Department of Mechanical Engineering

A Sensitivity Comparison of Neuro-fuzzy Feature Extraction Methods from Bearing Failure Signals

Jonny Latuny

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

November 2013

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature: …….

Date: ……

Abstract

Bearings play an important role in the operation of rotating machinery. They are critical machinery components that are subject to continuous load and harsh operational conditions and hence prone to failure during its lifetime. Bearing failure can lead to breakdown of the whole machine which could in turn lead to unwanted stoppage of an industrial production line. Therefore, the operational condition of a bearing must be monitored for the purpose of maintenance and avoidance of unwanted machinery stoppage that might be caused by such a bearing failure.

Better understanding of a bearing failure condition is useful in maintaining a continuous production line. In this context, a scheduled maintenance event, based on bearing fault diagnosis, is an advantage, preventing unwanted production line maintenance events. This is an important requirement in the prevention of revenue loss due to stoppage of the production line or in any rotating machinery related to human safety (i.e., transportation, etc)

The importance of bearing fault analysis and classification in relation to maintenance cost reduction and safety issues have been the motivation for intensive and wider research aimed at providing better methods for bearing fault analysis and classification.

A bearing fault diagnosis system which utilises advanced combinations of vibration signal processing and artificial intelligent methods has gained attention in recent years. In past decades, the trend was to combine digital signal processing with available artificial intelligence (AI) techniques to produce better and reliable bearing fault diagnosis systems. The application of signal processing and artificial intelligent methods is an open research field for investigation and exploration for the purpose of obtaining a new combined methodology.

A review of the literature, which includes vibration signal analysis in fault diagnosis, statistical parameter applications in feature extraction methods, wavelet transforms and artificial intelligence systems in fault diagnosis, is presented. Based on the

literature review, it was found that there is a possibility of proposing a new combined method for the purpose of building a bearing fault diagnostic system. In particular, the application of new feature extraction methods combined with artificial intelligence (AI) systems. There were no standard guidelines available in finding better systems for bearing fault analysis and classification through the utilization of combined feature extraction methods and AI applications.

This research reports an investigation process of building bearing fault classifiers for outer race, inner race and ball fault cases using a wavelet transform, statistical parameter features and an Artificial Neuro-Fuzzy Inference System (ANFIS). The building process started by acquiring and processing raw vibration signalfrom a bearing under investigation. The data acquisition process was carried out for both normal (fault-free) and faulty operation of a double row self-aligning ball bearing.

An accelerometer was used to collect the vibration data from a faulty bearing. The raw vibration data was processed using a wavelet transform employing a Daubechies wavelet filter to produce wavelet coefficients and their energy levels. The result was then processed to extract the statistical parameters (i.e., kurtosis, RMS, variance, standard deviation). The features generated from statistical parameters and wavelet transform scheme were then used to train the ANFIS.

In order to reduce the number of rules generated during the training process, only two inputs were used for the purpose of building the classifier. The selection of the most influential inputs for the training process of the ANFIS was achieved through the use of the ANFIS built-in capability of selecting the best correlation of two inputs towards one target output which best represented the bearing operating condition.

An extensive computation was used in the process of selecting the most influential input-output combination fromthe six inputs available. The number of input-output combinations tested was 720, being the permutations of six inputs. In the search for the best combination of input-output, the possible combinations of statistical parameters, wavelet coefficients and wavelet's level of energy were investigated extensively in order to obtain the best classifier for bearing fault diagnosis.

The ANFIS was then implemented to capture the input-output relation of the selected inputs to generate a suitable classifier that could be used to classify bearing operating condition. The classifiers generated were then tested to evaluate their ability and accuracy in predicting a faulty bearing.

The test results show that the ball fault (BF) classifier successfully achieved 100% accuracy without mis-classification, while the outer race fault (ORF), inner race fault (IRF) and no fault (NF) classifiers achieved mixed percentage between successful classification and mis-classification results.

Acknowledgements

The completion of this doctoral thesis has depended on valuable contribution and support of many people whom I would like to acknowledge.

I would like to thank my supervisor Dr. Rodney D. Entwistle who supervised and supported me in every stage of research and thesis writing process. His extensive academic expertise and technical knowledge are invaluable in directing me throughout my study.

My sincere thanks also extended to:
Mr. Graeme Watson of Mechanical Engineering Department workshop at Curtin University gave his excellent craftmanship for making the bearing test rig available.

Mr. Joseph Justin from Electrical Engineering Department workshop for his technical support on electronic motor controller for making the bearing test rig available. Mr. Derek Oxley of Mechanical Engineering Department workshop gave technical support on load cells calibration. Mr. David Collier of Mechanical Engineering Department workshop at Curtin University provided additional support.

Ms. Kim Yap, Ms. Margaret Brown and Ms. Diane Garth of Department of Mechanical Engineering office gave their excellent administration support.

AusAID provided the financial support for the opportunity to conduct a doctoral study at Curtin University through Australian Development Scholarships (ADS). Ms. Julie Craig of Curtin International Sponsored Students Unit provided continuous support for AusAID scholarships adminstration.

Finally, but most importantly, I would like to thank my wife *Yulianty Sukasly Ginting* who gave her support, encouragement and understanding during the course of the study. I present this great achievement to you.

The fear of the LORD is the beginning of knowledge

-Proverbs 1:7a

To my wife Yulianty Sukasly Ginting

Table of Contents

List of Figures

Page

List of Tables

Chapter 1 – General Introduction

1.1. Introduction

The success of the industrial sector,in terms of operation, haslong-depended upon the reliability of both the production line and industrial equipment. Production machinery should therefore be well-maintained in order to achieve the highest reliability level which in turn supports the production of cost-effective, high-quality products.

Machines are critical to the industrial production process. Many production machines have rotating components such as shafts which are supported by bearings that could deteriorate and fail during their operation. Therefore, maintenance is required in order to keep such machines in good operational condition. However, even when regular and efficient maintenance is undertaken, machinery may still deteriorate during operation.

The complexity of industrial equipment has increased over time due to vast technological developments in the area. These advances and the associated complexity of the machinery have added to the effort required to achieve high machine reliability. These factors are compounded by difficulties in identifying and predicting faults within acceptable time frames. This situation has led to the need for a more advanced and effective planning and maintenance strategy in managing installations and the production process.

A high standard of maintenance program is required, one that can ensure the operational reliability of industrial machinery and production systems whilst keeping profitability and competitiveness of the companies in question. As stated by Tu *et al*. (2001), it is crucial to have robust maintenance management systems with which an industrial organisation can maintain its equipment in an effective manner and at an optimum level, whilst at the same time lowering maintenance costs.

In addition, the urgency for the development and improvement of existing maintenance systems is motivated by the fact that industries need to maintain a high

degree of operational safety, provide better assets availability and reduce maintenance costs. There are also requirements within the current advanced industrial field to operate at low-risk to the environment while achieving maximum output.

Advanced fault monitoring and analysis methods are needed to monitor and analyse the operational condition of machines with rotating components. These monitoring and analysis methods are aimed at preventing unwanted machine stoppages caused by the failure of rotating components. An effective fault analysis method which accurately indicates bearing deterioration,or at the least the development of faults, would be advantageous in the planning of maintenance processes.

Fault analysis and diagnosis methods play an important role in monitoring the operational condition of rotating machinery. As such, they have become an attractive research field in condition monitoring that aims to produce a fault diagnosis scheme which is more accurate and efficient in its application than previous schemes.

The following section will briefly discuss types of maintenance methods, in order to place this research in context.

1.2. Maintenance Methods

Historically, maintenance methods may be grouped into at least three types: Breakdown Maintenance, Preventative Maintenance and Condition-Based Maintenance (CBM) (Heng *et al*. 2009).

1.2.1 Breakdown Maintenance

Breakdown maintenance, historically, is the earliest maintenance method, in which maintenance works, repair or replacement of machinery equipment is carried out only when parts have failed (Jardine *et al*. 2006). Breakdown maintenance is also termed unplanned maintenance. Its main disadvantage is in the inability to plan for an interruption to production, as machinery breakdowns and stoppage times are not predicted.

1.2.2 Preventative Maintenance

Preventative maintenance is a time-based maintenance method which aims to prevent critical failures and emergency shutdowns of machinery equipment. In preventative maintenance, work is performed periodically regardless of the operational condition of the machinery system. Maintenance activities include periodic machine inspections and maintenance without considering the machine's actual operational condition. Examples of periodic preventative maintenance works may include lubrication, calibration, inspection of equipment, drive belt and bearing replacement at regular intervals, or some other scheduled time, regardless of the health status of machinery equipment.

Optimal maintenance interval selection is important to the effective functioning of the predictive maintenance method. A mathematical optimisation method was proposed by Bazovsky (1961) to calculate the interval for the purpose of preventative maintenance. An optimal maintenance interval method was also proposed by Jardine *et al.* (1973). This was based on the analysis of the reliability of data (e.g., machine historical breakdown events) and information regarding costs incurred.

Gertsbakh (1976) proposed a method to determine required maintenance time. This was based on the assumption that the mean time between failures (MTBF) for machinery equipment could be determined statistically or inferred from experience of systems which are operated under normal conditions, with normal loading and usage.

The main disadvantage with preventative maintenance is that some techniques used are no longer suitable and cannot accommodate the practical requirements of the operation of modern industrial machinery. Where the products of an industry become more complex and demand high quality and high reliability, the use of a conservative prediction model in predictive maintenance has resulted in very high maintenance costs (Jardine *et al*. 2006). Furthermore, preventative maintenance requires greater resources in terms of labour and often results in unnecessary maintenance works, which still may not prevent the occurrence of critical failures (Heng *et al.* 2009).

1.3 Condition-based Maintenance (CBM)

Condition-based maintenance (CBM) is a maintenance program that is performed based on information collected through condition monitoring (CM) processes. The main objective of CBM is to perform maintenance works only if there are signs or indications of abnormal operation conditions in machinery installations. CBM can avoid unnecessary maintenance and uncertainty. In addition, it aims to prevent interruption of normal machine operations since maintenance works would only be carried out as needed or as dictated by signs or indicators of an abnormality that appears in the machines. CBM must be supported by efficient and reliable decision-making activities related to the need to carry out maintenance works based on diagnosis of the real-time condition of a machine.

An effective CBM scheme is based on three key elements:

- data acquisition - data related to machine health condition is collected and stored.
- data processing, conditioning - filtering and features extraction/selection.
- decision-making process - providing outputs and assessments regarding machine health condition which can be translated to maintenance actions based on diagnosis and prognosis processes (Jardine *et al.* 2006), (Yen and Lei, 2000).

1.3.1 Condition Monitoring (CM) for Condition-based Maintenance (CBM)

In general, condition monitoringcan be defined as a process which involves the technical activities of gathering or collecting information associated with machinery operation for the purposes of determining the integrity status and operational condition of the observed machine. It provides an ability to monitor current conditions and predict the future condition of a component or machine while it is in operation.

Making observations while a machine is actually operating is an advantage, since this will not affect or interrupt production schedules. However, this feature limits the collection parameters to those that may be obtained externally. Nonetheless the internal condition of the machine in operation is still represented.

Condition monitoring is essential to the achievement of an effective condition-based maintenance strategy and for planning the management of machinery installations and production processes. It is crucial to the support of a robust maintenance management system, by which a manufacturer can maintain their machinery or equipment in an effective manner and at an optimum level, while at the same time lowering maintenance costs where possible (Tu *et al*. 2001).

In relation to creating maintenance schedules, the time to carry out a maintenance event is determined by observing the operational condition of the components in the machinery system. This requires that the data representing several operational parameters of the system be measured and recorded either periodically (intermittently) or continuously.

In essence, condition monitoring for maintenance purposes includes five distinct stages which are: fault detection, fault diagnosis, fault development prognosis, post-fault or post-breakdown analysis and evaluation. Fault diagnosis is the process in which location, level of damage and cause of the machinery fault is determined. It is a very important part of condition monitoring.

1.3.2 Methods of Condition Monitoring

In applying condition monitoring to machine health condition evaluation, several condition-monitoring methods have been employed that are related to obtaining machinery operational data. These methods make use of interdisciplinary fields such as industrial vibration and noise, dynamics, Tribology and non-destructive testing (NDT). The methods/techniques include:

- Mechanical vibration signatures analysis,
- Lubricant analysis or oil particles density rate analysis,
- Acoustics emission signature analysis,
- Electrical measurements & analysis, and
- Non-destructive testing & analysis.

The focus of the research in this thesis is on mechanical vibration signatures. The analysis is based on the finding that when a machine is in operation, it generates vibration signals with unique patterns or signatures. Machines in operation generate mechanical vibrations and the signature of the mechanical vibration changes with operational conditions, be they normal or faulty or operating under particular or unusual loading conditions.

1.4 Thesis

This thesis addresses the generation of feature extraction methods for enhancing mechanical vibration signatures analysis in condition monitoring. The purpose of the methods is fault identification and classification. The methods are carried out by implementing a combination of Wavelet transforms, Artificial Neuro-Fuzzy Inference Systems (ANFIS) and Statistical Parameters.

The main objective of the research was to develop a neuro-fuzzy feature extraction method for bearing vibration signals. The method was then used to investigate and compare signal sensitivity in the application of bearing fault diagnostics.

The thesis reports on the design and development of a fault feature extraction technique through the use of wavelet transforms, statistical parameters and ANFIS. The generated features were used to train the ANFIS to construct a fuzzy inference system (FIS). The purpose of the FIS unit was to identify three types of rolling element bearing faults: outer race, inner race and ball faults. If successful, the discovered feature extraction technique could then be used to monitoring bearing condition and performance.

1.5 Report Structure

A thorough discussion is presented in the thesis. It explains the process of the design and development of feature extraction methods and their application to ANFIS training. The objective of ANFIS utilisation is to generate a fuzzy inference system (FIS). The FIS obtained through ANFIS training consists of fuzzy inference models that can be used to recognise the particular bearing fault (i.e., outer race, inner race, and ball faults).

The principal areas of the research consisted of:

- The vibration signals which were transformed using Daubechies (db-n) wavelet type, with seven features being generated from the wavelet transform results. The features generated were energy levels, RMS, kurtosis, dominant frequency of wavelet transform result, amplitude magnitude of dominant frequency of the wavelet transform result, standard deviation, and variance.
- Exhaustive processes were performed to select the best related data features that could be used in training the ANFIS model.
- The results of the ANFIS training produced FIS units which were then evaluated for their accuracy and db-n sensitivity in recognising each type of bearing fault.

1.6 Significance of the Research

In recent years, there has been increasing interest in finding better methods for failure detection in machine parts, especially the bearings which play an important role in the operation of rotating machinery. A great number of proposed techniques have been presented in the literature, and with the rapid development of these failure detection methods, there is a need to find a way of selecting a suitable method for a particular application.

In the pursuit of a superior method, a suitable feature extraction method for bearing vibration signals requires development. It must be sensitive enough for use with special types of bearings, and for specific loading conditions. The effects of operating conditions have not yet been fully investigated and modelled for fault diagnosis (Heng *et al*. 2009).). Even though this is an important gap in the knowledge relating to this research, the main aim of this thesis was to propose an intuitive new approach which employs a combination of several existing methods that aimed to extract additional patterns from post-processed bearing vibration signals. These could then be used to formulate bearing diagnostic models of wider applicability, that later, within the next phase of the research, might be used to also integrate bearing's operating parameters.

The aim of investigating the various methods was to understand their characteristics which in turn, might provide an effective new method consisting of the combination of several existing methods.

In fault diagnosis processes, the existence of many feature extraction methods for bearing fault analysis raises a need for selection guidelines. These would be used for implementing the automatic selection of features and intelligent diagnosis for bearing fault analysis.

1.7 Scope of Research

This research focused on the development of a specific feature extraction method for bearing fault identification that could be used for condition monitoring. The core objective of the research was to produce a Fuzzy Inference System (FIS) which would be generated through the training process of an Artificial Neuro-Fuzzy Inference System (ANFIS).

In this research, raw vibration signals acquired from a test rig installed with three types of bearing faults were processed using a Discrete Wavelet Transform (DWT) and the results were then processed to generate seven features that would be used in ANFIS training. Detailed explanations of the algorithm and the vibration signal pre-processing using wavelet transform and feature extraction techniques are presented.

The selection of the features that best related to a specific bearing fault condition was carried out automatically through ANFIS training. The ANFIS training process was aimed at automatically generating the FIS units that were used to recognise bearing faults. The generated FIS units that related to each type of bearing fault (i.e., outer race fault, inner race fault and ball fault) were evaluated to assess their accuracy and sensitivity at recognising the type of bearing fault that related to the FIS training data. The findings regarding characteristics and aspects of the feature extraction methods and FIS performance were presented for assessing their merits and disadvantages.

Chapter 2 – Review of Vibration Signal Analysis in Fault Diagnosis – Techniques and Literature

The performance of condition-based maintenance (CBM) processes is closely related to the performance of the associated condition monitoring. In turn, the performance of condition monitoring is determined by the quality of the fault diagnoses. Therefore, fault diagnosis plays an important role when used to assess the real-time or on-line condition of machines during condition monitoring. This indicates that there is a close relationship between CBM, condition monitoring and fault diagnosis. Therefore, the diagnosis of location, cause and severity level of a machine fault is a skill which allows maintenance management departments within an industry to utilise the condition monitoring method of maintenance.

Various condition monitoring techniques have developed health indicators from areas such as vibration, noise, dynamics, Tribology and other non-destructive methods. According to ISO 1991, vibration measurement, electrical measurement, and tribological measurement are the principal methods used in condition monitoring.

Vibration measurement is widely used due to its effectiveness and versatility (Patil *et al*. 2008). It is based on the common knowledge that the vibration level and pattern of a machine has a close relationship with the health condition of the machine. This correlation between vibration signal and machine health condition has been used widely in condition monitoring by various analyses of particular vibration signals (Yen and Lin, 2000).

The vibration signal in question is translated into the data measured and collected at selected intervals, thus forming a time series. It is collected via a data acquisition process where physical information is measured by a transducer (e.g., an accelerometer) that is attached to the machine. The vibration signal generated by the machinery component of interest is recorded and stored for the purpose of condition monitoring. The vibration signals are mainly processed and analysed for the purpose of diagnosis and/or prognostics in condition monitoring. The products consist of

particular information or indicators that are extracted from the vibration signal by means of signal processing techniques and algorithms.

The time series vibration signal is also called a time waveform and the data processing of the waveform of the vibration signal may be called signal processing.

As detailed in much of the literature, the waveform analysis of vibration signal can be divided into three main categories: *time-domain analysis*, *frequency-domain analysis* and *time-frequency analysis* (Jardine *et al.* 2006). These three main categories of vibration signal processing have a wide range of applications in the fault diagnosis field.

Vibration signal analysis and processing techniques in fault diagnosis are mainly aimed at extracting key features from the vibration signal that can be used for fault diagnosis and identification. Feature extraction techniques may involve domain transformation that extracts the desired signal features which may be hidden in the original time domain signal. Feature extraction also reduces the amount or dimensionality of the data that must be processed before performing the classification process (Yen and Lin, 2000).

In addition, feature extraction helps in speeding up computational processes and in shortening processing time. Feature extraction processes are also of use in cases where the feature is to be used by a classifier or a human inspector. The process ensures the accuracy of the information embedded in the vibration signal (Yen and Lin, 2000).

In relation to vibration signal analysis, feature extraction techniques or methods can be divided into three main groups based on the three approaches to vibration signal analysis ((Du and Yang, 2007), (Rai and Mohanty, 2007) and (Zhang *et al.* 2010)).

The following sections review some important aspects of the three categories of vibration signal analysis techniques.

2.1 Time-domain Analysis

Time-domain analysis of vibration signals deals directly with the time-based waveform of the vibration signal. The time-based waveform is acquired using an installed sensor. The acquired parameters are mainly velocity and acceleration of the vibration signal in the time-domain. Time-domain features can be calculated from the acquired time waveform signals using descriptive statistics like kurtosis, the root mean square (RMS), mean, standard deviation, crest factor, skewness, and others.

2.1.1 Standard Deviation

Standard deviation is defined as:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}} \qquad (2.1)$$

where x_i is each value in the data set, $\bar{x}$ is the mean of the data set and N is the number of data points (Seker and Ayaz, 2003), (Xu *et al*. 2009). An example of standard deviation used as a feature in bearing fault diagnosis is presented and can also be found in Mathew and Alfredson (1984).

2.1.2 Kurtosis

Kurtosis is defined as the fourth moment of the amplitude distribution which measures the "peaks" and "lows" of a distribution as compared to a normal distribution (Yang *et al*. 2003). It is defined as:

$$Kurtosis = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^4}{(N-1)\sigma^4} \qquad (2.2)$$

where x_i is i^{th} data point of the data set, N is the number of data points, $\bar{x}$ is the mean value of the signal and σ is the standard deviation (Howard, 1994). Example applications or the use of kurtosis as a feature in fault diagnosis can be found in Dyer and Steward (1978), Heng and Nor (1997) and Lee and White (1997).

2.1.3 RMS

The Root Mean Square is an indicator of the energy level or power level of a vibration signal. It also known as the quadratic mean of signal and it indicates overall signal energy level. It is defined as:

$$\text{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \qquad (2.3)$$

where N is the number of signal data points, $\bar{x}$ is the mean value of the signal and x_i is the i^{th} element of data set (Xi *et al*. 2000), (Xu *et al*. 2009). An example application of RMS as a feature in bearing fault analysis can be found in Xi *et al*. (2000). In addition, Howard (1994) discussed RMS application in detecting the presence of incipient bearing damage.

2.1.4 Variance

Variance is also known as second moment statistical measure and it is defined as:

$$Variance = \frac{\sum(x_i - \bar{x})^2}{N} \qquad (2.4)$$

where $\bar{x}$ = mean of the data set, x_i is each value in the data set and N is the number of data points. Examples of variance application in analysing vibration signals can be found in Loutridis (2008) and Rafiee *et al*. (2009).

2.1.5 Crest Factor

Crest Factor (CF) is defined as:

$$CF = \frac{Peak}{RMS} \qquad (2.5)$$

where $peak = \frac{1}{2}(\max(x_i) - \min(x_i))$ and RMS is as defined in Equation 2.2

A study regarding the use of the crest factor (CF) as a fault feature in detecting bearing faults was presented in Howard (1994).

2.1.6 Skewness

Skewness is defined as:

$$skewness = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^3}{(N-1)\sigma^3} \qquad (2.6)$$

where $\bar{x}$ is the mean, x_i is each value in the data set, σ is the standard deviation and N is the number of data points.An example of skewness application in detecting bearing faults can be found in Heng and Nor (1997).

2.1.7 Impulse Factor

Impulse Factor (IF) is defined as:

$$\text{Impulse Factor} = \frac{peak}{\frac{1}{N}(\sum_{i=1}^{N}|x(i)|)} \qquad (2.7)$$

where N is the number of data points, and $x(i)$ is the i-th element of the data. A discussion of the IF application in bearing fault diagnosis can be found in Howard (1994).

A detailed presentation regarding applications of the abovementioned statistical parameters in bearing fault diagnosis is presented in Section 2.2.

There are also other methods of analysis in the time-domain analysis area of vibration signals that go beyond simple statistical analyses. For example, time-synchronous averaging (TSA) and Autoregressive Moving Average (ARMA).

2.1.8 Time-Synchronous Averaging (TSA)

Another popular time-domain analysis method is time-synchronous averaging (TSA) (Jardine *et al*. 2006). In principle, TSA is an application of a grouped average of the raw vibration signal over a number of cycles synchronised to the rotation of a shaft. TSA attempts to remove or reduce the noise and effects from other sources in order to enhance the components of the signal that are of interest.

The principle of TSA is to use an ensemble average of the raw signals over a number of revolutions which are aimed to reduce noise. By reducing noise, the objective is to enhance the signal component of interest. A repetitive pattern in the signals obtained as a result of TSA process indicates the information related to faults.

In practice, TSA is carried out by averaging together a series of signal segments each of which corresponds to one period of a synchronised signal, triggered by a once-per-revolution trigger of a known phase (key-phasor). TSA is defined by (Randall and Antoni, 2011), (McFadden and Toozhy, 2000) as:

$$c(t) = \frac{1}{N}\sum_{n=0}^{N-1} x(t + nT_t), 0 \le t \le T_t \qquad (2.8)$$

where $x(t)$ denotes the signal, T_t is the averaging period and N is the number of samples used for averaging.

Details on TSA can be found in Randall and Antoni (2011) and in Dalpiaz *et al*. (2000). Several successful applications of TSA were obtained through synchronous averaging in the analysis of bearings and this was reported on in Wang and McFadden (1993), McFadden and Toozhy (2000) and Komgom *et al.* (2008).

2.1.9 Autoregressive Moving Averaged (ARMA)

The Autoregressive Moving Average (ARMA) is an advanced technique used in time-domain analysis of waveform data. The principle of ARMA is to fit waveform data to a parametric time series model and extract features based on the parametric model. The general model of ARMA can be found in Jardine *et al.* (2006). Example applications of the technique can be found in Poyhonen *et al.* (2004) where the autoregressive technique was used to model a vibration signal acquired from an induction motor. The autoregressive model coefficients were used as the extracted features. Another example of an ARMA application can be found in Zhan *et al*. (2003) where a vibration signal was analysed using a state space model representation of an autoregressive model.

There are also other time-domain techniques such as filter-based methods (i.e., demodulation and adaptive noise cancelling (ANC)) and stochastic and advanced methods (i.e., blind source separation). The following section presents a brief discussion regarding these techniques.

2.1.10 Filter-based Methods

ANC is a method of filtering signals corrupted by additive noise. It utilises two input signals, a primary input which contains the corrupted signals, and a reference input which contains noise correlated with the primary noise. Generally, the acquisition of these two types of signals has been carried out simultaneously. The reference signal may be acquired from a sensor located at a position in the noise area where the signal is very weak (Shao and Nezu, 1999).

The ANC method has been successfully applied in the detection of bearing faults (Chaturvedi and Thomas, 1982). An asynchronous ANC has also been successfully applied in detecting faulty self-aligning roller bearings(Shao and Nezu, 1999).

Modulation is a condition by which a sinusoidal signal, called a ‘carrier-signal’, has its amplitude or frequency made to vary with time. In this case, if the amplitude is varied then it becomes an amplitude modulation. If the frequency is varied then it becomes a frequency amplitude or phase modulation. Demodulation is the reverse of the modulation process. A demodulation method called ‘envelope analysis’ has been widely used in analysing the vibration signals of bearings and gears. The signal envelope is extracted by using amplitude modulation and its frequency is analysed to reveal the repetitive frequencies that are related to the faults (Randall, 2011).

A signal resonance demodulation technique for fault analysis was used by Wang (2001) to detect gear faults. Nikolaou and Antoniadis (2002b) utilised demodulation of vibration signals to diagnose defects in rolling element bearings.

The Prony method, originated from French scientist Baron de Prony in 1795 (Chen and Mechefske, 2000), extends Fourier analysis by directly estimating the frequency, damping, magnitude and relative phase of the modal components present in a given signal (Hauer *et al*. 1990).

A Prony-based model was applied to bearing faults diagnosis by Chen and Mechefske (2000), Chen and Mechefske (2002). It was used to analyse transient vibration signals generated by low speed element bearings fault. The method produced spectral plots by using small data sampling rate. Trending parameters was obtained and Prony parameters presented based on the spectral estimations. The results showed that Prony-based model able to analyse the transient impacted signal and the determining fault severity.

2.1.11 Advanced Methods

Advanced methods such as stochastic parameters (including chaos, blind source separation, etc) have been used to analyse time-domain vibration signals. For example, the irregularity (chaos)condition where computation parameters, known as correlation dimensions, were used to characterise several induced faults of varying severity in a rolling element bearing (Wang *et al*. 2001). Based on the correlation dimension, some basic information on the investigated dynamical system can be determined and the information used to classify differing fault conditions (Mevel *et al*. (2000).

Blind source separation (BSS) is a method for recovering signals from different physical sources and from several observed combinations which are independent of the propagation medium (Serviere and Fabry, 2004). BSS is a promising tool for non-destructive monitoring as it is able to recover the vibration signature of a single rotating machine from a combined vibration signature of several operating machines (Gelle *et al*. 2001). Its application for bearing fault detection was investigated and showed a potential and promising result (Gelle *et al*. 2001).

2.2 The Use of Time-domain Analysis of Vibration Signals for Fault Diagnosis

Time domain analysis of vibration signals for the purposes of bearing fault analysis is a straightforward process which can be done by measuring the level of statistical parameters like RMS, kurtosis, crest factor, variance, standard deviation, etc.Statistical parameters that have been commonly used for fault diagnosis in rolling element bearings are the root mean square (RMS), kurtosis, standard deviation, and skewness (Tandon and Choudhary, 1999).

These statistical parameters are indicators that can be used to show the shape of the amplitude distribution of the vibration data collected from a bearing. In addition, they have an advantage over time and frequency domain analyses by which the variation and speed conditions have a lower effect on their values (Honarvar and Martin, 1997).

An early application of feature extraction in time-domain analysis for fault diagnosis was presented by Dyer and Stewart (1978). Statistical features were used in the analysis of bearings. Skewness and kurtosis values were used to analyse outer race defects in rolling element bearings. It was found that the kurtosis value was close to 3 for an undamaged bearing. Kurtosis values greater than 3 could then be used as indicators of the existence of a defect.

In addition, the use of kurtosis values in fault investigation provides a low cost tool for both maintenance and quality control applications in fault diagnosis (Patil *et al*. 2008).

However, the main disadvantage in using kurtosis is that the value becomes lower as the defect becomes more severe. Hence, it was suggested by Dyer and Stewart (1978) that kurtosis values be used in selected frequency bands as a potential measure of bearing condition but combined with other maintenance measures such as schedules, spares availability, etc.

Braun and Datner (1979) investigated the use of the RMS of averaged bearing vibration waveforms for fault detection. Artificial localised defects were induced, using an electric-discharge machine, into the inner and outer race of the test bearing structure. The results showed that the proposed method had low sensitivity in detecting sharp repetitive transient signals generated by low loading conditions.

Another early application of the use of statistical parameters in the analysis of bearing vibration signals was conducted by Mathew and Alfredson (1984). In the study, peak amplitude, RMS, standard deviation, kurtosis, and mean values were used in the analysis of inner race, outer race, and ball defects of rolling element bearings. It was found that the values of kurtosis multiplied by RMS were the most

sensitive indicators of condition where impulses were present in the vibration signal. It was also found that a near-Gaussian distribution was obtained for the damaged bearings.

The effectiveness of the method of using kurtosis was also explored by White (1984), in which a simulated condition for generating a faulty vibration signal was used. The results suggested that kurtosis would be useful as a monitoring parameter in situations where pulses contribute significantly to the signal level.

Daadbin (1991) utilised kurtosis as a feature in fault detection in rolling element bearings. The fault detector was developed using a computer program. The results showed that the kurtosis analysis gave a clear indication of the damage. For healthy bearings the value of kurtosis was around 3.0 while all of the damaged bearings produced values of kurtosis higher than 3.0.

Lee and White (1997) investigated the use of higher-order statistics such as Wigner bi and tri-spectra in detecting faults in mechanical systems, based on the observation of signals with impulsive components. In line with other investigations, it was concluded that the impulsive components of signals tend to increase kurtosis values.

Heng and Nor (1998) successfully used kurtosis as a criterion to distinguish between the signals of faulty and normal bearings. Williams *et al*. (2001) explored the application of kurtosis and the crest factor in detecting and localising the damage in rolling bearings. It was found that effective bearing signals have a Gaussian distribution with an approximate kurtosis value of 3. This was also found by Dyer and Stewart (1978).

Tandon and Choudhary (1999) reported limited success in applying the RMS level and crest factor (i.e., the ratio of peak value to RMS value of acceleration) in obtaining the status of the operating condition of a bearing under test.

The trend of using statistical parameters as features in the fault analysis of bearings later shifted to a combination of statistical parameters with other techniques such as artificial neural networks (ANN), wavelet transforms and neuro-fuzzy techniques.

The combined techniques aimed to enhance the accuracy and reliability of fault diagnosis outcomes. Examples of this development are reviewed below.

Xi *et al*. (2000) investigated the use of statistical parameters such as RMS, kurtosis, impulse factors and crest factors as the basis for a trend analysis system using pattern recognition for the purposes of bearing defect classification. The classification features obtained from seeded fault bearing operations were used to construct the computational engine of the classification system. The classification system provides a visualisation of the diagnostic results on a two-dimensional plane and the classification space was constructed by a piecewise linear classification function. Numerical experiments using data with seeded defects showed that the method was effective in indicating both the location and the severity of bearing defects.

Altmann and Mathew (2001) used kurtosis values as features for training a neuro-fuzzy network adapted from Jang (1993) and Jang and Sun (1995). The purpose of the proposal was to enhance the detection and diagnosis of low-speed rolling-element bearing faults by using discrete wavelet packet analysis (DWPA), as presented by Daubechies (1990). The method included the automatic extraction of wavelet packets, containing bearing fault-related features, from the discrete wavelet packet analysis of machine vibrations. An adaptive neuro-fuzzy inference system (ANFIS) (Jang, 1993) was used in automating the selection process of the wavelet packets in question. The results showed that the DWPA multiple band-pass filtering significantly improved signal-to-noise-ratio which was useful to obtain the best possible isolation of the bearing transients information.

A generic method for analysing rolling element bearing faults by use of statistical features and pattern recognition was investigated by Sun *et al*. (2004). Statistical parameters such as kurtosis, crest factor, and RMS were calculated from the vibration signal generated from a bearing with faults (i.e., one with inner race, outer race and roller defects) and without fault. The statistical parameters were used as features for the purposes of fault pattern recognition. The generated features were then mapped to create feature integration, linear classification and diagnosis. An artificial neural network (ANN) scheme was used in parallel with the mapping scheme for to generate a classifier for bearing fault classifications. The proposed

methodology showed advantages in dealing with complicated signatures such as those present in the vibration signal of rolling element bearings.

Niu *et al*. (2005) proposed the application of a new uniform description of normalised statistical parameters of skewness and kurtosis for analysing vibration signals from a rolling element bearing. The results of simulation and experimental testing indicated that the two new statistical parameters were as effective as the original parameters (i.e., generic skewness and kurtosis) for detecting bearing failure.

Rafiee *et al*. (2007) employed standard deviation as a feature in their study of bearing fault diagnosis. Standard deviation values were calculated from wavelet transform results. The values of standard deviation were then used together with other features such as energy levels of wavelet transform results in order to train neural networks. The results showed a considerable increase in training convergence and network performance which contributed to the 100% accuracy gained in detecting and gear and bearing defects.

RMS and kurtosis were used as features by Zhao *et al*. (2009) in verifying a neuro-fuzzy model that was designed for bearing fault diagnosis. The test results showed that the neuro-fuzzy model might be used as a reliable forecasting tool since it could capture dynamic behaviour and was able to track features of the systems.

Stepanic *et al*. (2009) investigated the application of statistical pattern classification and recognition for bearing fault analysis. There were two classes of bearing defects that were used in the classification, defective and functional rolling element bearings. Statistical parameters such as kurtosis, RMS, and variance were combined with characteristic frequency components as the features. Fault classification and recognition processes were carried out using a fault classification system constructed based on several “if-then” statements as proposed by Fukunaga (1990) and Thedoridis and Koutroumbas (2003). The results showed that the fault classification system achieved an accuracy of 97.45% for a horizontally placed accelerometer and up to 99.49% for the vertically placed accelerometer.

Latuny and Entwistle (2010) used changes in kurtosis values in investigating early indications of a bearing fault. Prior to the kurtosis calculation, a wavelet transformation was applied to a pre-processed vibration signal. It was found that a very early sign of bearing deterioration level was indicated by the changes in kurtosis value of the wavelet.

2.3 Frequency-Domain Analysis

Transformation of signals into the frequency domain is the principle on which the frequency-domain analysis depends. Frequency-domain analysis provides an ability to identify and isolate certain frequency components of interest which are useful for fault diagnosis. For example, in a bearing with a damaged raced, one would expect there to be a frequency component corresponding to a ball pass frequency. In this context, frequency-domain analysis provides features which give it an advantage over time-domain analyses.

2.3.1 The Fast Fourier Transform (FFT)

The Fast Fourier Transform (FFT) is the most widely used conventional analysis in the frequency-domain area. It is of critical importance in a wide variety of applications, from digital signal processing and solving partial differential equations to algorithms for quick multiplication of large integers (Brigham, 1988).

The Fast Fourier Transform (FFT) is used to produce a complex spectrum of a sampled vibration signal. It is used to calculate spectra power levels and phases of the vibration signal by which the spectra result has a frequency range from zero to half of the sampling frequency.

In addition to the FFT technique, it is acknowledged that other frequency domain techniques are available, especially the ones that using higher order mathematical functions. A summary of developed frequency-domain techniques is presented in Table 2.1 (Yang *et al*. 2003).

Table 2.1 - Developed frequency-domain Techniques (Yang *et al*. 2003)

First order	Second order	Third order	Fourth order
Spectrum (FFT)	Power spectrum, Power cepstrum, Cyclostationary	Biocoherence spectrum, Bi-linearity	
Signal averaging, correlation of spectrum, Short Time Fourier Transform (SFFT)	Spectrogram, Wigner distribution	Wigner bi-spectra	Wigner tri-spectra

Detailed discussion and examples of the others frequency-domain techniques can be found in Yang *et al*. (2003) and Jardine *et al*. (2006).

2.3.2 Application of the FFT in Fault Analysis

In the area of fault diagnosis, the Fast Fourier Transform (FFT) is the most widely used frequency-domain technique for spectral analysis to identify frequency-based features of bearing fault signals (Tandon and Choudhury, 2000), (Igarashi and Hamada, 1982).

An early application of FFT in bearing fault analysis was proposed by McFadden and Smith (1984), by which a study to obtain the spectrum of an envelope signal was established. Its purpose was to separate vibration information generated by a defective component from the vibration information generated by other machine elements. It was shown that the procedures for obtaining the spectra of envelope signals were well established. However, the use of the proposed technique was limited due to an incomplete understanding with regard to which factors controlled the appearance of the spectrum.

Variations of the FFT such as the Fast Fourier Transform (FFT)-based Hilbert Transform have been used to generate the envelope of signals for fault diagnosis, and

this has been widely used in bearing diagnostics (Randall, 1987). For example, Ho and Randall (2000) investigated the use of the squared envelope to improve the envelope's spectrum performance in separating the background noise from signals. Results showed that if the random spacing fluctuation of the bearing fault signal was less than 1%, then analysing the squared envelope gave an improvement if the mean-squared ratio (MSR) of the bearing signal to noise ratio in the demodulated band was greater than a factor of 0.2.

Subsequently, Randall *et al*. (2001) used a Fourier transform of the average squared envelope of the signal in diagnosing rolling element bearing signals. It was concluded that the squared envelope analysis was a valuable tool for the analysis of signals which have statistical properties that vary cyclically with time.

Lim and Su (2006) used FFT signal processing to convert a steady state vibration signal to a frequency domain in order to extract the vibration characteristics of an electric motor. The proposed system used a statistical method to generate a reference model. The resulting motor residual vibration was stationary and the RMS values were used to compute the fault indicator. The results were promising, and demonstrated the effectiveness of the proposed method for induction motor condition monitoring and fault detection.

Rai and Mohanty (2007) used the conventional FFT for the purpose of condition monitoring of a rolling element bearing. Based on the results, it was suggested that a better fault diagnosis would be achieved by using an advanced signal processing technique like the Hilbert-Huang Transform (HHT) with a FFT of intrinsic mode functions (IMFs).

Seryasat *et al*. (2010) applied the FFT to feature extraction in a ball bearing fault multi-fault analysis. In the study, the FFT of the vibration signal was used to generate features of FFT energy and root mean squares (RMS) of different frequency bands which were used to identify ball bearing faults.

2.3.3 Drawbacks of the FFT

The FFT has several drawbacks. The main one is that it is not capable of analysing the frequency content of a defective bearing signal since the signal is non-stationary and it is amplitude-modulated (Rai and Mohanty, 2007). The FFT is less sensitive in detecting inner race faults at the incipient stage, since the fault produces a highly attenuated vibration signal (Purushotham *et al*. 2005). Another disadvantage of using the FFT is the inability to provide any information about time durations within the spectrum, since the results are averaged over the entire duration of the signal (Peng and Chu, 2004).

In addition, the signal to be analysed using FFT must be periodic or stationary. If it is not, the resulting Fourier spectrum will not produce distinctive information related to the signal (Seryasat *et al.* 2010). Therefore, FFT has limitations in analysing vibration signals generated from a bearing since the frequency component of the signal changes over time and represents non-linear processes (Seryasat *et al*. 2010).

2.4. Time-frequency Domain Analysis

Another tool utilised is the combined time-frequency domain. One of the widely used tools in time-frequency analysis of vibration signals is the wavelet transform and in the following section, a brief discussion of wavelet transforms is presented. More detailed examples of description may be found in Mallat (1989) and Daubechies (1990), amongst others.

2.4.1 Wavelets Transform

Wavelet transform (WT) is a signal processing tool which employs what is called wavelets. A wavelet is a waveform of effectively limited duration that has an average value of zero (Mathworks, 1997).

The mathematical formulation of the wavelet transform was first proposed in the form of a new orthonormal function by Alfred Haar in 1909. This then led to the invention of the simplest orthogonal wavelet, which was later named the Haar wavelet ((Graps,1995), (Li *et al*. 2007)).

The idea of using a wavelet transform in signal analysis is to meet the need for analysis of signals using different sized scaling factors (window or scale). A large scale refers to a big frame or window for viewing or analysing a signal, while a small scale refers to a small frameor window used for viewing the details of a signal. In this context, changing from a large scale (window or frame) to a small is equal to the zooming process and vice versa.

In Fourier analysis, a signal is decomposed into sine waves of various frequencies. Similarly, in wavelet analysis, a signal is decomposed into shifted and scaled versions of what is called the original (or *mother*) wavelet. The aim of the wavelet transform is to overcome the shortcomings of the Fourier transform such as signal-cutting (windowing) problems, along with the analysing of non-stationary signals.

Wavelet transform uses a fully scaleable modulated window to solve signal-cuttingproblems. The window is shifted along the signal and for every position, the spectrum is calculated, as shown in Figure 2.1. The process can be repeated many times with a slightly shorter (or longer) window or scalein every new cycle. The end result will be a collection of time-frequency representations of the signal, all with different resolutions.

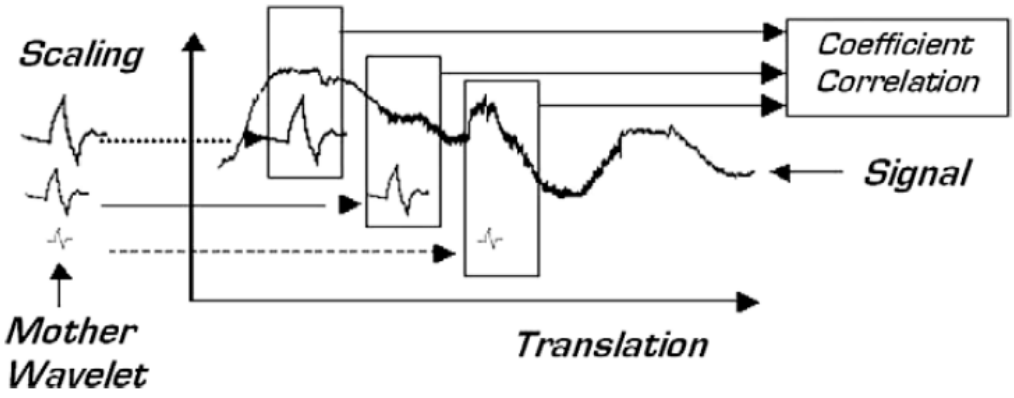

Figure 2.1 Wavelet transform principle (Saravanan and Ramachandran, 2010)

The mother wavelet with a defined scale size is translated from the beginning to the end of the signal to be analysed, as shown diagrammatically in Figure 2.1. The

process is iterated by determining a new scale of the wavelet function and the process is cycled again from the beginning to the end of the signal.

The process produces segments of signal which are comprised of "Approximated versions" and "Detailed versions" of the processed signal. This process is called the wavelet transform of the signal which provides the method for synthesising or disassembling the signal into two parts which are labelled approximation (a) and detail (d) parts.

The approximated version has a low frequency content which approximates the original of the processed signal, while the detailed version contains high frequency information on the processed signal.

The wavelet transform (WT) was proposed to address the limitations of the Fourier transform since the sine and cosine functions used in the FT are continuous and therefore not suited to particular needs. This is due to the fact that they are not localised functions and that they stretch out to infinity. In this context, if sine and cosine functions are used in the approximation of non-stationary signals the result is not satisfactory. In the WT, better results are achieved since the approximating functions used are limited to a finite time period (Graps, 1995).

The difference between the FT and the WT is at the ability of the WT to decompose signals using multi-scale analysis with dilation and translation in order to extract the time-frequency features or characteristics of the signal.

The wavelet analysis uses the wavelet function as the basic function to produce localised features of the original signal in a scaled domain (Paya *et al*. 1997). The basic functions comprise a family of functions which are derived from a single wavelet function called the mother wavelet.

The basic functions of the wavelet are useful in cases where there is a need to apply more suitable functions, other than sines and cosines, to approximate irregular signals or data with sharp discontinuities (Crandall, 1994). The basic function gives

the wavelet the capability of analysing the signal in localised time (or space) along with the frequency (or scale) domains (Al-Badour, *et al*. 2011).

An advantage of the wavelet transform is that it can be used to analyse signals in different frequency bands and to study each band with a resolution based on the wavelet scaling factor, as shown in Figure 2.1. If a longer scale of wavelet is used, the analysis produces low frequency information regarding a signal. The high frequency information is produced when a shorter wavelet scale is used (Graps, 1995).

There are many possible different types of wavelets; each of them is specified by its own coefficients. A common orthogonal set wavelet defined by Inggrid Daubechies (Daubechies, 1990) called the Daubechies *n* wavelet (abbreviated as db-n wavelet) is commonly used in various applications. An example of the practical implementation of wavelet transform using Daubechies wavelet is presented in Appendix 1.

The wavelet transform provides a time-scaled result of a signal which is different to the classical representation in the time-frequency used by FT.

In the process of wavelet transformation, the basic function (mother wavelet) is translated and dilated to provide improved time resolution for high-frequency information and simultaneously it provides limited time resolution for low frequency information.

The detail of time-frequency information of wavelet transform is depicted in Figure 2.2.

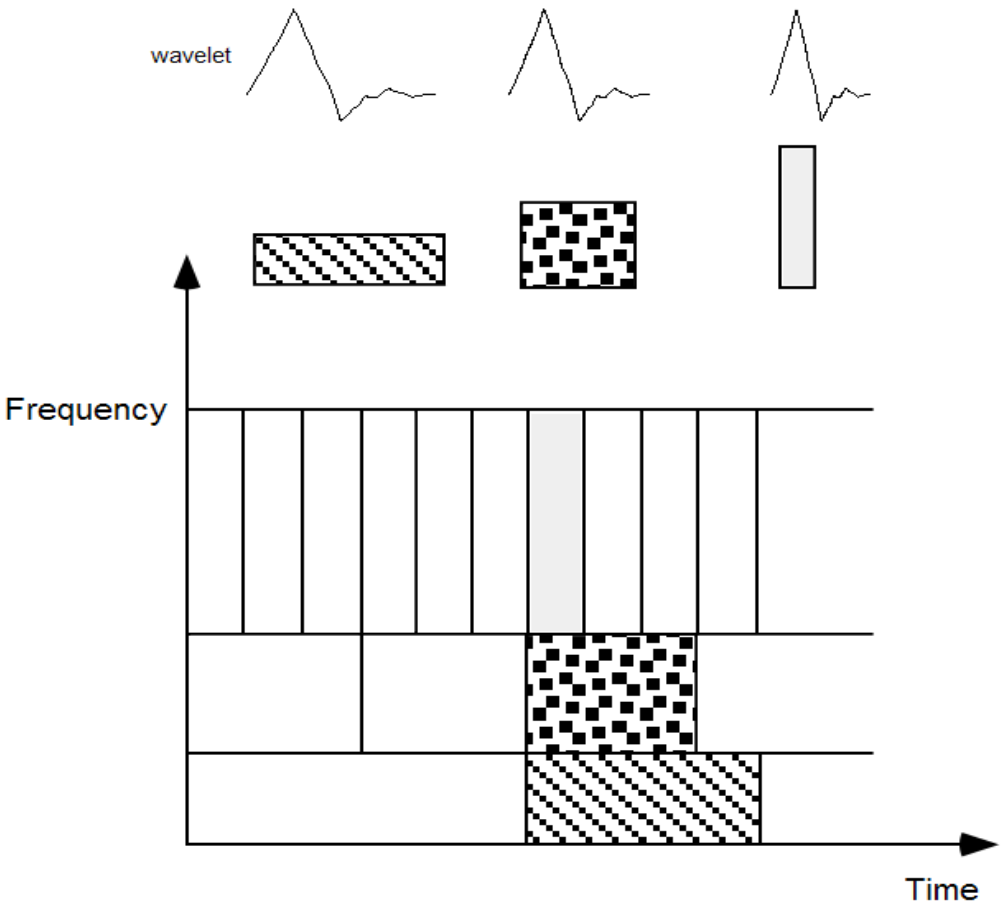

Figure 2.2 Daubechies Wavelet basis function, time frequency tiles and coverage of the time-frequency plane (Graps, 1995)

Figure 2.2 shows the coverage areas in the time-frequency plane with a Daubechies wavelet function. There are three different time scales of Daubechies wavelet, shown in the upper part of Figure 2.2. The shape of wavelet varies with the size of time-frequency coverage windows.

The Daubechies wavelet belongs to the family of orthogonal wavelets. An orthogonal wavelet is a discrete wavelet transform and it is defined by a maximal number of vanishing moments/points within a given support range. There is a scaling function for each wavelet type in this class or family which generates an orthogonal multi-resolution analysis. The functions of the Daubechies wavelet (Ψ) are shown in Figure 2.3 and the scaling (Φ) functions are shown in Figure 2.4.

The scale factor is an important parameter in the wavelet transform when used for processing vibration signals. As shown in the top section of Figure 2.1, there are three different Daubechies (db) wavelets with different scaling. Each of the wavelets has a particular time scale which produces differently sized time-frequency windows. Larger scale wavelets produce a "wider" time window but with a correspondingly low frequency coverage span. Smaller time scale wavelets produce a "narrower" time window but with a higher frequency coverage area. The number of transformation levels determines the frequency and time resolution segmentation. A higher transformation level will improve frequency resolution at the expense of decreased time resolution.

