "Identification of non-
linear dynamic systems using functional-link artificial neural
networks," *IEEE Trans. Systems, Man and Cybernetics – Part B*,
Vol.29, No.2, pp254-262, 1999.
- [72] C. L. Chen, S. R LeClair and Y. H. Pao, "An incremental adaptive
implementation of functional-link processing for function
approximation, time-series prediction, and system identification,"
Neurocomputing, **18**, pp11-31, 1998.
- [73] Y. Pao, *Adaptive pattern recognition and neural networks*,
Reading, MA: Addison-Wesley, 1989.

- [74] B. Widrow and A. Michael, "30 years of adaptive neural networks: perceptron, madaline and back-propagation," *Proceedings of the IEEE*, Vol.78, No.9, pp1415-1442, 1990.
- [75] M. Hagan, H. Demuth and M. Beale, *Neural network design*, Boston: PWS Publishing Co., 1997.
- [76] M. Bello, "Enhanced training algorithms and integrated training/architecture selection for multilayer perceptron networks," *IEEE Trans. On neural Networks*, **3**, pp864-875, 1992.
- [77] T. Samad, "Back-propagation improvements based on heuristic arguments," *Proceedings of International Joint Conference on Neural Networks*, **1**, pp565-568, 1990.
- [78] A. Superduti and A. Starita, "Speed up learning and network optimization with extended back-propagation," *Neural Networks*, **6**, pp 365-383, 1993.
- [79] V. Ooten and B. Nienhuis, "Improving the convergence of the back-propagation algorithm," *Neural networks*, **5**, pp465-472, 1992.
- [80] A. A. Miniani and R. D. Williams, "Acceleration of back-propagation through learning rate and momentum adaptation," *Proceedings of International Joint Conference on Neural Networks*, San Diego, CA, **1**, pp676-679, 1990.

- [81] R. A. Jacobs, "Increased rates of convergence through learning rate adaptation," *Neural Networks*, Vol.1, No.4, pp 295-308, 1988.
- [82] A. Salvetti and B. M. Wilamowski, "Introducing stochastic process within the back-propagation algorithm for improved convergence," In *Intelligent Engineering Systems through Artificial Neural Networks*, Vol.4, pp205-209, ASME Press, New York, 1994.
- [83] C. Charalambous, "Conjugate gradient algorithm for efficient training of artificial neural networks," *IEE Proceedings*, Vol.139, No.3, pp301-310, 1992.
- [84] S. Shah and F. Palmieri, "MEKA-A fast local algorithm for training feed-forward neural networks," *Proceedings of International Joint Conference on Neural Networks*, San Diego, CA, 3, pp41-46, 1990.
- [85] M. T. Hagan and M. B. Menhaj, "Training feed-forward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, Vol.5, No.6, pp989-993, 1994.
- [86] R. Battiti, "First and second order methods for learning: between steepest descent and Newton's method," *Neural Computation*, Vol.4, No.2, pp141-166, 1992.
- [87] *Neural network toolbox*, MathWorks, 2000.

- [88] F. Yu, N. Gupta and J. Hoy, "ANN-Based Modelling of Non-Intrusive Pressure Measurement System", *ELEKTRON (The SAIEE electronics, computing, information & communication engineering journal)*, Vol.24, No.5, pp15-17, May 2006.
- [89] F. Yu, N. Gupta and J. Hoy, "Non-intrusive pressure measurement based on ultrasonic waves," *Insight (Journal of the British Institute of Non-Destructive Testing)*, Vol.47, No.5, pp285-288, 2005.
- [90] F. Yu, N. Gupta, and J. Hoy, "Non-intrusive condition monitoring of diesel generators," *Energize (Power Journal of the South African Institute of Electrical Engineers)*, pp39-40, July 2004.
- [91] F. Yu, N Gupta, "A novel method for computer controlled condition monitoring of hydraulic systems," *Insight (Journal of the British Institute of Non-Destructive Testing)*, Vol.45, No.8, pp536-538, 2003.
- [92] F. Yu, N. Gupta and J. Hoy, "Non-intrusive pressure measurement: analysis of characteristics using neural networks," *IEEE Conference on instrumentation and Measurements*, 18-20 May 2004, Como, Italy, Vol.1, pp363-367, 2004.
- [93] F. Yu, N. Gupta, "Development of computer controlled non-intrusive pressure monitoring system for diagnosis of

hydraulically operated equipment,” *International Conference on Condition Monitoring*, St Catherine’s College, Oxford, July 2003, pp59-65, 2003.