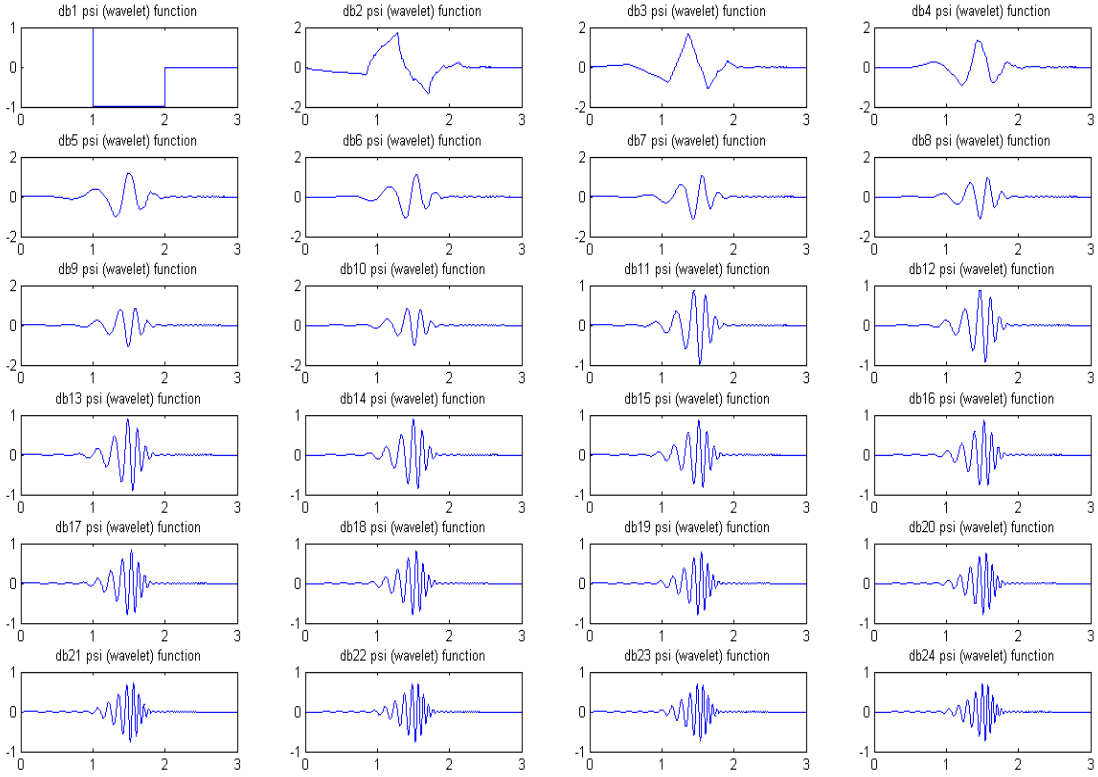

Figure 2.3 Daubechies wavelet (Ψ) functions

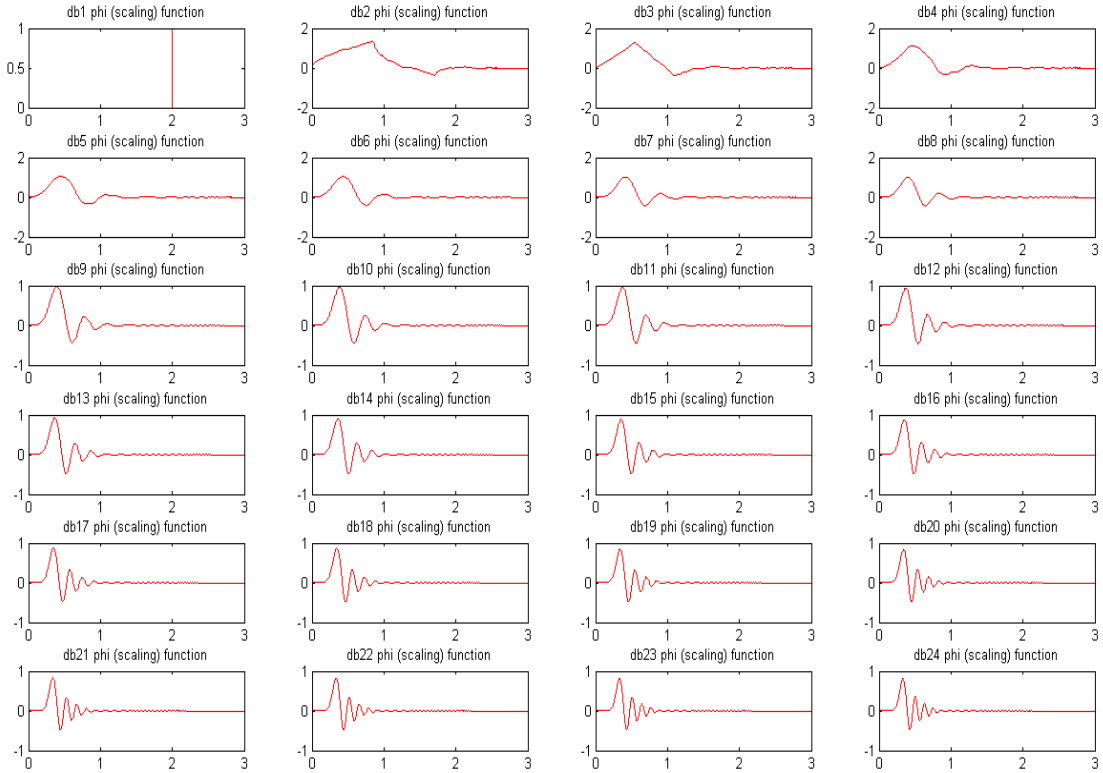

Figure 2.4 Daubechies wavelet scaling (Φ) functions

2.4.2. Wavelet Mathematical Notation

Wavelets are constructed in sets or families of functions and each of them is determined by a dilation process which affects the scaling parameters and the translation process. Translation involves the time when a single function, named the mother wavelet or analysing wavelet $\psi(t)$, is applied to the signal.

A wavelet $\psi(t)$ is a waveform which has a finite (and usually short) duration. It has an average zero value over time which is given by:

$$\int_{-\infty}^{\infty} \psi(t)\, dt = 0 \qquad (2.9)$$

It may be dilated by scale a and time-translated by factor b.

$$\psi_{(a,b)}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \qquad (2.10)$$

The parameter a is also called the scaling parameter and b is the time localisation parameter, while ψ is the "mother wavelet" function. The parameter for translation $b \in R$ and dilation $a > 0$ may be continuous or discrete.

The wavelet transform of f (finite energy signal $x(t)$) with the analysing wavelet $\psi(t)$ at scale a and position b is computed by correlating the transform with the analysing wavelet $\psi(t)$ or the convolution of $x(t)$ with a scaled and conjugated wavelet:

$$Wf(a,b) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{a}} \psi^* \left(\frac{t-b}{a} \right) dt \qquad (2.11)$$

where $\psi^*(t)$ stands for the complex conjugation of $\psi(t)$. W is the wavelet transform.

The wavelet transform $\psi_{(a,b)}$ can be seen as functions of translation b using scale a. Equation 2.10 shows that the wavelet analysis is a time-frequency analysis, or time-scaled analysis. The signal resulting from the wavelet transform $Wf(a,b)$ is presented in the $a-b$ plane, where a and b are used to determine the frequency and the time location of the wavelet in Equation 2.11.

In wavelet transforms, the analysing wavelet (basis function) traverses along the length of the processed signal as the value of b in Equation 2.10 increases. Local time span and frequency content information obtained from the processed signal is related to the increase or decrease of the a value in Equation 2.10 (Saravanan and Ramachandran, 2010). The chosen basis function (mother wavelet) is crucial, as it affects the resulting signal analysis to which the selected wavelet basis function is applied.

2.4.3 Continuous Wavelet Transform (CWT)

The continuous wavelet transform (CWT) is defined as the sum, over time, of the signal multiplied by scaled and shifted versions of the mother wavelet function (Peng and Chu, 2004). The CWT is defined in Equation 2.12.

$$c_f(a,b) = \int_{-\infty}^{\infty} x(t)\psi(a,b,t)\,dt \qquad (2.12)$$

where t is the time, a is the scale and b is the location or space parameter. The CWT is widely used in finding singularities in a signal which could be used to detect an impact fault in the signal (Al-Badour, 2011).

2.4.4 Discrete Wavelet Transform (DWT)

The discrete wavelet transform (DWT) was developed by Daubechies and Mallat through presentation of discrete versions of the CWT modulus of the wavelet transform (Mallat, 1989), (Daubechies, 1990). The DWT enables the application of the wavelet transform in signal analysis using its discrete form (Peng and Chu, 2004). In this context, the DWT is derived from the CWT by using discretisation of the wavelet $\psi(a,b)(t)$. The standard discretisation of the wavelet in the dyadic discretisation is given in Equation 2.13 (Mallat, 1989).

$$\psi_{(j,k)}(t) = \frac{1}{\sqrt{2^j}}\psi\left(\frac{t-2^j k}{2^j}\right) \qquad (2.13)$$

where 2^j replaces a and $2^j k$ replaces b (Mallat, 1989). The discrete wavelet method is needed in practical applications especially in the implementation of a computerised version of the wavelet transform.

A scaling function is used to generate an orthogonal multi-resolution analysis for each wavelet type in this family or class. The Daubechies wavelet function has been used in many applications which are intended to solve a broad range of problems, such as self-similarity properties of a signal or fractal problems and signal discontinuities (Saravanan and Ramachandran, 2010). Furthermore, the Daubechies and Meyer wavelets are the optimum wavelets for use in vibration signal analysis since the signals often have impulsive characteristics or form. These findings are based on the results obtained from simulation-based experiments aimed to select the best wavelets for vibration signals (Al-Badour, 2011).

2.4.5 Signal Decomposition and Reconstruction Using the Wavelet Transform

Signal transformation using wavelet transforms can be also interpreted as an application of a specific filtering process (Wu and Liu, 2009) in which a low and a high-pass wavelet filtering analysis is used in the filtering of a discrete signal. The concept of the filtering process is shown diagrammatically in Figure 2.5.

The basic filter function of a wavelet transform implementation is shown in Figure 2.5. The scheme can be applied from both directions, starting with either decomposition or reconstruction. Convolution is achieved by using a low pass filter *L*, and a high pass filter *H*, which are applied to the discrete signal *s*, producing two vectors cA_1 and cD_1. Elements of the vector cA are called *Approximation Coefficients* and the elements of the vector cD are called *Detail Coefficients*. In the decomposition process, the symbol (↓2) represents down-sampling, i.e., omitting the odd series elements of the signal. The process produces a number of elements in cA_1 and cD_1 which is approximately the same as the number of elements of the filtered signal, *s*.

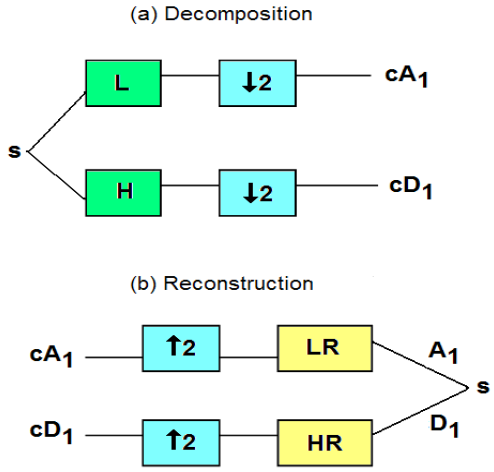

Figure 2.5 The wavelet transform using decomposition and reconstruction steps

In the reconstruction scheme, both vectors cA_1 and cD_1 are convolved with a pair of low-pass and high-pass reconstruction filters (*LR* and *HR*), respectively. The result of this process is two reconstructed signals, A_1 and D_1. These are known as *Approximation Coefficients* and *Detail Coefficients* respectively.

In the reconstruction process, an up-sampling process ($\uparrow 2$) is performed. This process involves the insertion of zeros between the elements of the vectors cA_1 and cD_1. The reconstruction of the decomposed signal s can be carried out by adding vectors A_1 and D_1 which produce a complete form of signal *s*. This step can be written as:

$$s = A_1 + D_1 \qquad (2.14)$$

The basic scheme can be performed iteratively on the approximation vector cA_1, and in sequence for each new approximation vector, cA_i, produced. The iteration process can be depicted as a wavelet tree in which i is the number of the iteration level. The number of coefficients for each vector A_i is approximately $ls/2^i$ where ls is the length of the signal, s and it covers a frequency band of $0 - Fs/2^{i+1}$, where Fs is the sampling frequency of the data. In general form, the reconstruction signals A_1 and D_1 match the equations;

$$A_{i-1} = A_i + D_i \qquad (2.15)$$

$$s = A_i + \sum d_j \qquad (2.16)$$

where i and j are positive integers (Nikolaou and Antoniadis (2002a)).

An example application of a wavelet transform used as a low and a high pass filter for filtering a noisy signal is shown in Figures 2.6 and 2.7.

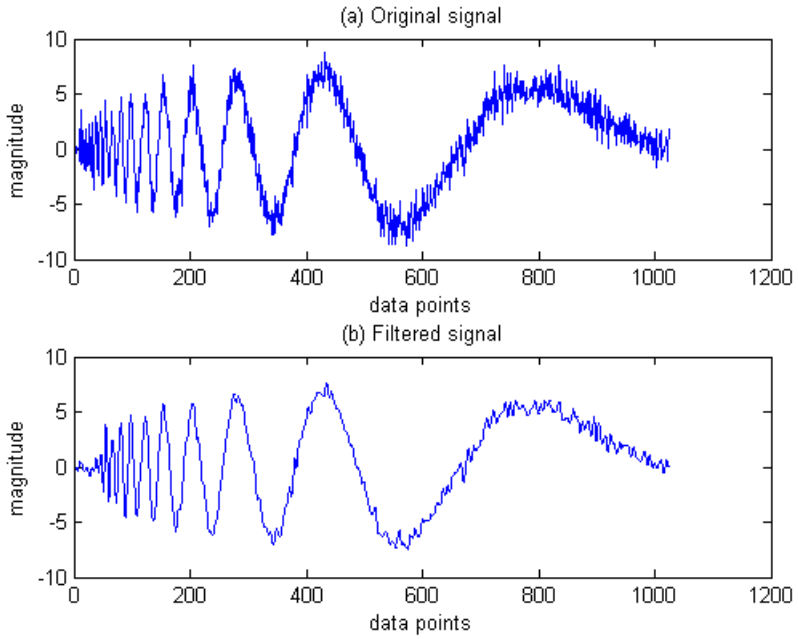

Figure 2.6 Filtering noise using wavelet transform (2 levels of decomposition)

A chirp signal that contains noise is processed (filtered) using Daubechies 4 (db4) wavelet up to two levels of decomposition. Figure 2.6a shows the original signal and Figure 2.6b shows the filtered or transformed signal. Noise is significantly reduced after the original signal is processed, using two levels of wavelet transform. Greater noise reduction is achieved by increasing number of level of the wavelet transform. Figure 2.7 (a) and (b) show the results of wavelet filtering of the same original signal obtained after five levels of wavelet transform.

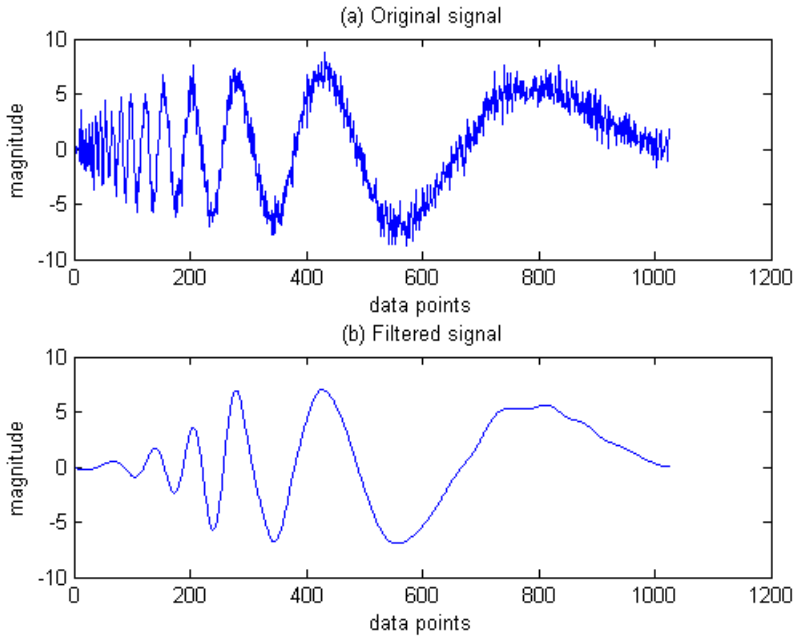

Figure 2.7 Filtering noise using wavelet transform (5 levels of decomposition)

Figure 2.7b shows a clean signal which is obtained by increasing the number of decomposition levels of the wavelet transform used to process the signal.

2.4.6 Multi-Resolution Analysis (MRA)

The DWT is used to process or transform a signal using Multi-Resolution Analysis (MRA). The scheme of DWT MRA decomposition of a signal up to 5 levels of decomposition is shown in Figure 2.8.

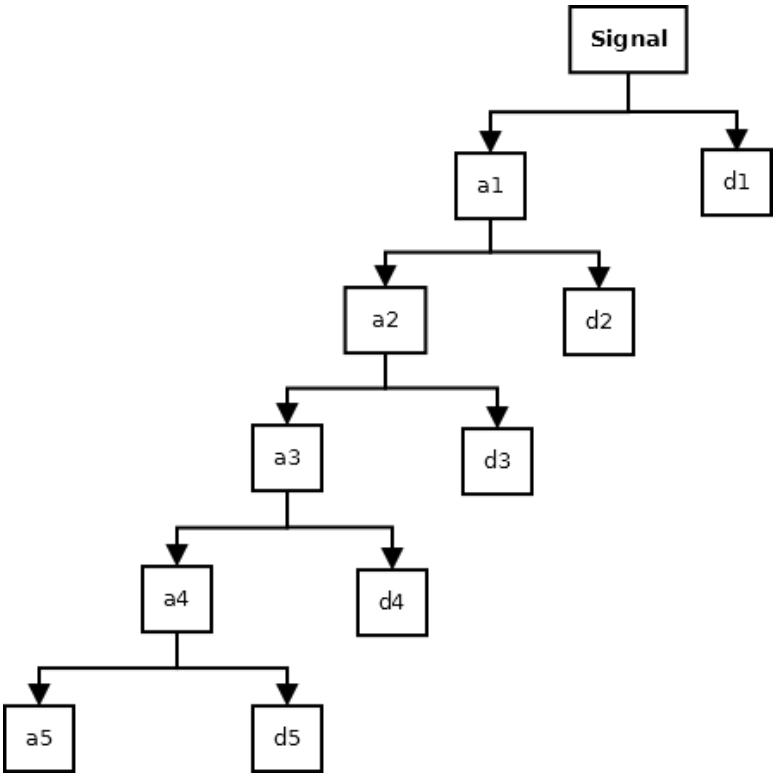

Figure 2.8 Signal decomposition scheme using 5 levels DWT MRA (Reda Taha *et al*. 2006), (Wu and Kuo, 2009)

The MRA scheme shows decomposition of a complete signal by using DWT which results in Approximated (a) and Detailed (d) parts. The corresponding index (a1 to a5) represents decomposition level at the related level.

The frequency sub-bands of a signal processed by a wavelet transform have standard divisions for each approximation (a) and detail (d) parts at each level of decomposition (Wu and Liu, 2009), (Bin *et al*. 2012).

An example of the MRA scheme is shown in Figure 2.8, using a signal with a maximum frequency span of up to 3000 Hz. If the signal is processed using five levels of MRA then the sub-bands division of *a* and *d* parts from levels 1 to 5 are given in Table 2.2.

Table 2.2 Frequency sub-bands of the DWT of five levels MRA results

Approximation	Sub-bands (Hz)	Detailed	Sub-bands (Hz)
a1	0 – 1500	d1	1500 – 3000
a2	0 – 750	d2	750 – 1500
a3	0 – 375	d3	375 – 750
a4	0 – 187.5	d4	187.5 – 375
a5	0 – 93.5	d5	93.5 – 187.5

At the first level of wavelet transform, the signal is transformed into two parts, namely *a1* and *d1*. The frequency sub-bands coverage of *a1* is from 0 – 1500 Hz and for *d1* from 1500 – 3000 Hz. The frequency sub-bands coverage for the subsequent levels of decomposition are given in Table 2.2.

Detailed results of the MRA implementation, using a signal corrupted with noise as an example, are shown in Figures 2.9 and 2.10.

Figure 2.9 shows the approximated (a) parts of the processed signal after it has been processed using 5 levels of wavelet transforms. Note that the amount of noise is gradually reduced from level 1 to level 5, as shown.

Figure 2.10 shows the detailed (d) parts of the processed signal after it has been processed using 5 levels of wavelet transforms. The detail parts contain high frequency components of the original signal. The high frequency parts of the original signal are related to the noise frequencies which contaminated the signal and the signals were filtered by a high-pass filter in the wavelet transform.

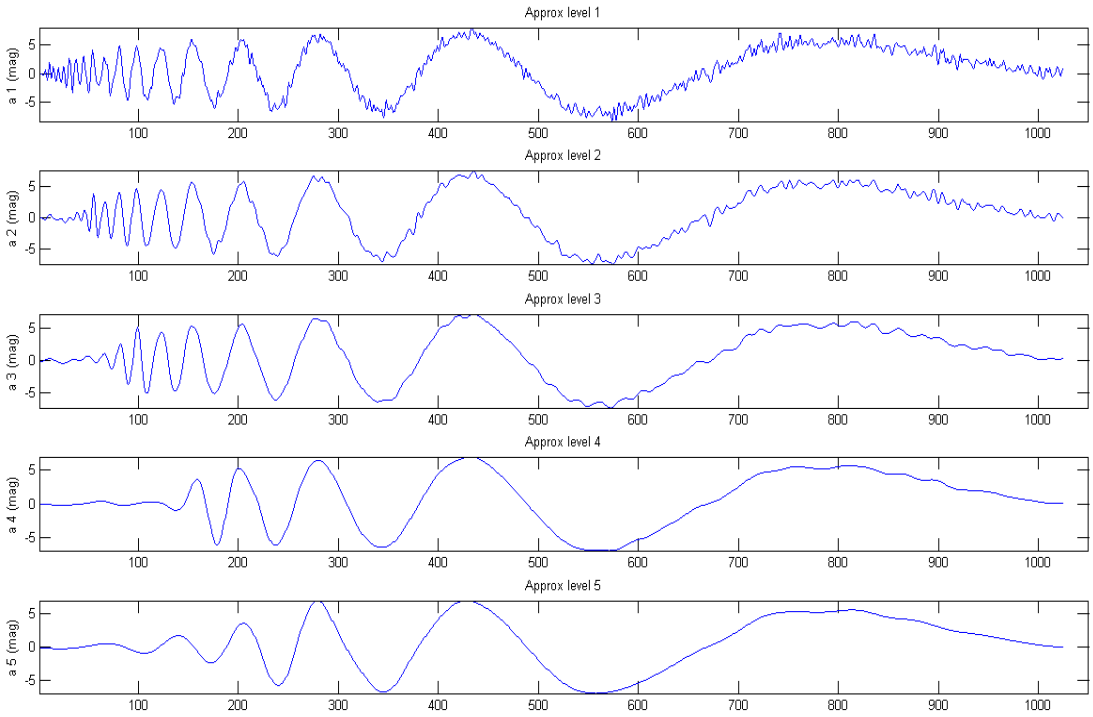

Figure 2.9 Approximated parts of signal with noise (a1 - a5)

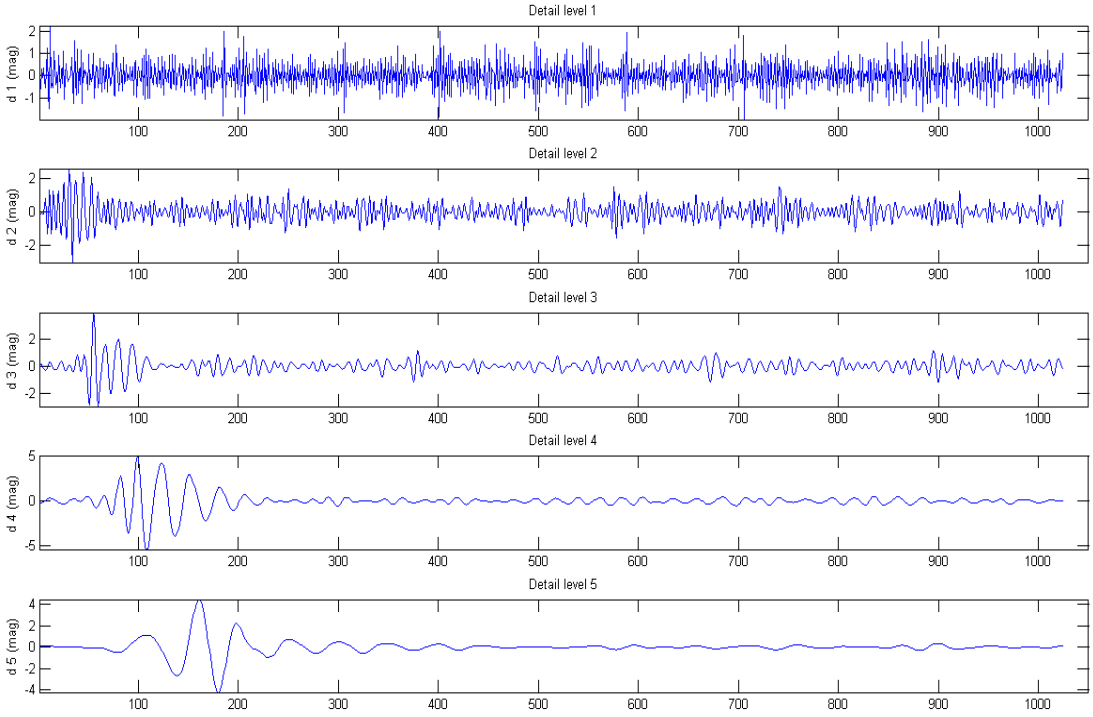

Figure 2.10 Detailed parts of signal with noise (d1 - d5)

2.4.7 Energy Level of Wavelet Results

The energy levels from the wavelets transform feature in fault analysis and have been utilised in fault diagnosis (Peng and Chu, 2004), (Yan and Gao, 2009) and (Seryasat *et al*. 2010). Wavelets been widely used due to their effective energy concentration properties which can be used as fault indicators. They are also advantageous by virtue of their compact support of wavelet based functions (Peng and Chu, 2004).

The energy level density analysis method has been applied to roller bearing fault diagnosis according to the characteristics of the fault vibration signals. Energy level density is an important physical variable in a signal. It represents a signal characteristic since energy level density of the signal is varied and distributed along the change of time and frequency domain (Junsheng *et al*. 2005).

Gaing (2004) employed Parseval's theorem to extract the energy distribution features from DWT results. The energy features were calculated using Parseval's theorem which states that the summation result of a signal $x(t)$ in the time domain components along its whole time span (t) is equal to the summation result in the frequency domain of all of frequency components (f) of the Fourier transform (FT). The method was used to propose a prototype of wavelet-based neural-network classifiers for the purpose of power disturbance recognition and classification. The experimental results showed that the proposed method had the ability to recognise and classify different power disturbance types efficiently. It was also shown to have the potential to enhance the performance of the power transient recorder with real-time processing capability.

Wu and Liu (2008) used Parseval's theorem, with energy calculated from DWT results, in investigating engine fault diagnosis and Wu and Kuo (2009) used the theorem in generating energy features for the purpose of automotive generator faults analysis. In their research, the energy level distribution features of Approximated version P_a and Detailed version P_d of DWT results were calculated using the approximation and detailed parts of the wavelet transform results.

Using Parseval's theorem energy calculation proposed by Gaing (2004), Wu and Hsu (2009) applied the method to gear-fault identification based on vibration signals. Vibration signals were processed using DWT and the energy features of the DWT results were calculated. The energy features were implemented to reduce the number of DWT data involved in the analysis. The features generated from DWT results were used to identify faults by using Fuzzy-logic inference. The experimental results indicated that the proposed system was effective in increasing accuracy in gear-fault-identification of the gear-set platform.

Another application that utilised the energy level of wavelet coefficients was also conducted by Yan and Gao (2009). It was found that by using a specific wavelet scale, wavelet coefficients could be used to identify defect-induced vibration patterns that were embedded in the vibration signal. Based on this concept, selection of the base wavelet can be guided by comparing the amount of energy extracted from the signal being analysed by different base wavelets. The wavelet that extracts the most energy from the signal will represent the most appropriate wavelet for defect-induced vibration analysis.

A modified version of the wavelet energy calculation, based on the original version presented by Goswami and Chan (1999), was used by Latuny and Entwistle (2010) in analysing vibration signals from bearings under a lifecycle test. In the analysis, the energy level of Detailed parts (cD) of the wavelet transform was calculated via Equation 2.12.

$$Energy\ level\ D_i = \left(\frac{1}{2}\right)^{p-i} \times \sum (D_i)^2 \qquad (2.12)$$

where:

p = next power of two based on data length,

i = decomposition level ($i = 1, \ldots, n$)

D = Detailed (cD) result of wavelet transform at i-th level.

By using the variations in energy levels, it was shown that early deterioration of bearings could be detected. The results were much improved compared to using changes of kurtosis to monitor early deterioration of bearings under life test.

2.4.8 Application of wavelets in Fault Analysis

The application of the wavelet transform to vibration signal analysis as a feature extraction method in machine fault detection has been widely investigated by many researchers. The following section provides a brief discussion of wavelet applications in fault diagnosis.

Newland (1994a) and Newland (1994b) presented the basic theory and methods of wavelets in a more systematic manner, and provided examples of how to use the wavelet in the analysis of vibration signals. These presentations contributed to wavelet transform becoming a preferred method in vibration signal analysis. The applications in vibration signal analysis for the purpose of machine fault analysis were undertaken by Peng *et al*. (2010).

An earlier suggestion to make use of wavelets in fault diagnosis was proposed by Pan and Sas (1996). They highlighted the importance of analysing non-stationary signals since these signals usually contain abundant information on machine faults. It was found that analysis of non-stationary signals and data with sharp discontinuities was best implemented using the wavelet transform since it provided better results.

Wang and McFadden (1996) investigated the applicability of wavelets in diagnosing mechanical faults, particularly in gearbox vibrations. It was found that the wavelet could be used to detect incipient mechanical failure and different types of faults at the same time.

A study by Liu *et al.* (1997) on ball bearings fault analysis implemented a wavelet packet-based method to generate wavelet coefficients. It was used as a feature extraction method for fault diagnosis and showed a high sensitivity to faults.

Wavelet analysis of time-domain sound signals generated by bearings was carried out by Shibata *et al*. (2000). The investigation produced a component of the signal that can be of use for fault indication.

The implementation of a method to obtain a better signal to noise ratio (SNR) of a low speed rolling bearing signal by using discrete wavelet packets for multiple band pass filtering was carried out by Altmann and Mathew (2001). It was found that the filter using the wavelets significantly improved SNR compared to its other high-pass counterpart.

An application of wavelet transform modulus maxima to detect sudden changes in vibration signals originating from bearings related to faults was investigated by Sun and Tang (2002). The outcome showed that the proposed method was able to significantly discriminate the noise from the signal. The method also showed an advantage in determining the severity of the bearing damage in qn operating condition where load and speed were varied.

Rubini and Meneghetti (2001) investigated applications of wavelets in fault analysis of double-row self-aligning ball bearings. The investigation used Gaussian (Dalpiaz and Rivola, 1997) and Morlet type wavelets (Chui, 1992) in order to construct an alternative method for the use of envelope analysis in the time domain. It was found that the wavelet transform was more suitable than the envelope analysis in extracting the impulsive effect due to short and low impacts buried in the signal noise.

Investigation of bearing faults using wavelet transforms was carried out by Prabhakar *et al*. (2002) in order to explore the capability of the wavelet transform in extracting impulses from the faulty ball bearing signal. Daubechies wavelet type 4 (db4) was used in the investigation. The original vibration signal was processed using the wavelet transform and the high frequency parts (detailed parts) of the wavelet transform results were investigated. The investigation results showed that periodic impulses related to bearing faults were enhanced and better presented in higher level parts of Discrete Wavelet Transform (DWT). The results showed that the DWT can be used effectively in detecting single and multiple faults in ball bearings.

2.5. Concluding Remarks

Vibration measurement is a common method used in fault diagnosis applications. In analysing vibration signals for the purposes of fault diagnosis, there are three categories of analysis that are commonly used. These categories are time-domain analysis, frequency-domain analysis and time-frequency domain analysis.

The frequency domain method is a popular method (Patil *et al*. 2008), in which the FFT method is the dominant processing tool in use (Seryasat, 2010). In addition, it has been found that the envelope analysis and amplitude demodulation of vibration signal have a wide range of applications in fault diagnosis (Randall and Antoni, 2011) and Luo *et al*. (2013).

Chapter 3 – Review of Artificial Intelligence (AI) Systems in Fault Diagnosis

The need to implement intelligent systems in fault diagnosis has become an essential requirement as industrial maintenance processes become increasingly complex. This complexity is a result of sophisticated developments in machinery systems along with the industrial processes involved. The amount of work now required to provide fast and reliable fault diagnoses has surpassed the capability of human operators to manage these processes in a timely manner.

In addition, due to the complicated nature of modern machinery, there is also an increased risk of failure which requires effective and efficient problem solving techniques. Dealing with machinery failure is an important factor in the accommodation of increasing demand for high quality products, production and cost reduction, timely availability of machinery, reliability of production equipment and most importantly, greater safety requirements in all elements of the production line.

These factors have become the motivation for the utilisation of intelligent diagnosis models in identification of faults in machinery systems.

Final condition identification is another task in fault diagnosis of rotating machinery. Diagnosis is mainly carried out by using artificial intelligence (AI) techniques. In this case, the AI techniques are used to provide an automated fault diagnosis process. AI method applications are mainly characterised by the implementation of neural-network-based systems.

In general, the building blocks of an intelligent fault diagnosis system consist of the following: data collection section, feature extraction section and fault detection and identification sections. The fault detection and identification section may employ the following systems: artificial neural networks (ANN), Fuzzy Logic/Fuzzy Sets, Expert Systems and Hybrid Artificial Intelligence Techniques.

The following sections briefly discuss the most widely used AI techniques in fault diagnosis.

3.1 The Artificial Neural Networks (ANN)

Perhaps one of the most widely used Artificial Intelligence techniques in fault diagnosis is the Artificial Neural Network (ANN) and its variants (Haykin, 1999).

Artificial Neural Networks, a major component of neuro-computing (Rao *et al*. 2012), were first explored by Rosenblatt in 1959 (Widrow and Lehr, 1990) and Widrow and Lehr (1960).

The ANN consists of interconnected groups of artificial neurons. These neurons use a mathematical or computational model for information processing. ANN is an adaptive system that changes its structure based on information that flows through the networks (Haykin, 1999).

The ANN learns patterns from a training data set that represents the relationship between inputs and outputs by the use of a learning method (Rao *et al*. 2012). An example of this is seen in a back propagation type of learning algorithm, first introduced by Werbos in 1974 (Widrow and Lehr, 1990).

The ANN comprises a layer of input nodes, one or more layer of hidden nodes and one layer of outputs. Each layer consists of processing elements called ‘nodes’ or ‘neurons’ and interaction between them is achieved using numerically weighted connections (Tsoukalas and Uhrig, 1997).

The intelligent diagnosis system which employs Artificial Neural Networks is generally based on the utilisation of a data processing system that comprises a number of simple and highly interconnected processing elements. These highly interconnected elements or nodes are arranged in an architecture that mimics the cerebral cortex in the human brain. The system is used in intelligent diagnosis by supplying known inputs in order to obtain an output that represents the machine fault characteristics.

An unknown function can be mapped through an iterative evaluation and adjustment of the connecting weighting values used by the ANN. In this context, the ANN can be used in the modelling of complex non-linear problems which may be implemented to approximate practical problems in fault diagnosis cases. In practical diagnosis applications, the ANN is trained to recognise the features extracted from recorded or acquired signals. During the training process, the ANN is instructed to mimic the input – output relationship of the data or information. That is, the ANN is trained to memorise the relationships which will enable it to later be used as a classifier where it is designed to classify input (data/information) patterns and related input(s) to the known output(s).

3.1.1 Advantages of ANN

There are several advantages to ANN which make it suitable for implementation in a wide range of applications.

Firstly, ANN is not dependent on prior principles, statistical data or models. Hence, it can be used directly in analytically-difficult modelling tasks which cannot be solved using conventional methods or approaches. In these cases, the use of conventional methods is not practical as the modelling tasks have non-linear characteristics, high-order, time-based dynamics variation and data input-output relationships which have no existing analytical model.

Secondly, the ANN provides an improvement to the fault tolerance capabilities of fault diagnosis systems through its adaptive features in which it has self-adjustment capabilities which compensate for the changes in data over time.

These advantages make ANN suitable for intelligent technique applications in fault diagnosis and identification which would otherwise involve modelling tasks and conventional analytical methodsthat are impractical.

Furthermore, the construction and the methods of information storage and information manipulation within ANN make it suitable for ongoing learning. This learning feature can be used to recognise patterns even if the information that makes up the pattern is noisy or incomplete. This is also known as the ANN feature of data

matching in high-dimensional spaces which the employs the effective interpolation and extrapolation ability of the ANN based on the learned data. Establishment of a complex regression function between a set of input-output data can be obtained by using the ANN training process.

For the purposes of fault diagnosis, an ANN is applied by using appropriate input data, in order to learn the data and produce fault information or characteristics which represent data features.

The ability to map input-output relationships through learning processes using presented data has made the ANN one of the most widely-used and preferred methods in intelligent fault diagnosis of machinery systems. The learning features have motivated a wide range of ANN applications in fault diagnosis where they are used to handle such problems using existing data obtained from machinery parts under investigation. In this case, the features extracted from acquired vibration signals from machinery related to a known fault condition are fed into an ANN for training. As a result of learning the system's features, the ANN can determine most of the related or specific characteristics embedded in the data, and the specific characteristic(s) can be used as a signature associated with the fault condition. The ability of an ANN to map input-output data by its training process allows it to be used as an on-line pattern recognition system useful in the fault diagnosis process.

3.1.2 Learning Methods

In the application of ANN, there are two learning or training methods. The first is supervised learning and the second is unsupervised learning (Peng, *et al*. 2010). Supervised learning is where complete examples of data or input-output pairs are presented to the ANN during its learning session (Haykin, 1999). The presented data contains the corresponding known target or output (Carpenter *et al*. 1992). Unsupervised learning is where there are no output (target) values and the learning task aims to obtain the characteristics of the process that generated the data (Kohonen, 1997), (Haykin, 1999).

The training and testing steps are two important procedures in the application of an ANN system. The learning step plays a very important role since it is here that the

ANN gains its useful abilities. The testing step is the process by which the inputs with known features are fed into the ANN to test its performance and to perform network weighting calculations in order to produce values for the last neuron stage. These values are then compared with the targeted output in order to verify the suitability of ANN design.

3.1.3 Types of ANN

There are several types of ANN (Jang *et al*. (1997)). They are:

- Back Propagation for Feed Forward Network (BPFFN)
- Multi Layer Perceptrons (MLP)
- Back Propagation Multilayer Perceptrons (BPMP)
- Radial Basis Function Networks (RBFN)
- Self Organized Map (SOM)

All of these types of ANN have been implemented in various fault diagnosis applications.

3.1.3.1 Back Propagation Feed Forward Network (BPFFN)

A typical Back Propagation Feed Forward Network is constructed from several series or successive layers of neurons. In the example shown in Figure 3.1, a typical 3 layer back propagation (BP) feed forward neural network (NN) construction is illustrated. x_1, x_2, … x_n are networks inputs, y_1, y_2, …, y_m are outputs. w_{ij} represent the connection weightings input layer of neural cell i and the hidden layer neurons layer y, and v_{jt} is the connection between hidden layer neurons j and the output neurons layer. Typically, sigmoid-type neurons, neurons with differentiable functions such as the hyperbolic tangent function (Haykin, 1999), are used in the layers as the transfer function of the BP network.

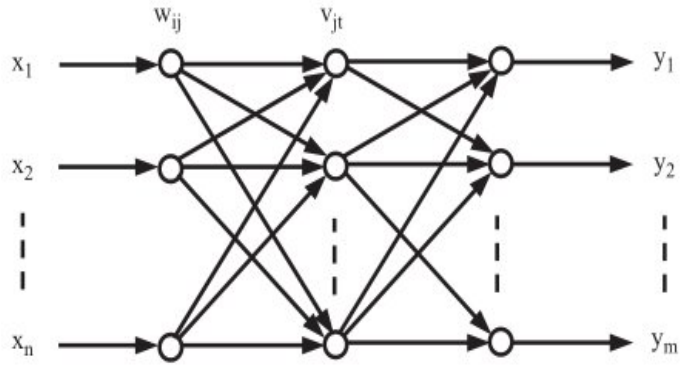

Figure 3.1 Typical three layers of BPFFN network
(Galushkin, 2007), (Bin *et al*. 2012)

Figure 3.1 show the layer configuration of the sigmoid-type neurons which are the building blocks a Feed Forward Neural Network where every neuron in a layer receives inputs from the outputs of all the neurons of the preceding layer.

The term back-propagation is applied to this type of ANN, as during the training or mapping of the input-output relationships of the data, the error is propagated backwards through each internal node. The error information is then used to calculate the weighting adjustment for the corresponding node. This calculation process is the core of the training process during which the information is forwarded from the input layer to the output layer and weighting values are changed until the error value is reduced to an acceptable limit.

The multilayer construction that fully interconnects the feed forward (FF) network uses the 'delta rule' to compute the weights between the actual output and the desired output. The desired output is optimised through a least square approach (Alguindigue, 1993). The basic algorithm for back propagation is presented in Rumelhart and McClelland (1986).

The realisation of the arbitrary non-linear mapping between the input and output by using FF a network is useful in areas such as pattern recognition, function approximation, and data compression. The mapping of input data output data is achieved using a multi-layered FF network. This mapping process is carried out by changing the connection weight of each neural neuron in a network. The change of

weight values aims to ensure that the output generated is consistent with the anticipated one. The process of weight value modifications is known as the training process of the network. The BP NN can be implemented to recognise a non-linear relationship between fault types and fault characteristic parameters of bearings (Bin, *et al*. 2012).

Use of either multi layer perceptron (MLP) networks or Radial Basis Function (RBF) networks for the common application of FF networks is optional (Meireles *et al*. 2003). They are used to map given sets of data points (input-output) using interpolation methods. For the purposes of pattern recognition, adoption of an MLP is preferred since it has a function which produces numerical results of 1 or 0 which are suitable for classification purposes.

In the design of the ANN, determination of the number of processing element within the input-output layers is generally based on the number of variables that are used as input and output entities. The determination of the number of processing elements for the hidden layer is based on the complexity of the problem. Several design criteria for the number of hidden layers of the ANN are presented in Kung and Hwang (1988).

Multi-layer FF networks can be used to map the input and output relationship pattern that exists in the data. Inputs are received by the input layer and are then modified based on the set of weights. The modified results are then sent to the hidden layers. Each hidden layer propagates the modified inputs to the subsequent layers before the modified inputs reach the output layer. The calculation of overall error takes place in the output layer.

3.1.3.2 Recurrent or Recirculation Neural Network (RNN)

The recurrent or recirculation neural network(RNN) and the generic FF networks have similar characteristics, except that the RNN employs additional feedback connections that delay and store information from previous steps (Wang *et al*. 2004b). The application of these feedback connections means that the training process of an RNN is carried out in cycles. The training is executed iteratively which takes longer for results to be obtained. In general, the RNN has fixed connection

weights of 1, and the dynamic response is achieved by delaying inputs and outputs (Evsukoff and Gentil, 2005). The structure of an RNN is shown in Figure 3.2.

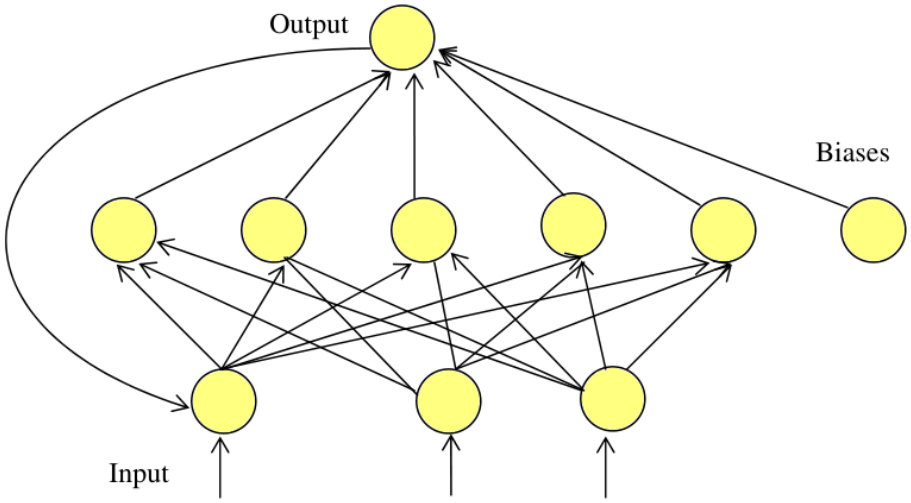

Figure 3.2 Recurrent Recirculation Neural Network structure (Graupe, 1997)

3.1.3.3 Self-Organizing Map (SOM)

The Self-Organizing Map (SOM) network comprises a forward two-layer network that employs a non-linear projection characteristic that maps the input signal, which is of arbitrary dimension, to a one or two-dimensional array of (neurons) nodes in which the array of nodes is related to a discrete map (Kohonen, 1997).

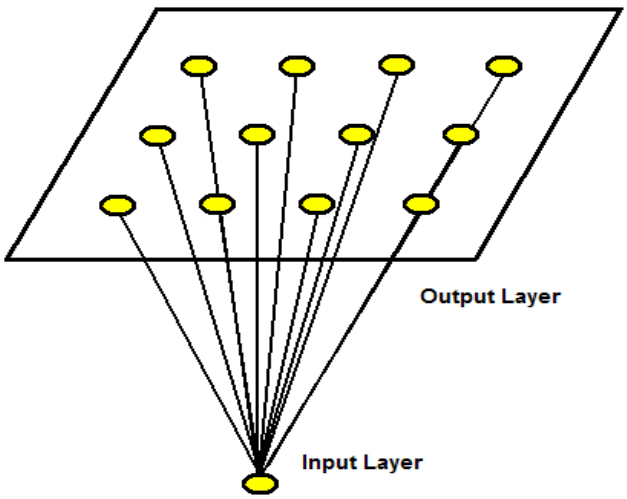

Figure 3.3 Self-Organizing Maps (SOM) (Haykin, 1999)

Figure 3.3 shows the basic configuration of the SOM network which consists of a two layer neural network using full connections between neurons in the output layer.

The SOM provides a mapping process which is based on the sequence of a high-dimensional distribution data on a regular or simplified low dimensional grid. This feature imparts to the SOM the ability to translate complex, nonlinear statistical relationships between data with high dimensions into a simple geometric relationship on a low-dimensional space or simplified graph.

The simplification of relationships is carried out while it preserves the most important structure and metric relationships of the data. This process can also be seen as a type of abstraction. The features of an SOM that provide abstraction and visual information of dimensional data can be used in a number of ways for applications that contain complex tasks such as process monitoring, process analysis and fault diagnosis (Kohonen, *et al*. 1996), (Zhong *et al*. 2005).

As an example, the Kohonen SOM network consists of three major learning step attributes: competition, co-operation and adaptation (Yang *et al.* 2004). In the competition step, the network compares and competes with other neurons based on the output values for a given input vector, modified by a chosen discriminating function. The discriminating function is later used to determine which output is closest to an input pattern. The competition step produces a selection among the output neurons. The neuron with the closest relationship to the input vector will be picked up and labelled as the winning (best-matching) neuron. The cooperation step is where the selected winning neurons are used to predefine a neighbourhood or group of neurons. This will provide the basis for the neighbouring neurons to cooperate by which only the weights of those neurons defined within the topological neighbourhood of the winning neuron will be updated or changed. The synaptic weighting, strength of a connection between two neurons, of neurons outside the neighbourhood will remain unchanged. This is followed by the adaptation step where the winning neuron within the group constantly changes its weight value to adapt to the values of the inputs pattern. This learning strategy provides the ability to evolve the synaptic weight vectors towards the distribution of the input vectors.

3.1.3.4 Radial Basis Function (RBF)

The radial basis function (RBF) network is a forward network with three layers: an input layer, a hidden radial basis layer and an output linear layer (Lei *et al*. 2009). The input neuron information is transferred to the neurons in the hidden layer. The RBF in the hidden layer responds to the input information, and the network outputs are then generated in the neurons of the output layer.

The RBF was first applied to neural networks by Broomhead and Lowe (1988). The RBF NN is a feed forward network which comprises three layers: an input layer, a hidden radial basis layer and a linear output layer. Information received by neurons in the input layer is sent to neurons in the hidden layer. The RBF neurons in the hidden layer respond to the input information and the output layer generates output.

The advantage of the RBF network is that the hidden neurons will produce non-zero outputs if the outputs values are within the minimum limit of the input values pre-defined range. Otherwise, the output will be zero. This network feature makes the number of active or used neurons small and the time required in training the network is shorter (Lei, *et al*. 2009).

3.2 Disadvantages of ANN

While ANNs have advantages, they also have disadvantages. Firstly, an ANN applied as a diagnosis tool acts as a black-box, meaning that it is difficult to deduce physical explanations for its output. Secondly, the training process becomes more complicated and time consuming once the size of an ANN increases. Thirdly, there is an issue in determining the number of hidden layers to be used along with the number of nodes in each layer (Brotherton, *et al*. 2000).

Finally, the ANN is lacking the semantics feature needed to process the imprecise and vague nature of information that is characteristic of human reasoning. In this context, the basic form of neural networks cannot be used to accommodate the use of expert knowledge gained from observation and learning of a physical process (Liu and Shi, 2001).

3.3 Applications of ANN in Rotating Machinery Fault Diagnosis

Artificial neural networks are currently the most commonly found data-driven techniques in fault diagnosis literature (Heng *et al.* 2009) and they have been widely implemented in the fault diagnosis of rotating machinery. Examples of ANN applications in fault diagnosis are listed in Table 3.1, where applications to common machine groups are reported.

Table 3.1 - Neural Networks Applications in Fault Diagnosis of Rotating Machinery

Neural Networks	**Motors**	**Pumps**	**Bearings**	**Turbines**	**Gears & Gearboxes**	**Shafts**
BPFFN	Mahamad and Hiyama (2011) , Li *et al.* (2004)	Ilott and Griffiths (1997)	Li and Wu (1989), Liu and Mengel (1992), Samanta and Al-Balushi (2003), Gebraeel *et al.* (2004), Sreejith, *et al*. (2008) Wang *et al*. (2010), Prieto *et al*. (2013)		Yang *et al.* (2008), Wu and Chan (2009)	McCormick and Nandi (1997), Kuo*et al*. (2002)
RNN			Yam *et al.* (2001), Malhi *et al.* (2011)	Mohamm-adi *et al*. (2011)		
RBF	Selaimia (2006), Onel and Benbouzid (2008)	Lu *et al.* (2011)		Rong *et al.* (2009)	Li *et al.* (2009)	
MLP	Paya *et al.* (1997)		Meesad and Yen (2000), Senguler *et al.* (2010)		Meesad and Yen (2000)	Meesad and Yen (2000)
Kohonen	Kowalski and Kowalska (2003), Bay and Bayir (2005)					
SOM	Yang *et al.* (2004), Premrudee preechach arn *et al*. (2002)		Zhang and Ganesan (1997), Zhong *et al*. (2005),	Hu *et al.* (2003), Wu *et al.* (2002), Donat *et al*. (2008)	Liao *et al*. (2005)	
LVQ			Zarei*et al*. (2008)		Abu-Mahfouz (2005)	Meesad and Yen (2000)

Several selected applications listed in Table 3.1 are explained below.

ANN systems have been used to support classification of fault analysis. An earlier application of ANNs in the field of bearing fault analysis can be found in Li and Wu (1989). In the investigation, a perceptron-type network was used to analyse the

experimental data from ball bearings. The results showed that the network recognised faults made on the outer race of the bearings with a percentage of error smaller than the one of the conventional methods. It was reported that the proposed technique achieved a 14 per cent better rate than the conventional methods.

In an investigation by Liu and Mengel (1992), it was shown that the perceptron network was capable of distinguishing between six different cases of ball bearing faults. The fault detection used the variations of the peak amplitude in the frequency domain, the peak RMS and the power spectrum parameters as the training data for the perceptron network.

Baillie and Mathew (1994) diagnosed rolling element bearing faults using artificial neural networks and a bearing fault diagnostics system was developed. The incoming vibration signal was presented to each neural network model in the system and the network model that best approximated the signal was chosen to indicate a type of fault. The system was trained to diagnose fault conditions such as imbalance, outer race faults, inner race faults and normal conditions. The neural network-based diagnostics system was tested and it was shown that the system achieved accuracy in 95 per cent in all the test data set.

Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) neural networks utilised to classify the condition of rotating machines were used by McCormick and Nandi (1997). In the classification tasks, the findings were that similar success rates were achieved by both MLP and RBF networks. Further detail shows that the RBF networks needed significantly shorter time for training compared to the time needed by the MLP network. However, the MLP network achieved faster operation time and used fewer neurons.

Application of MLP and Learning Vector Quantization (LVQ) as classifiers used to diagnose faults in gears, bearings and shaft was carried out by Meesad and Yen (1997). Both networks performed successfully. Off-line training and iterative data feed were needed to achieve a successful fault classification process.

Zhang and Ganesan (1997) applied a multi-variable trend estimation of fault development to predict RUL of a bearing system by using a self-organising neural network. Condition monitoring was performed via online vibration measurements and fault quantification was formulated into a multivariate trend analysis. Self-organising neural networks were used to perform the multivariate trending of the fault development. It was found that the accuracy of the proposed prediction algorithm was the same as one of the SOM algorithms.

Yam *et al*. (2001) investigated the trend in predicting machine condition by using a recurrent neural-network system.

Kowalski and Kowalska (2003) demonstrated a Neural Networks application for induction motor fault diagnosis. In this research work two kinds of NN were proposed as multilayer perceptron networks and self-organising Kohonen networks. The results of the experimental tests showed that neural networks could be effectively used for the recognition of stator, rotor, rolling element bearing and supply asymmetric faults by appropriate measurements and interpretation of FFT analysis of current vibration spectra.

Gebraeel *et al*. (2004) carried out investigations on thrust bearing prognosis in an attempt to determine the prediction of the actual bearing failure time. The investigation aimed to develop neural-network-based models for predicting bearing failures. An experimental setup was developed to perform accelerated bearing tests where vibration information was collected from a number of bearings that were run until failure. This information was then used to train neural network models for the prediction of bearing operating times. Vibration data from a set of validation bearings was then applied to these network models. The resulting predictions were then used to estimate the bearing failure time.

Sreejith *et al*. (2008) proposed an application of neural networks for automated diagnosis of localised faults in rolling element bearings. Kurtosis and log-likelihood classification (Goumas *et al*. 2001 and Abbasion *et al*. 2007) extracted from time domain vibration signals were used as an input feature for the neural network. The

results showed that the trained neural network was able to classify different states of bearing faults with an accuracy rate of 100%.

Wang *et al*. (2010) used the autoregressive (AR) method combined with the back-propagation neural network (BPNN) in rotating machinery fault diagnosis. A new fault diagnosis method was studied by using the differences in AR coefficients with BPNN. The obtained diagnosis results were compared with three methods, BPNN with AR coefficients, BPNN with AR coefficient differences and BPNN with AR coefficient distances. It was found that the diagnosis results obtained by using BPNN with AR coefficient differences were superior to the other two methods.

Prieto *et al*. (2013) combined statistical-time features and neural networks in the detection of bearing faults in electric motors. Statistical-time features such as root mean square (RMS), standard deviation, variance and crest factorwere calculated from acquired vibration signals. The discriminant analysis (DA) value was used for the purpose of feature selection. The final classification tasks were carried out using a hierarchical neural network structure and the effectiveness of the method was verified by experimental results obtained from different operating conditions. The proposed method achieved a 95 percent classification rate of the overall test set.

3.4 Fuzzy Logic-based Fault Analysis

The Fuzzy Logic (FL) concept is based on the Fuzzy Set theory introduced by Lotfi A. Zadeh of the University of California, Berkeley, in 1965 (Zadeh, 1965). Fuzzy Set theory is about vagueness or uncertainty and it provides a method of using imprecise information within mathematical concepts.

3.4.1 Fuzzy Set and Conventional Set Theory

Fuzzy set theory extends conventional set theory to follow human-like reasoning. In conventional set theory, a fixed boundary is used to define membership of an object into a set in a classical or conventional set theory. For instance, a classical or conventional set, *A*, of real numbers greater than 10 can be expressed as:

$$A = \{x \mid x > 10\} \qquad (3.1)$$

In this case, the membership or inclusion of an object into a set is determined by a characteristic function. The characteristic function for "x is greater than 10" is defined as:

$\mu_A(x) = 1$, if x is an element of the set A and, (3.2)

$\mu_A(x) = 0$, if x is not an element of the set A (3.3)

Conventional set theory is a very important theory in mathematics, computer science and applied sciences applications. However, the nature of human-like reasoning is not represented well by the classical sets theory. This is because the form of human reasoning is mainly abstract and includes vagueness or imprecise information. For instance, in classic set theory, the set of high room temperatures is defined as a group of numbers which is greater than 25°C.

$$\mu_{High}(T) = \left\{ \begin{matrix} 1 \geq 25^{\circ}C \\ 0 otherwise \end{matrix} \right\} \quad (3.4)$$

Equation 3.4 defines A = "room temperature" and x = "high temperature" and by using classical theory represented by Equation 3.4, it is clear that 25.1°C is classified as high room temperature, but 24.9°C is not. The visual classification for the term high and not-high room temperature is illustrated in Figure 3.4.

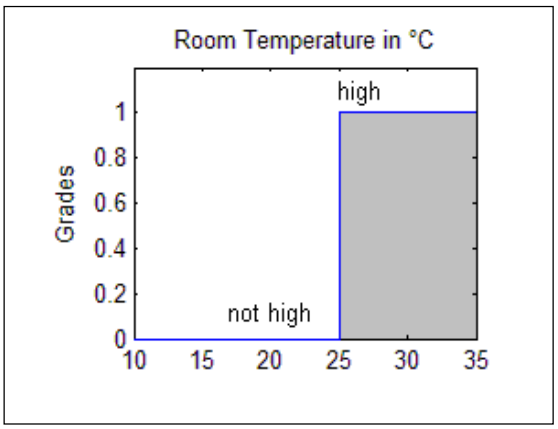

Figure 3.4 Non-fuzzy membership grading for room temperature

The characteristic function for high and low room temperature for conventional set theory is defined as presented in Equation 3.4.

The abrupt temperature transition shown in Figure 3.4 is not followed by a human's temperature sensor as humans cannot clearly detect the difference between 24.9 and 25.1°C. Therefore, it is not common to use the condition in Figure 3.4 as a description of high room temperature based on the human temperature sensor.

The disadvantage of conventional set theory as applied in this example is shown in defining the abrupt or sharp transition between a member inclusion and exclusion in a set. In real life human reasoning situations, a gradual transition from not-high to high temperature is employed.

In contrast to a conventional set, a fuzzy set, as its name implies, is a set without an abrupt or sharp boundary of memberships. It employs a gradual transition from "member of a set" to "not member to a set". Therefore, the concept of a gradual transition in temperature can be represented well by using a fuzzy set since it allows gradual membership of an object or element in a set. The gradual transition given to the membership functions is what gives fuzzy sets the flexibility to model commonly used linguistic expressions, such as "the temperature is low" or "the speed is high".

In fuzzy sets, the values of membership function are real numbers in the interval [0, 1], where $\mu_A(x) = 0$ means the object (x) is not a member of the set (A), and $\mu_A(x) = 1$ means a full membership to the set, $0 < \mu_A(x) < 1$ means partial membership.

Additional details on Fuzzy Sets theory are presented in Appendix 2.

3.4.2 Fuzzy Set to Fuzzy Logic

Based on the Fuzzy Set theory, the concept of FL aims to deal with vague information in reasoning. In his seminal paper Zadeh (1973) proposed that one of the reasons why humans are better at controlling systems compared to existing machines, is that humans have the ability to make effective decisions based on imprecise linguistic information. Therefore, Zadeh considered that it should be

possible to improve the performance of electromechanical systems by implementing the method humans use to deal with the imprecise information (Schwartz *et al*. 1994).

At its core, fuzzy logic (FL) is a method for problem-solving which provides solutions to problems which have imprecise, vague, ambiguous or inaccurate data that contains high levels of noise generated by fluctuations in a process. It provides a method of reasoning, in a human-like model, through an intuitive way of employing incomplete or inaccurate information. The main advantage is in its ability to describe qualitative knowledge or information, embedded linguistic knowledge and to approximate reasoning.

The difference between fuzzy logic and traditional logical systems is that fuzzy logic aims to adopt the spirit of the human thinking processes and natural human language. Fuzzy logic, which is the logic on which fuzzy reasoning is based, is much closer in nature to human thinking and natural language than the traditional logical system.

FL is useful in cases in which a decision-making expert system cannot fully determine an outcome. This is because the true/false statement as stated in the classic predicate logic is not suited to the nature of the decision making requirements. That is, not all decisions can be framed as true (fully belonging) or false (not fully belonging) in the domain of a decision.

In the practical world, it is not always possible to define conditions (mathematical sets) and their associated membership (degree of belongingness) in such a precise manner. It is not either 0 (not belong/false) or 1 (belong/true). Reasoning processes using qualitative and imprecise statements that can be included in rule bases are enabled by using FL.

Fuzzy sets, which are the foundation of FL, provide a complementary way of dealing with this situation by allowing partial set membership or partial “belongingness” which is based on a parameter called ‘degree of membership’ or ‘degree of truth’.

The FL reasoning statements can be incorporated into rule bases which provide simple and more intuitive models. FL accommodates the need to handle problems that cannot be solved using exact mathematical solutions. It is useful for problems which are difficult to solve using mathematical models.

Similar to expert systems, FL systems employ simple IF-THEN rules which are derived empirically to solve problems. The rules are descriptive since they are composed using words. However, unlike expert systems, the rules are made imprecise. For example, a typical fuzzy process logic statement may be composed such as '*IF (temperature is too hot)* AND *(heating power is too high)* THEN *(Reduce heating process faster)'*.

In practice, fuzzy systems consist of a knowledge based block, a fuzzy rule base and an algorithms block. Information is received from various sensors and pre-processed, then converted into fuzzy forms (fuzzification process). The fuzzification process is needed so that the information can be evaluated using fuzzy rule sets and in general, fuzzy rules are constructed manually.

In the fuzzification process, membership function is used to determine the mapping of input data into specific fuzzy variables. The final output of a fuzzy logic-based process is defuzzified in order to have a crisp or a precise numerical output. The defuzzification process involves the use of another membership function to calculate the precise numerical output of the fuzzy system.

Automatic control engineering is an area that uses fuzzy systems extensively. The input into the controlled system is the current state of the controlled device, and the output is fed to the controller which drives the device's actuator to adjust specific parameters related to the controlled device. A list of several early applications of fuzzy logic in the area of automatic control field is presented in Lee (1990).

3.5 Disadvantage of Fuzzy Logic

Fuzzy logic has several disadvantages in its application as a fault diagnosis tool. One of the disadvantages is the high dependency on acquisition of intuitive experience from an expert operator and this affects the subjectivity of fault diagnosis (Yang *et*

al. 2002). In addition, fuzzy logic systems lack the capability of any self-learning feature which is an important feature that is in great demand in on-line or real-time fault diagnosis processes. A self-learning feature is also important in practical fault diagnosis applications, especially in high-demand real time-time systems which require high-precision results (Gao and Ovaska, 2001).

Another disadvantage of fuzzy logic-based fault diagnosis systems occurs in its design process where the construction of the fuzzy rules that represent behaviour of the modelled system are critically dependent on the intuitive knowledge and experience obtained from an expert or operators. Expert experience is essential in developing the representation of each variable related to the characteristics of the problem and the resulting fuzzy membership functions. Consequently, the developed fuzzy rules cannot be guaranteed to be optimal.

In order to overcome its disadvantages, FL is usually combined with other techniques in its practical applications to fault diagnosis. The most widely used technique in combination with FL is artificial neural networks (ANN) which forms what is called a neuro-fuzzy system.

3.6 Application of Fuzzy Logic in Rotating Machinery Fault Analysis

FL offers the ability to deal with uncertainties in maintenance and scheduling processes and it has been used to improve the performance of fault detection and prediction in mechanical systems. It has been widely implemented in fault diagnosis applications because of its advantages in approximating reasoning and in linguistic knowledge implementation.

In mechanical equipment monitoring tasks, fuzzy logic theory was applied in cases where precise mathematical models were unavailable or too complex, but where there was still some vague, subjective and empirical knowledge related to the problem under investigation (Wang and Lei, 2001). The existing knowledge is generally constructed as a set of fuzzy relationships or fuzzy rules on which the overall fuzzy system is based. These rules or fuzzy relationships can be constructed based on information supplied by human experts.

In cases where only partial or incomplete fuzzy rules could be supplied by human experts where a set of problems or system input-output data were available, then it was deemed preferable to extract fuzzy relationships or fuzzy rules from the system-based data and combine the data, where possible, with human knowledge and experience. The combination could then be used to construct a complete and relevant set of fuzzy rules.

An early application of FL in fault diagnosis was investigated by Goode and Chow (1995), in which a hybrid neural-fuzzy fault detector was used to detect motor faults. In the investigation, a neural-fuzzy fault detector was used to monitor the condition of the motor bearingswear and the stator winding insulation failure. Fuzzy IF-THEN rules for bearing wear classification by the detector were constructed heuristically for three ranges of membership: low, medium and high. The trained neural-fuzzy fault detector was able to provide accurate fault detection results and could also provide the heuristic reasoning behind the fault detection process and the actual motor fault conditions.

Another early application of fuzzy logic (FL) in fault diagnosis was proposed by Goddu *et al*. (1998) in which a fuzzy logic-based method was used to interpret vibration signals from an electric motor in order to diagnose bearing faults. Spectrum data of the vibration signal was entered into the fuzzy decision system and a valid fault diagnosis result obtained. It was suggested that incorporation of neural networks or genetic algorithms with fuzzy logic to improve the capabilities of the fuzzy decision system, would be beneficial.

Vicente *et al*. (2001) presented a work on an automatic diagnosis system for detection and classification defects in rolling bearings using fuzzy logic. The measured vibration signals were analysed using spectral and statistical techniques. The variables used as inputs for the fuzzy system included: radial load, shaft speed, kurtosis, skewness and RMS. The designed system was able to classify three types of pre-established defects in rolling element bearings which operated under several shaft speeds and load conditions. The results showed that the designed system was able to diagnose 97% of the test database to distinguish between normal conditions

and a fault case. The system achieved 95% accuracy in the classification of fault cases of normal, pit corrosion, and scratched condition.

Miguel and Blazquez (2005) applied fuzzy logic in a model-based diagnosis application for a DC motor controller. Fuzzy logic was used to handle the uncertainty of the system model, noise and others variables that reduce the reliability and robustness of the fault diagnosis method. The FL fault detection and isolation system was successfully applied in a laboratory in which the uncertainty caused by disturbances and modelling errors was reduced.

Celik and Bayir (2007) studied the application of a complementary fuzzy logic system in fault diagnosis of an internal combustion engine. The fuzzy rules of the system were constructed by using theoretical knowledge, expert knowledge and the experimental results. The accuracy of the fuzzy logic classifier was tested by experimental studies which were performed under differing fault conditions. Using the developed fault diagnosis system, ten general faults which were observed in the internal combustion engine were successfully diagnosed in real time.

Wu and Hsu (2009) studied the development of a gear fault identification scheme using vibration signals with fuzzy logic inference and discrete wavelet transforms (DWT) for an experimental gear-set system. A proposed scheme that employed the combination of signal feature extraction using discrete wavelet transform techniques and fault identification using fuzzy logic inference was investigated. The fuzzy logic inference was proposed to develop the diagnostic rules of the database in the fault identification system. The experimental works were performed to evaluate the effect of fault diagnosis in a gear-set system under various operation conditions. The experimental results showed that the proposed fault diagnosis scheme was effective as it increased the accuracy in gear fault identification of the gear-set system. The aim of using FL was to overcome difficulties in the fault diagnosis of rotating machinery in a complex and noisy environment and to reduce the need for the knowledge of an experienced technician.

Saravanan *et al*. (2009) used a fuzzy classifier that was obtained from intuitive information and related domain knowledge of fault characteristics for a bevel gear

fault diagnosis scheme. A decision tree was used in selecting the best statistical features that could discriminate the fault condition of the gearbox vibration signals. It was followed by the formation of a rule set from the extracted features and then given to a fuzzy classifier for the fault classification process. The results of the fuzzy classifiers were found to be encouraging.

3.7 Hybrid AI Techniques in Fault Diagnosis

The combination of two or more of AI techniques to construct a hybrid model is based on the objective of simultaneously integrating the advantages of each individual approach and overcoming their weaknesses or shortcomings. The most common combination found in the application fault diagnosis systems combines ANN and Fuzzy Logic (FL) techniques. It forms a hybrid system that employs a combination of ANN and FL which is called a neuro-fuzzy system.

3.7.1 Neuro-Fuzzy System

A neuro-fuzzy system refers to the system by which various learning techniques are used by using neural networks and are integrated with a fuzzy inference system model. The system aims to use the advantages of fuzzy and neural networks in order to provide an accurate initialisation of the neural network in terms of the fuzzy reasoning scheme (Zhao *et al*. 2009).

The main objective of the integration of neural networks and fuzzy systems is to combine the strengths of both methods in order to achieve adaptive learning in diagnostic systems with a transparent knowledge representation (Leonhardt and Ayoubi, 1997).

Implementation of the combined advantage features of fuzzy logic and neural networks provides a superior diagnosis process of a system. In addition, the advantage of combining the features provides the fault detector with an adaptability feature which gives greater solution accuracy in different operating conditions (Altug*et al*. 1999).

The integration aims to utilise the fuzzy logic feature in handling imprecise information originating from imprecise conditions, and to provide initial modelling

requirements for neural-networks systems that have trainable capability, can perform calculations in parallel and have noise tolerance. The integration provides a robust fault diagnosis system that combines both numerical (quantitative) and symbolic (qualitative) information (Patton *et al*. 2000).

3.7.2 Application of Neuro-fuzzy System in Fault Diagnosis

Garga *et al*. (2001) introduced a hybrid reasoning method that integrated machinery data into a feed-forward neural network. The system used the training process to obtain a representation of the explicit domain knowledge for prognosis of faults of a gearbox. The method combined neural networks and fuzzy inference systems which constructed a neuro-fuzzy predictor. By using this combination, the ability of the fuzzy inference system to use linguistic descriptions is enhanced by the learning procedures used by neural networks.

A neuro-fuzzy system was used by Wang *et al*. (2004a) to evaluate the condition of spur gears during operation. The structure of the fuzzy inference system (FIS) was produced by using experts' knowledge, and a neural network training procedure was used to generate the related fuzzy membership functions. The test results demonstrated that the proposed neuro-fuzzy system significantly improved diagnostic accuracy due of its adaptability and robustness.

Results of a comparison between recurrent neural networks (RNN) and neuro-fuzzy (NF) inference systems used to predict fault propagation trends was presented by Wang *et al*. (2004b). It was found that a properly trained neuro-fuzzy (NF) system performed better than RNNs in forecasting accuracy and training process efficiency.
Chinnam and Baruah (2004) presented an application of neuro-fuzzy systems to estimate the remaining useful life (RUL) of a drilling tool. In the study, the existence of a specific failure definition of the model, and failure data were not available for the system under investigation. However, a neuro-fuzzy system, along with the knowledge and experience of an expert in the domain, were needed to construct an estimator model. The estimator system was trained using information derived from domain experts who had suitable knowledge of the process. The estimator model was constructed by using a neuro-fuzzy scheme and produced a good result in carrying out the RUL estimation.

Satish and Sarma (2005) proposed a combination of neural-networks and fuzzy logic to form a fuzzy back-propagation network to identify the present condition of a bearing, and estimate the remaining useful life (RUL) of an electric motor. In their work, the results of the fuzzy back-propagation network were compared to a generic neural network. It showed that the hybrid approach was preferable for use in assessing the present condition of the bearing and the time available before replacement was required.

A neuro-fuzzy modelling approach was presented by Kothamasu and Huang (2007), in which the system was based on adaptive learning using a Mamdani fuzzy model for system diagnosis and prognosis. It was designed to function as a decision-making tool in assisting condition-based maintenance. It had a feature allowing an adaptation to provide continuous improvement through interaction with users.

Castejon *et al*. (2010) implemented a combination of FL and ANN systems to form an integrated neuro-fuzzy system aimed at handling complex fault classification problems. It used the relationship between a set of patterns and fault types without the need to model the internal processes explicitly. The combined system performed well in handling uncertain information, which was very similar to the human reasoning process. ANN was used since it has abilities of real-time learning, parallel computation and self-organisation. Fuzzy logic was chosen, based on its ability to deal with imprecise or inexact information which is useful in handling vague or imprecise classification tasks.

3.7.3 Adaptive Neuro-Fuzzy Inference System (ANFIS)

A variant of neural-network-based systems which is commonly used in bearing fault diagnosis research is the Adaptive Neuro-Fuzzy Inference System (ANFIS). It was first introduced by Jang (1993). It is one of the most widely used hybrid intelligent systems and provides the advantages of both fuzzy logic and neural networks. It uses a given input-output data set to construct a Fuzzy Inference System (FIS),the membership function parameters of which are tuned (adjusted) using either a back-propagation algorithm, or in combination with a least squares type of method. This allows the fuzzy system to learn from the data used for the modelling purpose.

The neuro-adaptive learning technique incorporated in ANFIS is based on a method by which the fuzzy modelling procedure is able to learn information about the data set. It then computes the parameters of membership function that model the given input-output data set.

In terms of fuzzy logic, it provides a mapping feature between inputs and outputs. The mapping process is achieved by using membership functions (MFs) of the input and its associated parameters in relation to the MFs of the output. It is a tool that is used to accomplish the creation and adjustment of fuzzy logic membership functions in combination with the ANN learning feature.

ANFIS formulated a Sugeno fuzzy inference model (SFIM), which was originally proposed in Takagi and Sugeno (1985) and Sugeno and Kang (1988). The SFIM is a special case of the Mamdani FIS in which each rule's consequent is specified by a fuzzy singleton (or a pre-defuzzified consequent) (Jang *et al*. 1997). It is embedded into the framework of a multi-layer ANN in which synaptic weights (the connection strength between two neurons) are not used. Instead, it uses adaptive and non-adaptive nodes (Reddy and Mohanta, 2007).

ANFIS functionality is equivalent to an SFIM. As an SFIM, ANFIS uses set of input–output training data pairs in order to regulate the membership function and other associated parameters by using a back-propagation gradient descent or a least square type method. ANFIS represents an integration of a Sugeno fuzzy model where optimisation of the final fuzzy inference system is achieved through the ANN training.

ANFIS is useful in cases where there is a need to apply FIS to a system for which a collection of input-output data sets are available and the data is intended to be used for modelling, model-following, or similar needs.

In practice, there are some modelling cases of fault diagnosis in which the construction of the fuzzy logic membership functions can be carried out easily only by manually discerning the available data. There are also cases where determining fuzzy logic membership functions manually is not preferred due to the complexity of

input-output relationships of the data set. In these cases, selection of the parameters associated with a given membership function is mainly carried out by trial and error procedures. By using ANFIS, a pre-determined model structure based on characteristics of variables of the designed system is not needed.

The preferred method is to use a parameter selection process in a way that can create the membership functions automatically, based on the input-output data relationships.This method is also able to accommodate variations that exist within the data. This is where a neuro-fuzzy learning technique such as ANFIS is useful.

The expertise of a human operator, or experts, can be used on the modelled system to generate the initial membership functions and the rules for the Fuzzy Inference System (FIS). ANFIS is then used to refine the initial fuzzy IF-THEN rules and membership functions to best match the relationship between the inputs and the output characteristics of the system or data (Lou and Loparo, 2004).

This feature provides the ability to use predetermined input–output training data sets which employ ANFIS, in the regulation of the membership functions and other associated parameters by means of back-propagation gradient descent and least square type methods. These methodologies provide greater objectivity in using the ANFIS model since they provide a more systematic way that is less dependent on expert knowledge, which is useful in fault diagnosis (Zhang *et al*. 2010).

3.7.4 ANFIS Structure

A fuzzy reasoning scheme implemented in an ANFIS model is shown in Figure 3.5. Figure 3.6 shows the layer structure of ANFIS. Figures 3.5 and 3.6 are used to provide a brief explanation of ANFIS. For simplicity, an example of ANFIS which has two inputs x and y and one output f is used for the purpose of explaining its structure.It is assumed that the rule base contains only two if–then rules of the first order Sugeno fuzzy inference model.

The concept of an ANFIS structure can be represented by using a simple rule base which is defined as follows:

Rule 1: If x is A_1 and y is B_1, then $f_1 = p_1 x + q_1 y + r_1$,

Rule 2: If x is A_2 and y is B_2, then $f_2 = p_2 x + q_2 y + r_2$,

where x and y are the inputs, and A_1, A_2, B_1 and B_2 are fuzzy sets which represent linguistic labels such as *small*, *medium*,*large*. These fuzzy sets would be determined during the training process. p_1, q_1, r_1, p_2, q_2 and r_2 are design parameters which are also determined during the training process.

Figure 3.5 illustrates the fuzzy reasoning scheme of a two input and one output parameter. The related equivalent ANFIS architecture is shown in Figure 3.6. The nodes and layers functions are explained below:

Layer 1: This layer is also termed an adaptive node. The nodes in this layer represent input nodes. The membership grade is generated by nodes in this layer using functions:

$$O_{1i} = \mu_{Ai}(x) \quad i = 1,2.$$
$$O_{2i} = \mu_{Bi}(y) \quad i = 1,2.$$

where O_{1i} and O_{2i} are the fuzzy membership grades which are used to specify the degree of "belongingness" of the given crisp inputs x and y in terms of the linguistic labels A_i and B_i. A_i and B_i which are fuzzy sets constructed using their membership functions μ_{Ai} and μ_{Bi}. The selected shape of the membership functions are normally bell-shaped with a maximum equal to 1 and a minimum equal to 0, as defined in Equation 3.5,

$$\mu_{Ai}(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \qquad (3.5)$$

where a_i, b_i and c_i are membership function parameters that determine the form of the membership function based on linguistic labels. These parameters are in the premise part of the if–then rules, and hence are called premise parameters.

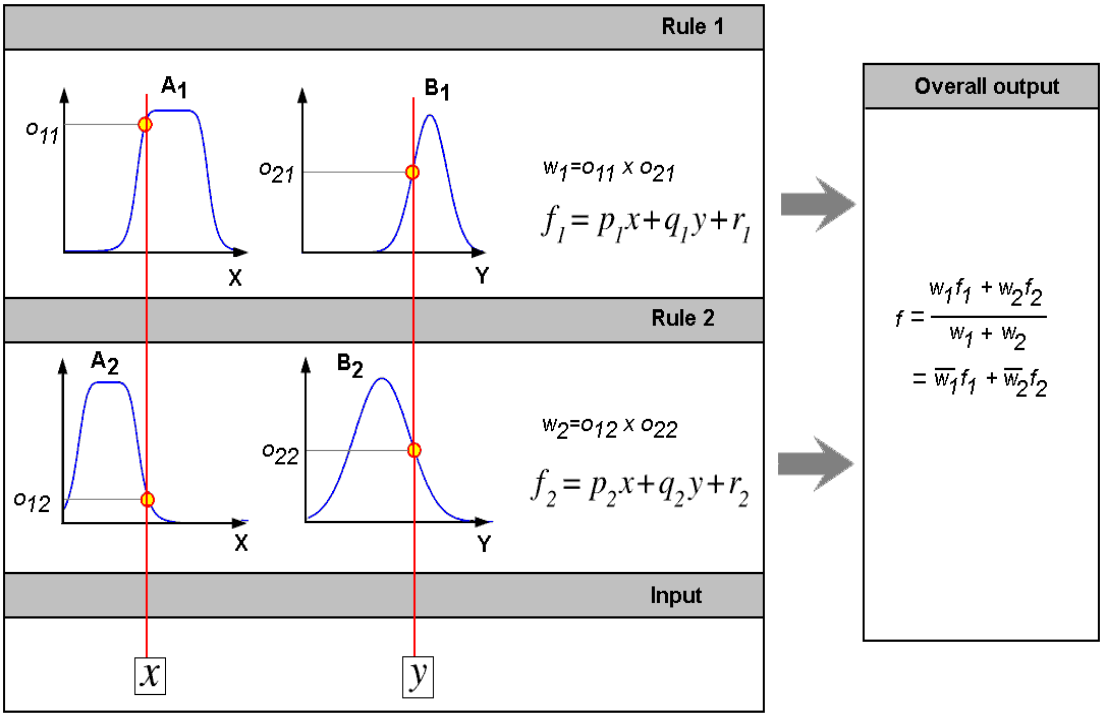

Figure 3.5 Fuzzy reasoning scheme (Jang, 1993), (Zhang *et al*. 2010)

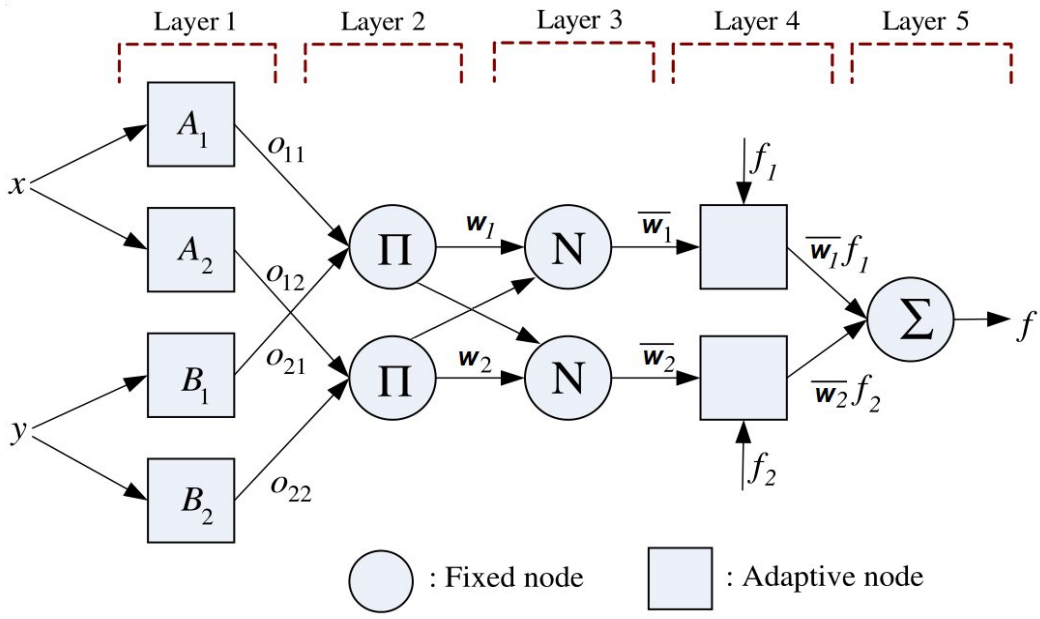

Figure 3.6 ANFIS layer structure (Zhang *et al*. 2010)

Layer 2: This layer contains fixed nodes labelled Π which are used to multiply the incoming signals and to obtain the product. For instance,

$$w_i = \mu_{Ai}(x) \times \mu_{Bi}(x) \qquad (3.6)$$

A rule "firing strength", the membership grade of the antecedent of the rule, in this layer is represented by the output of each node in this layer.

Layer 3: This layer contains fixed nodes labelled N which are used to calculate the ratio between the firing strength of the i^{th} rule and the sum of all rules' firing strengths. The ratio is called the normalised firing strength:

$$\overline{w}_i = \frac{w_i}{w_1 + w_2} i = 1,2. \qquad (3.7)$$

Layer 4: This layer contains the adaptive nodes. Each node produces the product of the normalised firing strength and (for a first order SFIM) a first order polynomial. The outputs of Layer 4 are defined by:

$$\overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) i = 1,2. \qquad (3.8)$$

where $\overline{w}_i$ is the output of layer 3, and p_i, q_i and r_i are linear parameters of the first order SFIM, which are referred to as consequent parameters.

Layer 5: There is only a single node in this layer. It is a fixed node which is labelled Σ. As its label suggests, it performs the summation of all incoming signals. Hence, the total output of the model is defined by:

$$f = totaloutput = \sum \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \qquad (3.9)$$

The five layers explained above function as a first order SFIM in which there are two adaptive layers, i.e., the first and the fourth layer. The first layer contains Premise

parameters $\{a_i, b_i, c_i\}$ and the fourth layer contains consequent parameters $\{p_i, q_i, r_i\}$. These parameters are obtained through a hybrid learning algorithm. The algorithm used is a combination of the gradient descent approach and least squares estimation which is comprises two stepsin order to improve training efficiency and eliminate possible trapping due to local minima.

In the first step, the premise parameters are assumed to be fixed. The optimal consequent parameters are then obtained by using the least squares estimate. In the second step, the consequent parameters are assumed to be fixed. The premise parameters are then updated by using the back-propagation gradient descent method, based on the error values.

3.7.5 Advantages of ANFIS

The expertise of human operators or experts on the modelled system or data can be used to generate initial membership functions and rules for the Fuzzy Inference System. ANFIS is then used to refine the initial fuzzy if-then rules and membership function to best match the relationship between the input and output characteristics of a complex system or data set (Lou and Loparo, 2004).

Using predetermined input–output training data sets, an ANFIS can be used to regulate the membership function and other associated parameters by using back propagation gradient descent and least square type of methods. These methodologies provide better objectivity in using the ANFIS model since it provides a more systematic way and that is less dependent on expert knowledge which thus providing more objectivity to the problems (Zhang *et al*. 2010).

3.7.6 Application of ANFIS in Fault Diagnosis

The idea of using ANFIS in fault diagnosis is to obtain a more robust diagnostic method which can integrate several raw features generated from the data.

There are various fault diagnosis systems that use ANFIS in automating the final condition identification of faults and several examples are presented below.

Zhang and Morris (1996) presented a recurrent neuro-fuzzy model for long-term forecasting, which was very similar to ANFIS, where the function outputs were local linear autoregressive models. The fuzzy models were implemented by fuzzy neural networks which combined the capability of fuzzy reasoning in handling uncertain information and the capability of neural networks in learning from examples. The fuzzy sets were used to represent process abnormalities which were then used in a fuzzy approach in representing process abnormalities that made the diagnosis system more resistant to measurement noise.

A method to enhance the detection of diagnostics information from low-speed rolling element bearing faults was presented by Altmann and Mathew (2001). It was based on the application of an adaptive neuro-fuzzy inference system that was employed to automatically select suitable wavelet packets that matched the fault features.

A neuro-fuzzy system was used by Wang *et al.* (2004a) to evaluate several faults of a spur gear set during operation. The neuro-fuzzy system was used to identify fault types, such as a cracked gear, with 20% and 40% tooth root width and a chipped gear with 20% and 50% tooth surface area. In this work, the structure of the FIS was produced using expert knowledge, while the neural network training procedure was employed to generate the related fuzzy membership functions. Through experimental tests, it was found that the developed neuro-fuzzy classifier produced promising results due to its adaptation capabilities and robustness.

Zhang *et al*. (2010) studied the application of ANFIS and multi-scale entropy (MSE) for the purposes of feature extraction and fault recognition in the diagnosis of faults in electric motor bearings. Several scales of MSE were calculated from vibration signals. There were five statistical parameters like maximum value, minimum value, arithmetic mean value, geometric mean value and standard deviation which were obtained from MSE and used as features. The features were then presented to the ANFIS for fault classification. The experimental results indicated that the ANFIS classifier could obtain higher diagnosis results. Thus, the proposed approach had the possibility for incipient fault diagnosis in bearings.

3.8 Concluding Remarks

In this chapter, a review of Artificial Intelligence (AI) techniques that are widely used in fault diagnosis has been presented. The three most-used AI techniques discussed are, artificial neural networks (ANN), Fuzzy Logic, and Neuro-Fuzzy (ANFIS).

The types of ANN and list of fault diagnosis applications in rotating machinery have been reviewed by presenting several applications of various ANN according to machinery component groupings. The background and an introduction to fuzzy logic concept were presented. The advantages and disadvantages of fuzzy logic techniques in fault diagnosis application were also detailed.

The last section of the chapter presented motivation and needs for implementing a hybrid or combined AI technique for better performance in fault diagnosis. The neuro-fuzzy technique, which is one of the most used AI techniques, was discussed. The aforementioned is a hybrid system that is formed to incorporate the advantages of ANN and FL for the purposes of fault diagnosis.

Chapter 4 – Review of Features Extraction Techniques in Bearing Fault Diagnosis

Feature extraction from vibration signals is a significant and challenging field of research in engineering, especially in the area of fault diagnosis of rotating machinery. It is one of the most fundamental issues in intelligent monitoring (Rafiee e*t al*. 2010). Feature extraction processes are required prior to performing fault diagnosis or fault classification. The processes aim to reduce the dimensionality of the data and to produce a transformation by which hidden signal features in the original time-domain are extracted (Yen and Lin, 2000). In the feature extraction process, an initial processing of sensor data measurement is carried out to obtain suitable parameters to indicate whether a pattern of interest is appearing.

In the development of feature extraction, a wide range of new techniques has been proposed in recent decades. Each technique has different theoretical foundations and produces different results. Furthermore, in a particular system, the implementation of some techniques may be better suited than others in relation to operational conditions (Bin *et al*. 2012).

The motivation behind the wide range of combined applications of feature extraction techniques for bearing fault diagnosis is in the importance of selecting the most effective and suitable techniques – those producing reliable diagnostic results. This motivation has led to continuous new developments of feature extraction techniques aimed at finding better and more applicable feature extraction and generation techniques for bearing fault diagnosis.

4.1 Feature Extraction Applications

Examples of combined feature extraction techniques are presented in Table 4.1. As the table shows, many feature extraction applications combine several techniques which are enhanced by using artificial intelligence methods such as artificial neural networks (ANN) and fuzzy logic systems (neuro-fuzzy). It shows that the structures of combined (hybrid) systems and the methods are varied. Among the proposed combinations, the most common techniques are combined feature extraction

techniques using time-domain, frequency domain, frequency-time domain, neural networks and fuzzy logic methods.

Table 4.1 - Summary of Combined Feature Extraction Techniques

References	Objects	Defects Considered	Techniques	Features	Classifiers	Results
Dyer and Stewart, (1978)	Rolling element bearings	Outer race defect	Time-domain analysis	Skewness, Kurtosis	NA	NA
Mathew and Alfredson, (1984)	Rolling element bearings	Outer race defect, Inner race defect	Time-domain analysis	Peak, RMS, standard deviation, RMS * Kurtosis, arithmetic mean, & geometric mean values	NA	recommend the combination of several parameters in diagnostic process
Liu and Mengel, (1992)	Ball bearings	Inner race defect, outer race defect, ball defect, combinations	Time-domain analysis	Peak Amplitude of the freq. domain, power percentage, peak RMS values	Artificial neural networks	Success rate up to 97 percent for classifying six categories.
Liu*et al*. (1996)	Bearings	Inner race defect, outer race defect, rolling element defect	Time-domain analysis	Statistical features: Kurtosis, Crest factor	ANN, Fuzzy Logic	A PC-based expert system produced. Fuzzy system achieved 100 percent accuracy
Paya *et al*. (1997)	Bearings and Gears	Defects on inner race of bearing and gear tooth irregularity	Daubechies 4	Ten wavelet numbers indicting both time and frequency and their 10 corresponding amplitudes	Artificial neural networks	96 percent
Xi*et al*. (2000)	Tapered roller bearings	Single spall in inner race, single spall in outer race & broken roller	Time-domain analysis	Peak, RMS, Crest factor (Cf), Kurtosis, Impulse factor, Shape factor.	Two dimensional feature spaces	provide a visualization of the diagnostics results on a two-dimensional plane of the computer
Nikolaou and Antoniadis, (2002a)	Rolling element bearings	Inner race & outer race	Daubechies 12	Mean and standard deviation of wavelet packet coefficients	NA	NA

Prabhakar *et al*. (2002)	Rolling element bearings	One scratch mark each on inner race (on the track) and outer race (on the track), two scratch marks on outer race (180° apart on the track), one scratch mark on each of inner race and outer race (on the track)	Daubechies 4	RMS, Kurtosis	NA	NA
Rubini and Meneghetti, (2000)	Three double row self aligning ball bearings	Inner race, outer race & ball defects (each bearing only)	Gaussian & Morlet	spectrum of the average of the wavelet transform	NA	NA
Seker and Ayaz, (2003)	Bearings	Failure caused by electric discharge to bearing	Wavelet basis used: Daubechies 15 & Daubechies 20	Statistical parameters: Mean, Standard deviation, Skewness, Kurtosis & RMS values (of cD)	NA	RMS values of d1 increase as the motor bearing degrades toward failure
Samanta and Al-Balushi, (2003)	Rolling element bearings	Outer race defect	Daubechies 4 & Time-domain analysis	Statistical parameters: RMS, variance, skewness, kurtosis and normalised sixth central moment.	Artificial neural networks	The test achieved success rate between 98 - 100 percent
Lou and Loparo, (2004)	Ball Bearings	Normal conditions, inner race faults, a ball fault, & outer race faults. Loads: 0,1,2 & 3 HP	Daubechies 2 (db4) and 10 (db20) - data normalised (0,1)	Feature vectors v generated from wavelet transform cD parts - (v1, … , v6), Euclidean vector distance, vector correlation coefficients	Adaptive Neural Networks Fuzzy Inference System (ANFIS)	Some characteristic components associated/investigated increase in fault severity level.
Sun *et al*. (2004)	Rolling element bearings	Inner race defect, roller defect, outer race defect	Time-domain analysis	Statistical parameters: Crest factor (Cf), Kurtosis value (Kv), clearance	Artificial neural networks	To generate boundary curves separating different classes in 2-

				factor (Cl) and Impulse factor (If). Energy level using RMS, absolute amplitude using peak-to-peak values.		dimensional classification space.
Purushotha-m*et al*. (2005)	Rolling element bearings	Single and multiple point defects on inner race, outer race, ball fault and combination of these faults	Daubechies wavelet	Mel Frequency Complex Cepstrum (MFCC) coefficients	Hidden Markov model Classifiers	Best efficiency obtained as 99 percent
Rafiee*et al*. (2007)	Gears and bearings	Three different fault conditions on gears (slight-worn, medium-worn and broken tooth), faulty bearings	Daubechies 4 (wavelet packet)	Standard deviation of wavelet packet coefficients	Artificial neural networks	best efficiency obtained as 100 percent
Wu and Liu, (2008)	Internal Combustion Engine	Five synthetic faults: air leakage on intake manifold, electronic control thermal (ETC) sensor fault, cam-shaft sensor fault, one cylinder misfiring & tow cylinders misfiring	Daubechies 4, 8 & 20 (DWT)	Energy spectrum of DWT components based on Parseval's theorem	Artificial neural networks	Recognition rate obtained: over 95 percent. Fault finding technique using db20 is superior than the one of db4 and db8
Li*et al*. (2008)	Rolling element bearings	Single point faults with different dimensions to balls and races.	Daubechies 20 - db20 (WPT-DWT)	Kurtosis	NA	NA
Saravanane *t al*. (2008)	Gears	gear tooth breakage, gear with crack at root and with face wear	Morlet wavelet	Standard error, sample variance, kurtosis and minimum value	Support Vector Machines (SVM) & Proximal SVM (PSVM)	best efficiency obtained as 100 percent

Wu and Liu, (2009)	Internal Combustion Engine	Sound emission recorded for without fault condition, air leakage of the intake manifold, camshaft sensor fault, electronic control thermal (ETC) sensor fault, one cylinder misfiring, and two cylinders misfiring	Daubechies 4, 8 & 20 (wavelet packet)	Shannon entropy of each wavelet coefficients (psi function)	Artificial neural networks	Recognition rate obtained: over 95 percent. Fault finding technique using db20 superior than db4 and db8
Wu*et al*. (2009)	Gears system	General fault among several gears	Daubechies (db4, db8, db20)	Energy spectrum of DWT components based on Parseval's theorem (4 levels decomposition)	Adaptive Neural Networks Fuzzy Inference System (ANFIS)	The total recognition rates are over 96 percent.
Zhao*et al*. (2009)	Bearings	Life accelerated test condition	Wavelet transform, time domain, frequency domain	Statistical features: value error (AVE), root mean square error (RMSE) and mean absolute percentage error (MAPE), Kurtosis, RMS, Energy index	Radial-based Function NN & Neuro-Fuzzy	Neuro-fuzzy classifier performs better than RBF NN classifier
Stepanic*et al*. (2009)	Rolling element bearing (6203Z)	Normal and defective condition	Time-domain analysis	Statistical parameters: arithmetic mean value, RMS, SM, Skewness, Kurtosis, C-L-S-I factors. Frequency domain features (bearing characteristic frequencies)	Linear and Quadratic classifiers (Fukunaga, 1990)	Achieved accuracy between 97.45 - 99.49 percent
Rafiee*et al*. (2010)	Gears and bearings	Ball, cage, inner race, outer race defects on bearings and three different	324 mother wavelets from various wavelet families like Haar, Daubechies,	Variance, standard deviation, kurtosis and 4th central moment of CWC-SVS	Artificial neural networks	Recommend that the best efficiency can be achieved using db 44 for gear and

		fault conditions on gears (slight-worn, medium-worn and broken tooth)	Coiflet, Morlet, etc.			bearing fault diagnosis (> 96 percent)
Castejon*et al*. (2010)	Ball bearing (FAG7206B)	Normal conditions, inner race faults, outer race faults, and ball faults	Daubechies 6 (wavelet packet)	cD of level 5 (cD5) chosen as characteristic features. cD5 normalised in [-1 1] interval	Artificial neural networks	classifier accuracy around 85 percent
Jayaswal *et al*. (2010)	Bearings	Inner race, outer race & ball defects (constant speed & load)	Daubechies 8 - db8 (WPT)	Statistical features: Mean, Max, Min values of RMS at node-n	ANN, Fuzzy Logic	Proposed Fuzzy BP network can be used for diagnosis and prognosis of bearing condition.
Marichal*et al*. (2010)	Roller bearings (FAG 7206B)	Rolling element fault, inner race fault & outer race fault	Frequency-domain analysis	Demodulated signal via FFT-based Hilbert Transform	ANFIS	Output value of training data set in range [0,1]
Kankar*et al*. (2011)	Rolling element bearings	Spall in inner race, outer race, rolling element and combined defects	Daubechies 44, Meyer, Coiflet5, Symlet2, Gaussian complex Morlet and Shannon wavelets	Statistical features: kurtosis, skewness, and standard deviation from wavelet coefficients corresponding to scale maximising energy to Shannon Entropy ratio	Support Vector Machines (SVM), artificial neural networks, self-organising maps	The best efficiency obtained using complex Morlet wavelet and SVM classifier as 100 percent
Sugumaran and Ramachandran, (2011)	Roller bearings (4 units of SKF 30206)	Normal condition, inner race defect & outer race defect	Time-domain analysis	Histogram of time-domain signal, Decision tree method to generate fuzzy rules	Fuzzy Logic system	The results are found to be encouraging. The results shows a good performance
Prieto *et al*. (2013)	Rolling element bearing in electric motor	Single point defects, combined single point defects and generalised degradation over different speed and torque.	Time domain analysis combined with 'curvilinear components analysis' (CCA)	Statistical features: kurtosis, variance, standard deviation, crest factor, skewness, RMS, etc.	Artificial Neural Networks (hierarchical multi-layer perceptrons)	Successful classification ratio of 95 % achieved

Present work	Rolling element bearing FAG 2705 TVH	Normal condition, inner race defect & outer race defect	Hybrid method: wavelets (db4, db8, db12, db22 & db44), statistical parameters, ANFIS	Statistical features: kurtosis, variance, standard deviation. Also level of energy of wavelet results, RMS of wavelet results.	Artificial Neural Networks Fuzzy Inference System (ANFIS)	To Be Concluded

Several important investigations that use combined feature extraction techniques for the purpose of fault diagnosis of rolling element bearings, as seen in Table 4.1 are explained in chronological order in the next section.

4.2 Combined Feature Extraction Methods in Bearing Fault Diagnosis

An early application of bearing fault analysis that combines the application of statistical and other parameters of time-domain signals with an artificial neural network (ANN) classifier to identify the characteristics of bearing faults was proposed by Liu and Mengel (1992). The features used to train the ANN classifier were peak amplitude of the frequency domain, peak RMS, and power spectrum percentage values. The type of neural network used was a feed forward network. In the study, Liu and Mengel identified faults in ball bearings based on indices which consisted of the features mentioned above. These features were used as indirect indices to develop a system for monitoring and classifying ball bearing defects which employed a feed-forward neural networks system. The results showed that all the trained neural networks were capable of distinguishing normal bearings from defective bearings with a 100 percent success rate. The networks could also classify the bearing condition into six different states with a success rate of up to 97 percent.