- [94] B. Benmokrane, M. Chekired and H. Xu, “Monitoring behavior of grouted anchors using vibrating-wire gauges,” *Journal of Geotechnical Engineering-ASCE*, **121**, pp466-475, 1995.
- [95] D. Coutts, J. Wang and J. Cai, “Monitoring and analysis of results for two strutted deep excavations using vibrating wire strain gauges,” *Tunnelling and Underground Space Technology*, **16**, pp87-92, 2001.
- [96] J. Bordes and P. Debreuille, “Some facts about long-term reliability of vibrating wire instruments,” *Transportation Research Record*, **104** (1), pp20-26, 1995.
- [97] L. Xu and H. Liu, “Signal collecting and data processing of vibrating-wire sensors,” *Industrial Safety and Dust Control*, No.1, pp16-19, 2000.
- [98] Y. Gao, G. Ruan and Y. Huang, “Development of model SF-1 force meter using vibrating-wire sensor,” *Building Science*, Vol.14, No.2, pp27-30, 1998.
- [99] GB/T 13606-92, *Vibrating-wire pressure sensors for geotechnical engineering*.

- [100] Y. L. Cui, T. L. Deng, F. Yu, *Theories and measurement techniques of resonant sensors*, Beijing: China Mining Industry Press, 1997.
- [101] F. Yu, N. Gupta, "An efficient model for improving performance of vibrating-wire instruments," *Measurement*, **37**, pp278-283, 2005.

9 Appendix-Publications

1. F. Yu, N. Gupta and J. Hoy, "ANN-Based Modelling of Non-Intrusive Pressure Measurement System", *ELEKTRON (The SAIEE electronics, computing, information & communication engineering journal)*, Vol.24, No.5, pp15-17, May 2006.
2. F. Yu, N. Gupta and J. Hoy, "Non-intrusive pressure measurement based on ultrasonic waves," *Insight (Journal of the British Institute of Non-Destructive Testing)*, Vol.47, No.5, pp285-288, 2005.
3. F. Yu, N. Gupta, "An efficient model for improving performance of vibrating-wire instruments," *Measurement (Journal of the International Measurement Confederation)*, Vol.37, pp278-283, 2005.
4. F. Yu, N. Gupta, and J. Hoy, "Non-intrusive condition monitoring of diesel generators," *Energize (Power Journal of the South African Institute of Electrical Engineers)*, pp39-40, July 2004.
5. F. Yu, N Gupta, "A novel method for computer controlled condition monitoring of hydraulic systems," *Insight (Journal of*

the British Institute of Non-Destructive Testing), Vol.45, No.8, pp536-538, 2003.

6. F. Yu, N. Gupta and J. Hoy, "Non-intrusive pressure measurement: analysis of characteristics using neural networks," *IEEE Conference on instrumentation and Measurements*, 18-20 May 2004, Como, Italy, Vol.1, pp363-367, 2004.
7. F. Yu, N. Gupta, "Development of computer controlled non-intrusive pressure monitoring system for diagnosis of hydraulically operated equipment," *International Conference on Condition Monitoring*, St Catherine's College, Oxford, July 2003, pp59-65, 2003.
8. F. Yu, N. Gupta, "Correction of Non-Linearity for Sensors Implemented with a Microcontroller", *ELEKTRON (The SAIEE electronics, computing, information & communication engineering journal)*, in press.
9. F. Yu, N. Gupta and J. Hoy, "Estimation of Characteristics for Non-Intrusive Measurement of Hydraulic Pressure", Submitted to *IEEE (Instrumentation and Measurement)*.