Application of a PC-based fuzzy expert system for failure detection of rolling element bearings using an expert system was explored by Liu *et al*. (1996). The system aimed to provide specific knowledge of various aspects related to bearing monitoring. These aspects included defection of frequency, failure selection, diagnostic methods, fuzzy-based bearing fault classification. Experiments were conducted to investigate the effectiveness of the system. The results showed that there were five values of frequency response in the high frequency region (5-22 kHz)

which were the best features for the detection of rolling bearing defects. The system achieved a 100% reliability rate for the detection of roller bearing defects using fuzzy reasoning.

Another early application of hybrid systems in which wavelet transforms and artificial neural networks were combined in analysing bearing fault conditions was presented by Paya *et al*. (1997). In the investigation, the vibration signal was processed using wavelet transforms; this was followed by a process that recorded thresholds. The thresholds were based on the levels of the most dominant values of the reference signals (valid bearing signals). The ten most dominant features were selected as input vectors to the back-propagation neural network. The results showed that the combination of the wavelet transform with an artificial neural network provided a useful tool for intelligent diagnostics of faults in rotating machinery. The results also showed that the combination of wavelets with neural networks achieved an overall success classification rate of 96 per cent.

Garga *et al*. (2001) introduced a hybrid reasoning method that integrated machinery data into a feed-forward neural network through a training process based on the representation of explicit domain knowledge for the prognosis of gearbox faults. This method combined neural networks and a fuzzy inference system which resulted in a neuro-fuzzy predictor. By using this combination, the ability of the fuzzy inference system to use linguistic descriptions was enhanced by the learning procedures of the neural networks. The outcome showed that the proposed hybrid automated reasoning system was able to combine domain knowledge and machinery operational and test data.

Investigations of feature extraction for the purposes of bearing fault diagnosis by the use of statistical parameters calculated from wavelet packet transforms were performed by Nikolaou and Antoniadis (2002a). In the investigation, standard deviation and mean values were calculated from all wavelet coefficients and used as features. The feature values were selected based on the thresholding process. The resulting standard deviation and mean values from the thresholding process were then used to calculate the energy vectors. The diagnosis stage continued with the selection of the best energy vector and the taking of the FFT of the results. The

investigation aimed to propose a method that could utilise the underlying modulated features present in the vibration signal of faulty bearings. The method used the Wavelet Packet Transform (WPT) and the systematic parameter selection criteria that reduced any interventions of the end user in the fault analysis process.

Application of the wavelet transform as an agent for multi-resolution analysis of bearing vibration signal was studied by Seker and Ayaz (2003). In the study, bearing vibration signals, recorded from an accelerated fluting ageing condition in an induction motor, were transformed using wavelet multi-resolution analysis (MRA). The MRA produced segmented selected frequency bands that were used to extract information from the signal with minimum distortion. In the analysis, the RMS of the vibration signal was compared with the wavelet Detailed (c*D*) part of the wavelet transform results. It was found that the ratio between the RMS of the vibration signal and the c*D* part increased as the motor bearing degraded toward failure.

Samanta and Balushi (2003) investigated the application of statistical parameters generated from bearing vibration signals used as features in an application of artificial neural networks (ANN) as a bearing fault classifier. The values of RMS, variance, skewness, kurtosis and normalised sixth central moment (skewness multiplied by 2) were calculated directly from the time-domain vibration signals and used as features to the node inputs to the ANN system. The study also included investigation of the effect of pre-processing techniques such as high-pass filtering, band-pass filtering, envelope detection (demodulation) and wavelet transformation of the vibration signals. These investigations were carried out prior to the feature extraction process. The proposed method aimed to reduce the number of features needed in ANN training in order to achieve a faster ANN training process. The results showed that an ANN could be used effectively in the diagnosis of the bearing's condition.

The applications of hybrid techniques in bearing fault diagnosis have continued, producing more variations of combined methods aimed to increase the accuracy of diagnosis. Combined or hybrid systems that utilise a combination of neuro-fuzzy (ANFIS), statistical parameters and wavelet transform results have been widely

investigated and evaluated. Several examples of these combined methods are presented below.

A neuro-fuzzy system was used by Wang *et al*. (2004a) to evaluate the condition of a spur gear operation. In this work, the structure of the FIS (fuzzy inference system) was produced by experts, while the neural network training procedure was used to generate the related fuzzy membership functions. Using experimental tests, it was found that the developed neuro-fuzzy classifier produced promising results. In addition, the test results showed that the proposed learning algorithm could effectively update the fuzzy system, which was necessary to improve the diagnostic performance.

Results of the comparison between recurrent neural networks (RNN) and neuro-fuzzy (NF) inference systems used to predict the fault propagation trend were presented by Wang *et al*. (2004b). It was found that a properly trained NF system performed better than recurrent neural networks, both in forecasting accuracy and training process efficiency.

A method for analysing localised defects in ball bearings using statistical parameters, wavelet transforms and a neuro-fuzzy classifier was studied by Lou and Loparo (2004). In the study, the vibration signal was acquired from a motor-driven experimental system with normal bearings and then with bearings with inner race, outer race and ball faults. The first feature vector was calculated for a given signal by using Discrete Wavelet Transform (DWT) results and their standard deviation values. The second was calculated by using Euclidean vector distance (calculated from the square of Euclidean distance metric) and vector correlation coefficients (Lou and Loparo, 2004). The feature generation method led to a fast fault detection scheme that was developed using the standard deviation of the wavelet decomposition parts. A fuzzy neural inference technique was implemented to provide a reliable diagnostic decision. The fuzzy neural inference technique was based on an adaptive neural-fuzzy inference system (ANFIS) proposed by Jang (1993). In comparing the results of the proposed method, Lou and Loparo used the Euclidean vector distance method and the vector correlation coefficient method. The findings

showed that the proposed diagnostic method could be used to identify different fault conditions under a variety of load conditions.

A combination of neural-networks and fuzzy logic which formed a fuzzy back propagation network which was used to identify the present condition of a bearing and estimating the RUL of an electric motor was proposed by Satish and Sarma, (2005). In their work, the results of a fuzzy back-propagation network were compared with the neural network. The hybrid approach result showed proved more effective in assessing the present condition of the bearing, and the time available before the replacement of the bearing was required.

The exploration of applications for hybrid feature extraction methods has continued with more methods proposed in the area of bearing fault diagnosis. For instance, Rafiee *et al*. (2007) investigated the use of a single statistical parameter which was calculated from wavelet packet coefficients. It was used in an intelligent condition monitoring system which included general fault analysis of a bearing using ANN. The vibration signal was processed using the wavelet transform, and the feature vectors were generated by calculating standard deviation values of the wavelet packet coefficients. The type of wavelet used was Daubechies wavelet type 4 (db4), the decomposition level of vibration signal was taken up to the 4^{th} level. The features for the ANN training process were generated by calculating the standard deviation of the wavelet packet components. Instead of using the energy level of the wavelet as a coefficient, the investigation used a new feature vector that employed standard deviation as a feature to train the neural network and identify the faults.

A neuro-fuzzy modelling approach was presented by Kothamasu and Huang (2007) in which the system was based on adaptive learning using the Mamdani fuzzy model for diagnosis and prognosis of a system. The system was designed to function as a decision making aid for condition-based maintenance. It also had a feature to allow some general modifications to be made with continuous improvement through interaction with users.

A combination of kurtosis and wavelet transforms applied in fault diagnosis of rolling element bearing was investigated by Li *et al.* (2008). The input signal was

processed using both Discrete Wavelet Transform (DWT) and Wavelet Packet Transform (WPT) and followed by using inverse DWT and WPT to obtain the time-dependent signals of each frequency-band. The process continued with the calculation of the kurtosis value of each frequency band signal in order to obtain the kurtosis curves of DWT and WPT. The results showed that the proposed method was able to distinguish faulty bearings from those in good working order.

The application of neuro-fuzzy systems in bearing life prognosis was explored by Zhao *et al*. (2009). In their work, the features or parameters of the time domain, frequency domain and wavelet domain were extracted by using the corresponding signal processing and filtering methods. The extracted features were then compared in order to select the best one for prediction. The features used in the selection process were RMS, kurtosis, energy index and peak-to-peak. Based on the best selected feature (i.e., RMS) a prediction model for bearing health condition was constructed using a neuro-fuzzy scheme.

An approach in implementing statistical pattern recognition in constructing the classifier for rolling element bearing faults was carried out by Stepanic, *et al*. (2009). In this implementation, the signal's time-varying statistical parameters and characteristic rolling element bearing fault frequency components were obtained through the envelope analysis method and then used as the vectored features. A linear and quadratic statistical parameters recognition method was used in building the classifier. The results showed that the proposed approach achieved a classification accuracy rate of 97.45% - 99.49%.

Wu and Kuo (2009) used feature extraction with Parseval's theorem (energy level distribution), calculated from the discrete wavelet transform (DWT) results for fault analysis of an automotive generator. Prior to the calculation of energy level, the vibration signal was processed using a wavelet transform with up to 9 frequency bands (9 levels). The feature extraction scheme was performed by using db2, db4, db10 & db20 Daubechies wavelets. The features were used to train neural network systems which were later used to identify the fault signals. The proposed method was able to detect and classify the distorted signal with high accuracy. It was found that

the most accurate results were obtained when the energy feature was calculated from the Daubechies wavelet db4.

Rafiee *et al*. (2010) focused their study on finding applicable features for bearing and gear fault detection and diagnosis. It was posited that feature extraction and feature reliability were important factors in intelligent systems. Furthermore, since there were no standard rules on practical feature extraction of vibration signals, a proposed method which used continuous wavelet transform (CWT) was used in the research. The focus was on using Wavelet transform (WT) that was capable of processing stationary and non-stationary signals simultaneously in the time and frequency domains for the feature extraction process. Continuous wavelet coefficients (CWC) were calculated for the gear and bearing segmented signals using 324 mother wavelet candidates from different wavelet families: Haar, Daubechies, Symlet, Coiflet, Gaussian, Morlet, complex Morlet, Mexican hat, bio-orthogonal, reverse bio-orthogonal, Meyer, discrete approximation of Meyer, complex Gaussian, Shannon, and frequency B-spline wavelets. The results showed that db44 had the most similarity in shape with signals across both faulty gears and bearings. It matched with random high impact signals (e.g., broken-tooth gear). Also, it was found that standard deviations and variance of the CWCs were suitable for use as features for bearings fault diagnosis. On the other hand, kurtosis of the CWC showed that it did not have any consequential relation that related to the fault which was presented in the bearing dataset.

The detection of an incipient fault in bearings was studied by Castejon *et al*. (2010). In the study, an automated rolling bearing fault analysis was implemented, based on the analysis and classification of signature vibrations. Multi-resolution analysis (MRA) was used in a first stage in order to extract the most interesting features from signals. The features were used in a second stage as inputs of a supervised neural network for classification purposes. The vibration signals were processed using the multi-resolution analysis (MRA) of the wavelet transform. The application of MRA aimed to extract the most interesting features from signals. Features obtained were then used in supervised training of a neural network for classification process. The pattern selection process was employed to select the best pattern of the wavelet coefficients. The selected pattern was then used to construct a classifier which

employed an ANN training procedure. The experimental results showed the ability of the method to detect four bearing conditions (normal, inner race fault, outer race fault, and ball fault) in a very early stage.

Jayaswal *et al*. (2010) applied the wavelet transform and statistical parameters as a bearing fault analysis system. In their investigation, statistical parameters like mean and RMS values were calculated from parts of the wavelet transform of the bearing vibration signal. The features obtained from mean and RMS of the wavelet packets was used to build the classifier for bearing faults based on a fuzzy back-propagation neural networks system. The results showed that the proposed hybrid method successfully diagnosed the bearing faults.

Kankar *et al*. (2011) used wavelet-based feature extraction to fault diagnose ball bearings which had localised defects on the various bearing components. A wavelet-based methodology was constructed using Relative Wavelet Energy (RWE) and Shannon entropy criteria (Shannon entropy measures the diversity of a possibility series (Shi *et al*. 2004)). These criteria were used to calculate the statistical features needed for the training and testing of artificial intelligence techniques.RWE criterion and Maximum Energy to Shannon Entropy ratio criterion was calculated from several wavelets. It was used to select suitable wavelets that had maximum energy, for the purpose of building a bearing fault classifier. The wavelets which had maximum energy were then used to generate three statistical features, i.e., kurtosis, skewness, and standard deviation. The features were then used to train three machine learning techniques, i.e., Support Vector Machine (SVM) (Jack and Nandi, 2001), ANN and SOM. The results showed that the wavelet selected using Maximum Energy to Shannon Entropy ratio criterion (Meyer wavelet) produced a better classification efficiency than others.

Sugumaran and Ramchandran (2011) carried out a time-domain analysis of vibration signals generated by a faulty roller bearing and obtained the amplitude magnitude histogram of the signal. The decision-tree method was used to generate features based on the best selected histogram that differentiated the fault conditions of the bearing. Fuzzy logic was implemented as a bearing fault classifier and the rules for the fuzzy logic system were generated automatically based on the features by using

the decision-tree method. The fuzzy classifiers were tested for their ability to classify normal conditions, inner race faults and outer race faults. The fuzzy logic classifiers showed promising results in identifying faults.

Wang and Chen (2011) explored the application of fuzzy neural networks with frequency-domain features of the vibration signals in order to process the ambiguous relationship between the symptom parameters and the fault types. The exploration aimed to construct a system that could automatically identify bearing faults. Automatic identification was carried out using possibility theory, fuzzy neural networks and frequency-domain features extracted from the bearing signals. The findings showed that the non-dimensional fault symptom parameters described in the frequency-domain could represent the characteristics of the signals measured for fault diagnosis in a rolling bearing. The proposed method was successfully applied in the fault diagnosis of a rolling bearing used in a centrifugal blower.

Prieto *et al*. (2013) combined statistical-time features, curvilinear component analysis (CCA) (Demartines and Herault, 1997) and neural networks in the detection of bearing faults in an electric motor. Statistical-time features such as root mean square (RMS), standard deviation, variance and crest factorwere calculated from acquired vibration signals. The discriminant analysis (DA) value was used for the selection of the most significant features based on the value of the DA. A large DA value implied that the investigated feature(s) contributed to a proper representation of the measurements in the data space, hence the features(s) were significant and useful. The final classification tasks were carried out using a hierarchical neural network structure which employed hierarchal multi-layer perceptrons (hMLP) (Vasquez *et al*. 2009). The effectiveness of the method was verified by experimental results obtained under different operating conditions. The proposed method achieved a success rate of 95 per cent of the overall test set.

4.3 Concluding Remarks

The literature reviews presented in this chapter have shown the technique trends that are being applied to feature extractions in fault diagnoses. There are several important findings from the results of the reviews.

- One of the most important applications of the wavelet transform is feature extraction (Wang *et al*. 2011), and despite the amount of previous research on the wavelet transform, the selection of the mother wavelet function, which is a significant topic in signal analysis, is still open to question (Rafiee *et al*. 2010). Hence there are possibilities for exploration of the implementation and combination of various wavelet functions in the feature extraction process.
- Even though feature extraction is important in the application of intelligent systems, there is no clear standardised rule which can be used as guidance for its implementation in the application to vibration signals. In this case, there is no standard methodology for finding and generating reliable features (Rafiee *et al*. 2010). Hence it is an open area in which new methodologies may be proposed to process the vibration signals for the purposes of studying the feature extraction process.
- Even though the application of the Daubechies wavelet (db) has been covered by several researchers, they have mainly referred to the applications of low-order db types (db1-db20) (Wu and Liu, 2007). The application of high-order Daubechies (db) wavelets is rare (Antonino-Daviu *et al*. 2006, Rafiee *et al*. 2009b). It can be inferred that the application of high-order Daubechies wavelets (i.e., db22, db44, etc) is open for investigation.
- One of issues in the application of wavelets in fault analysis is that the selection of the base wavelet function has remained largely an *ad hoc* process. For example, there was a suggestion that shape matching be used as a way of selecting the best base wavelet for vibration signal analysis (Ling and Qu, 2000, Yan and Ren, 2004). The suitable basis function that determines the shape of a wavelet is related to the sensitivity characteristics of the extracted features technique. It determines the ability of the feature extraction method to enhance the detection of faults. The selection of a suitable basis function for particular fault detection is still an area which is open to investigation.
- The research in constructing the automatic selection of an integrated feature extraction technique for bearing fault diagnosis through

application of artificial intelligent techniques is still developing. This is particularly the case in the application of hybrid methods that combine neuro-fuzzy (ANFIS), wavelet transforms and statistical parameters as features in fault diagnosis.

Chapter 5 – Adaptive Neural-Fuzzy Inference System and Wavelet-Based Feature Extraction

In this research a new method of wavelet-based feature extraction combined with an adaptive neural-fuzzy inference system (ANFIS) is proposed and investigated based on developments and findings obtained from the extensive literature review presented in Chapter 3 and Chapter 4.

The feature extraction method combines the use of statistical parameters calculated from wavelet MRA results and ANFIS for learning and building bearing fault classifiers. The proposed method was influenced by previous work in the related areas. The proposed method generated several statistical features calculated from wavelet transforms using the MRA technique. There were four statistical features in use; RMS, kurtosis, standard deviation and variance. In addition, there were three others features investigated: the energy levels of wavelet transform results, the dominant frequency of each wavelet transform result and the amplitude of the dominant frequency of each wavelet transform result. These seven features were used as input-output data for ANFIS training for the purposes of building a bearing fault classifier for three fault conditions.

Several statistical feature applications for bearing fault diagnoses, as presented in Table 4.1 of Chapter 4, were the motivation for further investigation. The research extended the application of ANFIS for bearing fault diagnosis, and the background that led to the construction of the proposed method was based on variants of feature extraction methods and the ANFIS applications listed in Table 4.1. Specific findings are presented below.

The option to use Daubechies wavelet in the feature extraction process in this research was based on findings that it is rarely used. Hence its usefulness for the purposes of fault analysis could be explored further. This led to the use of five Daubechies wavelets (i.e., db4, db8, db12, db22 and db44). The research extended the fault diagnosis applications investigated and used by Wu and Liu (2008) for

engine fault diagnosis, Wu *et al*. (2009) and Rafiee *et al*. (2009b) for gear and gearbox fault diagnoses.

The ANFIS was used in this research since it harnesses the advantages of fuzzy logic and neural network modelling. In addition, it was chosen because of its wide range of applications in fault diagnoses. Examples of an induction motor fault may be found in Tran *et al*. (2009), for a gear faultWu *et al*. (2009), and for an electrical bearing fault Zhang *et al*. (2010).

In addition, this proposed project introduced and investigated a new scheme for the selection of dominant or related fault features using the *Exsrch* function of Matlab.

Furthermore, the proposed method aims to contribute additional information regarding the use of db wavelet functions in bearing fault diagnosis. The investigation might be used to add information to the selection of the most suitable Daubechies type for certain fault diagnoses in the context of feature extraction. This still remains a challenging and open area for research.

The research also aims to investigate the application of statistical parameters generated from a wavelet-based feature extraction method and used in the construction of a fault classifier with a neuro-fuzzy system (i.e., ANFIS). The applications of statistical parameters, wavelet transforms and ANFIS have been various (Lei *et al*. 2007, Zhao *et al*. 2009, and Kankar *et al*. 2011) and this research intends to expand upon and explore different schemes for using the combined feature extraction of statistical parameters, wavelet transforms and ANFIS.

The feature extraction method and the construction of bearing classifiers, using ANFIS proposed in this research, form an integrated investigation based on the abovementioned findings which were obtained from the literature review.

5.1 The Proposed Method

The proposed feature extraction technique utilises Daubechies wavelet to pre-process (transform) the time-domain vibration signals acquired from a bearing test rig. The pre-processing uses the multi-resolution analysis of the wavelet transform.

In the proposed scheme, the vibration signals acquired from a bearing test rig were pre-processed using the discrete wavelet transform (DWT). The wavelet type adopted was Daubechies (db). There were five types of Daubechies (db-n) wavelet utilised: 4 (db4), 8 (db8), 12 (db12), 22 (db22) and 44 (db44). A schematic diagram of the new feature extraction method is depicted in Figures 5.1 to 5.3.

Figure 5.1 shows the first step of the proposed feature extraction method. The process began by acquiring a signal from the test rig through the data acquisition process. The resulting raw vibration signal was then processed using Daubechies (db-n) wavelet transform, applying up to 10 levels of decomposition using the MRA technique. There were two parts produced by the db-n transform; cA (Approximated coefficients) and cD (Detailed coefficients) parts. In this research, only the cA parts of each db-n transform were used to generate the features since the cA parts contained an approximation of the transformed vibration signals.

The outputs of each db-n wavelet transform were the 10 cA parts. These cA parts were used to generate seven features, being Energy level, Root Mean Square (RMS), Kurtosis, cA dominant frequency (labelled *cA_x*), Amplitude of cA dominant frequency (labelled *cA_y*), Standard Deviation and Variance. All of these features were obtained from the cA parts of the db4, db8, db12, db22, and db44 wavelets in decomposition levels 1 to 10.

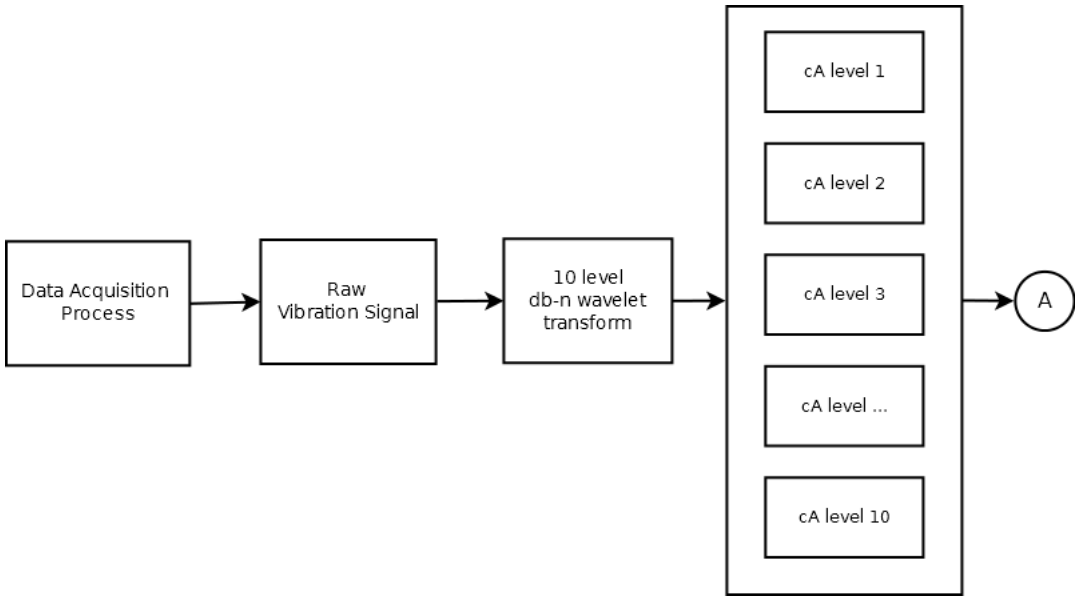

Figure 5.1 The first stage of the proposed method

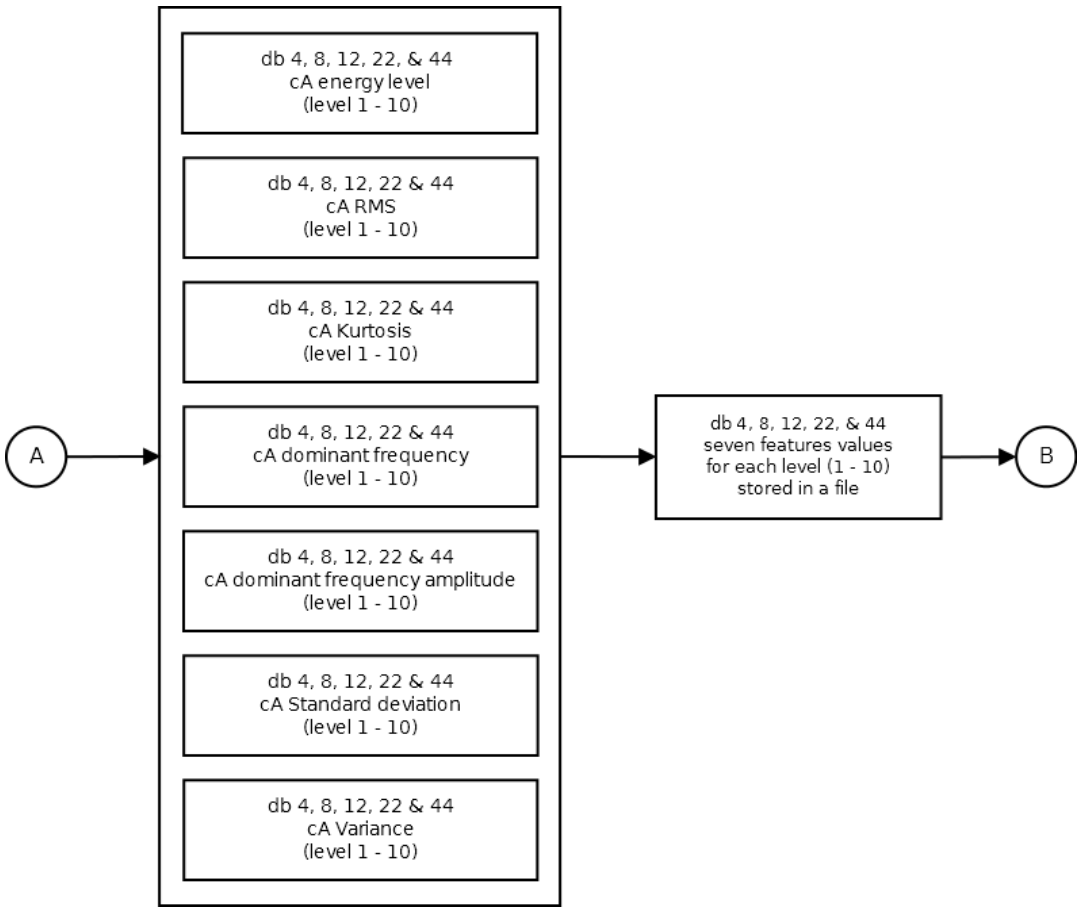

Figure 5.2 The second stage of the proposed method

The second stage of the proposed method is shown in Figure 5.2; a continuation of the feature generation process of Figure 5.1. The second stage depicts the generation

of the seven features. There were 10 groups of the seven features that resulted from 10 decomposition levels for each db-n wavelet transform. Each group contained the seven features produced from each type of db-n transform (i.e., db4, db8, db12, db22, and db44). All of the db-n 10 values of the seven features were then stored into text files for later use in subsequent stages of the process.

The third stage of the proposed feature extraction method is depicted in Figure 5.3. In this stage, the process began by selecting the highest energy level of db-n features in a data group from level 1 to 10. Only the feature data group which had the highest energy level was selected to be used in the next step. The calculation of the energy levels was carried out for all of 10 groups of the db-n feature data. The energy level results determined which particular data group would be used. There was only one data group out of the 10 groups of each db-n features data that was used in ANFIS training, based on its energy level.

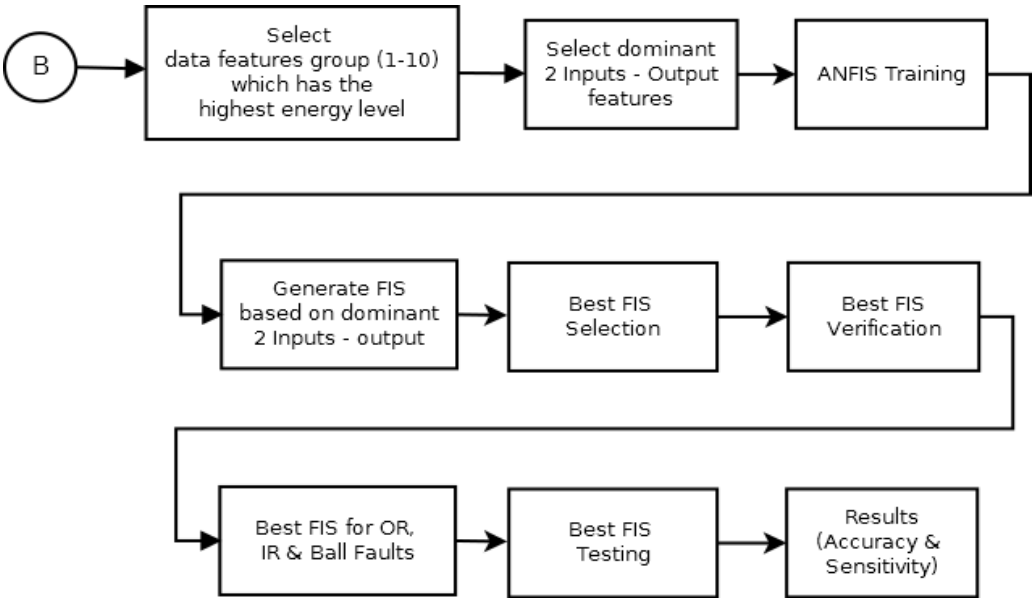

Figure 5.3 The third stage of the proposed method

As shown in Figure 5.3, prior to the ANFIS training process, two dominant inputs were selected that related to a Target Output from the seven features available. The following ANFIS training procedure aimed to produce a Fuzzy Inference System (FIS) with units based on the two most highly-related inputs and a single Target

Output of the seven available features. The process then continued with selection of the best FIS units as shown in Figure 5.3.

The best selected FIS units were then verified. The results of the verification process produced FIS units that related to the fault vibration data in use. There were four types of fault vibration signal used in the process: outer race fault (ORF), inner race fault (IRF), ball fault (BF) and no fault (NF) vibration signals. The process produced four types of FIS unit related to the four types of bearing fault respectively.

The selected FIS units for each type of fault were then tested. The test results were used to analyse the Target Output accuracy and db-n sensitivity of the corresponding FIS units.

5.2 Extended Multi-Resolution Analysis (MRA)

In the proposed method, a wavelet multi-resolution analysis (MRA) of the vibration signal was used. This technique was adopted previously by Zhu *et al.* (2009), Wu and Kuo (2009), Qiu *et al*. (2006) and Wu and Hsu (2009). However, in this case it was extended up to ten decomposition levels. This number of decomposition levels was chosen as a trade-off that aimed at applying a reasonable number of inputs for the purpose of the neuro-fuzzy system application. It also aimed to ensure the availability of adequate data so that loss of essential information from the original vibration signal was avoided (Marichal *et al*. 2011).

The MRA scheme, using ten decomposition levels, is depicted in Figure 5.4. The resulting cA parts (i.e., cA1 – cA10) were used to generate the seven features that were needed to train the ANFIS model for the purpose of fault diagnosis and classification.

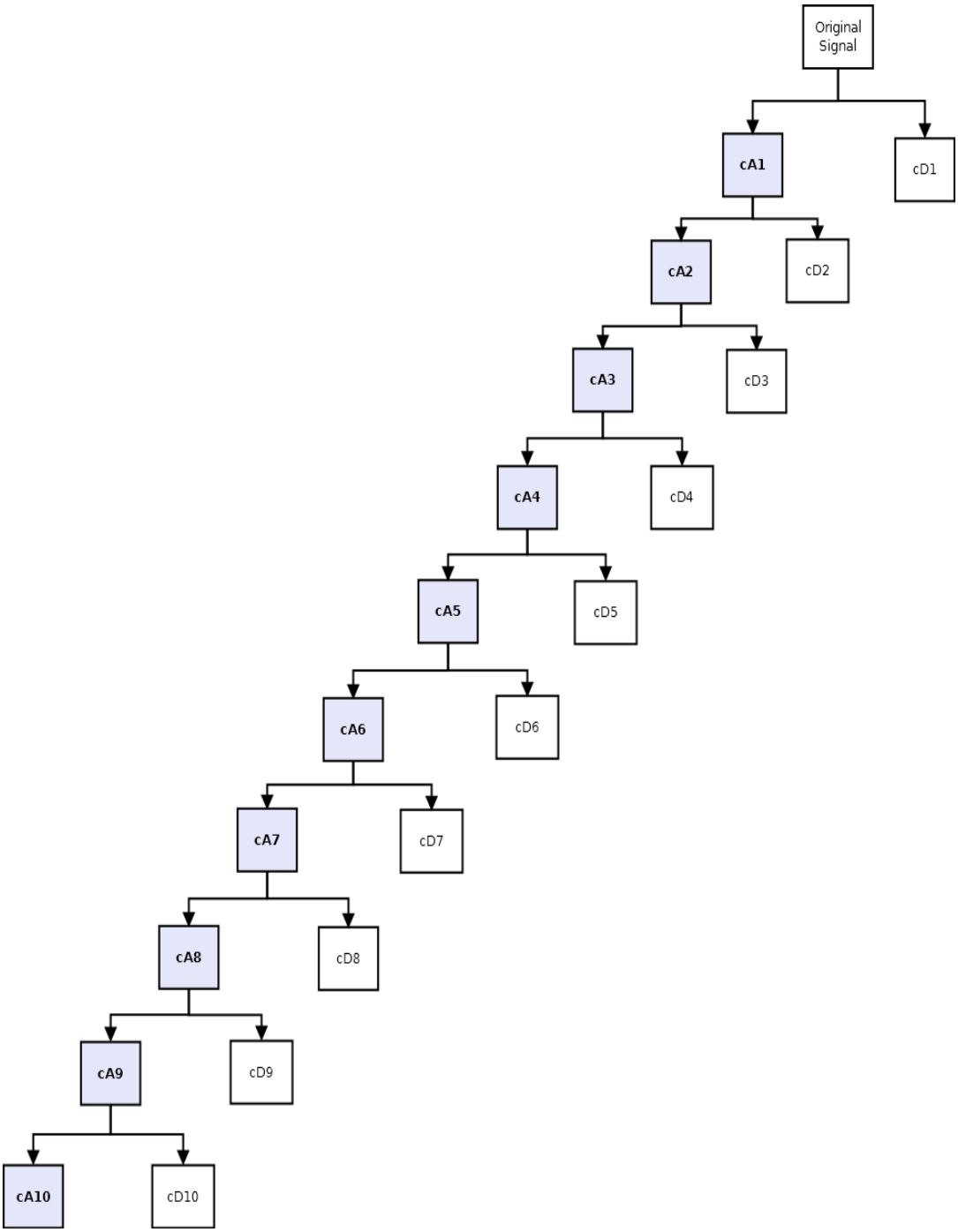

Figure 5.4 Ten-level multi-resolution analysis (MRA)

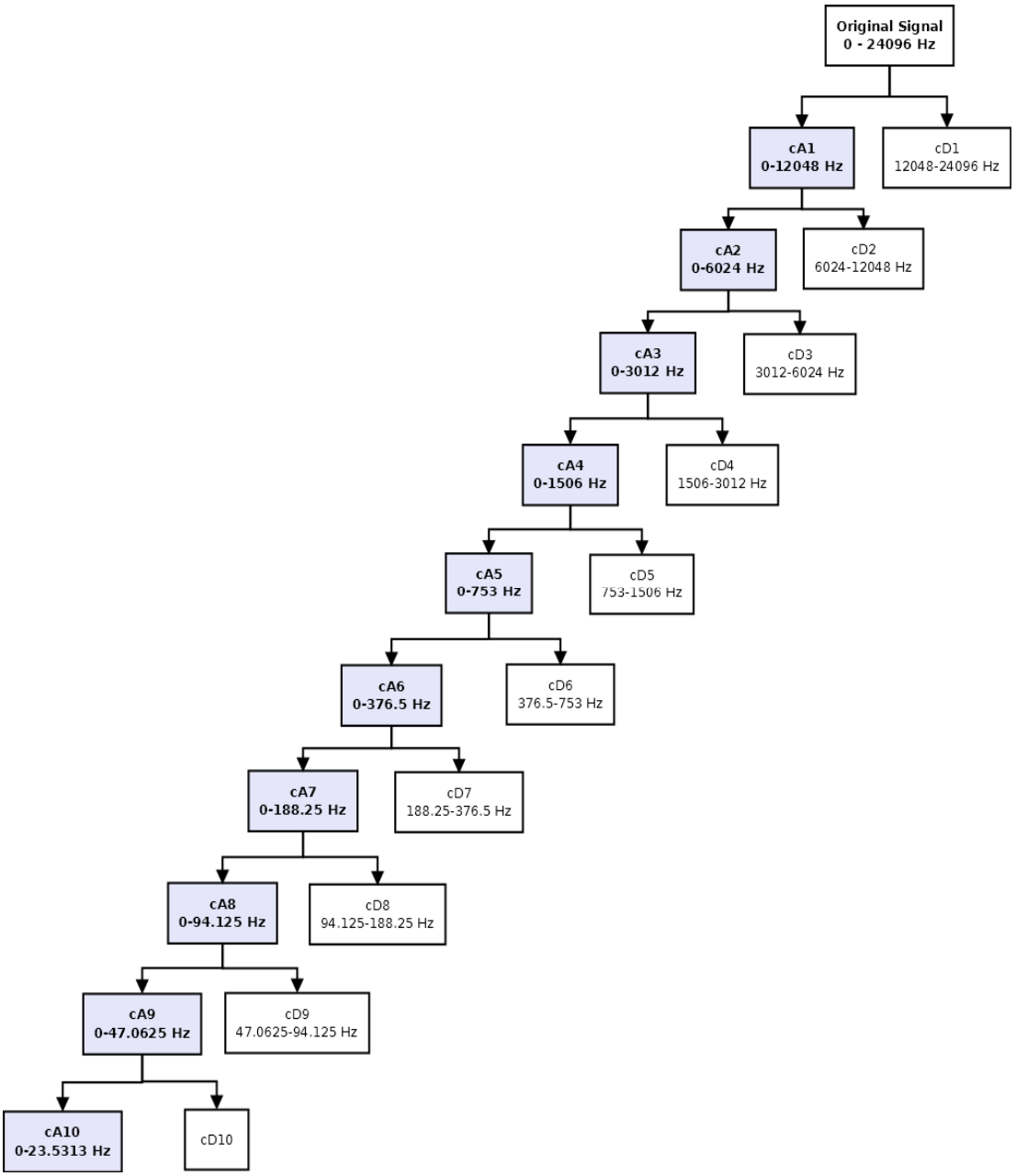

Figure 5.5 Frequency band separation of 10-level MRA

Figure 5.5 shows the frequency band divisions of the ten levels of the MRA. The tabulated values are shown in Table 5.1.

Table 5.1 – Frequency Bands of cA Parts

Segments / parts	Frequency Bands (Hz)
Original signal	0 – 24096
cA1	0 – 12048
cA2	0 – 6024
cA3	0 – 3012
cA4	0 – 1506
cA5	0 – 753
cA6	0 – 376.5
cA7	0 – 188.25
cA8	0 – 94.125
cA9	0 – 47.0625
cA10	0 – 23.5313

In this research, the computation of the wavelet transforms was carried out using a freeware Wavelet Toolbox for Matlab called Uvi Wave, http://cas.ensmp.fr/~chaplais/UviWave/About_UviWave.html , which was developed by the University of Vigo in Spain and used by Jensen and La Cour-Harbo (2001). Version 3 of Uvi Wave was used for wavelet computation in this research. The Uvi Wave wavelet toolbox was preferred over Matlab's version since the codes of the function scripts were open, hence the scripts were easier to modify to suit the research needs.

5.3 The Seven Features

Each of cA-i^{th} parts resulting from i^{th} decomposition level was used to calculate the seven features that were used in the ANFIS training process. The training process was implemented to identify the most dominant features that represented the characteristics of the vibration signals. Figure 5.6 shows the features that were generated using the proposed method.

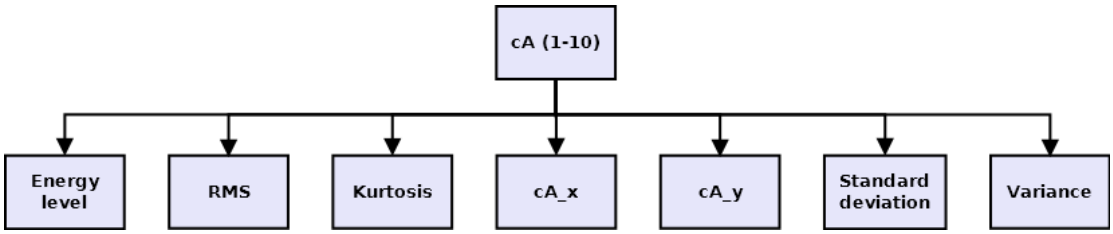

Figure 5.6 The seven features generated from cA parts of level 1 - 10

The explanation of each feature is given below.

5.3.1 Feature 1: Energy level

In this research, energy levels were one of the features generated from the wavelet transform result. They were calculated for each cA part of the wavelet results, using the modified formula of Equation 5.1, reported in Latuny and Entwistle (2010).

$$Energylevel A_i = \left(\frac{1}{2}\right)^{p-i} \times \sum (A_i)^2 \qquad (5.1)$$

where

p = next power of two based on data length

i = decomposition level ($i = 1, \ldots, n$)

A_i = Approximation (A) result of wavelet transform at i^{th} level

5.3.2 Feature 2: RMS

The RMS of the cA-n part was calculated using the formula (Howard, 1994). It evaluated:

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \qquad (5.2)$$

where N was the number of signal data points. $\bar{x}$ was the mean value of the signal and x_i was the i^{th} element of the data set.

5.3.3 Feature 3: Kurtosis

Kurtosis was calculated using the available Matlab built-in function. It was evaluated as follows:

$$Kurtosis = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^4}{(N-1)\sigma^4} \qquad (5.3)$$

where x_i was i^{th} data point of the data set, N was the number of data points, $\bar{x}$ was the mean value of the signal and σ was the standard deviation.

5.3.4 Features 4 & 5: *cA_x*&*cA_y*

The feature labelled *cA_x* is the dominant frequency (the frequency which has the highest amplitude magnitude) and *cA_y* is the amplitude of that dominant frequency for each cA-n wavelet transform result.

These two features were introduced to investigate their significance as features suitable for bearing fault diagnosis. The motivation for investigating these two features was that for each cA of the wavelet transform result, the frequencies were found to be different from one condition to another. The inclusion of these two features was also based on the findings by Rafiee and Tse (2009) which showed that different classes of faulty signals produced different amplitudes in their dominant frequencies and harmonics as well as in the related side bands.

The *cA_x* and *cA_y* were extracted from the FFT result of each cA-n wavelet transform part. The dominant frequency and the corresponding amplitudes of the cA parts were taken as two additional features.

5.3.5 Feature 6: Standard Deviation

The standard deviation was calculated using the available Matlab built-in function. It evaluated:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}} \qquad (5.4)$$

where x_i was each value in the data set, $\bar{x}$ was the mean of the data set, N was the number of data points.

5.3.6 Feature 7: Variance

Variance was calculated using the available Matlab build-in function. It evaluated

$$Variance = \frac{\sum (x_i - \bar{x})^2}{N} \qquad (5.5)$$

where $\bar{x}$ = mean of the data set, $x_i$was each value in the data set and N was the number of data points.

5.3.7 Numerical examples of the Seven Features

The feature values shown in Table 5.2 were calculated from the data set dt2111 at sequence number (data set number) 492 using a db4 wavelet type. This data set is explained fully in Section 5.4. The row labelled cA7 is highlighted, showing that the maximum energy level yielded was in the cA7 results.

Table 5.2 – Sample of db4 Numeric Values of the Seven Features

db4	**Energy** (mag)	**RMS** (mag)	**Kurtosis** (mag)	**cA_x** (Hz)	**cA_y** (mag)	**Std Dev** (mag)	**Variance** (mag)
cA1	0.000202	0.01172	15.78845	8001	37.09885	0.01172	0.000137
cA2	0.001086	0.019216	22.29063	3538	78.15219	0.019216	0.000369
cA3	0.005221	0.029791	27.68568	2277	141.6847	0.029791	0.000887
cA4	0.002023	0.013113	22.93234	1267	70.06699	0.013113	0.000172
cA5	0.001475	0.007918	7.300608	473	67.00256	0.007918	6.27E-05
cA6	0.006111	0.011395	12.47108	282	117.1511	0.011395	0.00013
cA7	**0.014211**	0.012288	9.98194	183	147.3439	0.012288	0.000151
cA8	0.01276	0.008233	4.198542	51	1.39E+02	0.008233	6.78E-05
cA9	0.011877	0.005617	3.117982	45	105.0766	0.005617	3.15E-05
cA10	0.001383	0.001355	3.561562	20	22.99631	0.001355	1.84E-06

5.4 Visualisation of the Seven Features

Examples of these features are shown in Figure 5.7 and 5.14. The examples were based on vibration signals which coded dt2111 at sequence number (data set number) 492. Figure 5.7 shows the cA parts FFT results of levels 1 - 5. The dominant frequency or the frequency which had the highest amplitude magnitude is highlighted with a red circle. Figure 5.8 shows the cA parts FFT results of levels 6 - 10.

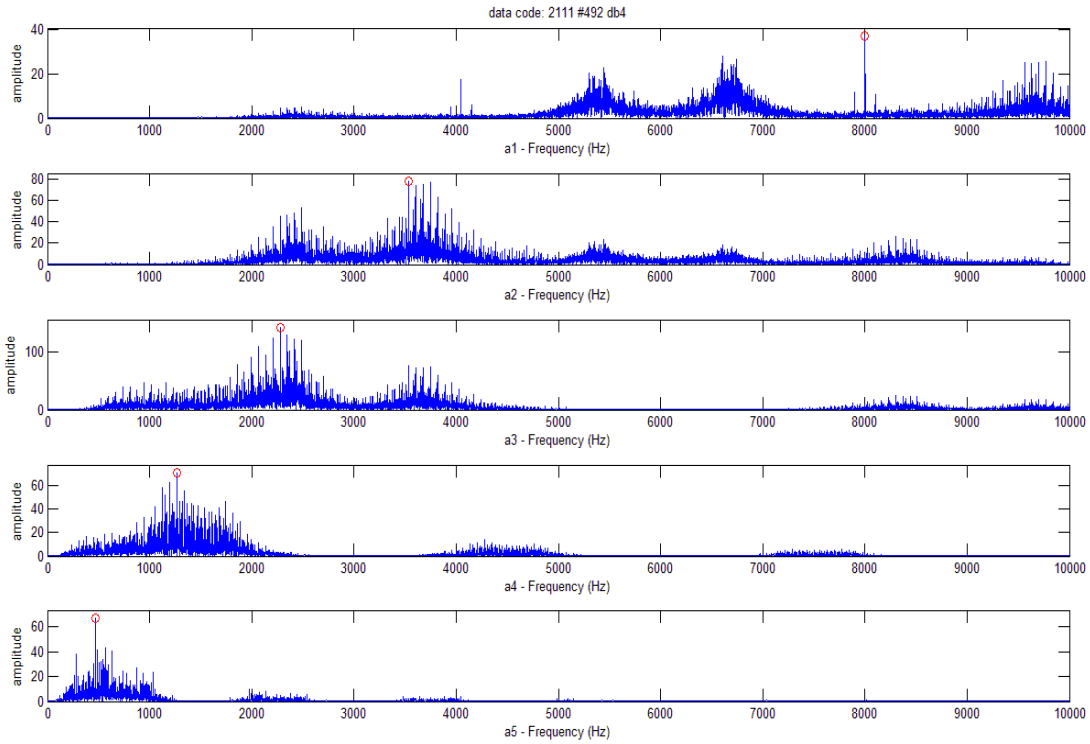

Figure 5.7 Sample of FFT results of cA parts (cA1 - cA5)

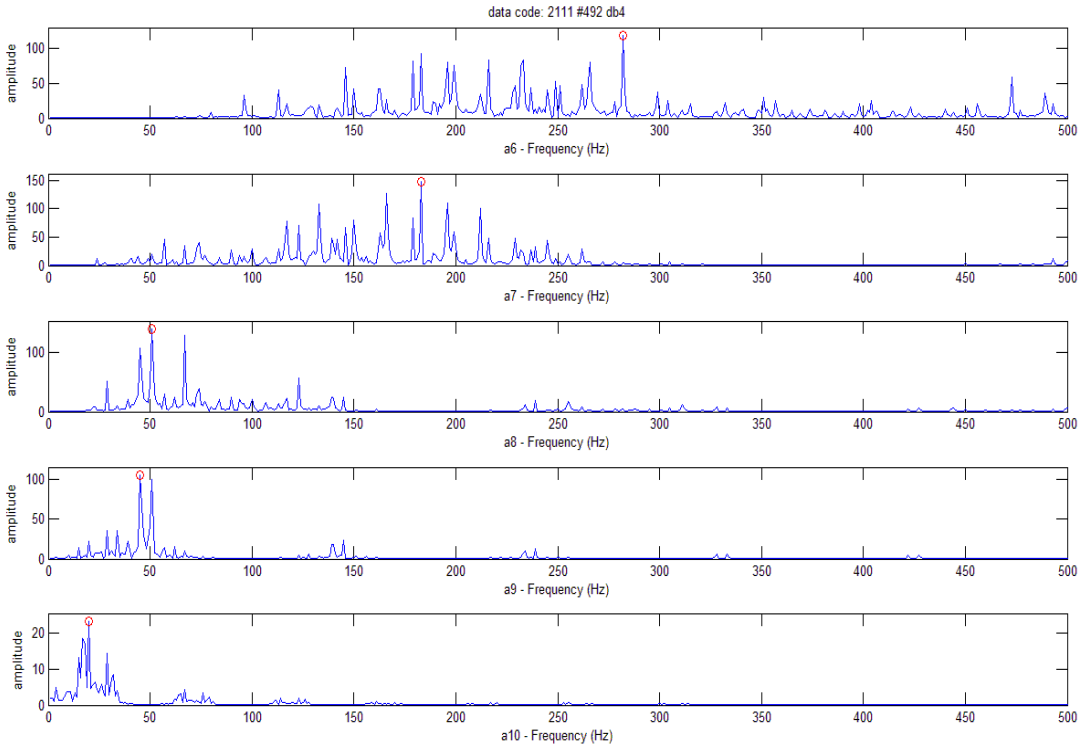

Figure 5.8 Sample of FFT results of cA parts (cA6 - cA10)

The visualisation of values in Table 5.2 is depicted in Figure 5.9 and 5.10.

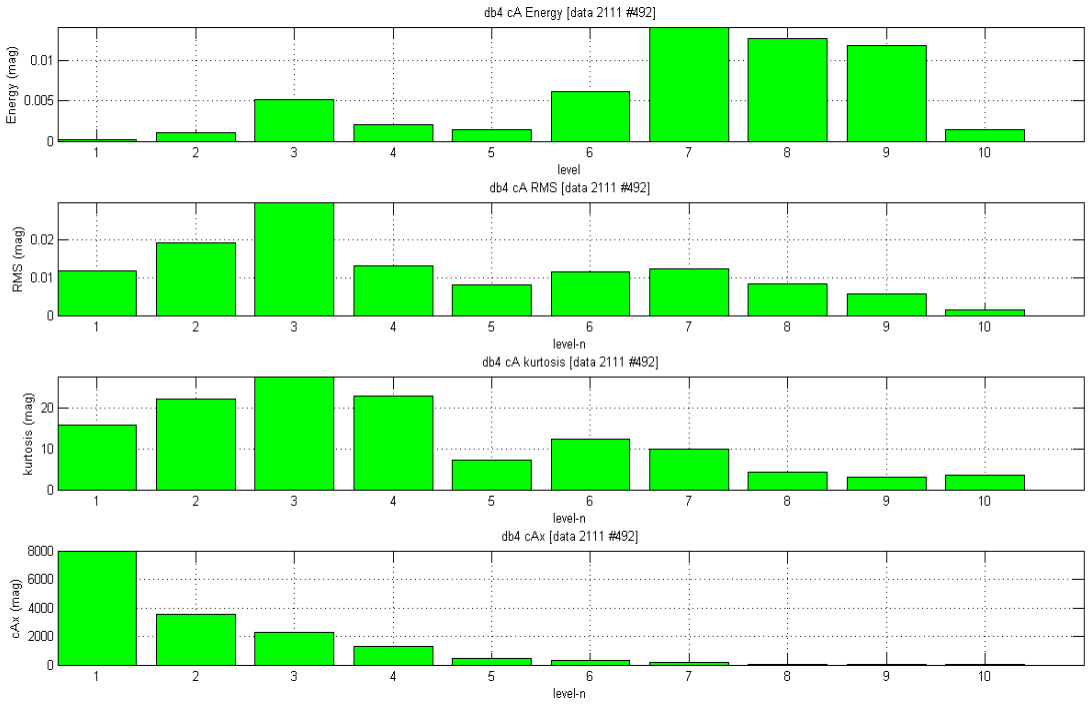

Figure 5.9 Plot of features of Table 1 (part 1)

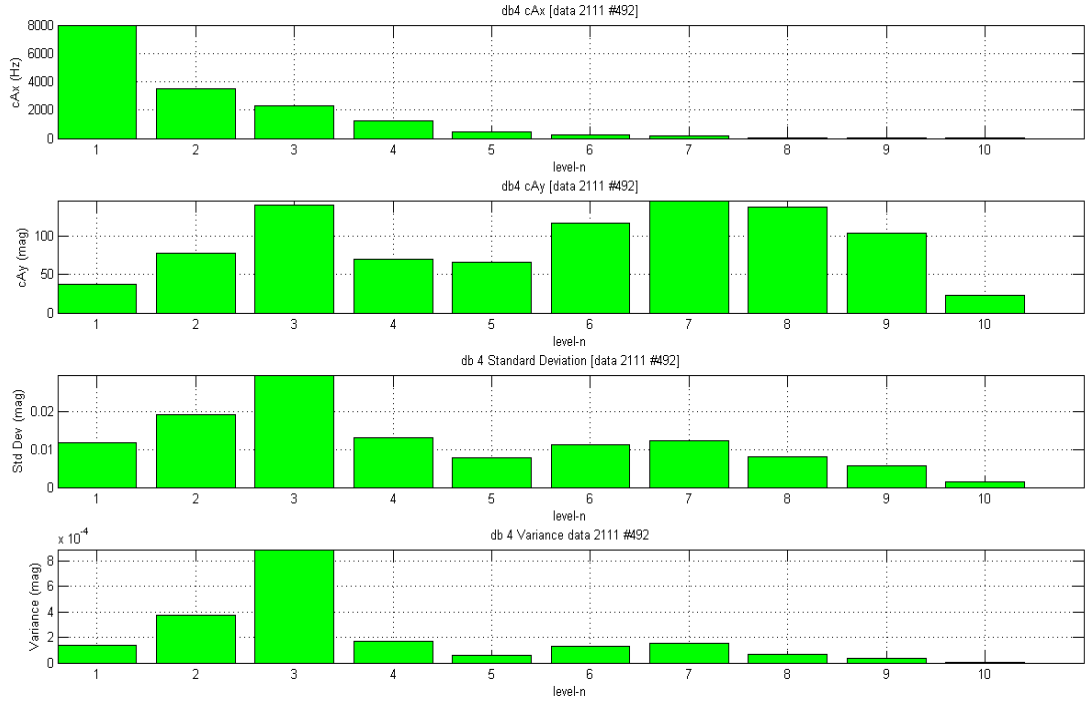

Figure 5.10 Plot of features of Table 1 (part 2)

Samples of the seven features applied in this research for outer race faults (ORF), inner race faults (IRF), ball faults (BF) and no fault (NF) cases are shown in Figures 5.11 –5.14.

Visualisation of the features samples is presented to show the randomness in the characteristics of each feature for each fault case investigated. It is difficult for a human expert to draw any particular pattern from each feature's characteristics except for the *cA_x* which shows a pattern.

In order to overcome the difficulty in assessing the randomness of the feature patterns, the ANFIS was then used to learn and obtain conclusions regarding the relationships of the seven features.

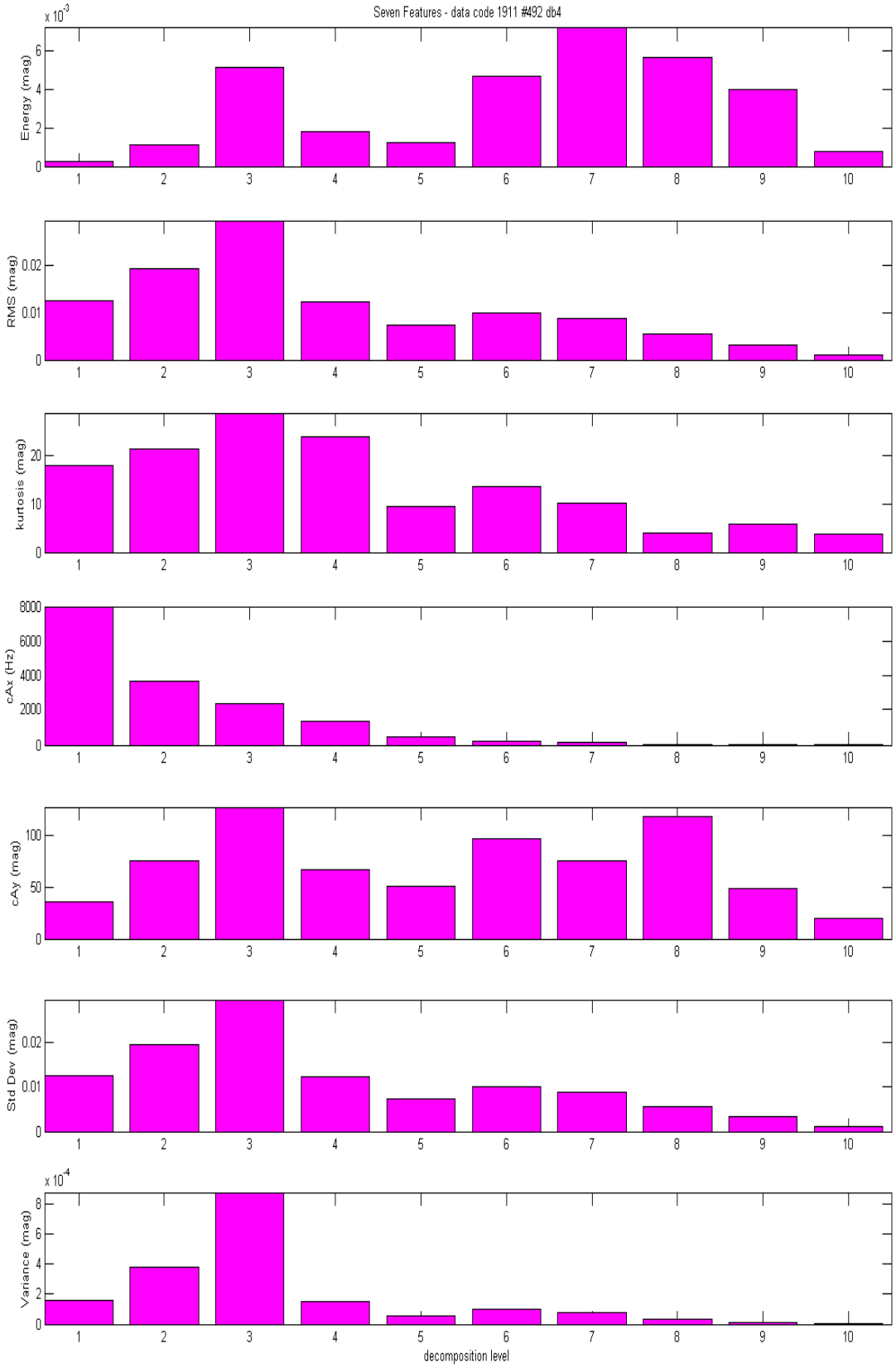

Figure 5.11 Sample of seven features of ORF

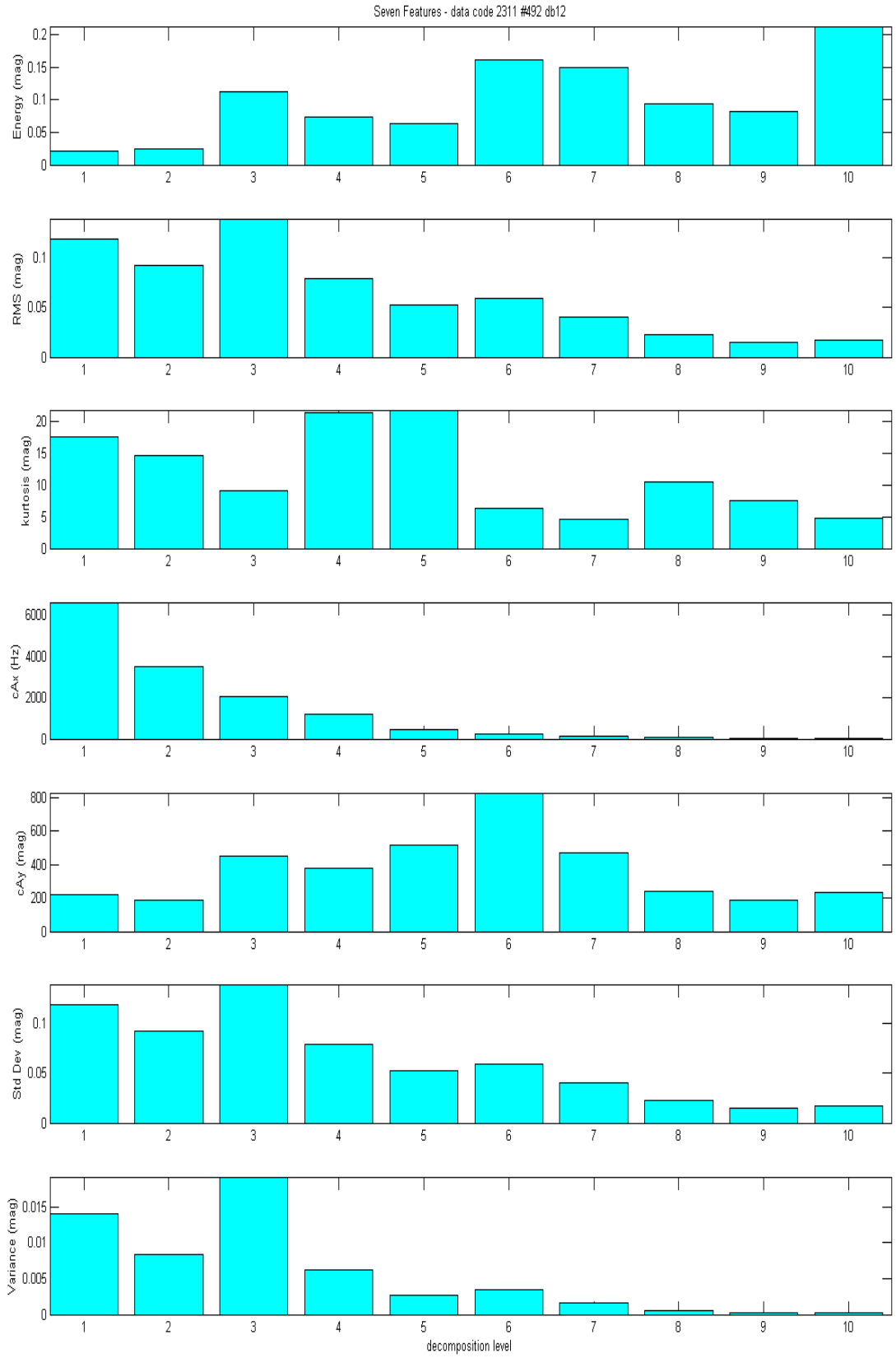

Figure 5.12 Sample of seven features of IRF

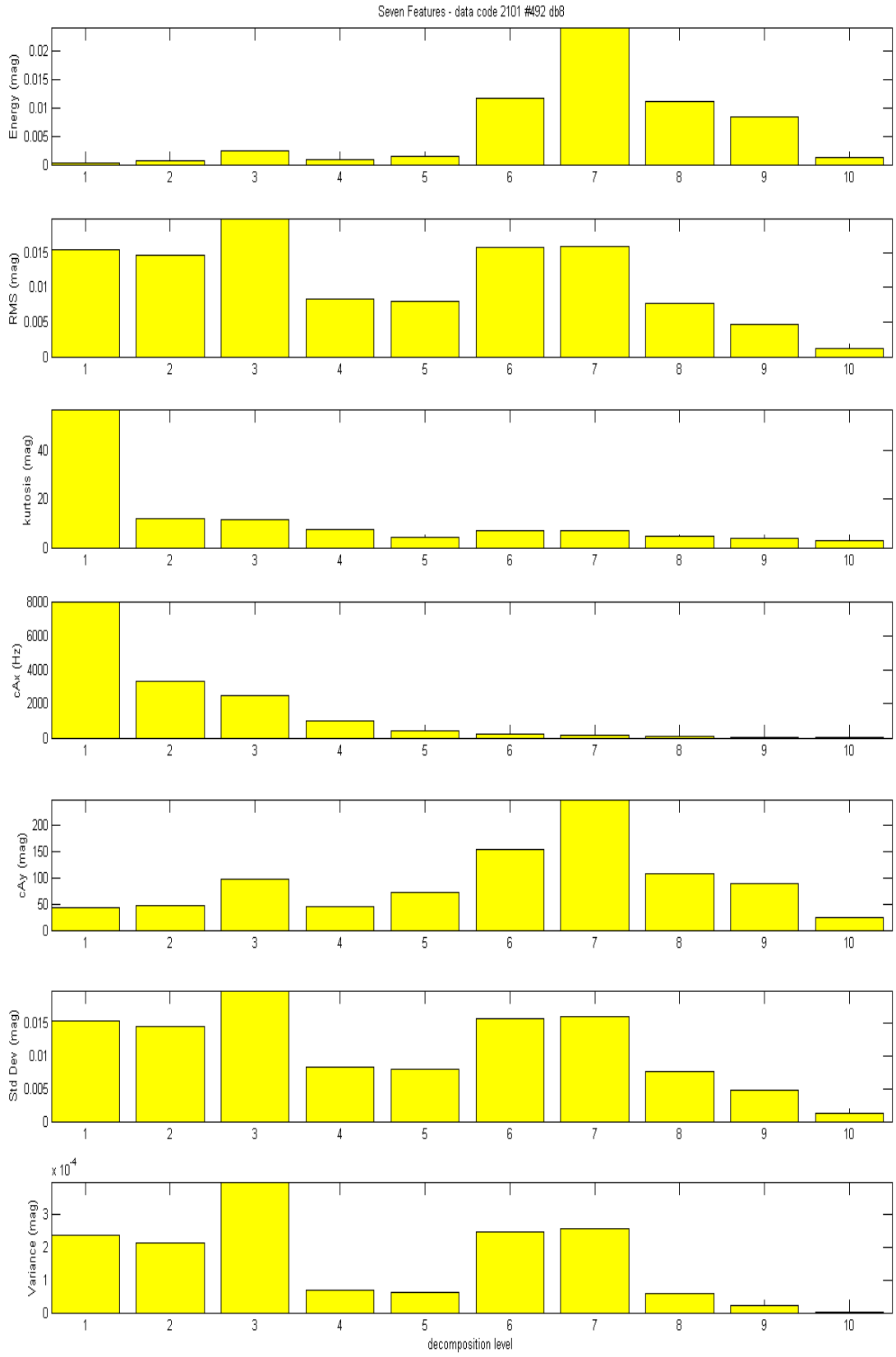

Figure 5.13 Sample of seven features of BF

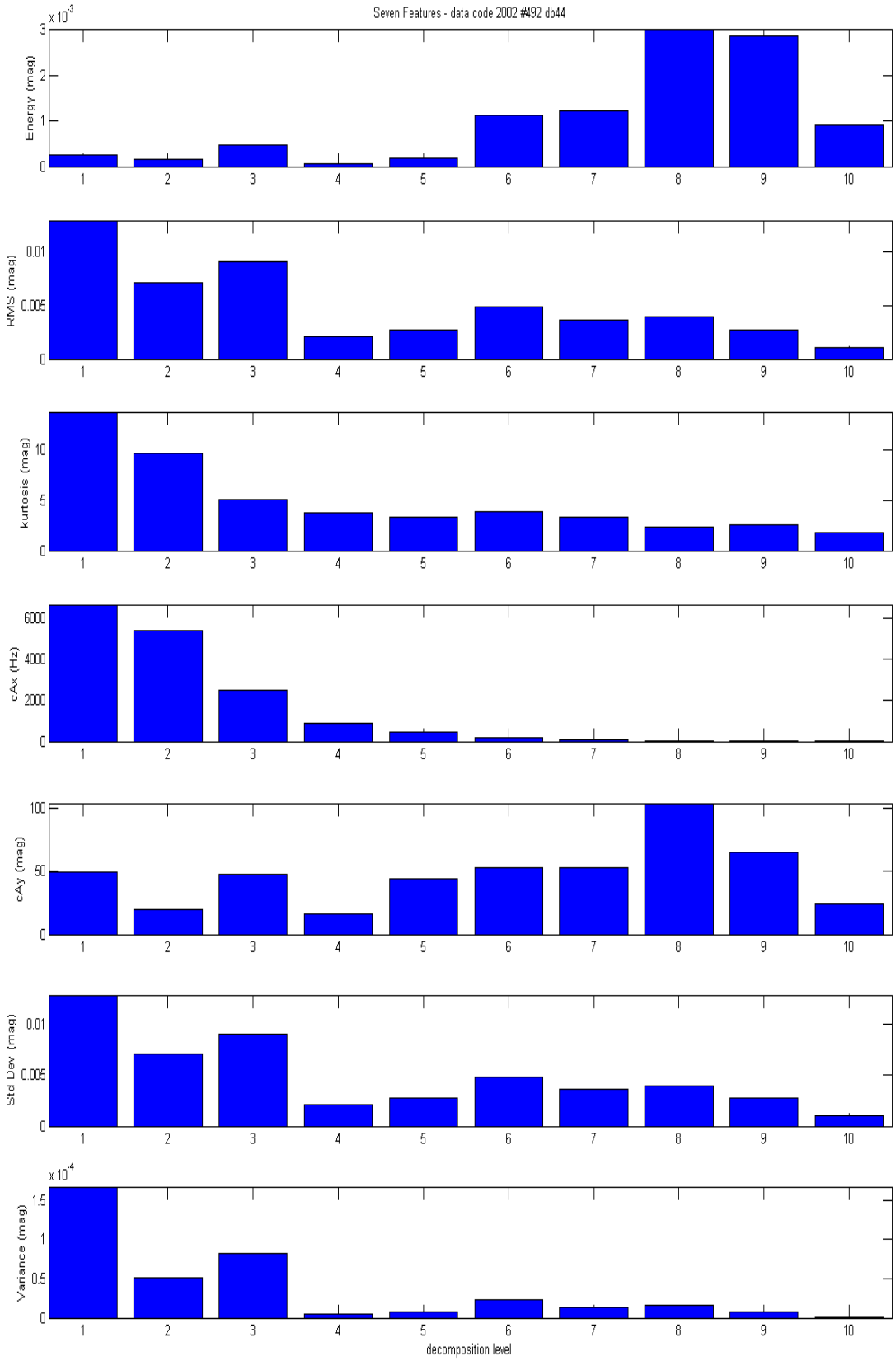

Figure 5.14 Sample of seven features of NF

The integrated training scheme consists of two main parts. The first part (part A) is shown in Figure 5.15, and part B is shown in Figure 5.16.

All of the raw vibration signals analysed in this study which comprised Outer Race (OR), Inner Race (IR) Ball Fault (BF) and no fault (NF) vibration signals, were processed using this integrated scheme. The Matlab code for the integrated training is listed in Appendix 3.

5.5 Integrated ANFIS Training Scheme

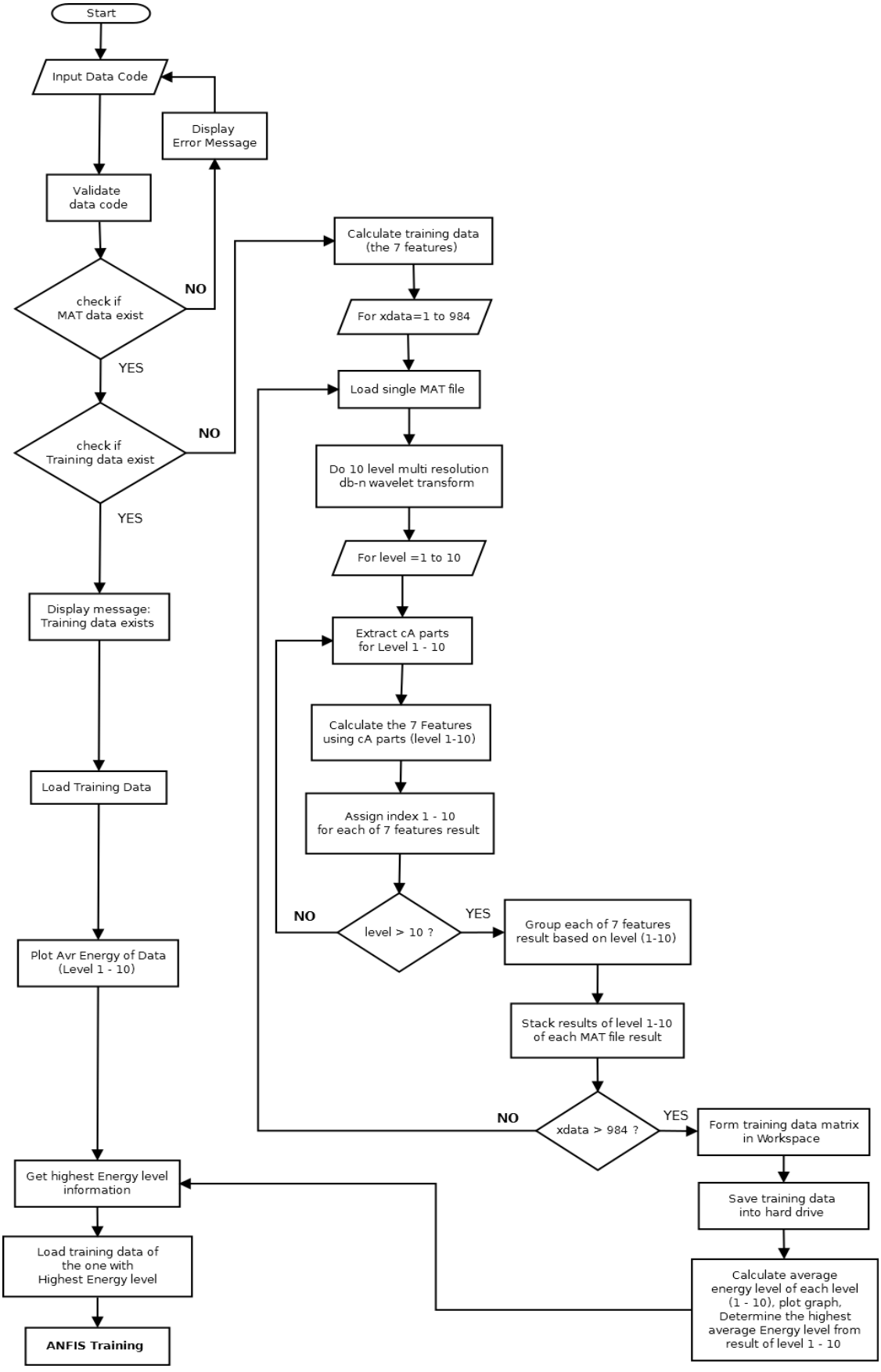

Figure 5.15 Integrated ANFIS training scheme (part A)

The process starts by choosing the data code that corresponds to the data identifier of a stored raw vibration signal acquired from the test rig. A routine check was performed to determine whether the input data code referred to an existing data file (mat file) of the stored raw vibration signals. The script would show an error message where a problem was identified. The process continued in the form of checking to confirm the data that had already been processed.

A routine loaded each single mat data file of the 984 mat data files and the vibration signals were processed from data segment no. 1 up to data segment no. 984. This cycle is shown in Figure 5.15. Each of the loaded mat data file (raw vibration data) was then transformed using a db-n wavelet transform up to 10 decomposition levels. The result of this wavelet transform using multi-resolution algorithm produced a composite matrix in a Matlab workspace that contained both Approximation (cA) and Detailed (cD) parts for decomposition level 1 to 10.

The process to extract the corresponding cA and cD parts was carried out in ten cycles (cycle 1 to 10), and each cycle produced corresponding cAn and cDn parts (i.e., cA1 & cD1, cA2 & cD2, …, cA10 & cD10). All the data for each vibration signal from sequence 1 to sequence 984 was processed using five Daubechies (db-n) bases, i.e., db4, db8, db12, db22 & db44, and the decomposition level used was up to 10 levels.

The process continued with the calculation of the seven features using the Approximation (cA) part of the wavelet transform. It is reiterated that in this study, only the cAn part of the multi-resolution wavelet transform was used to generate the seven features.

An indexing scheme was provided to index each features calculated within cycle 1 – 10. For instance, there was rms1, rms2, …, rms 10 which represented the RMS feature at decomposition levels 1, 2, …, 10. A similar indexing scheme was also implemented for the other features calculated from the cA parts in decomposition levels 1 to 10.

After 10 cycles of extraction and indexing of the calculated features, the results were grouped based on the type of each feature (i.e., 10 values of energy level, 10 values of RMS, etc) and each of 10 value groups were stacked as a matrix in the Matlab workspace for the later use.

For each single mat data file processed, a matrix of 1 row × 70 columns of feature data was produced. Since there are 7 features, 70 columns are needed to store each of 10 values (7 × 10 = 70).

The 70 columns contain values for Energy magnitude for levels 1 – 10 which were stored in column 1 to column 10 of the data matrix. Columns 11 – 20 of the data matrix contained 10 values of RMS magnitude, and the sequence continued like this for all of the features.

The overall process continues until 984 mat data files of the selected raw vibration data were processed. It produced a data matrix with 984 rows × 70 columns for each data code identifier (raw vibration data) processed. This was saved as a text file into the hard drive, and each file was given an appropriate naming scheme for the purposes of identification and later use in the ANFIS training in part B of the integrated training scheme.

The calculation of the average energy level value for each decomposition level value was carried out using the existing calculated features data. The average energy level information for all decomposition levels was then evaluated to obtain which had the highest level. For the purposes of visual interpretation, a plot was drawn up to show the average energy level of levels 1 – 10.

The information on the highest energy level was obtained from the data of levels 1 – 9 only (Level 10 data was not included because of its high value trend (Wu and Liu, 2008)). The information was then used to determine which decomposition level data of the seven features would be used for the ANFIS training procedure (in part B). For example, if the highest average value of energy level existed in the result of level 8 then the features data of level 8 would be used in the ANFIS training process.

5.6 Detail of ANFIS Training Process

Part B of the integrated process used in building the core bearing fault classifier is the training of the ANFIS. The flow chart of the ANFIS training procedure is shown in Figure 5.16.

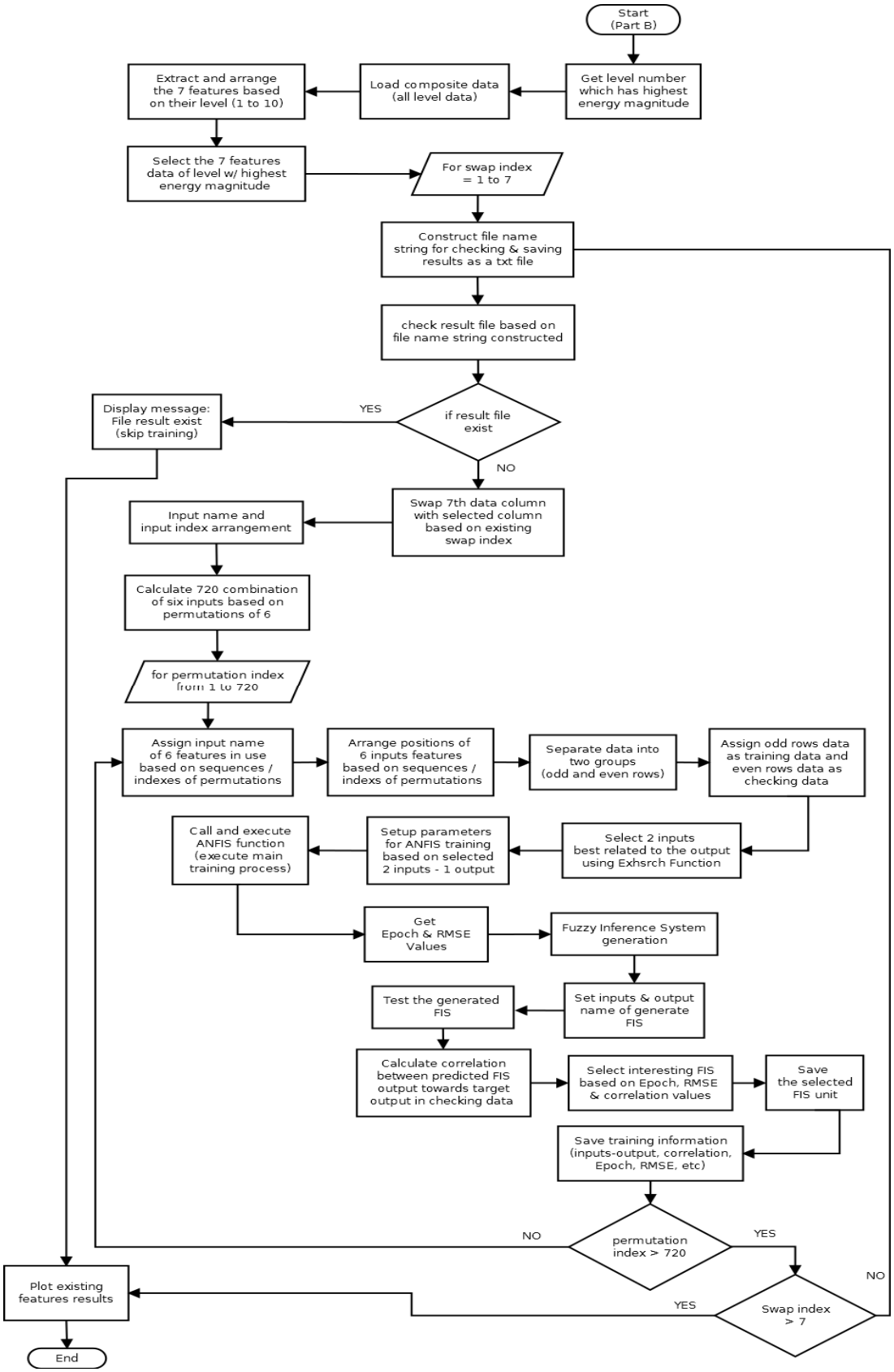

Figure 5.16Integrated ANFIS training scheme (part B)

The ANFIS training procedure continued from part A. The process began by retrieving the information of the highest energy level that occurred in a particular level of the seven-feature data. This was used to determine which corresponding level features data would be used in the ANFIS training.

The process continued with the grouping process of the seven features data according to their decomposition levels (1 – 10) with each of the data calculated for ease of storage and retrieval. This process produced 10 groups of matrix data in the workspace. Each of these 10 matrix groups contained the values of the 7 features at levels 1 to 10.

The size of each matrix group was 984 rows × 7 columns. The seven columns represented the seven features in the 984 rows representing the number of original raw vibration data sets used to generate the features by using a multi-resolution db-n wavelet transform.

The selection of which matrix data group (among group 1 - 10) would be used in the ANFIS training was determined by the index (number) of the highest average energy level that was calculated in part A. The number or index of the decomposition level that achieved the highest average energy level was used as an index to select among the 10 matrix group data sets available. For example, if the highest average energy magnitude existed in the result of the 8^{th} decomposition level, then seven features of level 8 would be used as training data for the ANFIS training process.

The process entered seven iterations or loops; the number of cycles needed to accomplish complete position swapping for each of the seven features. At the beginning of each cycle, a routine procedure was used to check whether the results data of the particular swap-cycle existed or had been calculated before. The results data of a complete seven-feature calculation were stored in the form of text files on the hard drive. If the results data file existed, a message was generated to show that the result existed and the script would skip the calculation process for the particular swap cycles. The whole procedure was then resumed by returning to Part A and beginning again.

There were seven data columns that represented the seven features in the data matrix, as shown in Figure 5.17. The arrangement in Figure 5.17 was the initial column arrangement, by which column c1 was assigned to energy level values, column c2 to RMS values, column c3 to kurtosis, column c4 to cA_x values, column c5 to cA_y values, column c6 for Standard Deviation values, and column c7 to Variance values.

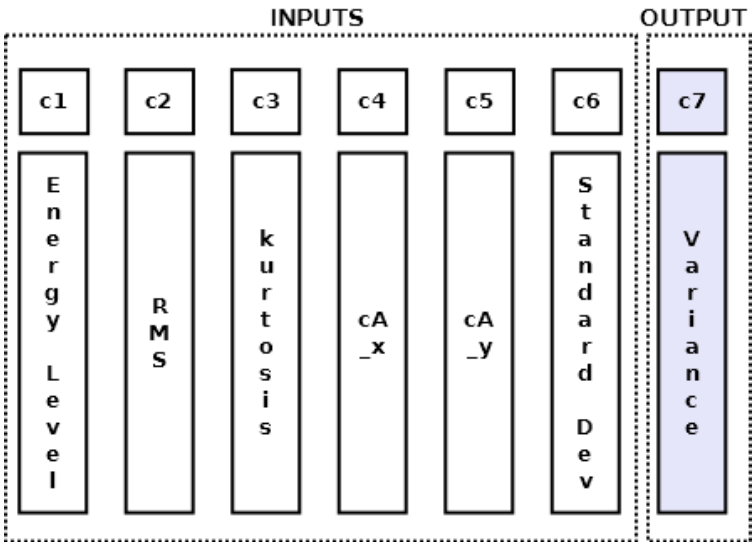

Figure 5.17 Columns assignment for ANFIS training data

All of the features in column 1-6 also needed to be rotated in turn to the Target Output except the one which was already at column 7.

The assignment of the last column as the Target Output was based on the requirement for ANFIS training data in the Supervised Training scheme. The arranged data was configured as 6 inputs – 1 output format, and the arrangement of the column data that matched the ANFIS training requirements was constructed as shown in Figure 5.17.

In Figure 5.18, the first 6 columns of data matrices were assigned as inputs and the last column (column 7) was assigned as the Target Output as required for the construction of the ANFIS training data.

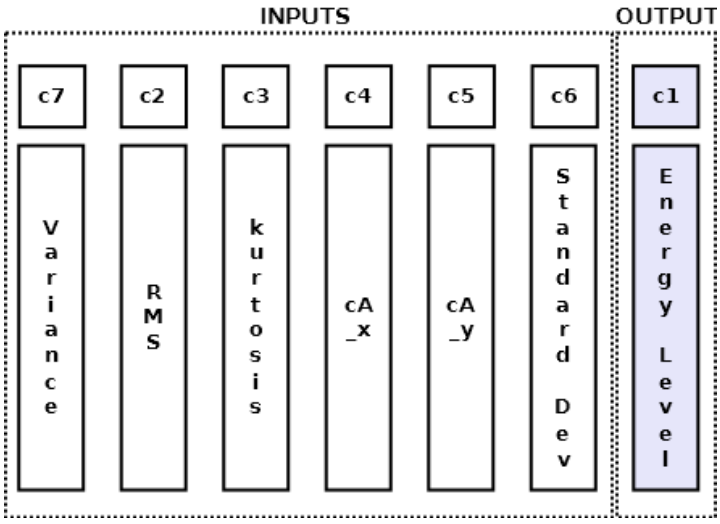

Figure 5.18 Swapping data of column 1 to column 7

The data sets were rotated in sequence so that each had its turn of being the target output. The order of the remaining six sets remained intact, as illustrated in Figure 5.19.

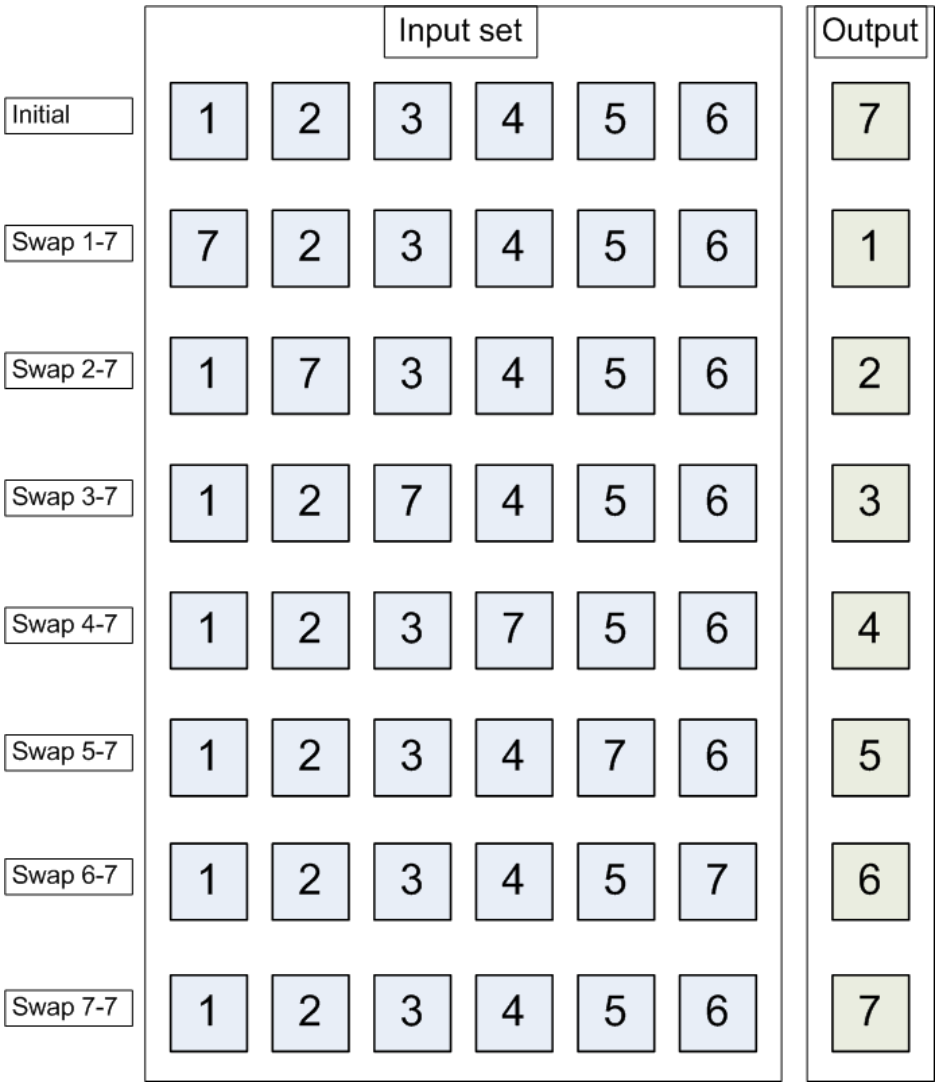

Figure 5.19 Graphical illustration of swapping process

5.7 Permutations Check for Six Inputs

In each of the seven swapping process iterations, a routine to check possible combinations of the six inputs was carried out. The purpose of the permutations check was to test all of the possible combinations in which 6 input features were arranged in the data column matrix. This was required as the ANFIS function is sensitive to the order / sequence / position of the data column when used in a training session.

This checking procedure was needed to ensure that all possible combinations of six inputs and a single Target Output were completed and presented to the ANFIS during the training process.

In each of the permutation combinations check process, an FIS unit was produced based on the corresponding six inputs and a single Target Output arrangement determined by the indices of the six permutations.

Using the permutation indices of six variables (input), there were 720 possible combinations of six input arrangements.

This would produce 720 FIS units during each rotation (swapping) of the Target Output. The 720 FIS units resulting from a complete permutations check process were later processed in order to select the best FIS candidates to be used as bearing fault classifiers.

The permutations check of 6 input combination positions were processed in each swapping step as shown in Figure 5.20.

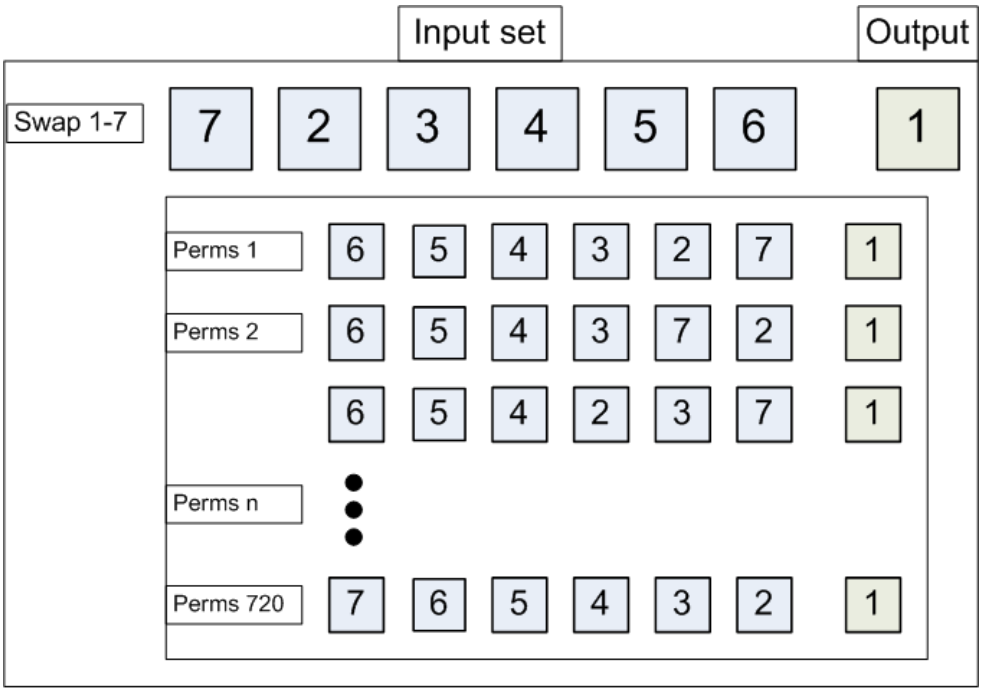

Figure 5.20 Permutation check of possible six input combinations in swap 1-7

Figure 5.20 shows an example using the swap 1-7 procedure. There were 720 possible combinations of the 6 inputs arrangement. For permutation #1 (Perms 1), there were 6 inputs which were arranged in the sequence of [6 5 4 3 2 7]. It was presented together with a feature (feature #1) which had been swapped to column 7

as the Target Output. The original data column 7 had been swapped to the position of data column 6 in the inputs data of the sequence [6 5 4 3 2 7].

The process then continued to enter a swap 2-7 procedure and the detail of the 6 inputs – 1 output arrangements are shown in Figure 5.21.

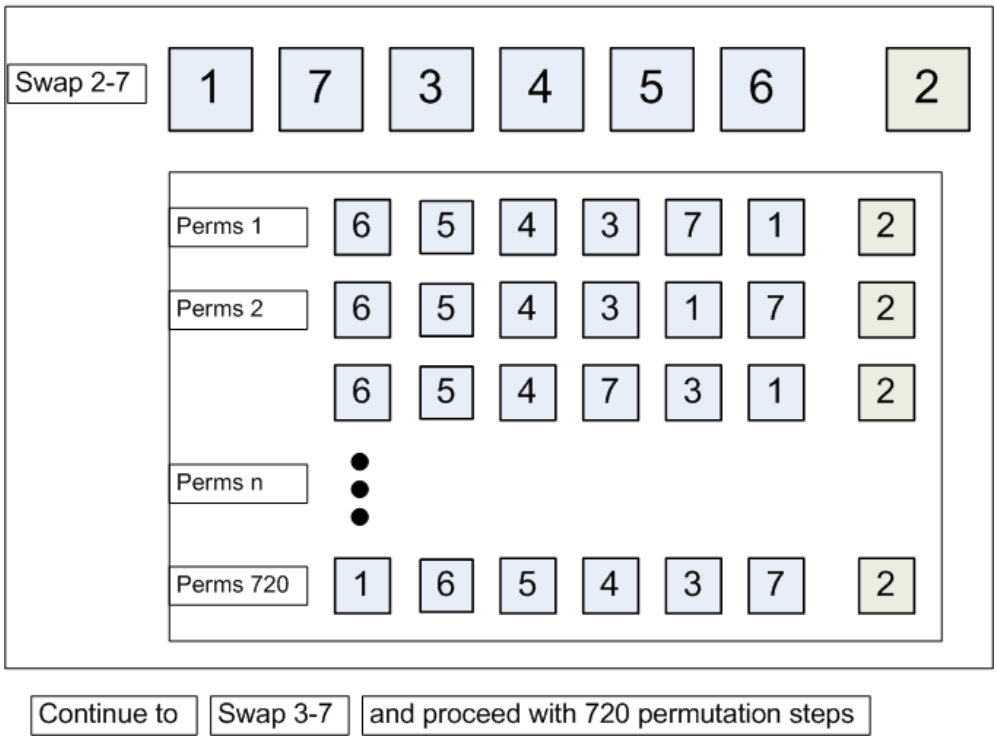

Figure 5.21 Permutations check of possible six input combinations in swap 2-7

Figure 5.21 shows the continuation of swapping procedures, combined with the permutation check in the swap cycle 2-7. In the swap 2-7 process, permutation #1 (Perms 1) had the 6 input index sequence of [6 5 4 3 7 1]. The 6 inputs, arranged in the sequence of [6 5 4 3 7 1] were tested against feature variable #2 which had been swapped to data column 7 to become the Target Output. The previous data column 7 had been swapped to data column 5 in the inputs of the sequence [6 5 4 3 7 1].

The process was carried out until 720 iterations had been completed for each of the swap cycles which started from Perms 1: [6 5 4 3 7 1], continued to Perms 2: [6 5 4 3 1 7], Perms 3: [6 5 4 7 3 1], and finished at Perms 720: [1 6 5 4 3 2 7], as shown in Figure 5.21.

5.8 Selection of the Dominant Two Inputs and Output

Prior to the main training procedure, a process to determine the two best inputs that produced the best input-output correlation was carried out by using the *Exhsrch* function. Exhsrch is a built-in function in Matlab available via the Fuzzy Logic Toolbox. According to the Matlab documentation, Exhsrch performs an exhaustive search within the available inputs to select the set of inputs that most influences the Target Output data.

The search for the two most related inputs was carried out by the *Exhsrch* function based on the lowest error achieved by combining two input features as shown in Figure 5.22. This example is a snapshot of the process of searching for the best two inputs which had the lowest RMS Error (RMSE). In this example, Variance and Standard Deviation are shown to have achieved the lowest error among all the other combinations of the two input features. Therefore, in this case, Variance and Standard Deviation were the two features most related to the Target Output, Energy level.

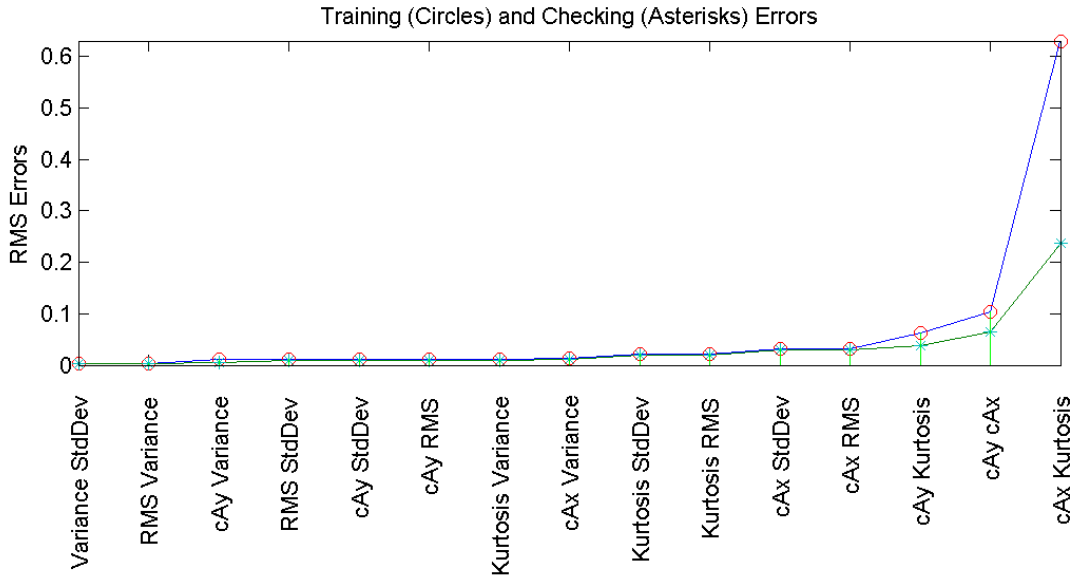

Figure 5.22 Search for the lowest error using Exhsrch function

The *Exhsrch* function was invoked for each permutation test for the six inputs to determine the two best related inputs towards a Target Output.

In detail, Exhsrchwas used to select the two best inputs among the 6-input feature which had the most influential connection with any given Target Output of the features data. Exhsrch may also be seen as a process for testing for the dominant relation between two selected features (out of 6) and a Target Output (swapped feature between column 1 - 7).

The information regarding the two best selected inputs (labelled Input 1 and Input 2) in relation to the Target Output were then used in the main training procedure. For this purpose, indices of the two selected inputs representing numbers of column positions of the training data and checking data were used. The column number of Target Output data was permanently assigned to the data column positioned at column 7. The content of data column 7 changed based on the swap-cycle sequence (1 - 7) encountered in the training process.

In the core process, the Exhsrch function built an initial ANFIS model for each input combination and carried out the training process. The training was performed using only a single epoch and reported the results achieved.

A fuzzy inference system (FIS) unit was generated in the Matlab workspace as the result of the initial training via Exhsrch. The generated FIS unit contained information of the best two selected input features which had the dominant relationship to a feature in data column 7 which was assigned as the Target Output.

Indices of the two selected inputs and the Target Output were extracted from the initial FIS. The index information was then used for the core ANFIS training in part B of the integrated training procedure.

5.9 Training and Checking Data

The core of the ANFIS training process required two types of data to be used. The first, called the Training data, was used in the training process to build an FIS unit. The second, called the Checking data was used to prevent the ANFIS from being trapped in the generalisation of the data (i.e., learning data too well). It was also used for the purpose of testing any FIS produced during the training process.

The provision of these two types of data was carried out by dividing the overall data matrix (which contained 984 rows) of data into two parts. This produced two groups of data where each contains half of the original data (492 rows).

The Training data part was generated by taking only the odd rows from the 984 available rows, whilst the Checking data was generated by taking only the even rows. The process of creating Training and Checking data was carried out in each cycle of the permutations check sequence numbered 1 to 720.

5.10 ANFIS Training Parameters

The ANFIS parameters utilised in the core training process are shown in Table 5.3.

Table 5.3 List of ANFIS Parameters

Parameters	Values	Remarks
Number of membership function	4	
Membership function type	gbellmf (Gaussian bell shaped)	
Training method	1	hybrid
Maximum epoch	100	
Step size	0.01	default
Error tolerance	Matlab default	default
Display result option	0 (not displayed)	default
Step size of decremental rate	0.5	default 0.9
Step size of incremental rate	1.5	default 1.1
Initial FIS type	genfis1	default

The initial FIS needed in the ANFIS training was generated by using the genfis1 function in Matlab. The function generates a single-output Sugeno-type fuzzy inference system (FIS) using a grid partition on the data (no clustering). FIS is used to provide initial conditions for the ANFIS training.

The new FIS generated in each permutation test sequence was then tested using the data of the corresponding two inputs (Input 1 and Input 2) taken from checking data parts. The parameters used to determine whether the new generated FIS in a particular permutation cycle could be used for further evaluation was determined by the sets of numerical requirements which are as follows:

- Root Mean Squared Error: $< 10^{-3}$
- Number of epoch cycle: $5 \leq Epoch < 75$
- Correlation factor: > 0.99

The correlation factor was calculated based on the correlation between the Target Output predicted by the evaluated FIS compared to the Target Output of the training features data.

Each generated FIS that had testing results that satisfied these numerical requirements was then recorded and saved onto a hard drive using the appropriate file naming scheme. The file naming scheme used to save each FIS unit was designed for ease of tracing and subsequent recall (load).

The naming scheme included the following information: a four-digits vibration data code (trn_out_*nnnn*), type of Daubechies wavelet (db*n*) used to process the data, decomposition level index of the data (L*n*), number of swap index sequence (S*n*), and the number of epochs (training cycles) (E*n*) in which the FIS was generated.

An example of the FIS unit name generated is *trn_out_2112_db44_L8_S7_E11_a.fis*, which can be interpreted as the FIS generated, based on ANFIS training using data 2112 (trn_out_2112), features / training data calculated using Daubechies 44 (db44) at decomposition level 8 (db44_L8). The corresponding FIS unit was generated during the swap sequence 7 (S7) at Epoch 11 (E11). In addition, strings *_a* were used for the purpose of an internal identification process in showing which scheme of FIS algorithm was used. The strings *.fis* were the standard file name extensions of the Matlab Fuzzy Logic Toolbox used in saving a FIS file.

The corresponding text file employed the file names of four digits of data code used (fis_rec_*nnnn*), wavelet type (db*n*), decomposition level (L*n*) and swap cylcle number (S*n*).

An example of the text file name was *fis_rec_2112_db44_L8_S7.txt*, by which fis_rec_2112 represents four digits of the original vibration data used, db44

represents the type of Daubechies wavelet used to generate the features data, L8 represents the decomposition level of the features data, and S7 represents the swap index sequence in use during the training step. The extension *.txt* represents the type of file in use in the operating system (i.e., Windows).

All of the .fis and .txt files were later used in the detailed final selection process of all the FIS units obtained. The purpose of the final selection process was to select only the best FIS candidates.

5.11 RMS Error (RMSE) of ANFIS Training and Checking Data

There were two types of data used in the ANFIS training process. The first type of data was training data and the second was checking data.Training data was used to train the ANFIS model in order to produce FIS unit based on the data. The checking data was used to prevent model over-fitting during the training process (Mathworks, 1998). Over-fitting or data generalisation was prevented through the use of the lowest RMS error of the checking data. In this case, the model parameters associated with the minimum value of the checking were used to generate a FIS unit. An example of the lowest checking error value is shown in Figure 5.23.

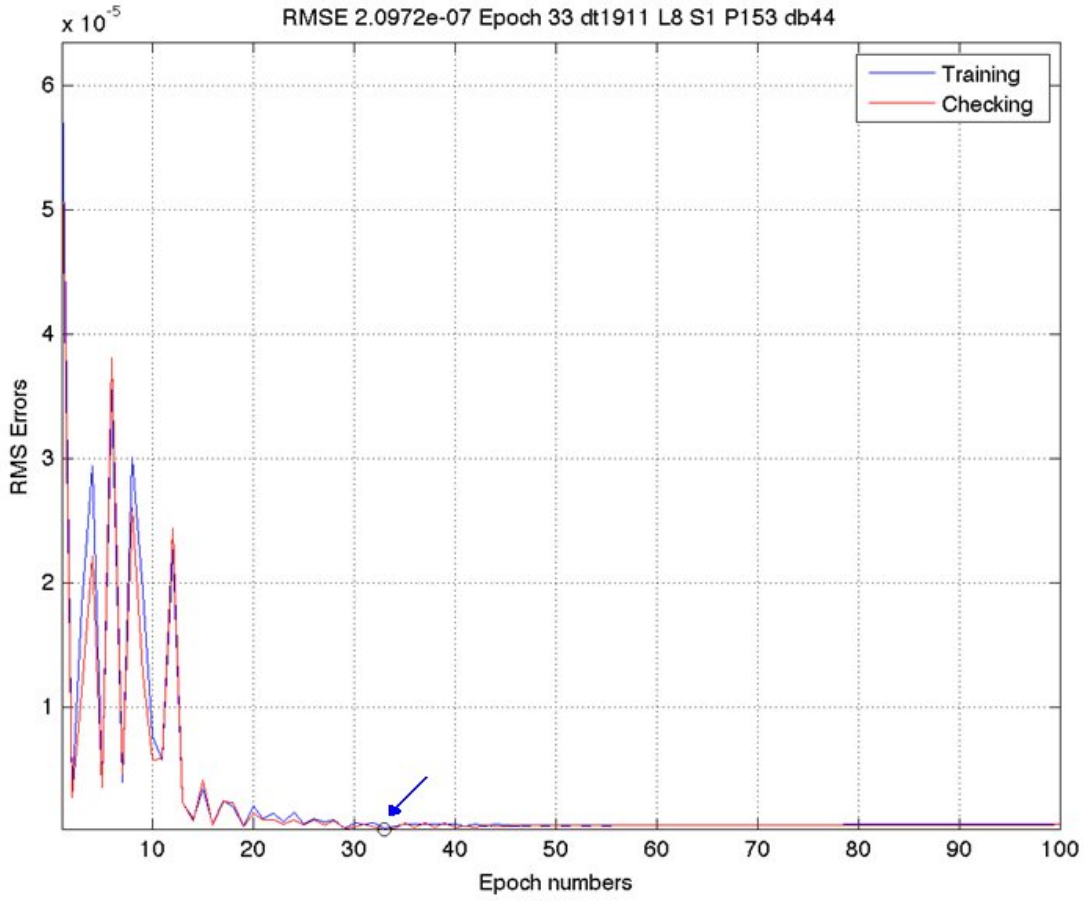

Figure 5.23 RMS error (RMSE) of training and checking data

The lowest RMS error at epoch (iteration) number 32 is shown in a circle in Figure 5.23. At this epoch, the FIS unit was selected as the appropriate result of the training process. FIS units constructed after an epoch of the lowest RMSE were not used or selected in order to prevent the FIS units having over-fitted characteristics. This shows the importance of using checking data in the ANFIS training process utilised in this research.

5.12 Selection of the Best FIS Units

There were hundreds of FIS units produced in the ANFIS training processes. Therefore, a process to reduce the number of the FIS units that needed to be further examined was needed.

The reduction of the number of FIS units was carried out in the second evaluation and selection process. The second evaluation process could be seen also as a

verification process. The process was carried out by increasing the requirements or standards to filter the best useful candidate of the FIS units.

The verification process used an algorithm to load and test the stored and recorded FIS units of each vibration data set processed. The process utilised all of the information generated and stored from the main training procedures. In this case, information from all of the generated FIS in the form of .fis files (FIS file) and .txt files (FIS name) were loaded and used. The .fis files were the FIS units from the training processes while the .txt files contained information (names) of the FIS units.

The saved FIS units generated from a particular training session can be loaded into the workspace by using their name stored in a text file (.txt).

Each of the loaded FIS units was then tested using a similar scheme of requirements as those used during its generation in the main or core ANFIS training process. However, the evaluation process used a higher value of correlation check in selecting only FIS units which had correlation values > 0.99. The correlation check was calculated by comparing the correlation between the FIS Target Output results and the Target Output of the feature data used.

In the evaluation process, the same training data used to generate a FIS was used to evaluate its characteristics. Only the FIS units which produced correlation factors > 0.99 between the FIS predicted Target Output and the training data Target Output were selected for the next processing step.

5.13 Final Evaluation Process

The final processing step aimed to evaluate the performance of the best FIS units. There were two parts in the final evaluation process. The first part (part A) is shown in Figure 5.24. In this part, the FIS units were evaluated using the same features (training) data which were used to generate the corresponding FIS units. In addition, the first part of the evaluation process aimed to get detailed information on the selected FIS units, such as which features were selected as Input1 and Input2, and which feature was selected as the Target Output.

In the second part of the process, the FIS units were evaluated using different feature (training) data segments. Evaluation using different features (training) data aimed to test the generalisation characteristics of the FIS units.

5.13.1 Final Evaluation Process (Part A)

Unique FIS units were selected from storage and loaded. The procedure then continued to load the corresponding training data used to generate the FIS units. At this stage, the same training and checking data sets that were used to train the evaluated FIS were used again to check whether the FIS could reproduce a Target output (i.e., energy level) that matched that in the training data. The output of energy levels predicted by the evaluated FIS was then compared using a correlation relationship toward the energy level values of the same training data. The unity correlation value would show that the energy level predicted or produced by the evaluated FIS matched the one in the training data. The FIS units and correlation / recognition factors were obtained and stored in the workspace.

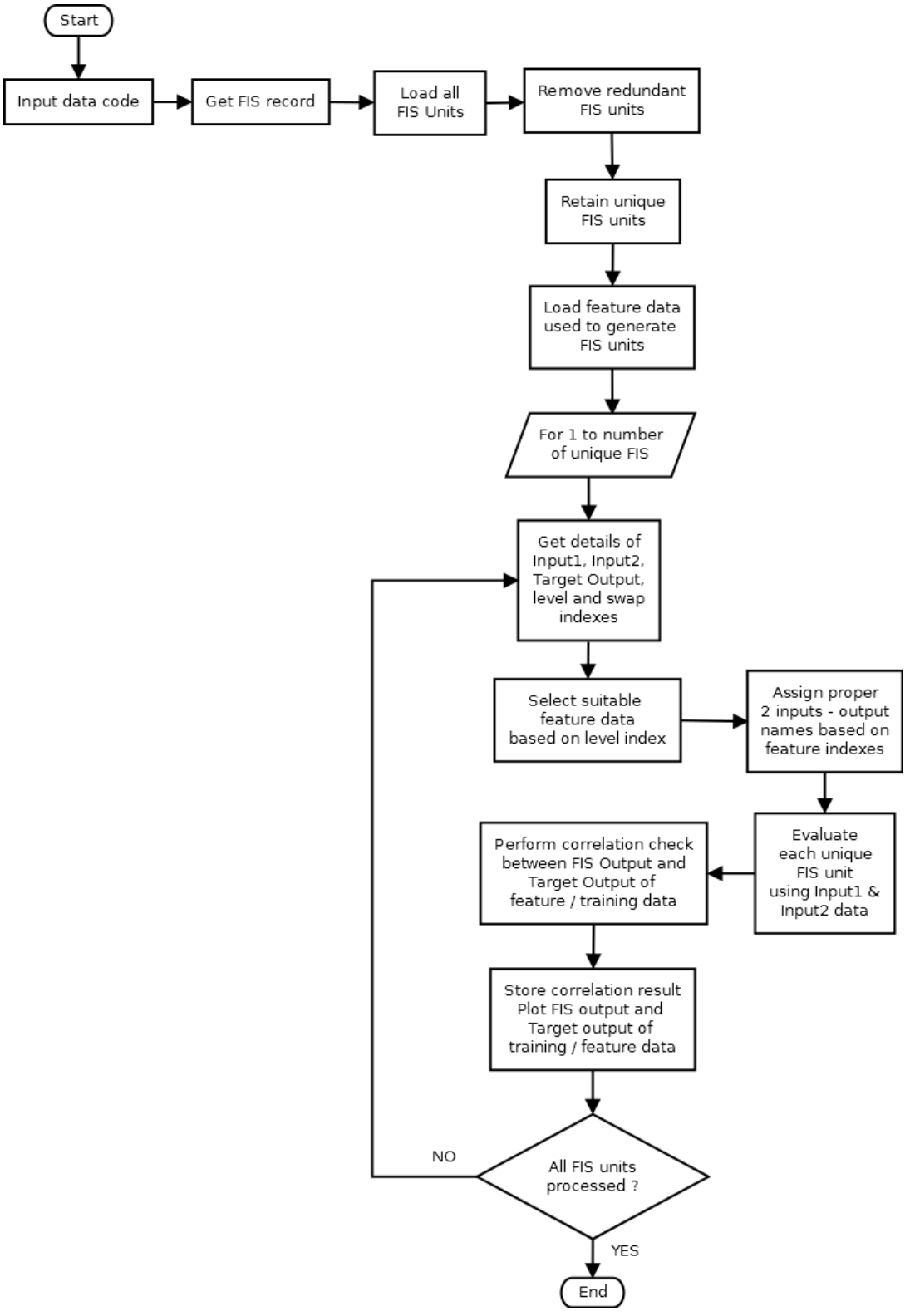

Figure 5.24 FIS selection and evaluation scheme

The next step was the removal of any redundant FIS units leaving unique FIS units for evaluation. This was followed by loading the similar training / features data that had been used to generate the corresponding FIS units.

The process entered a cycle in order to process all of the selected unique FIS units. In each cycle, a routine was used to extract information from the FIS to be evaluated. The information extracted was Input1, Input2, Target Output, Level of decomposition data, Swap indices. These indices referred to the details of an FIS unit which were embedded in the FIS when it was generated during the ANFIS training process. The information (indices) extracted showed which of the seven features were selected as Input1, Input 2 and the Target Output. It also showed which decomposition level data (i.e., 1 – 9) and Swap index (i.e., 1-7) were used during the ANFIS training.

The information gathered from the FIS unit under evaluation was used to determine which level of feature/training data would be used to evaluate the FIS unit. In particular, the information of Input 1 and Input 2 was used to assign which columns in the feature (training) data would be used as Inputs in the evaluation process. For instance, if the Input 1 index was 5 and the Input 2 index was 2 then the two inputs would be taken from column 5 and 2 of the feature (training) data. Column 5 refers to Standard deviation and column 2 refers to RMS. In this case, the FIS unit would be evaluated using Standard deviation and RMS features.

The index of Target Output determined which one of the seven features was selected in relationship to Input 1 and Input 2 during the production process of the FIS in the ANFIS training process. For instance, if the index of Target Output was 1, it referred to column 1 of the feature data which was the Energy level. In this example, it was shown that the FIS under evaluation would use the Standard deviation and RMS as inputs (Input 1 and Input 2) and it would produce the Energy level as the Target Output.

Once the suitable feature data was loaded and proper inputs and output values and names were assigned using the indices information obtained, then the FIS unit was evaluated. The evaluated FIS unit then produced the intended Target Output values.

The process continued with a step to compare the similarity between the FIS Target Output and the one in the features (training) data, and the similarity was measured using a correlation factor. A unity correlation factor meant that the FIS Target Output

and the one in the features (training) data were similar. It showed that the evaluated FIS unit was 100% accurate in producing the Target Output based on the two input features supplied.

The process continued with saving the correlation factor results and plot graphs of the FIS Target Output and the Target Output of the feature (training) data. The process was then continued on its cycle if there were further FIS units to be evaluated. Once all of the FIS units had been evaluated the process ended.

5.13.2 Final Evaluation Process (Part B)

All routines in part B of the evaluation were similar to the ones in part A with the exception that the features (training) data used were different than the one used to generate the evaluated FIS through ANFIS training.

The objective of this final evaluation part was to test the performance of an FIS unit when it was supplied with a different feature data set. The different data is data that was not used to generate the evaluated FIS. This data was obtained using similar bearing fault conditions but it was acquired under different loading conditions and different shaft speeds. Part B of the evaluation process provided the ability to assess the generalisation of the evaluated FIS through the utilisation of different data segments in the evaluation process.

Chapter 6 - Bearing Test Rig and Vibration Signal Acquisition

The fault detection scheme outlined in previous chapters was trialled using vibration data acquired from a test rig containing a bearing, sequentially and with and without fault. The conditions were tightly controlled with the faults being artificially introduced and exactly known. This controlled test data was then used to train the ANFIS to evaluate the scheme's effectiveness and suitability for condition monitoring applications.

The data acquisition scheme that was used to acquire sets of bearing vibration signal is shown in Figure 6.1.

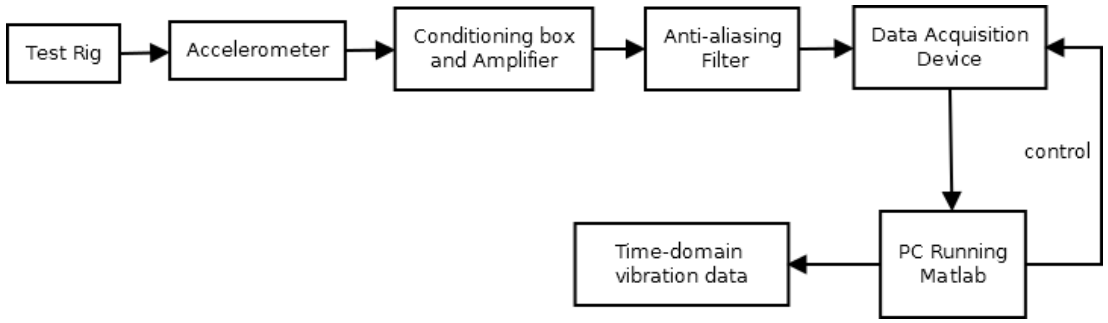

Figure 6.1 Data acquisition scheme

6.1 The Test Rig

A photograph of the bearing test rig is shown in Figure 6.2. It consists of a shaft driven by a variable-speed AC motor. The shaft was supported by two roller bearings that were known to be in good condition. The overhung shaft carried the test bearing within its floating housing. The bearing can be loaded in the radial and thrust directions by adjustable screws, the load being measured with strain gauge load cells. The rotational speed was set to be constant at 35 Hz (2100 RPM).

This particular speed was chosen based on the precaution to eliminate possible interference from half the AC power line frequency of 60/2 = 30 Hz. A constant speed was used for convenience, especially for reducing the number of variables that needed to be used or presented in the feature extraction process.

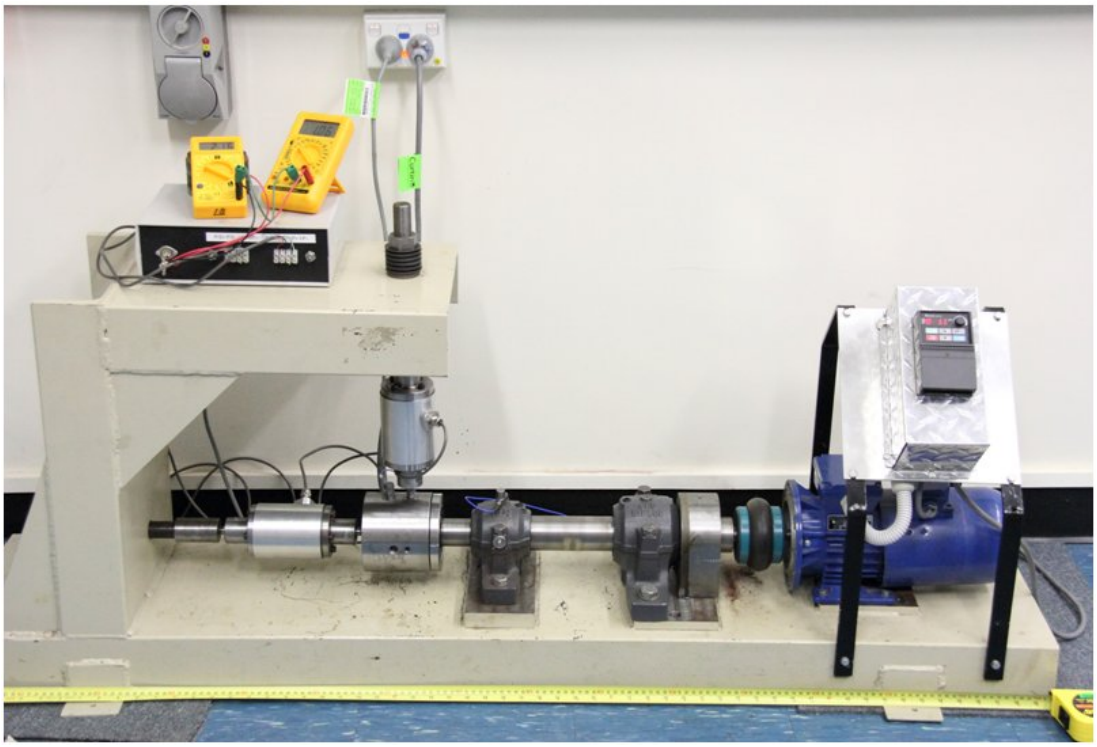

Figure 6.2 Bearing test rig

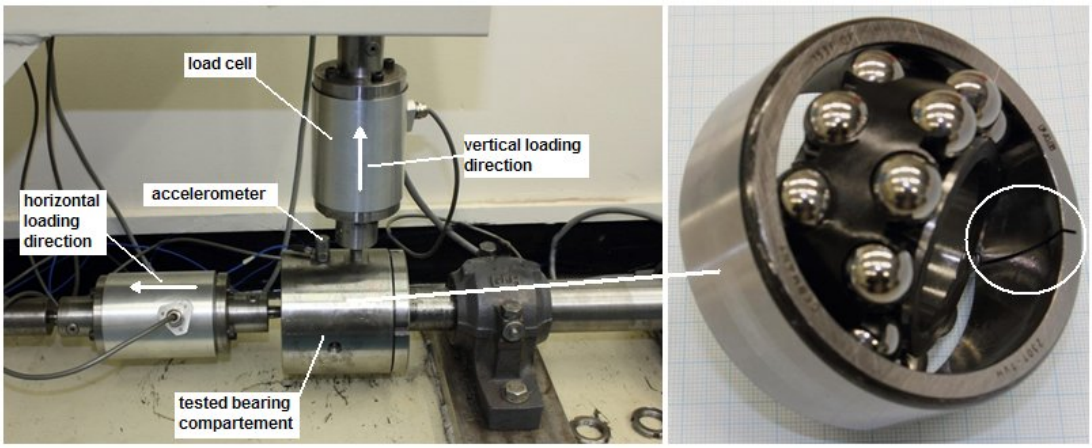

Figure 6.3 Detail components of the test rig

6.2 Test Bearings

The bearing under test was a self-aligning double-row bearing (FAG 2307-TVH) as shown in Figure 6.3.

This type of bearing was chosen for two reasons. Firstly, to accommodate the self-inclination construction of the bearing test compartment of the test rig, and secondly, because this type of bearing is rarely used in fault diagnoses applications.

The three types of faults used in gathering faulty vibration signals were outer race faults (ORF), inner race faults (IRF) and Ball faults (BF). Faults were introduced (individually) into the test bearing by using an EDM wire-cut machine to cut a .7mm groove on the surface of the outer race, inner race and two axially aligned balls. Photographs of the test bearings and the types of faults are shown in Figure 6.4 to 6.6, respectively.

Each bearing with its specific fault was then installed into the floating housing in the test rig. The bearing was then loaded by adjusting the radial and axial loading mechanisms. The load was displayed in Newtons by means of two calibrated load cells attached to the radial and axial pull rods. The radial load was set to approximately 1100 N and the thrust load to approximately 2100 N. Smooth loading adjustment was achieved by using a stack of cup springs (Belleville washers).

An accelerometer was attached at the upper surface of the tested bearing compartment with its active axis aligned with the radial load vector. This accelerometer was used to record the vibration signals.

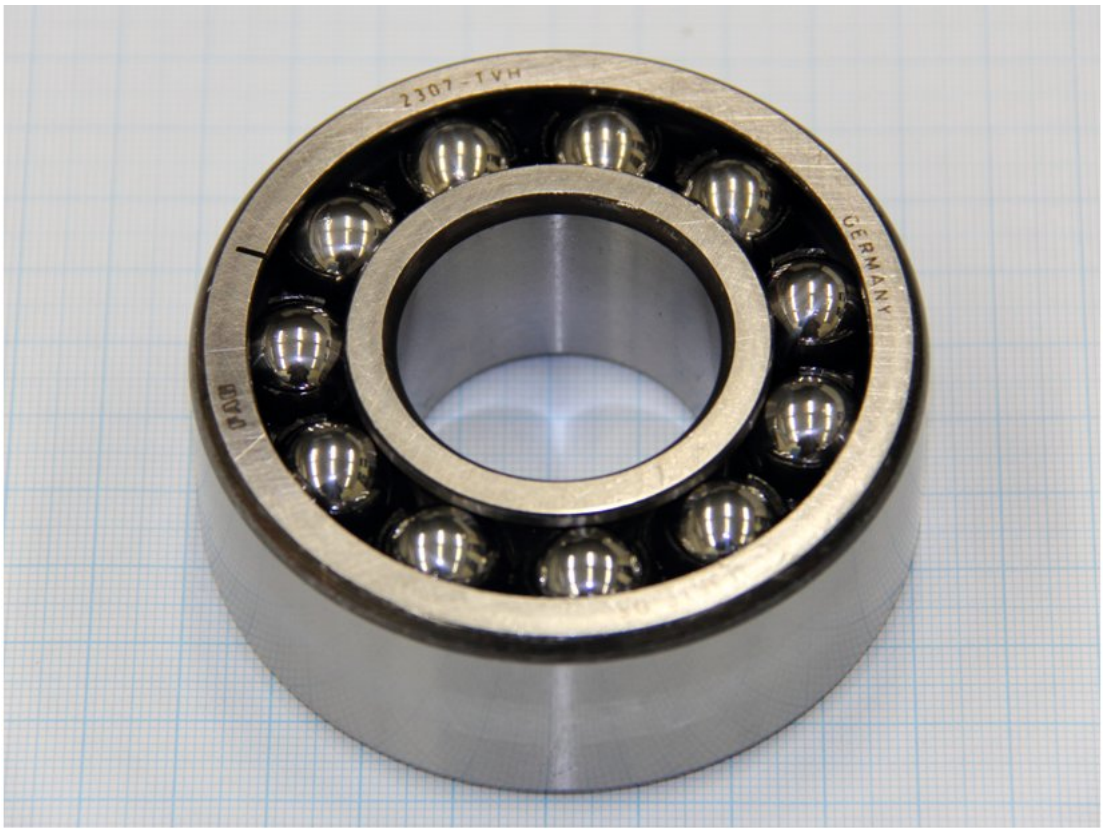

Figure 6.4 Test bearing with outer race fault

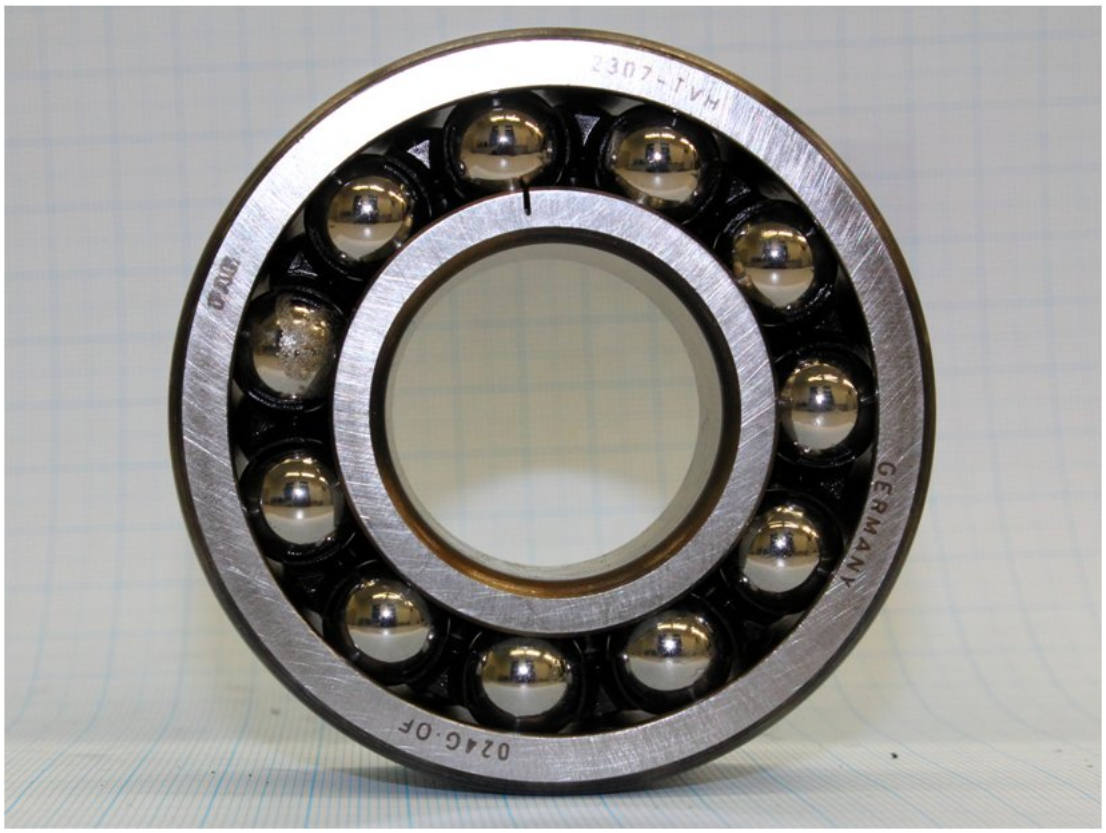

Figure 6.5 Test bearing with inner race fault

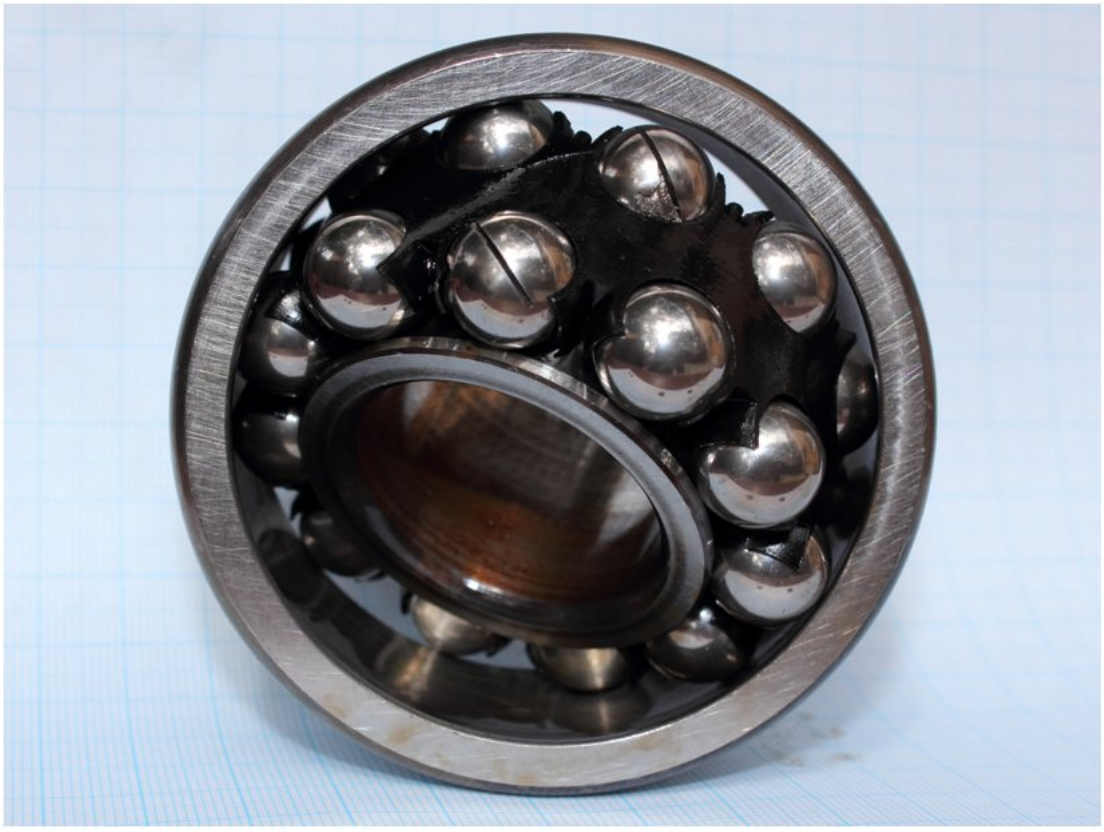

Figure 6.6 Test bearing with balls fault

6.3 Accelerometer and Signal Conditioner

The accelerometer in use was an ICP Accelerometer model-353B03 from PCB Piezotronics. The accelerometer was stud-mounted on the test-rig bearing housing and connected to a conditioning box which also served as an amplifier. The signal conditioner in use was battery-powered ICP® sensor signal conditioner, model 480E09 from PCB Piezotronics. Using the stud-mounting method, the accelerometer could achieve a bandwidth of up to 100 kHz according to its datasheet.

6.4 Analogue Anti-aliasing Low Pass Filter

An anti-aliasing filter with a cut-off frequency of 10 kHz was used to filter the vibration signal from the accelerometer. The analogue electronic filter in use was a Sallen-Key low-pass filter which was constructed using an LM 324 integrated circuit. A circuit schematic and details of this filter are presented in Appendix 4.

6.5 Data Acquisition Device

The data acquisition hardware used was a National Instruments USB-6251 device which was driven and controlled using the Matlab Data Acquisition Toolbox. Matlab source codes used to control data acquisition process are listed in Appendix 5.

The vibration data of each fault type was acquired in multiple segments of 1 second in duration, using a sampling rate of 48 kHz (48192 data points per second) (Qiu *et al*. 2006). The time domain vibration signals from each tested bearing were saved in the form of binary files.

The sampling rate of 48 kHz was chosen for convenience, its only disadvantage being an increased file size. The anti-aliasing filter was rolled off at 10 kHz which is sufficient to provide useful vibration signals with a bandwidth below 10 kHz. This bandwidth is similar to the one used in Qiu *et al*. (2006).

6.6 Vibration Data Structure

For each test of each type of bearing fault, case data acquisition was carried out for 1 second and then paused for 1 second. A 1 second pause was used to ensure that all of the acquisition data and process had been processed before continuing. This produced 984 data sequences of 48kHz data in the Matlab mat file format. There were 984 parts of 1 second's duration acquired. The vibration data was then used to generate the seven numerical features viadiscrete wavelet transform (DWT), as described in the previous chapter.

Chapter 7 – Fuzzy Inference System (FIS) and ANFIS Training Results

This chapter presents detailed characteristics of the Fuzzy Inference System (FIS) units obtained using the proposed method along with the related sensitivity comparison of db-n wavelet types used in the feature extraction method. The application results of the FIS units in the complete bearing fault classification system are presented.

There were four types of FIS unit produced by the proposed ANFIS training. These were based on the four types of fault feature data used in the ANFIS training procedures that aimed to build the FIS units, namely outer race faults (ORF) FIS, inner race faults (IRF) FIS, ball faults (BF) FIS and no fault (NF) FIS.

7.1 Outer Race Fault (ORF) FIS Units

The main characteristics of the FIS units obtained using the ORF feature data through ANFIS training was the association between Standard Deviation as Input 1, RMS as Input 2 and Energy level as the Target Output. Hence the structure of the ORF FIS units was based on Standard Deviation and RMS as Input 1 and Input 2 and Energy level as the Target Output.

By using the proposed method, the ANFIS model had the ability to capture the core relationships of the ORF feature data presented in the training session. As a result, it associated ORF features with the input-output relationships between Standard Deviation, RMS and Energy level features. Visualisation of the FIS unit structure obtained through the ANFIS training session using ORF features data is shown in Figure 7.1.

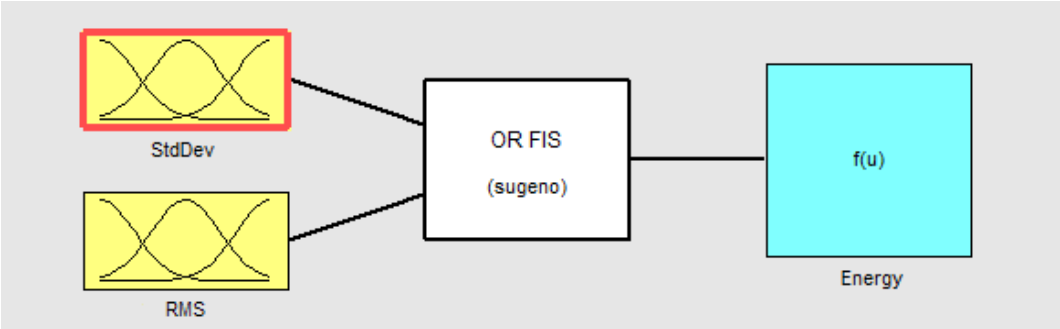

Figure 7.1 Outer race fault (ORF) FIS unit structure

In Figure 7.1, the ORF FIS obtained from the ANFIS supervised training process had two associated inputs in relation to the single Target Output. These two inputs and the single output were selected and associated by the ANFIS model during the training session.

The input and output association of the RMS, Standard Deviation, and Energy level were the best available characteristics selected by ANFIS from the ORF features data through the training process.

As an independent check of the FIS model, the ORF FIS units were supplied with the two suitable feature inputs (i.e., Standard Deviation and RMS features) and produced a matching pattern of Energy Level (Target Output). In this case, the Target Output pattern was independently produced by the ORF FIS units by using two suitable inputs that were generated through the wavelet-based feature extraction method utilised.

The Target Output (Energy level) pattern produced by the ORF FIS unit was matched to the Energy level pattern of the related feature in the training data. The matching was achieved due to the ANFIS training procedures which had successfully taught the ORF FIS to map the relationship between Standard Deviation (Input 1) and RMS (Input 2) towards the Energy level as the Target Output. The relationships between these three features were the best available characteristics the FIS units obtained through ANFIS training utilising wavelet-based ORF feature data.

It was concluded that the result of the ANFIS training using the ORF features data produced FIS units that found associations between two inputs (RMS and Standard Deviation) as Input 1 and Input 2, with the Energy level as the Target Output.

7.2 FIS Units of Inner Race Fault (IRF) Training Results

The main characteristics of the FIS units obtained using IRF feature data through the ANFIS training process is shown in Figure 7.2.

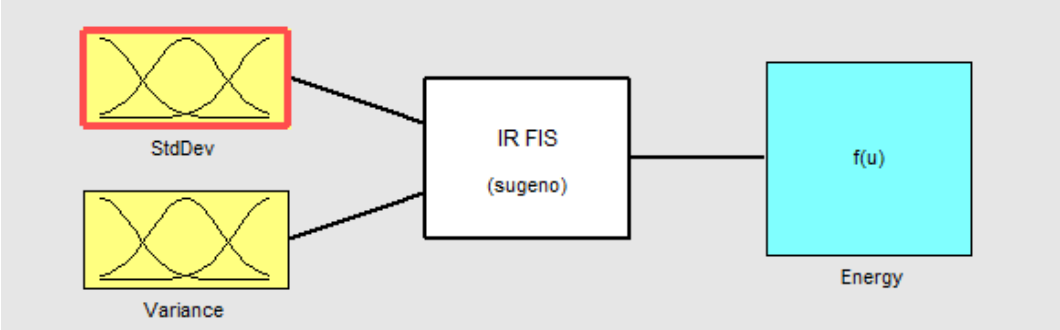

Figure 7.2 Inner race fault (IRF) FIS unit structure

The IRF FIS units had an association of Standard Deviation as Input 1, Variance as Input 2 and Energy level as the Target Output. This input-output association gave the best available characteristics which were automatically obtained through ANFIS training using the IRF feature data.

It was concluded that the result of the ANFIS training using IRF feature data produced FIS units that found associations of two inputs (Standard Deviation and Variance) as Input 1 and Input 2 with Energy level as the Target Output.

7.3 Ball Fault (BF) FIS Results

Figure 7.3 shows the main characteristics of the FIS units obtained using BF feature data through the ANFIS training process.

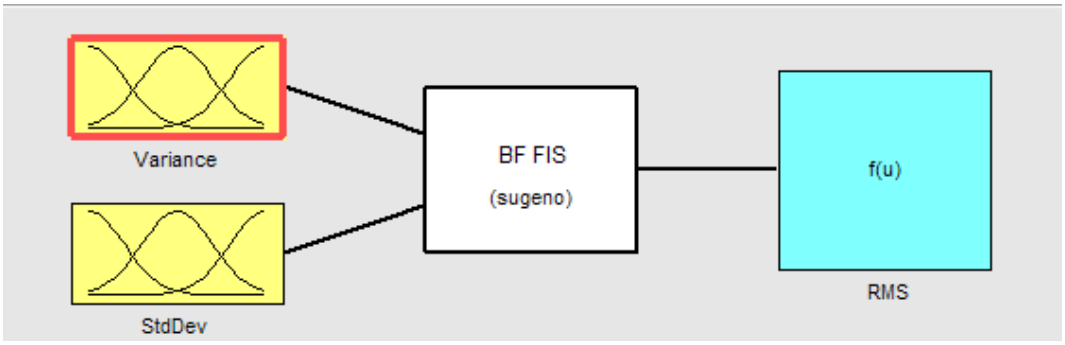

Figure 7.3 Ball fault (BF) FIS unit structure

The BF FIS units had an association of Variance as Input 1, Standard Deviation as Input 2 and RMS as the Target Output. This input-output association gave the best available characteristics which were automatically obtained through the ANFIS training process using BF feature data.

7.4 No Fault (NF) FIS Results

Figure 7.4 depicts the structure of the NF FIS units.The FIS units had an association of RMS as Input 1, Kurtosis as Input 2 and Standard Deviation as the Target Output

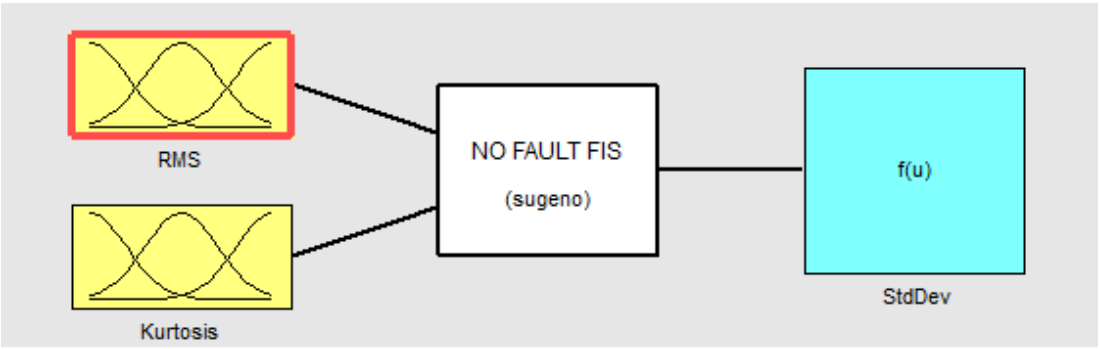

Figure 7.4 No fault (NF) FIS unit structure

This input-output association gave the best available characteristics which were automatically obtained through the ANFIS training process using NF feature data.

7.5 The summary of FIS units Input-Output Relationships

A summary of the selected two Inputs and Target Output features of the FIS units for all the fault cases based on the processed fault data is presented in Table 7.1.

Table 7.1 Summary of FIS Selected Inputs – Target Output Relationships

Fault Type	**Input 1 feature**	**Input 2 feature**	**Target Output feature**
Outer Race fault (ORF)	Standard Deviation	RMS	Energy
Inner Race fault (IRF)	Standard Deviation	Variance	Energy
Ball fault (BF)	Standard Deviation	Variance	RMS
No Fault (NF)	RMS	Kurtosis	Standard Deviation

7.6 ANFIS Model Structure

The ANFIS model structure of ORF, IRF, BF and NF FIS units consist of five neuron layers as shown in Figure 7.5. This structure was similar to the ANFIS general structure proposed by Jang *et al.* (1993). The first layer was the input layer, the second was *inputmf* (input membership function), the third was the rule layer, the fourth was the *outputmf* (output membership function) and the last was the output layer. There were two inputs and a single output used as they were generated based on input-output features provided in the training and checking data.

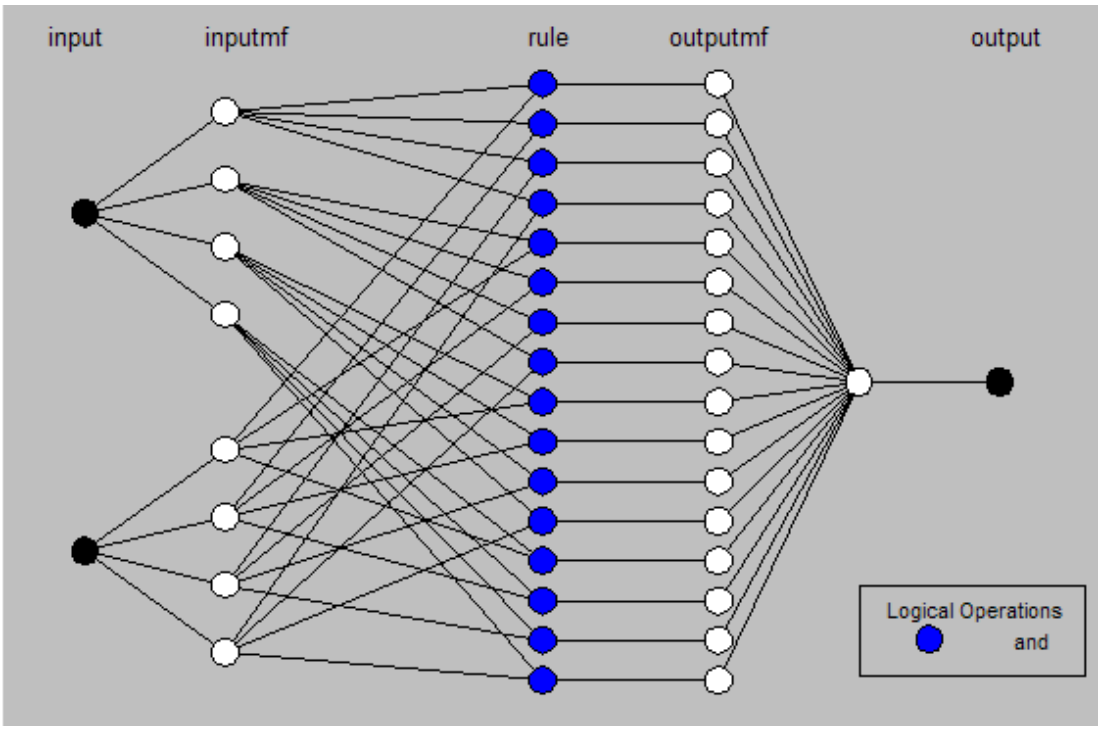

Figure 7.5 ANFIS Model Structure

There were 16 rules used and each rule connected to one of 16 outputmf (output membership functions). All of the 16 rules were constructed using AND logic. There were no OR and NOT logic operators used.

7.7 FIS Unit Evaluation and Selection Procedures

The evaluation process was carried out to assess performance of the FIS units obtained. The aim of the procedure was to collect the FIS units which had the ability to produce an accurate Target Output. The accuracy level of the Target Output was set to a required level which was measured using a correlation factor value. In this case, the Target Output produced by a particular FIS unit was compared to the Target Output of the training data which was used to train and generate the corresponding FIS unit under investigation. The matching rate between these two Target Outputs was measured using correlation factors. An FIS unit which produced a matching rate ≥ 0.9 was selected as the best candidate to be used as a core classifier in a bearing fault classifier.

The evaluation process was carried out to assess the ability of the FIS units to produce matching Target Output patterns using only two suitable input features (i.e.,

Input 1 and Input 2) which were generated from different fault data segments than those recorded in Table 7.1.

The ability to produce a matching Target Output was an important aspect that was used as an indicator to classify a particular bearing fault case. A bearing fault classification case was determined based on the ability of the FIS unit (classifier) to generate a matching Target Output using the suitable two inputs (Input 1 and Input 2). The ability to produce a matching Target Output was based on the embedded fuzzy rules generated and tuned by the proposed ANFIS training procedures, using wavelet-based feature data of the ORF, IRF, BF and NF cases.

The accuracy level of an FIS unit in producing a matching Target Output was measured as the error level which was obtained by comparing the Target Output produced by an FIS unit with one of the checking data sets. The error values between these two Target Outputs were used to assess the performance of the FIS under investigation. The assessment process was carried out to select the best FIS units that were later used in the classification of fault cases.

Another objective of the evaluation process was to obtain information on the dominant db-n wavelet type in generating features data for ORF, IRF, BF and NF cases. The FIS units selected as the candidates for the final evaluation process were based on the lowest error achieved from across all of the db-n results obtained (i.e., db4, db8, db12, db22 and db44).

A block diagram of the selection scheme process that was used in the evaluation and selection processes of the best FIS unit is presented in Figure 7.6.

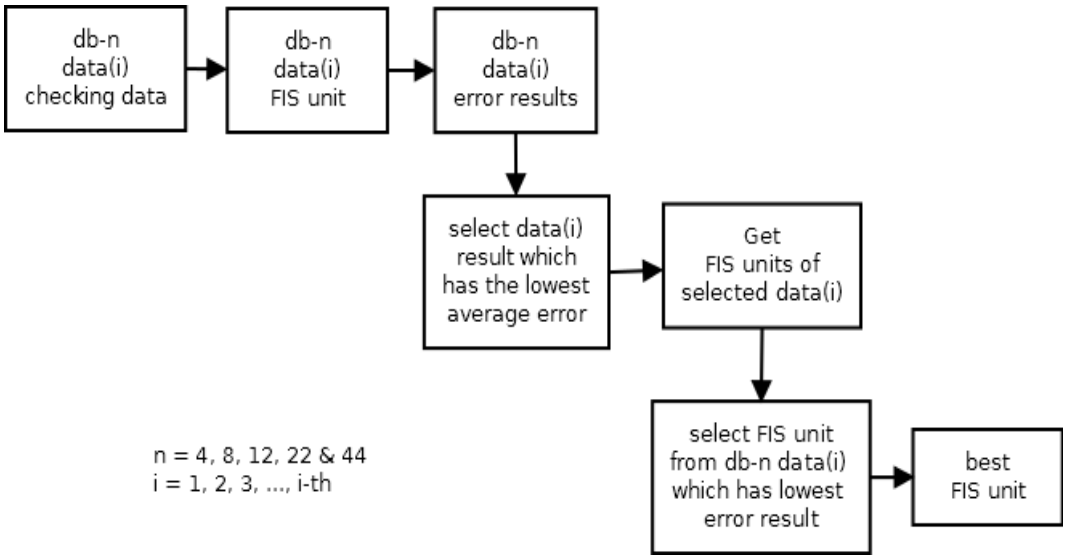

Figure 7.6 Evaluation and selection scheme of FIS units

In Figure 7.6, the Data(i) represents the checking data segment used in the evaluation process. Db-n data(i) represents the group of checking data(i) which was generated based on db-n wavelets (i.e., db4, db8, db12, db22 and db44). Each of the db-n data(i) data was supplied to the FIS units. The resulting error rate of the corresponding FIS unit under investigation was then calculated. The process was carried out until all of the db-n data(i) was processed.

The results of FIS units of db4, db8, db12, db22 and db44 were then evaluated to obtain the lowest average error rate. The db-n FIS unit that had the lowest average error was then selected as the best FIS for the corresponding fault case.

7.7.1 Outer Race Fault (ORF) FIS Evaluation Result

In order to assess the performance of FIS units obtained from the ANFIS training sessions using ORF features data, the procedures explained in Section 5.2, with details depicted in Figure 7.6, were carried out.

The following section explains the evaluation results of the ORF FIS units generated using wavelet-based features that employed db4, db8, db12, db22 and db44 wavelets.

There were five groups of ORF FIS units obtained based on the number of db-n wavelet types used. Each group related to a Daubechies (db-n) wavelet-based feature

data set. These five groups of ORF FIS units were db4 ORF FIS units, db8 ORF FIS units, db12 ORF FIS units, db22 ORF FIS units and db44 ORF FIS units.

The performance evaluation process was employed to test and select the FIS units obtained based on the lowest error of Target Output criteria. The error level of the Target Output was obtained by comparing the error between the Target Output of FIS unit and one of the checking data sets.

The ORF FIS performance evaluation was carried out by testing the FIS units using seven checking ORF data sets. The testing was applied for five FIS units groups obtained from db4, db8, db12, db22, and db44 wavelet-based features.

The results of the evaluation process produced information on errors. The error information was obtained by comparing the FIS Target Output to the Target Output of the checking data. The FIS units that had the lowest error results during the testing process were the ones selected for use. In this process, the lower the error values, the more they represented a well-performing FIS unit.

In the evaluation process, two suitable inputs from each feature data set were supplied to the ORF FIS units. The corresponding Target Output produced by the ORF FIS units was then compared to the Target Output of the checking data sets. The error between the two Target Output values was then calculated. The generated error from the ORF FIS unit for each checking data set was visualised, as seen in Figures 7.7 to 7.11.

In Figures 7.7 to 7.11, the chkdata set axis represented seven checking data sets that were used to evaluate the FIS units obtained. The *FIS unit* axis represented the eight FIS units under evaluation. The z-axis represented the error or difference between the FIS Target Outputs and those of the checking data.

The data set chkdata no. 1 was the feature data set used in the production of the corresponding FIS during the ANFIS training session. The other chkdata sets were different ORF checking data sets which were acquired under different loading conditions.

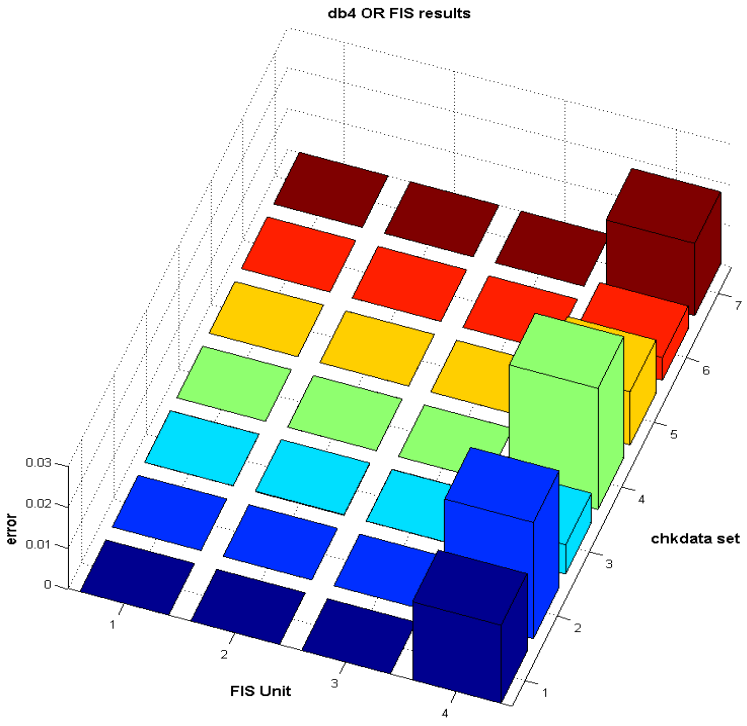

Figure 7.7 Evaluation results of db4 ORF FIS units

Figure 7.7 shows the evaluation results of ORF FIS units that were generated through ANFIS training using the features processed using db4 wavelet-based features.

Figure 7.7 shows that FIS unit no. 1, 2 and no. 3 produced lower error values across the checking data. These two FIS units were selected for comparison with the other evaluation results of the db8, db12, db22, and db44.

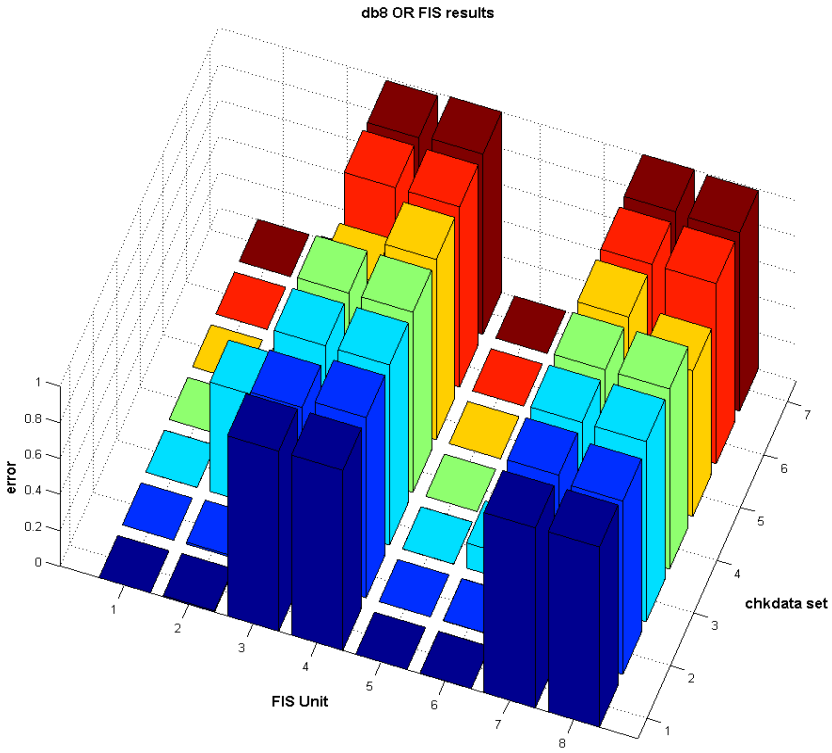

Figure 7.8 Evaluation results of db8 ORF FIS units

Figure 7.8 shows the evaluation results of the ORF FIS units that were obtained through ANFIS training, using the features data processed using the db8 wavelet. The FIS units no. 1 and no. 5 produced lower errors compared to the others. Based on these results, FIS units no. 1 and no. 5 were selected for further comparison. These two FIS units were compared to the evaluation results of ORF FIS db4, db12, db22, and db44 units.

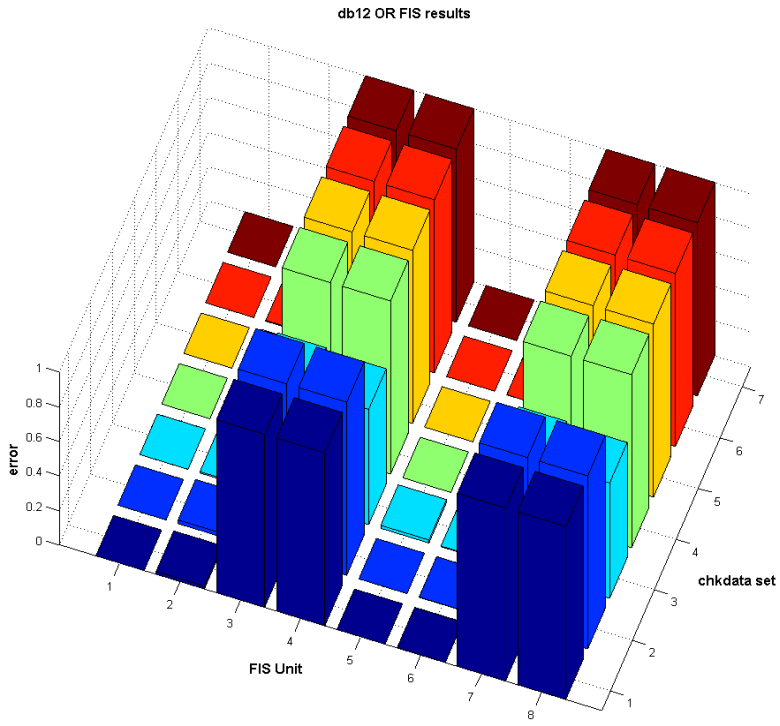

Figure 7.9 Evaluation results of db12 ORF FIS units

Figure 7.9 shows the evaluation results of the ORF FIS units that were obtained through ANFIS training using the features data that was processed using the db12 wavelet. FIS units no. 1 and no. 5 produced lower errors compared to the others. FIS units no. 1 and no. 5 results were compared to the evaluation results of ORF FIS db4, db8, db22, and db44 units.

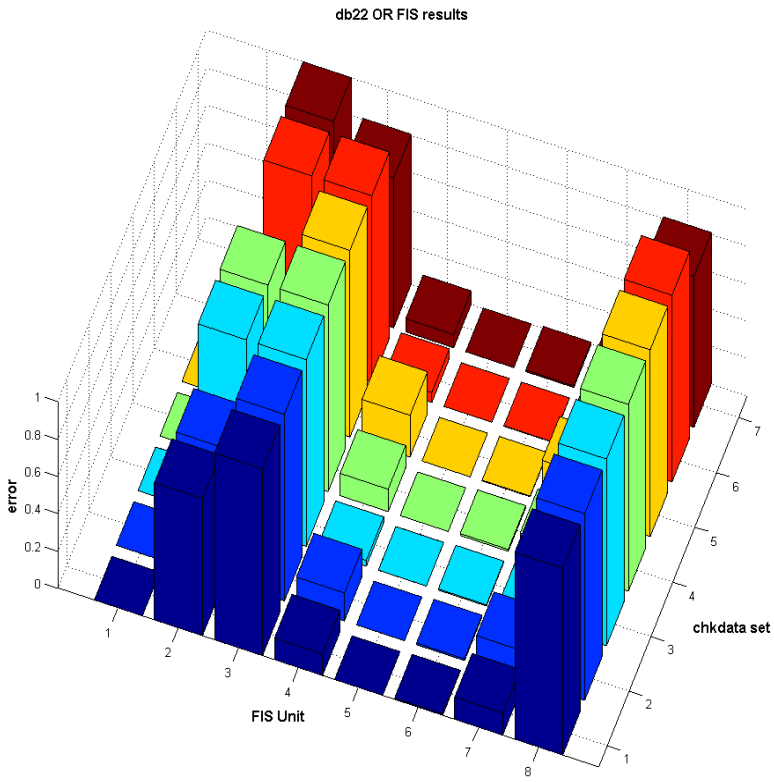

Figure 7.10 Evaluation results of db22 ORF FIS units

Figure 7.10 shows the evaluation results of the ORF FIS units that were obtained through ANFIS training using the feature data processed using db22 wavelet. FIS units no. 1, no. 5 and no. 6 produced lower error values compared to the others. The results of FIS units no. 1, no. 5 and no. 6 were selected for a further comparison with results from the evaluation results of ORF FIS db4, db8, db12 and db44 units.

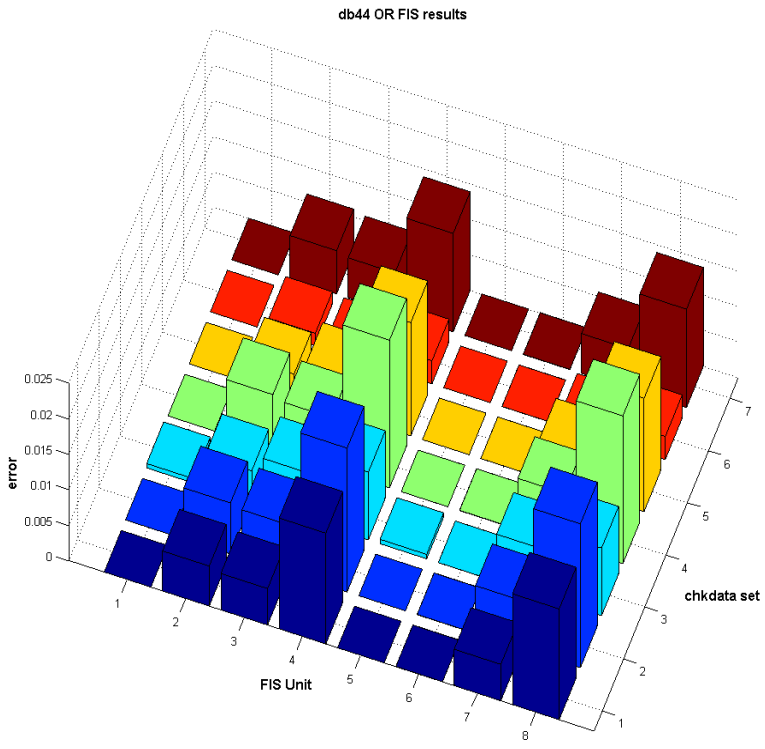

Figure 7.11 Evaluation results of db44 ORF FIS units

Figure 7.11 shows the evaluation results of the ORF FIS units that were obtained through ANFIS training using the feature data processed using db44 wavelet. FIS units no. 1, no. 5 and no. 6 produced lower errors across the checking data used. FIS units no. 1, no. 5, and no. 6 were selected for further comparison with the evaluation results of db4, db8, db12, and db22 ORF FIS units.

After evaluating the ORF FIS results, all evaluation results obtained from db4, db8, db12, db22 and db44 of ORF FIS units were compared. It was found that the overall lowest error results were produced by the FIS units generated using db4 wavelet-based features.

The visual information on the average errors is shown in Figure 7.12. The lowest error was achieved by the db4 ORF FIS units group. The FIS units of this group were selected as the best candidate FIS units for use as fault classifiers. The FIS units were produced through the utilisation of statistical and wavelet-based features data generated from outer race fault (ORF) bearing vibration signals.

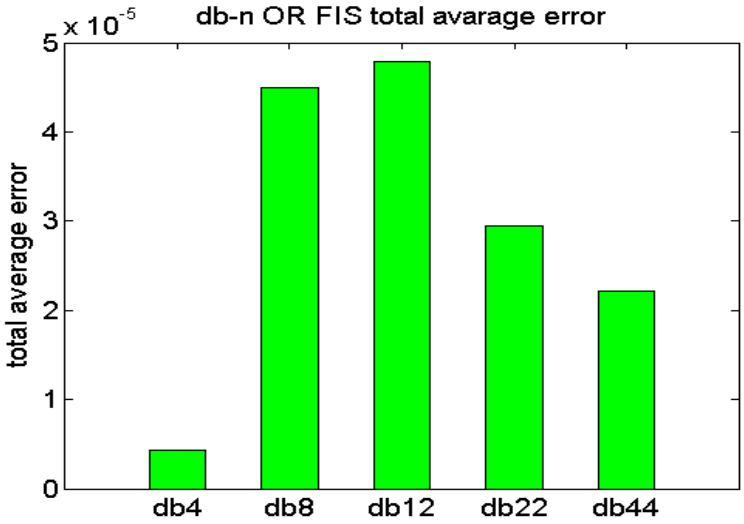

Figure 7.12 Average errors of all db-n results of ORF FIS units

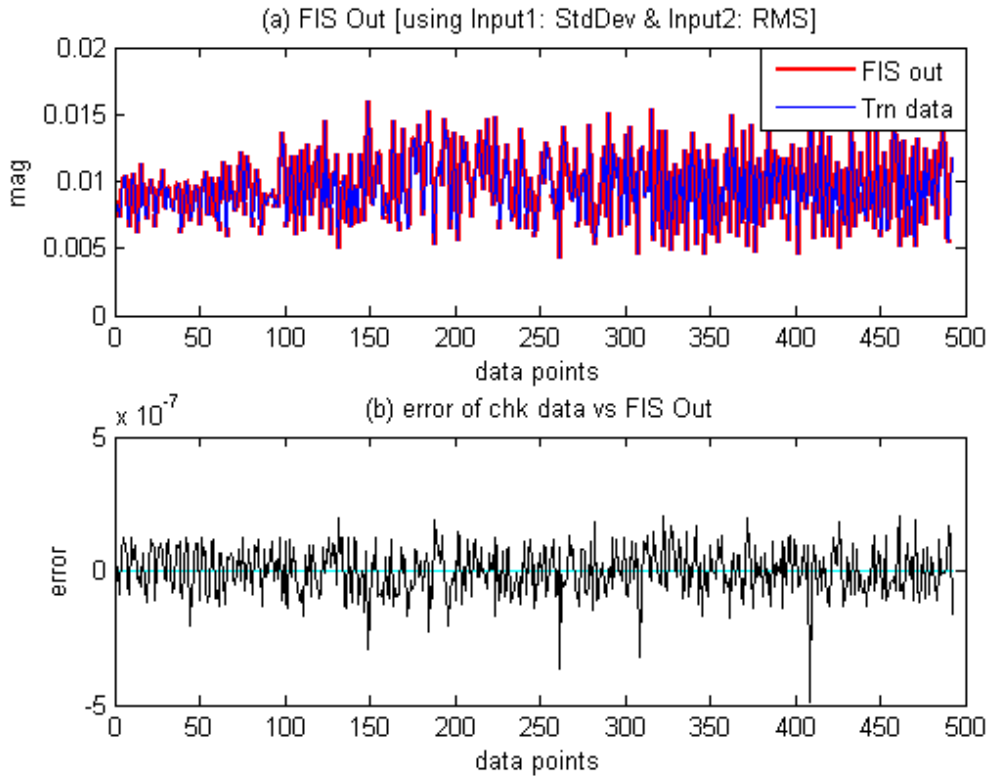

Figure 7.13 (a) ORF FIS Target Output and training data Target Output, (b) error between FIS Out and training data Target Output

The performance of the selected ORF FIS units in producing the intended Target Output (FIS Out) values based on two inputs (Input 1 and Input 2) is shown in Figure 7.13 which shows the ORF FIS unit output pattern in comparison to the Target

Output of the checking data used to evaluate the corresponding selected ORF FIS unit. The *x*-axis represents number of training data points (498) and the *y*-axis represents the amplitude of the Energy level.

Figure 7.13(a) shows the Target Output of ORF FIS under evaluation when presented with Standard Deviation as Input 1 and Variance as Input 2 and the Target Output of the checking data that had to be matched by the ORF FIS unit under evaluation. Figure 7.13(b) shows the error between the FIS output and the Target Output. The error was of the magnitude 10^{-5} order which was considered to be valid.

The characteristics of the ORF FIS units in producing the Target Output values based only on Input 1 and Input 2 represented the ability to capture the most distinct relation of input-output of the seven features that were presented in the training data. The assignment of Standard Deviation and RMS as Input 1 and Input 2 related to Energy level as the Target Output was the characteristic that was automatically obtained by the ANFIS model through the training process. In this case, the ANFIS training process was able to learn and map the unique relationships of the three features out of the seven provided in the ORF training data. The ORF input-output relationship was an important characteristic for the purposes of ORF case classification since the unique input-output relationship applies only to the ORF case.

7.7.2 Inner Race Fault (IRF) FIS Performance Evaluation

The procedures used to evaluate the IRF FIS units was similar to the one used for the OR FIS units, as explained in Section 7.7.1.

It was found that the overall lowest error results were produced by the IRF FIS units generated using db12 wavelet-based features. The finding was based on the evaluation of the IRF FIS results that were obtained by comparing all evaluation results of db4, db8, db22 and db44 of IRF FIS units.

The test results of the IRF FIS obtained through ANFIS training by using db12 fault features data is shown in Figure 7.14. There were eight FIS units evaluated using 14 checking data sets. The lower error trend results are shown by IRF FIS no.1 and FIS no. 4.

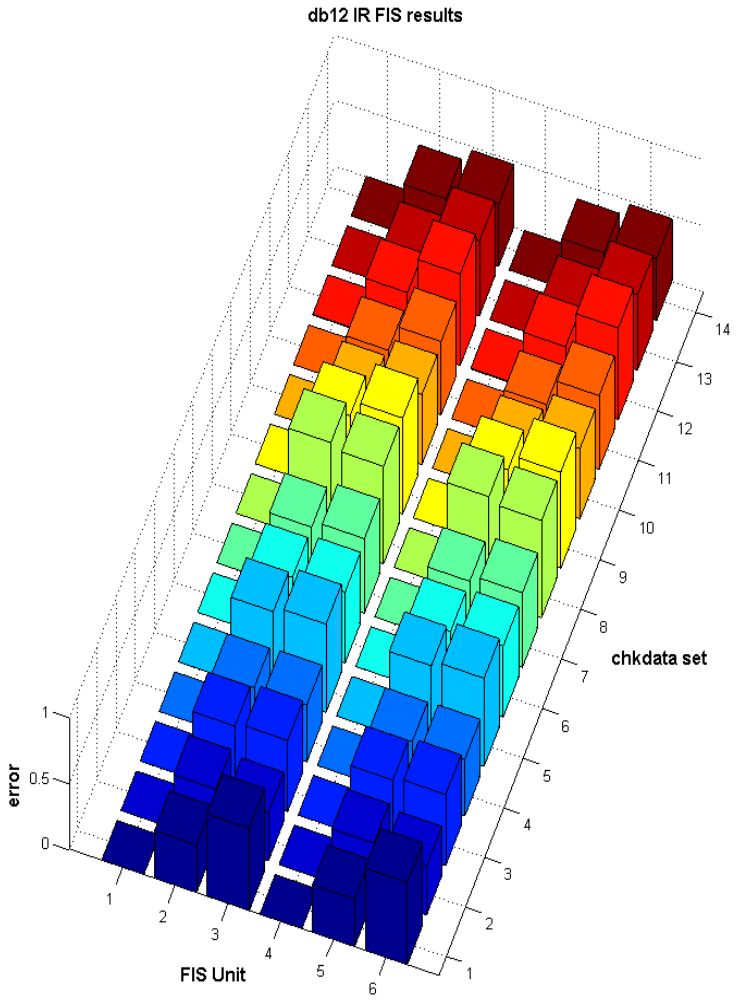

Figure 7.14 Test results of db12 IRF FIS units

The evaluation results of the six selected db12 IRF FIS units through the use of 14 checking data sets were shown in Figure 7.14 which shows that the lowest error was achieved by FIS units no.1 and 4.

Information in Figure 7.14 was used to evaluate and select IRF FIS units of db12 that were intended for use as fault classifiers. Figure 7.14 shows that a lower error value was produced by FIS no. 1 and FIS no. 4 across the testing results of a 14 checking data set. The lowest error values produced by the db12 IRF FIS units were used as a standard in selecting the FIS units.

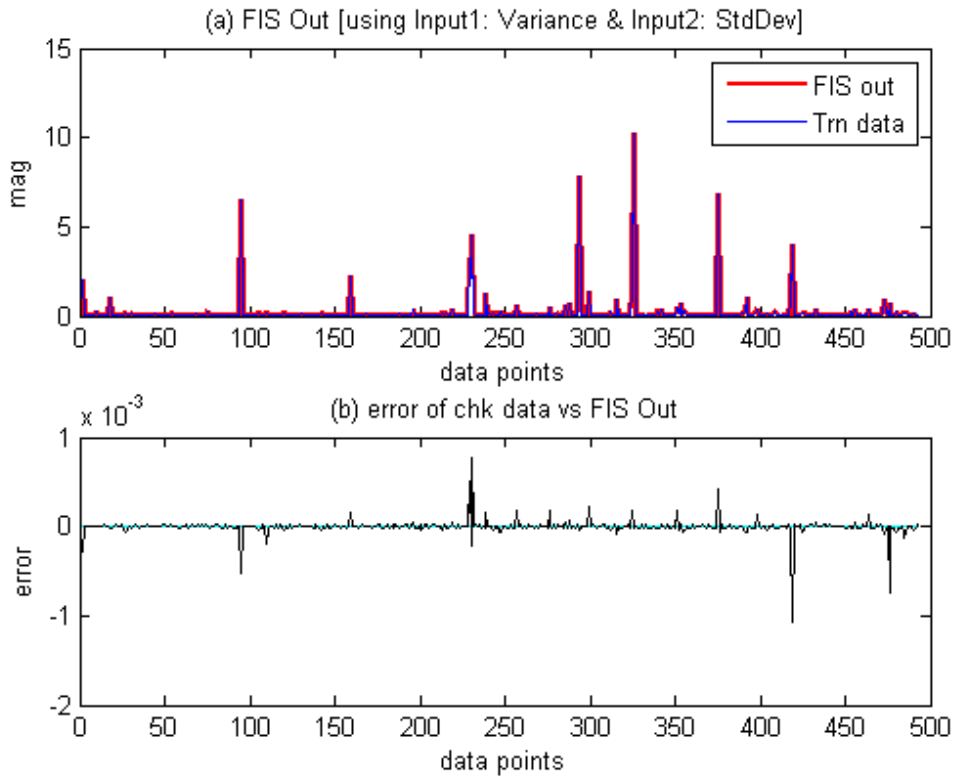

Figure 7.15 (a) IRF FIS Target Output and training data Target Output, (b) error between FIS out and training data Target Output

Figure 7.15 shows IRF FIS unit output pattern based on Input 1 and Input 2 that was compared to the Target Output of the checking data used to evaluate the corresponding selected IRF FIS unit. The *x*-axis represents the number of training data points (498) and the *y*-axis represents the amplitude magnitude of the Energy level.

Figure 7.15(a) shows the Target Output of the IRF FIS under evaluation when it was supplied with Standard Deviation as Input 1 and Variance as Input 2. It also shows the Target Output of the checking data that had to be matched by the IRF FIS unit under evaluation. The error between the FIS Output and the training data Target Output is shown in Figure 7.15(b).

The characteristics of the IRF FIS units able to produce the Target Output based only on Input 1 and Input 2 represented the ability to capture the most distinct input-output relation of the seven features presented in the training data.

7.7.3 Ball Fault (BF) FIS Performance Evaluation

The overall lowest error results were produced by ball fault (BF) FIS units which were generated using db8 wavelet-based features. This finding was based on the evaluation of all BF FIS performance results that were obtained by comparing all evaluation results of db4, db12, db22 and db44 of BF FIS units.

The performance evaluation results of BF FIS units are presented in Figure 7.16 which show eight FIS units that were evaluated using 19 checking (chkdata) data sets. The lowest error was produced by BF FIS units nos. 1, 2, 5 & 6. An additional evaluation process was carried out to select the lowest average error results among the results of FIS units nos. 1, 2, 5 & 6.

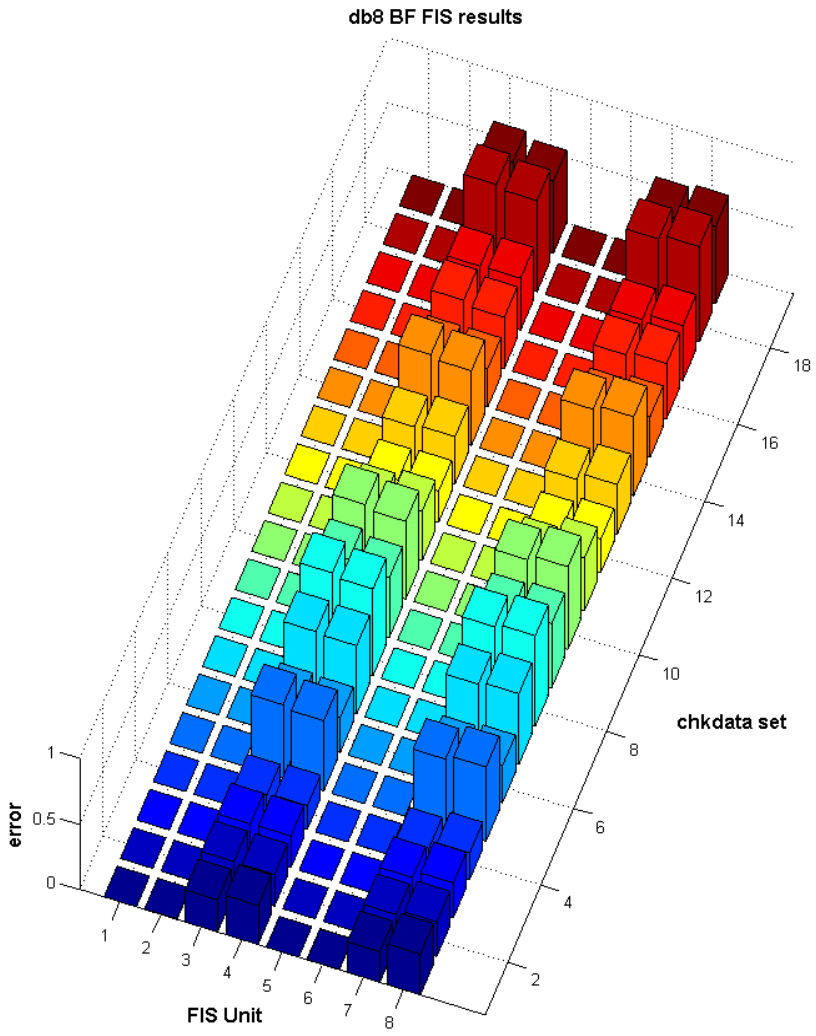

Figure 7.16 Test results of db8 BF FIS

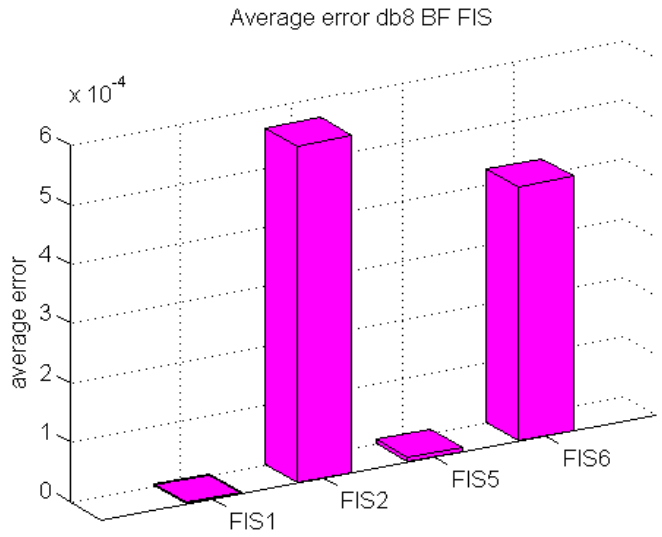

Figure 7.17 Average errors of db8 BF FIS nos. 1, 2, 5 & 6

The results of the additional selection process for BF FIS units nos. 1, 2, 5 & 6 is depicted in Figure 7.17. This shows that the lowest average error resulting from overall performance using 19 checking data sets was yielded by BF FIS no. 1, which hadthe lowest average error. Hence the BF FIS no. 1 was selected as the candidate for the fault classifier of BF case.

Figure 7.18 shows the BF FIS unit output pattern in comparison to the Target Output of the checking data used to evaluate the corresponding BF FIS unit. The *x*-axis represents the number of training data points (498) and the *y*-axis represents the amplitude magnitude of the Energy level.

Figure 7.18(a) shows the Target Output of the BF FIS under evaluation when it was supplied with Standard Deviation as Input 1 and Variance as Input 2. It also shows the Target Output of the checking data that had to be matched by the BF FIS unit under evaluation. The error between the Target Output and the FIS output is shown in Figure 7.18(b).

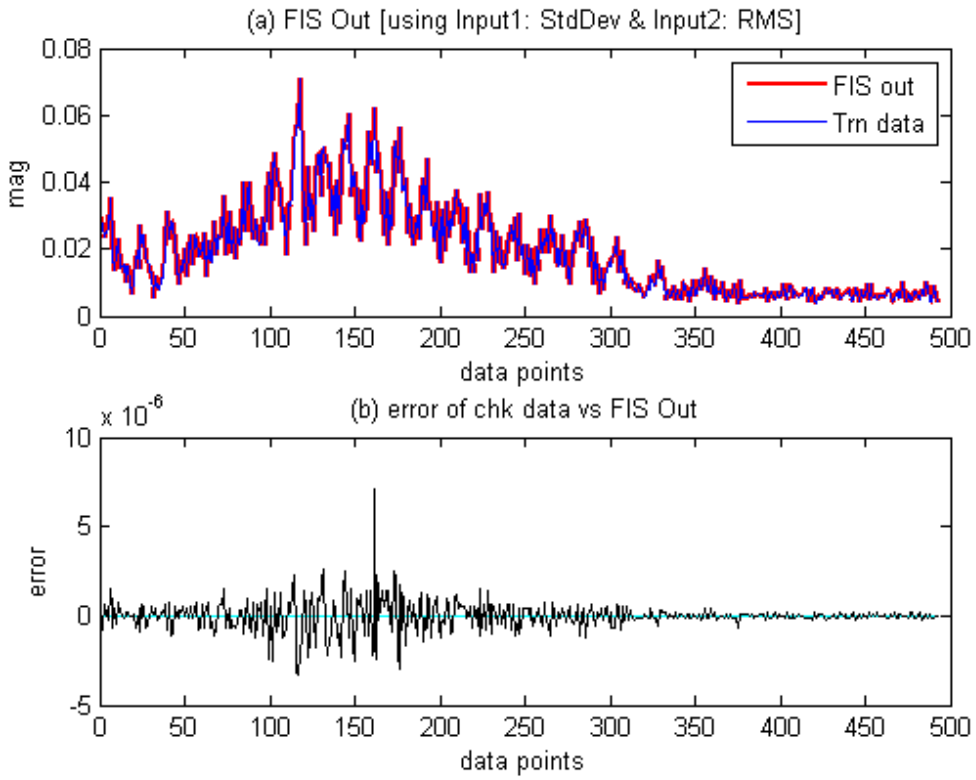

Figure 7.18(a) the BF FIS Target Output and training data Target Output, (b) error between the FIS out and the training data Target Output

The characteristics of the BF FIS units able to produce the Target Output values based only on Input 1 (Standard Deviation) and Input 2 (Variance) represents the ability to capture the most distinct input-output relationship of the seven features presented in the training data.

Note that even though Input 1 and Input 2 of the BF FIS are similar to those of the IRF FIS, the Target Output for the BF FIS is RMS which is different to that for IRF FIS (i.e., Energy level).

7.7.4 No Fault Performance Evaluation

The overall lowest error results were obtained from the no fault (NF) FIS units generated using db44 wavelet-based features. The performance evaluation results of these NF FIS units are presented in Figure 7.19.

Figure 7.19 shows that the lowest average error using six checking data sets was yielded by NF FIS nos. 1, 2, 5 &6.

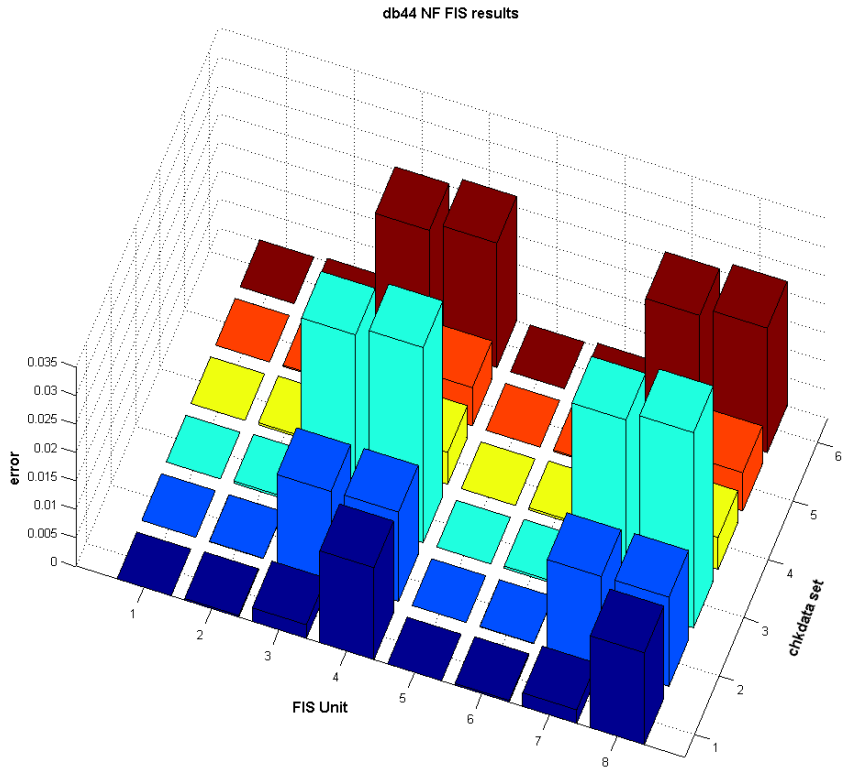

Figure 7.19Test results of db44 NF FIS

There were eight NF FIS units that were evaluated using 6 checking (chkdata) data sets. The lowest error trend was shown by NF FIS unit nos. 1, 2, 5 & 6. An evaluation process using only data from the NF FIS unit nos. 1, 2, 5 & 6 was carried out to obtain the lowest average error results among the results of the FIS unit nos. 1, 2, 5 and 6.

The graphical results of the average error of NF FIS unit nos. 1, 2, 5 & 6 are shown in Figure 7.20.

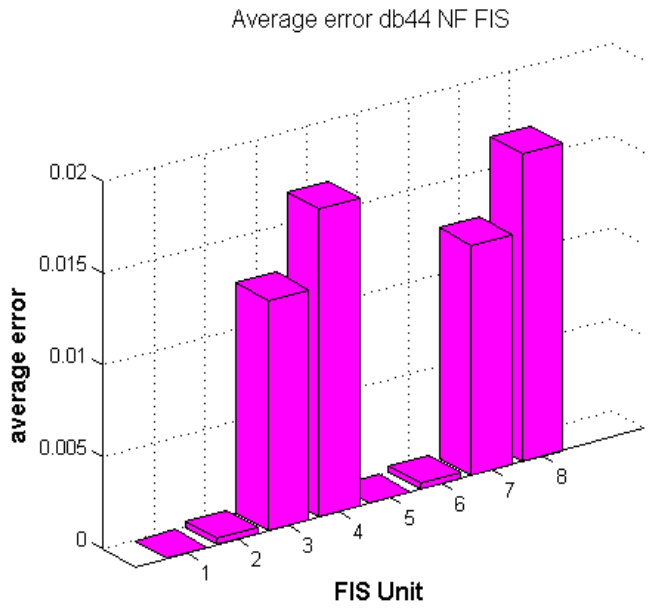

Figure 7.20 Average error of db44 FIS nos. 1, 2, 5 & 6

Figure 7.21 shows the NF FIS unit output compared to the Target Output of the checking data sets which was used in the evaluation process.

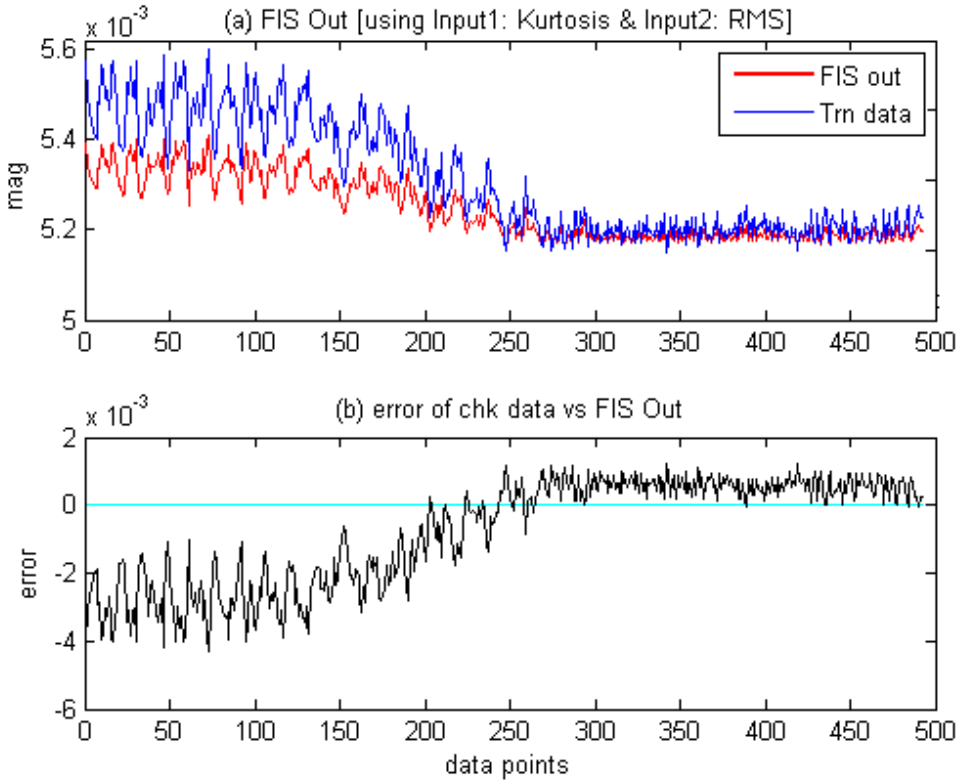

Figure 7.21 (a) NF FIS Output and training data Target Output, (b) error between NF FIS output and training data Target Output

Figure 7.21(a) shows the NF FIS output when supplied with RMS as Input 1 and Kurtosis as Input 2. It also shows the training data Target Output that must be matched by the NF FIS unit under evaluation. The error between the FIS output and the training data Target Output is shown in Figure 7.21(b).

The ability of the NF FIS units to produce the Target Output based only on Input 1 and Input 2 represents the most distinct input-output relationship of the seven features presented in the training data. This input-output relationship was an important aspect for the purpose of NF case classification since it is uniquelyapplied in the NF case.

7.8 Sensitivity Comparison of db-n Wavelet-based Feature Schemes

The db-n sensitivity comparison of the ORF, IRF BF, and NF FIS units aimed to find which type of Daubechies (db-n) wavelet was sensitive in relation to the feature extraction of the seven features for each of the ORF, IRF, BF and NF cases.

The investigation focused on finding which type of db-n wavelet used to generate feature data yielded the lowest error in the FIS unit implementation for ach fault case.

The FIS units of the db-n based wavelet which gained the lowest average error were used as indicators showing that a particular type of db-n was the most sensitive or suitable for use in producing the seven feature data sets provided for the ANFIS training.

The sensitivity comparison results were obtained by investigating the average lowest error among the FIS units which were generated using db4, db8, db12, db22 and db44 wavelets for each type of fault condition.

7.8.1 Outer Race Fault (ORF) db-n Sensitivity Comparison

The db-n output error comparison of the ORF FIS is depicted in Figure 7.22. There were 7 checking data sets used to evaluate the db4, db8, db12, db22 and db44 FIS units.

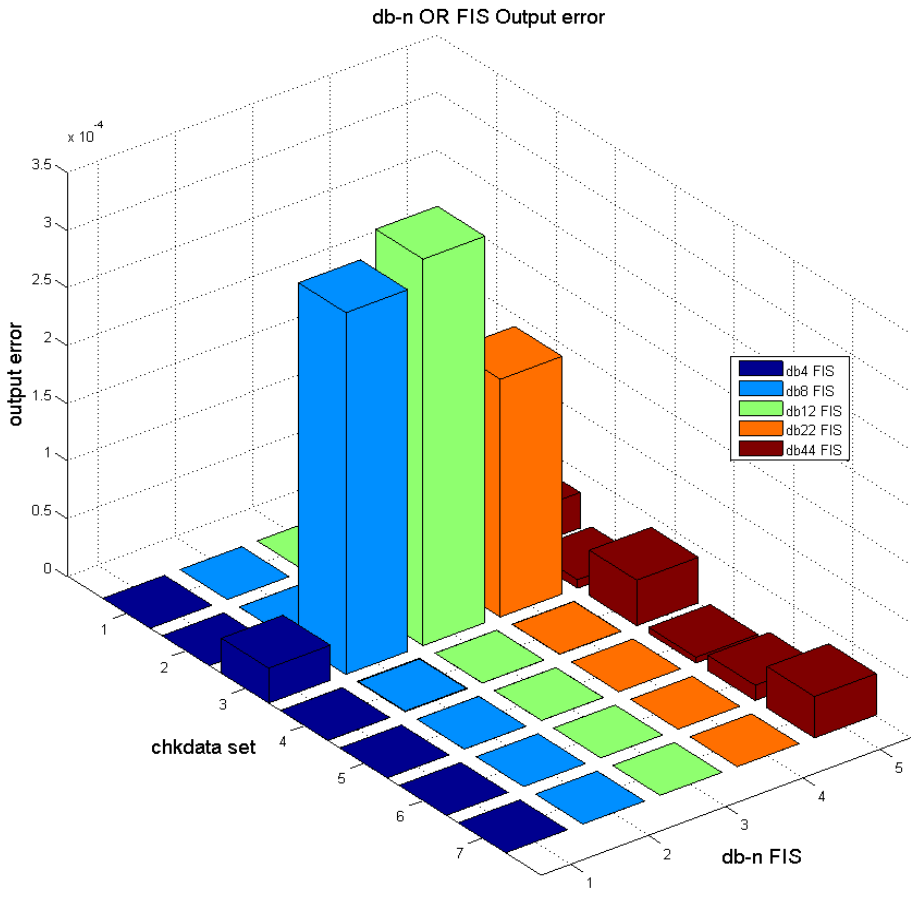

Figure 7.22 Average errors of all db-n results of ORF FIS units

The comparison of the overall db-n sensitivity was based on the error values obtained from all db-n ORF FIS Target Outputs and compared to the Target Outputs of the chkdata.

In Figure 7.22, the overall lowest error of Target Output was achieved by FIS units produced using db4 features data. The finding implies that Daubechies 4 (db4) wavelet was the most sensitive db-n wavelet (the best wavelet function) for use in generating the seven feature data sets for the ORF in the ANFIS training process.

The total average error of the results of Figure 7.22 is summarised in Figure 7.23.

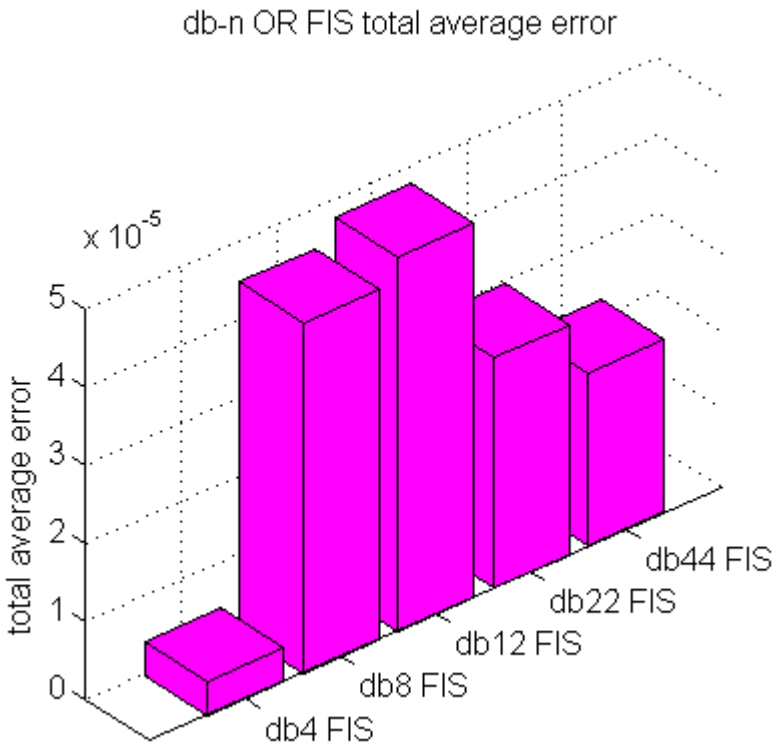

Figure 7.23 Average errors of all db-n results of ORF FIS units

In Figure 7.23, the db4 FIS unit yielded the lowest total average error. Hence it was concluded that the best wavelet type to generate seven features was the db4.

7.8.2 Inner Race Fault (IRF) db-n Sensitivity Comparison

The db-n output error comparison of the IRF FIS is depicted in Figure 7.24. There were seven checking data sets used to evaluate db4, db8, db12 and db44 FIS units. It was noted that there were no FIS unit candidates available for the db22 group.

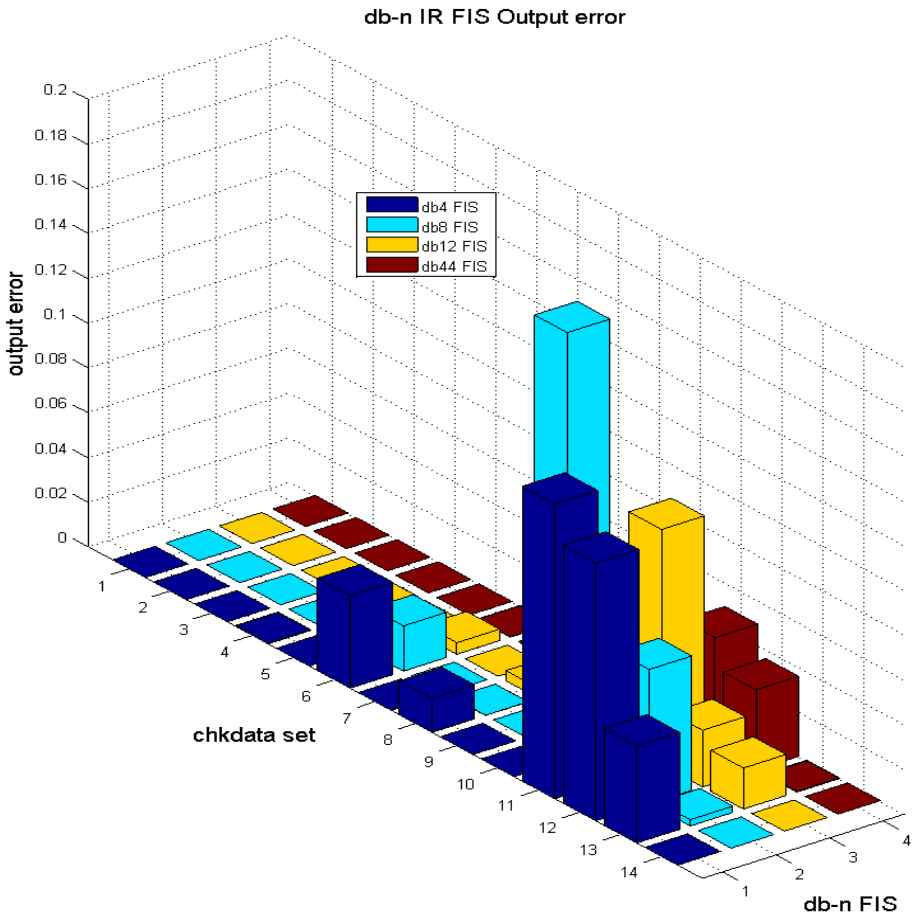

Figure 7.24 Average error across all db-n results of IRF FIS units

The comparison results of the overall db-n sensitivity (suitability) based on the error values among db-n IRF FIS Target Output compared to the Target Output of chkdata are depicted in Figure 7.24. The lowest error of the Target Output was achieved by the FIS units producing db44 features data. Hence it was implied that the Daubechies 44 (db44) wavelet was the most sensitive (the best wavelet function) for use in generating the seven-feature data for IRF for the ANFIS training process.

The total average error of the results of Figure 7.24 is summarised in Figure 7.25.

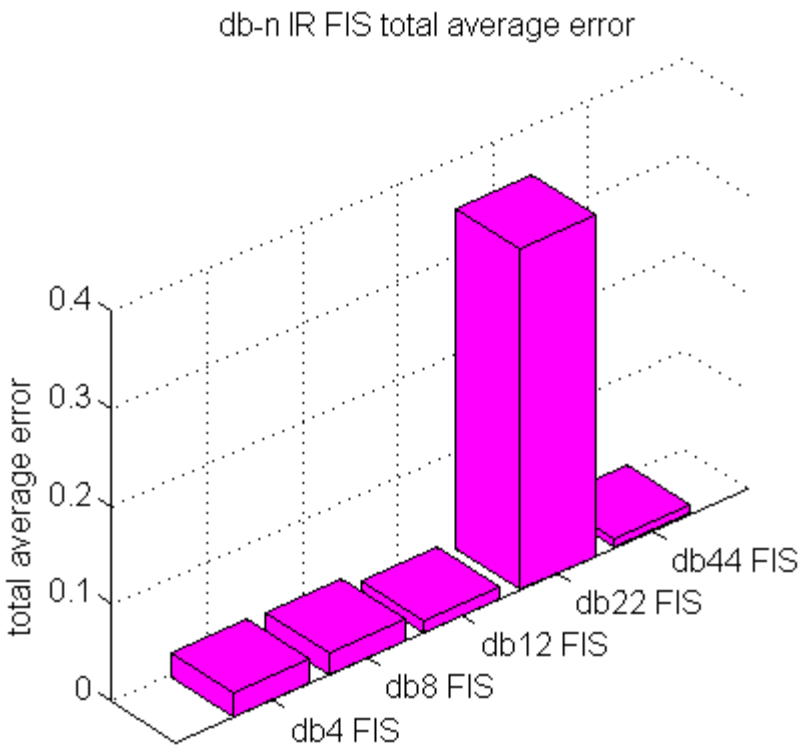

Figure 7.25 Average error of all db-n results of IRF FIS units

In Figure 7.25, the db44 FIS units yielded the lowest total average error. It was concluded that the db44 wavelet was the most sensitive (the best wavelet function) to generate the seven features data for the IRF case.

7.8.3 Ball Fault (BF) db-n Sensitivity Comparison

The db-n output error comparison of the BF FIS is depicted in Figure 7.26. There were seven checking data sets used to evaluate db4, db8, db12 and db44 FIS units.

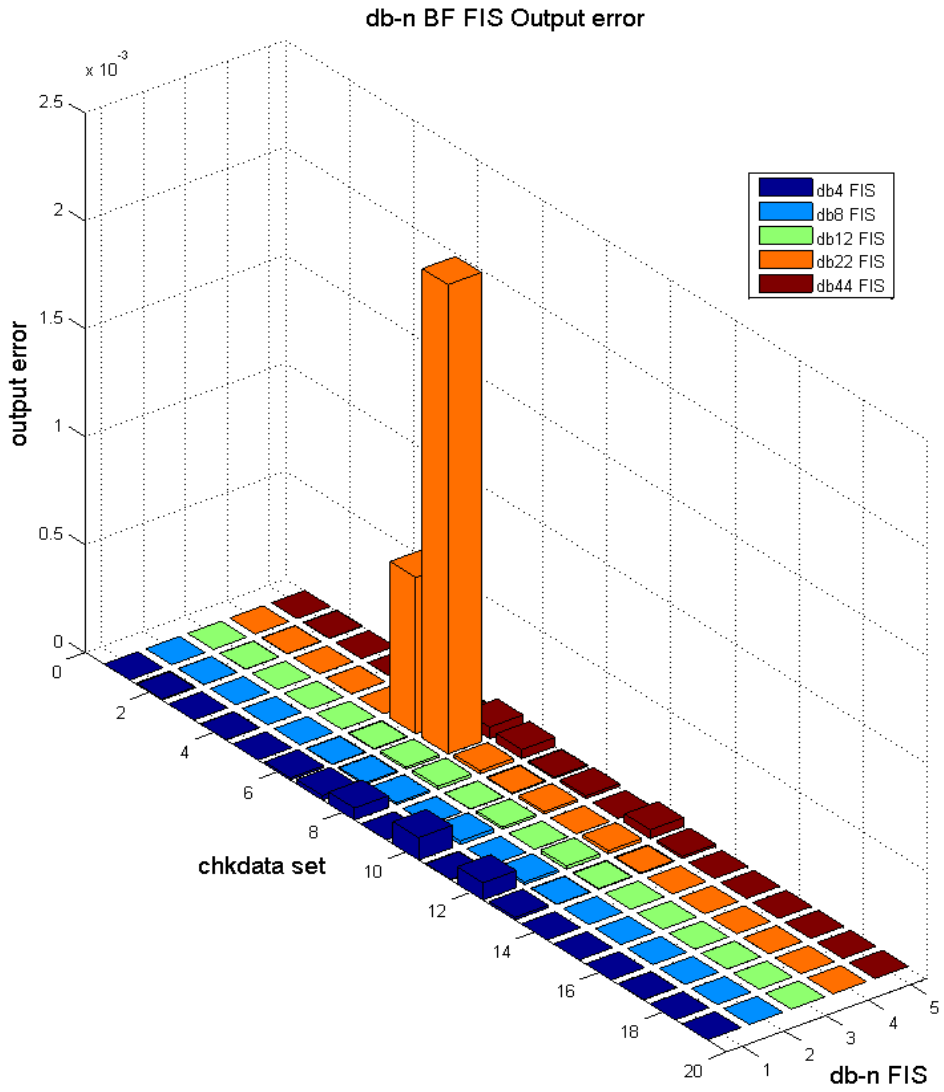

Figure 7.26 Average error of all db-n results of BF FIS units

The comparison results of the overall sensitivity, based on error values for all db-n BF FIS Target Outputs was compared to the Target Output of chkdata set. The results are shown in Figure 7.26. The lowest error of Target Output was achieved by the BF FIS unit producing the db8 features data. It was concluded that the Daubechies 8 (db8) wavelet was the most sensitive (suitable) wavelet type for use in generating features data for the IRF case.

The total average error of the results of Figure 7.26 is summarised in Figure 7.27.

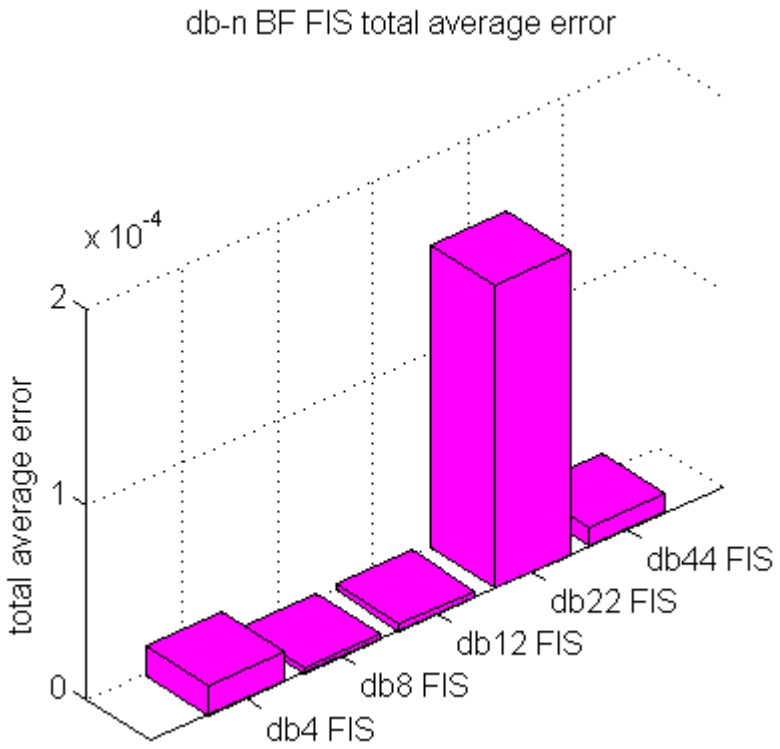

Figure 7.27 Average error of all db-n results of BF FIS units

7.8.4 No Fault (NF) db-n Sensitivity Comparison

The comparison results of the overall sensitivity based on error values of all db-n NF FIS Target Output was compared to the Target Output of chkdata set. The results are shown in Figures 7.28 and 7.29. The lowest error of Target Output was achieved by NF FIS units producing db44 features data. Hence it was concluded that Daubechies 44 (db44) wavelet was the most sensitive or suitable wavelet type to be used in generating features data for the NF case.

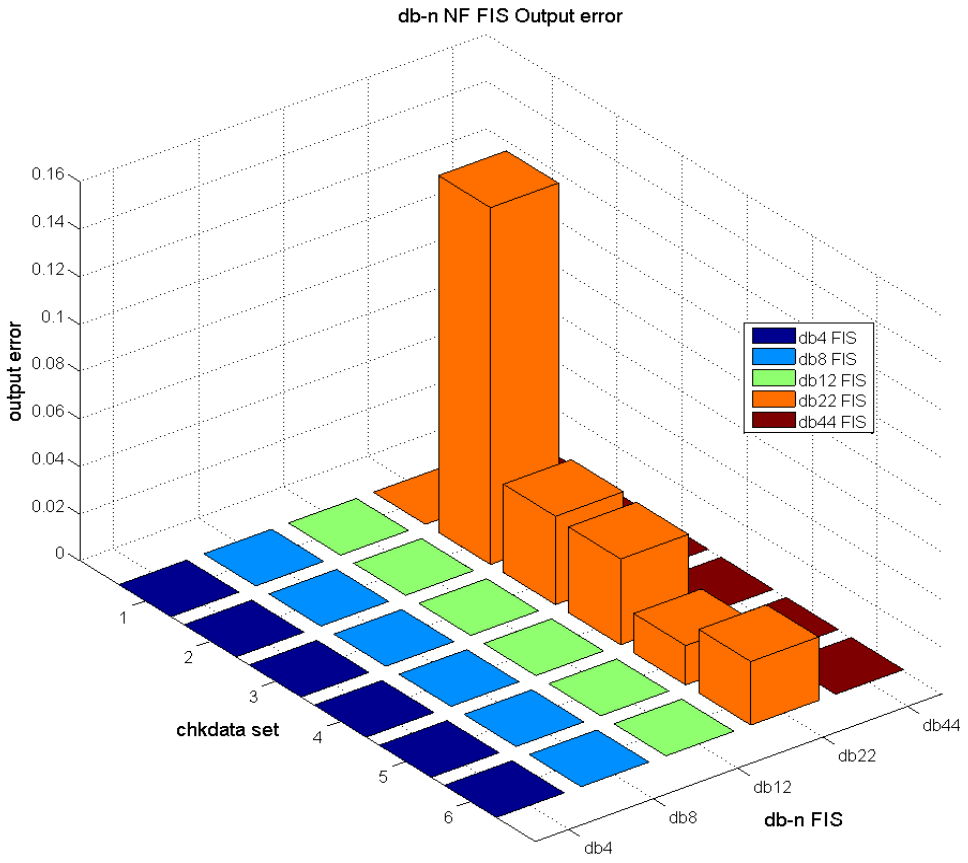

Figure 7.28 Average error of all db-n results of NF FIS units

The total average error of the results of Figure 7.28 is visualised in Figure 7.29.

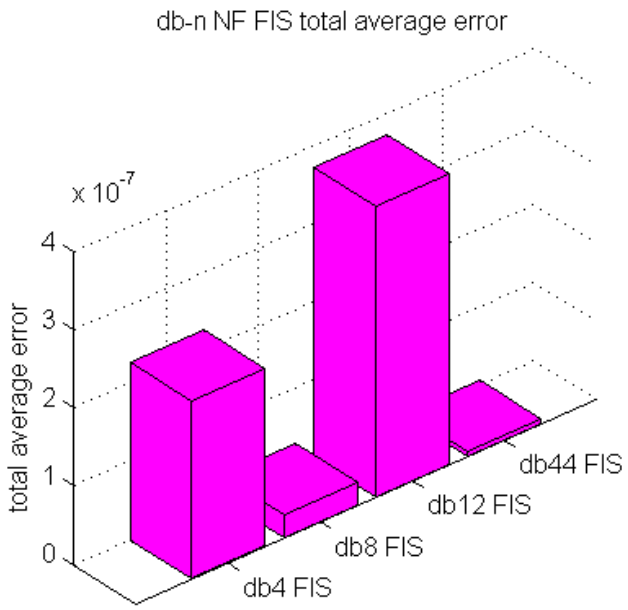

Figure 7.29 Average error of all db-b results of NF FIS units

In Figure 7.29, the db44 FIS unit yielded the lowest total average error. It was concluded that the db44 wavelet was the best wavelet for generating the seven features data for NF case.

7.9 The Application of the FIS Units as Fault Classifiers

In this study, the FIS units of ORF, IRF BF and NF were applied and tested in an automatic bearing fault classification system. The scheme of the system is shown in Figure 7.30. The Matlab code for the fault classification system in Figure 7.30 is listed in Appendix 6.

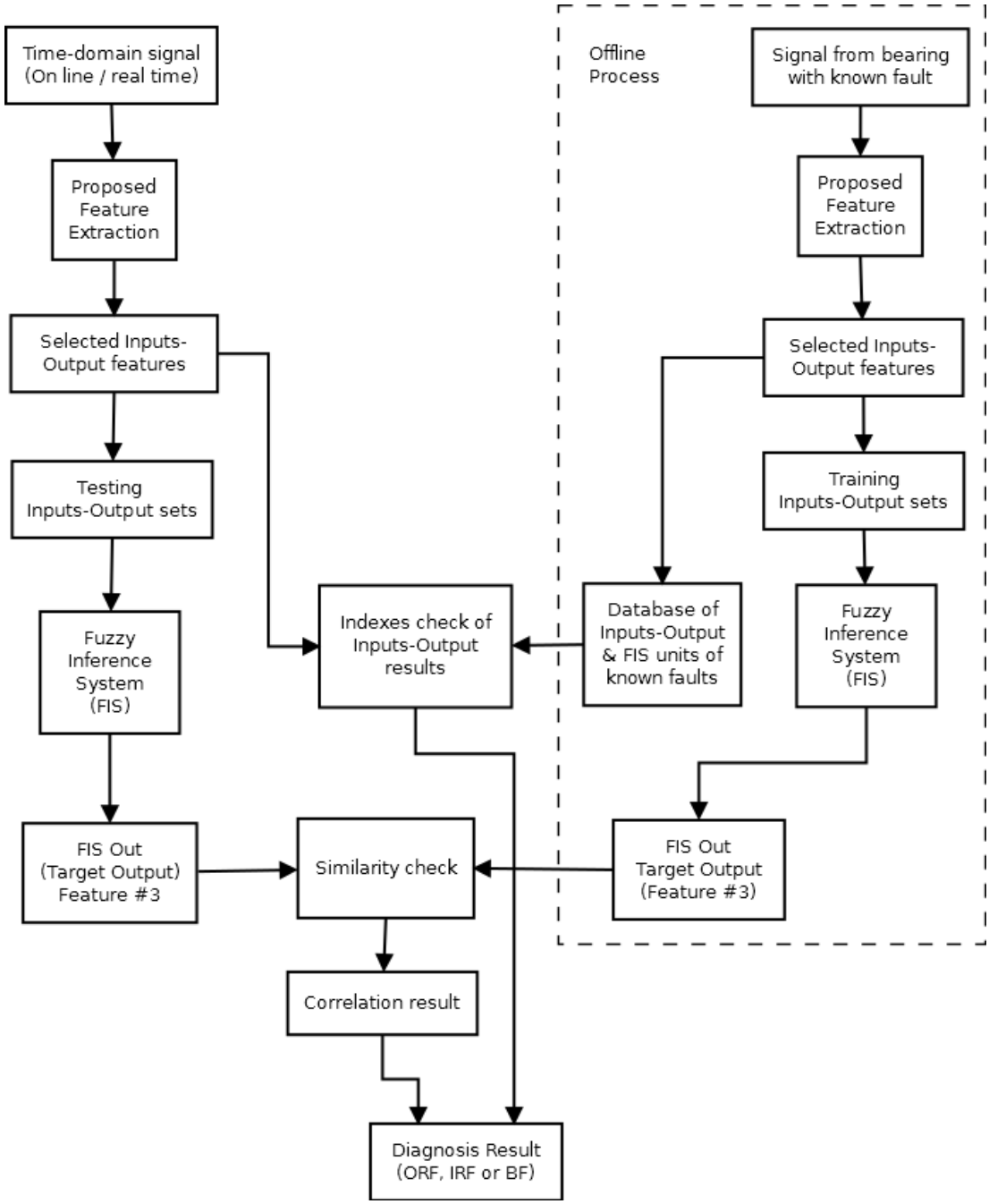

Figure 7.30 Application scheme of the FIS unit as a bearing fault classifier

In the application of the system, vibration signals being investigated were processed using the proposed wavelet-based feature extraction to generate the seven extracted features. The features were generated using the most sensitive Daubechies (db-n) wavelet type for each type of fault case. The proposed feature extraction method was used to provide input-output information (input features and output feature) previously obtained for each fault case. The information was used to assemble the input-output features data set. The assembled features data was then sent to the

appropriate FIS unit (core classifier). The FIS unit (core classifier) then produced the particular Target Output based on the presented parameters of Input 1 and Input 2.

The information of the two dominant inputs (Input 1 and Input 2) for each fault case was taken from the information of the known two dominant input-outputs related to a specific fault case (i.e., ORF, IRF, BF or NF). This input information is shown in Table 7.2.

Table 7.2 Summary of FIS Selected Inputs – Target Output features

Fault Type	**Input1 Feature**	**Input2 Feature**	**Target Output Feature**
Outer Race fault (ORF)	Standard Deviation (6)	RMS (2)	Energy (1)
Inner Race fault (IRF)	Standard Deviation (6)	Variance (7)	Energy (1)
Ball fault (BF)	Standard Deviation (6)	Variance (7)	RMS (2)
No Fault (NF)	RMS (2)	Kurtosis (3)	Standard Deviation (6)

As shown in Table 7.2, the statistical and Energy level features showed input-output relationships in each fault case. These relationships were the results obtained automatically by the ANFIS model. The relationships were used to construct the FIS units for all fault cases investigated.

The two dominant inputs (Input 1 and Input 2) listed in Table 7.2 were used to instruct the classifier (FIS unit) to generate or produce the Target Output in each fault case classification process. The Target Output was later used as a parameter or indicator in determining the type of bearing fault. The online Target Output was then compared to the Target Output of fault patterns stored in the database (training data). The online Target Output and that stored in the database were compared for

similarity. The similarity results of these two Target Outputs were presented as correlation numbers. The correlation result was used as an indicator to determine the type of fault or diagnosis results in conjunction with matching the checked results of the Input indexes.

7.10 Fault Classification Results

The overall fault classification results of the system shown in Figure 7.31 have been investigated using the complete sets of vibration features data. The results are shown in Table 7.3.

The success-rate of the FIS classifiers to differentiate non-related fault features data was varied. The classifier system shown in Figure 7.31 was tested by using different types of fault data (i.e., different loading conditions) to test the different FIS units implemented. There were cross checking processes carried out to evaluate how ORF, IRF, BF and NF FIS units performed when these FIS units were subjected to non-related fault features data. For instance, the ORF classifier (FIS unit) was subjected to IRF, BF and NF features data that was generated for each corresponding vibration signal that contained each type of fault.

The ability of ORF, IRF, BF and NF FIS units to differentiate non-related features data is shown as a "percentage of recognition" in Tables 7.3a –7.3d.

Table 7.3a Summary of NF FIS Classification Rates

FIS	db-n	Recognition Rate	Remarks
NF FIS	**db12**	**100%**	**(NF self recognition) fault case**
NF FIS	db4	27%	(ORF recognition) non-related case
NF FIS	db4	27%	(IRF recognition) non-related case
NF FIS	db4	77%	(BF recognition) non-related case

Table 7.3b Summary of BF FIS Classification Rates

FIS	db-n	Recognition Rate	Remarks
BF FIS	**db22**	**100%**	**(BF self recognition) fault case**
BF FIS	db12	0%	(ORF recognition) non-related case
BF FIS	db12	0%	(IRF recognition) non-related case
BF FIS	db12	0%	(NF recognition) non-related case

Table 7.3c Summary of IRF FIS Classification Rates

FIS	db-n	Recognition Rate	Remarks
IRF FIS	**db12**	**93%**	**(IRF self recognition) fault case**
IRF FIS	db8	32%	(ORF recognition) non-related case
IRF FIS	db22	95%	(BF recognition) non-related case
IRF FIS	db22	89%	(NF recognition) non-related case

Table 7.3d Summary of ORF FIS Classification Rates

FIS	db-n	Recognition Rate	Remarks
ORF FIS	**db44**	**99%**	**(ORF self recognition) fault case**
ORF FIS	db4	56%	(IRF recognition) non-related case
ORF FIS	db4	88%	(BF recognition) non-related case
ORF FIS	db4	77%	(NF recognition) non-related case

The high percentages for all self-recognitions (fault cases) showed that the FIS classifiers achieved a high recognition of the related fault case, which the units which recognised and classified for the purposes of bearing fault diagnosis.

Clearly, low percentages are preferred for the successful identification of all non-related cases of fault feature data for the classifiers. These conditions are preferred in the classification process as they produce better performance and reduce misclassification results.

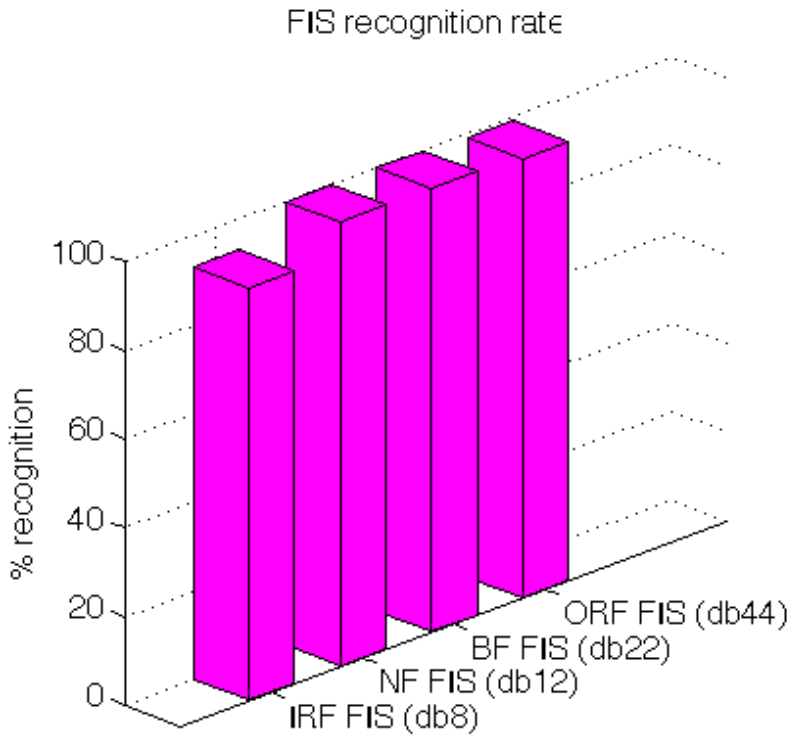

Figure 7.31 Overall FIS classifiers recognition rate

The summary of recognition rate (classification rate) shown in boldface font in Table 7.3a through Table 7.3d is depicted in Figure 7.31. The recognition rate represents the successful classification rate of each fault case achieved by the FIS units used in the bearing fault classification system. The classification rate for NF and BF FIS units achieved an overall recognition rate of 100%. The recognition rate of IRF units achieved 93% and ORF FIS units achieved a 99% recognition rate.

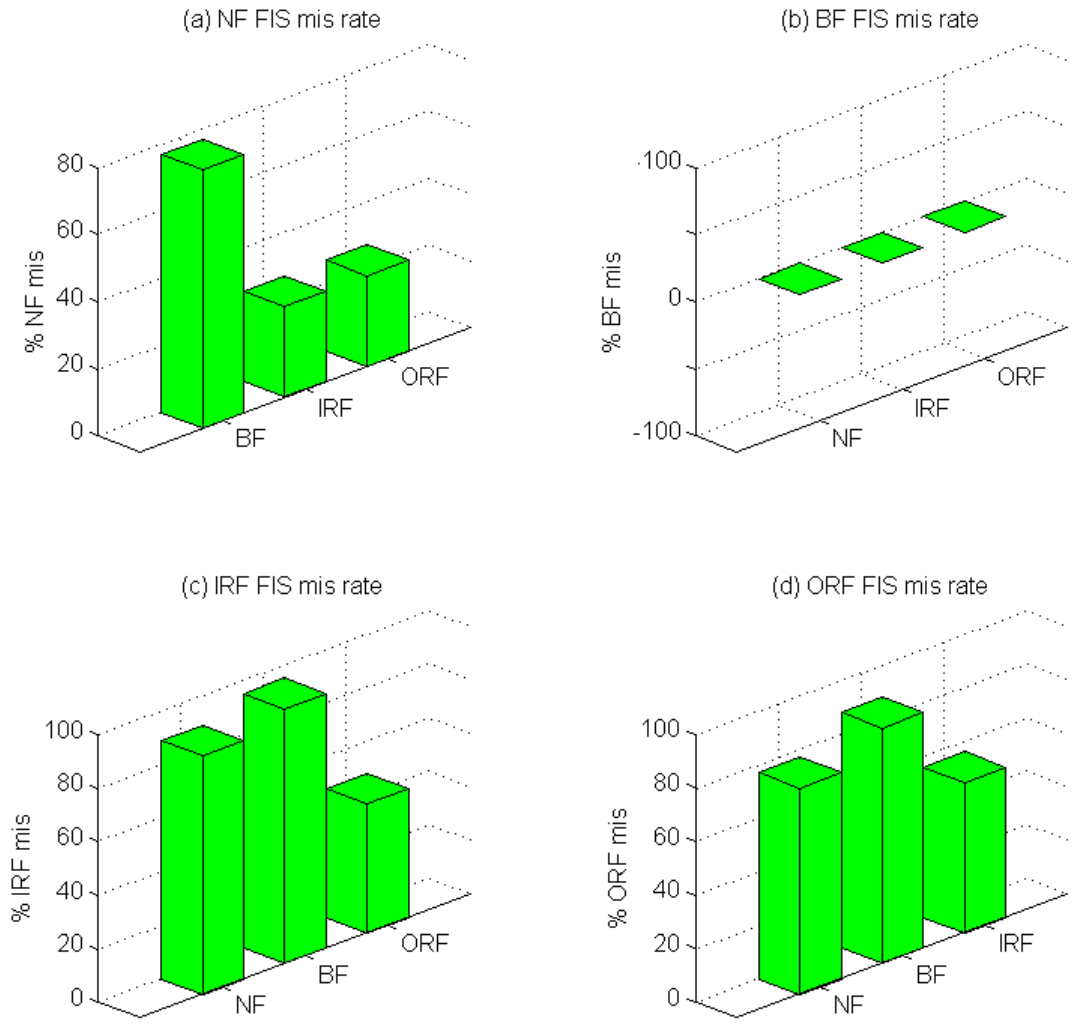

Figure 7.32 Overall FIS classifiers mis-classification rate

A summary of the non-related case classification rate is shown in Table 7.3a through Table 7.3d and visualised in Figure 7.32. Only the BF FIS units (classifiers) achieved 0% mis-classification when tested with non-related cases data. The non-related data was fault feature data that was not generated from a vibration signal containing the same fault case. In this context, the BF FIS unit used in the bearing fault classification system successfully identified the BF case as intended. The ORF, IRF and NF cases were completely rejected. The BF classifier reacted only to classify the BF case and it was therefore not confused with other cases.

The mis-classification rate (confusion rate) of NF, ORF and IRF varied, as shown in Figure 7.32(a), 7.32(c) and 7.32(d). The mis-classification rate ranged from a low of 48% to 95% at the highest. The FIS units obtained for NF, ORF and IRF cases had

the highest mis-classification rate for BF cases. The NF, ORF and IRF FIS classifier had a high tendency to mis-interpret BF feature data as NF, ORF or IRF cases.

There are several findings which are highlighted based on the information shown in Tables 7.3a – Table 7.3d. The findings were as follows:

7.10.1 No Fault (NF) Classifiers

The NF FIS units used as classifiers achieved 100% recognition for a signal with No Fault (NF). The rejection percentage of NF classifiers towards non-related cases of ORF and IRF cases was 73%. However, it was found that the rejection percentage towards BF cases was low at 33%. This showed that the NF classifier had a tendency to misclassify a BF case as an NF case 77% of the time. It was also found that NF FIS generated from db12 was chosen for the purpose of classifying the NF cases, while NF FIS units generated from db4 were used to classify other non-related cases (ORF, IRF, and BF).

This means that in the application of the NF classifier, it produced 100%, or a unity correlation in the classification result for its related case (i.e., NF case). For other non-related cases, it would not produce a result > 73% (correlation > 0.73) for the three other fault cases (i.e., ORF, IRF and BF). Preferably, the results for the other (non-related) cases would be < 50%. It would therefore be useful to distinguish the NF case from the others.

7.10.2 Ball Fault (BF) Classifiers

The BF FIS units that used as classifiers achieved 100% recognition for signals with a ball fault (BF). The rejection percentage of BF classifiers towards non-related cases of ORF and IRF cases was 100%. This meant that the BF classifiers had 0% tendency toward misclassification for all non-related cases. BF FIS units generated from db22 were chosen to to classify the BF cases and for ORF, IRF, and NF cases.

In the testing process of the BF FIS, It produced correct classification for BF cases with a 100 per cent success rate or unity correlation. For non-related cases of ORF, IRF and NF cases, it produced non-unity correlation, which meant that the classifier would not produce any classification result for the fault cases which were not

intended to be classified. This is preferred as the BF FIS does not react to produce any correlation for the other fault cases which are not intended to be classified, while producing 100% correlation for the BF case.

7.10.3 Inner Race Fault (IRF) Classifiers

IRF FIS used as classifiers achieved a 93% recognition rate for signals with an inner race fault (IRF).

This means that the IRF FIS would only produce a result of 93% (0.93 correlation) out of the preferred 100% (unity correlation) for the intended classification of the IRF case.

The rejection percentage of IRF classifiers towards non-related cases of ORF was 32%. It meant that the IRF classifiers had a 68% tendency to misclassify an ORF case as an IRF case.

The classifier produced a 68 % (0.68 correlation) result for the non-intended ORF case which was preferred to be 0 % (non unity correlation result).

For non-related cases, or non-related signals of BF, the rejection percentage of the IRF classifier was 5%. It meant that the IRF classifiers had a 95% tendency to misclassify in BF cases.

The classifier produced 95 % (0.95 correlation) result for the non-intended BF case which was preferred to be 0% (non unity correlation) result. This result could produce confusion in conclusions regarding faults, since the correlation result of non-intended fault cases (i.e., BF case) is higher than the correlation result for the intended cases. It is clear that further refinement is needed to reduce the correlation produced by the IRF classifiers for the non–related BF case. Hence it would then not produce such a high correlation classification result for the BF case which could lead to mis-classification results in using this classifier.

For non-related cases of NF, the rejection percentage of the IRF classifier was 11%. This meant that the IRF classifiers had an 89% tendency towards misclassification of

NF cases. That is, the classifier produced a 0.89 correlation for the non-intended NF case.

This result implies that a further refinement for the classifier is needed to reduce the correlation result of the classifier (preferably < 50%) for the classification of the non-related IRF case.

IRF FIS units generated from db12 were chosen to classify the IRF case. IRF FIS units generated from db8 were used to classify non-related case of ORF. IRF FIS units generated from db22 were used to classify non-related cases of BF and NF.

7.10.4 Outer Race Fault (IRF) Classifiers

ORF FIS units used as classifiers achieved a 99% recognition rate (0.99 correlation) (of successful classification) for signals with outer race faults (ORF). The rejection percentage of ORF classifiers for non-related cases of IRF was 44%. This meant that the ORF classifiers had a 56% tendency toward misclassification or showed a 0.56 correlation for IRF cases.

This result is preferred since the difference between the intended ORF case correlation and the non-intended ORF case was sufficient distinguish them.

The rejection percentage of ORF classifier for non-related case of BF was 12%. It meant that the BF classifiers had 88% tendency of misclassification for BF case. This result is a not preferred one since the classifier produced a 0.88 correlation to the BF case which is not intended for it. A refinement process could be done to improve the rejection characteristics of the ORF classifier towards the BF case.

The rejection percentage of ORF classifiers for non-related cases of NF was 23%. This meant that the NF classifiers had a 77% tendency toward misclassification for NF cases.

These results imply that a refinement process should be carried out to reduce the high percentage (i.e, 77%) of mis-classification tendency (0.77 correlation) of the ORF classifier towards the NF case.

ORF FIS generated from db8 was chosen to classify the ORF case, while ORF FIS units generated from db4 were used to classify non-related cases (IRF, BF and NF).

7.11 Unrelated Features

The introduction of *cA_x* and *cA_y* parameters into features data for ANFIS training showed no significant relationship to other features as these two parameters were not selected by the ANFIS model during training sessions. The parameters or features were not related to Inputs or Output in the ANFIS features selection used to construct a FIS unit of each investigated fault case. This implied that these two features could have been left out as features without detriment to the building of fault classifiers through ANFIS training.

7.12 Test of ClassifiersUsing Different Data

The test of classifiers (FIS units) using external vibration data obtained from another test rig at NASA was also carried out. The test aims to assess the applicability of the produced classifier.

Suitable data for the test was found at the URL:
http://ti.arc.nasa.gov/m/project/prognostic-repository/bearing_IMS.zip.
The details regarding the bearing data is available on a readme document provided within the zip file on the site. This data was also used in a study of roller bearing prognostics by Qiu *et al*. (2006).

There are three sets or group of data available. However, only one set of data could be used as it had clear documentation regarding the type of fault. The data set number 2 (second test data) was seen as suitable for the purposes of testing the FIS classifier. This data contained an ORF case which was acquired during the "go to fail" life-test of a roller bearing. The data set also contained the file number that matched the file number used in this research.

The second group or the second test of the NASA repository contained 4 columns of data sets. All of this column-data was used to generate the required features (i.e., the seven features) through the use of wavelet transforms and statistical parameters, as proposed in this research. The seven features were then used to test the

corresponding classifier (i.e., the ORF classifier). The related two dominant inputs (Input 1 and Input 2) and the Target Output which represent the characteristics of the ORF case embedded in the structure of the classifier were used to select the correct input-output. In this case, for the ORF case, the two inputs were Standard Deviation (Input 1) and RMS (Input 2). The Target Output was the Energy level.

The results were similar for all four columns of data sets used. An example of the results is shown in Figure 7.33.

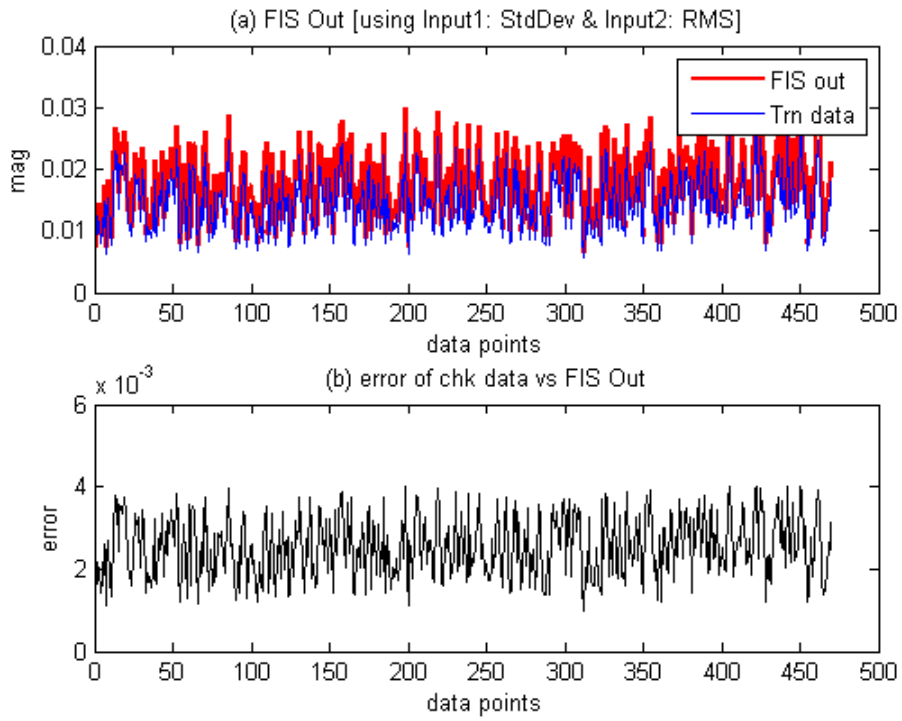

Figure 7.33ORF FIS Target Output and training data Target Output, (b) error between FIS Out and training data Target Output using NASA bearing data

It is shown from the results depicted in Figure 7.33 that the ORF FIS classifier shows similar characteristics to those shown by using in-house vibration data. The ORF classifier produced an accurate Target Output with an error in the magnitude of 10^{-3} which was based on Standard Deviation and RMS features. This result shows that in general, the ORF classifier is able to produce an accurate Target Output for the features data of similar fault cases used in the training. The ORF classifier shows a general applicability to an ORF case. It was able to recognise the relationships

between input-output of features data, even if the features data was generated from a different test rig or from a different fault investigation project.

The next step of the test, which was conducted using the external data, was to test the ORF classifier with different sets of data which were not related to an ORF case. In this case, the first test data set from the NASA repository, which was not an ORF case, was used. This data set contained vibration signals from a spall fault in the roller bearing which was subjected to a life-span test.

The data set contains 8 columns each. There were only four data columns used for the simplicity of generating the seven features using the proposed wavelet transform and statistical parameters techniques. The features data was then used to test the ORF classifier. The results were similar for all four columns of data sets used. An example of the results is shown in Figure 7.34.

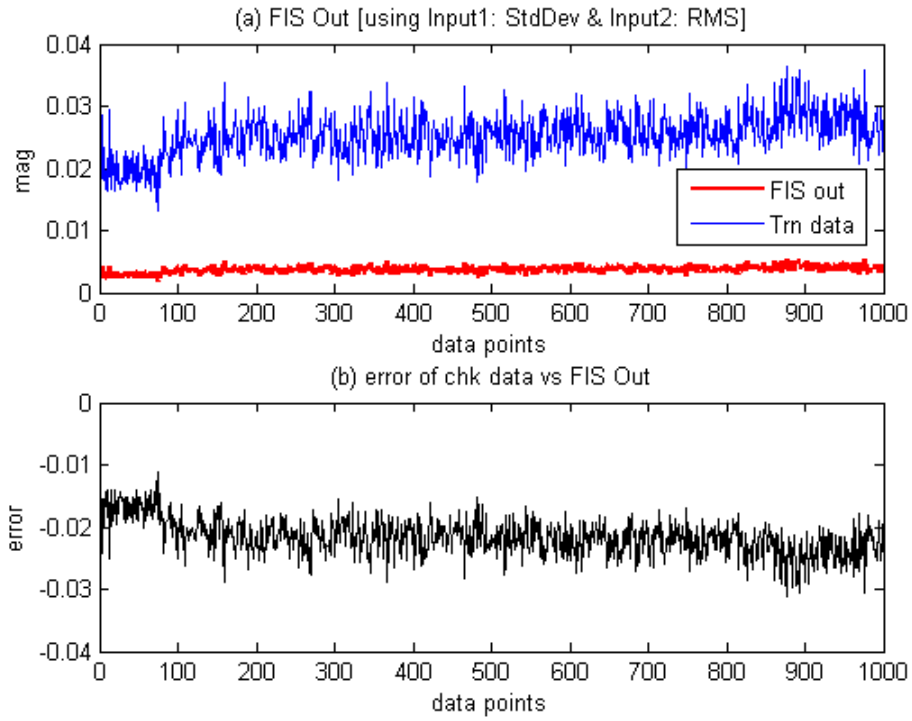

Figure 7.34 ORF FIS Target Output and training data Target Output, (b) error between FIS Out and training data Target Output using NASA bearing data for a non related case

The ORF classifier produced a larger scale of error when it was tested using non-related features data. Figure 7.34(b) shows this finding in which the error scale of the

Target Output for non related cases was of the magnitude of 10^{-1}. In Figure 7.33(b), for the ORF (related) case, the error scale was of the magnitude of 10^{-3}.These findings suggest promising results in the context of the general applicability of the proposed method. The Target Output errors for the ORF case and the non-related cases show a distinct level of error which may be useful for the purposes of further fault classification.

Chapter 8 - Conclusions and Future Research

8.1 Conclusions

The proposed feature extraction and classification scheme for bearing fault diagnosis was investigated in order to further explore bearing fault diagnosis techniques. This area has not yet been fully researched and is still open to new discoveries. The investigations in this paper produced new knowledge on the advantages and disadvantages of using wavelet transforms and ANFIS models in building bearing fault classifiers. The findings of the investigation provided new knowledge and methodology regarding the feature extraction of vibration signals from rolling element bearings. These findings have also contributed to the application of artificial intelligence methodology which interprets extracted features.

The investigation of a new feature extraction method was carried out and this led to the construction of the bearing fault classification system reported on in this thesis. The features extraction method combined the application of generated features based on wavelet-transform results, their energy levels, and other statistical indicators. The bearing fault classifiers generated by using the ANFIS training procedures in the proposed method were used successfully in the determination of the relationships between the chosen seven statistical features.

The development of wavelet-based features extraction, combined with an ANFIS model, aimed to search for a new features extraction method that might lead to the construction and application of new bearing fault classifiers.

The proposed wavelet extraction method produced a set of unique features from the original vibration signals. The unique features were then used as training data for an ANFIS model. The results of the ANFIS training procedures were FIS units which were capable of recognising the related features which represented the unique features of input-output relationships. These FIS units were used as the core classifiers in a classification system for bearing faults.

The procedure tested four classifiers which were generated for four types of bearing fault signals in order to determine the performance of each classifier. The test results showed that the BF case classifier achieved 100% accuracy in identifying a BF case. The rejection rate or mis-classfication rate of the BF classifier towards non-related cases (i.e., ORF, IRF and NF) was 100%.

The lowest accuracy level in the classification results was 93 %, although this result was obviously quite high. This percentage occurred in the test results of the IRF classifier. The NF and ORF test results showed that both classifiers achieved 100% accuracy for fault classification in the cases of NF and ORF.

However, the test results showed also that the ORF, IRF and NF classifiers were affected by mis-classification or confusion in classifying feature data which did not belong to the particular classifiers. This condition refers to the situation where a classifier had the tendency to classify a non-related fault in a particular fault case of its own. The highest mis-classification trend, found in the test results of the ORF classifier towards BF features data, was 95%. The lowest (and hence the best) result was 27% which was found in the test results of the NF classifier towards the IRF, ORF and BF features data, respectively.

The mis-classification characteristics were related to the features data used to generate the FIS units in the ANFIS training process. The mis-classification of a classifier unit of a particular non-related case was caused by some level of similarity with the features data used to generate the FIS classifier. A high similarity level of features data produced a high level of ambiguity or confusion in the classification process. Therefore, it was important to provide a higher level of unique characteristics in the features data in the ANFIS training for each fault case investigated.

The following sections document the final conclusions of the investigation of the proposed ANFIS and wavelet-based feature extraction process.

As was inferred from the results of the literature review, feature extraction methods in the area of bearing fault diagnosis are still open to exploration, especially in the

area of the combination between wavelet-based features extraction and artificial intelligence models such as a neuro-fuzzy system (i.e., ANFIS).

This research focused on exploring a new method of generating the related features of bearing faults by using a combination of wavelet transforms and an ANFIS model. There were seven features generated from the wavelet transform results of a vibration signal from a faulty bearing acquired from the test rig. An ANFIS was used to select the three most dominant features that were mapped as two Inputs towards a single Target Output. These three most dominant features were produced for each of the four fault cases investigated. Further, core FIS units were implemented in a complete bearing classifier system that was generated by using the ANFIS training procedure proposed in this research.

The test results show that only the BF classifier achieved 100 % accuracy for the BF case classification. In addition, it also achieved 0% mis-classification when presented with other non-related cases (i.e., ORF, IRF and NF cases). This means that only the BF classifier could uniquely determine the presence of a BF fault.

The statistical parameters (i.e., RMS, kurtosis, standard deviation and variance, Energy level, dominant frequency and maximum amplitude of dominant frequency) calculated from the wavelet transform results were used as fault features. The features were introduced into the ANFIS model in order to select the highest correlated features that might best represent a particular fault characteristic. The relationships between three features selected were learned by the ANFIS model, and the learned result was represented as an FIS unit for each training session.

The application of the ANFIS in the selection process of the most dominant Input-Output features provided an enhancement to the feature extraction process and reduced the required post-processing tasks which are usually conducted by human experts. ANFIS offered an automatic selection of the most related or dominant features that best represented a particular fault case. In this case, the unrelated features which were not best characterised as a fault case, were rejected or removed prior to the construction of FIS units.

Test results for the classifier units for each type of bearing fault case shows that only the BF case can determine a fault without mis-classification. The results also exposed the limitations of the proposed feature extraction method and FIS classifiers. It was found that the separation or distinction rate of the FIS units in identifying signals that were not intended to contain the related fault was low in several FIS units.

There was no single db-n wavelet type tested that proved successful for all feature extractions in the ORF, IRF, BF and NF cases. Rather, each fault case had to be detected with a different db-n FIS unit.

The feature extraction methods and bearing fault classifiers proposed in this research achieved a mixed success rate across ORF, IRF, BF and NF classification cases. The mis-classification in ORF, IRF and NF classifiers hindered the ability of these classifiers in identifying each of the fault cases correctly. The only exception was found in the BF classifiers, which produced a 100 % successful classification rate, i.e., there were no mis-classifications.

To conclude this section and in answering the research questions posed, it can be said that the proposed feature extraction scheme can be used for the purpose of fault identification and classification, with several limitations as pointed out the previous discussions. That is, the BF classifiers can be used immediately after they have been produced, while the other classifiers cannot.

8.2 Future Work

There are several areas related to this research that could be investigated further in order to explore refinements of the proposed feature extraction method and the bearing classifiers.

There is a need to investigate the use of different types of wavelets to generate the data features in the proposed feature extraction method beyond those used here. For instance, the utilisation of wavelet types such as symlet and coifflet in the feature extraction process may be beneficial.

Investigation of the effect of db-n wavelet types used to generate wavelet-based features is needed, since the results show that there were certain db-n types that produced better classification results for each particular bearing fault type. On the other hand, the results also showed that there were certain db-n types which were not suited for use in feature generation for a particular fault case. For example, there were no FIS units that matched the error and correlation standard in the selection of FIS units for the NF case that was generated using the db22 wavelet during ANFIS training. This implies that the db22 wavelet is not suited for the purpose of feature extraction for the NF case, since all of the obtained FIS units failed to produce a correlation > 0.99 to the Target Output of the training data.

The findings infer there was a relationship of sensitivity in the application of db-n type wavelets investigated in this research. The sensitivity of a db-n wavelet type used to generate features data for the ANFIS training was related to the shape of the mother wavelet function. Further investigation is needed to explore the results which showed that the mother wavelet function with higher similarity to the "peakness" of the vibration signal was more sensitive to be used in features extraction. The mother wavelet shape's similarity to the "peakness" of the vibration signal enhanced the characteristics of the vibration signal. The enhanced characteristics were produced in the form of the resulting wavelet transform of the processed vibration signal.

An investigation to explore the possibility of streamlining the permutations check procedures is advisable. This would shorten the ANFIS training time and reduce computation loads. A method of selecting only the useful permutation combinations could be investigated with the aim of obtaining a method that would select only useful permutation combinations (indices). These indices need to be tested during the training process and others that are not useful would be rejected.

Further exploration of the selection procedures for features data is needed, with the aim of increasing the accuracy of the FIS classifiers, especially in the classification of cases that showed a low rejection rate for the non-related cases. The initial step towards this future work would be started with the analysis of the characteristics of the features data. Specifically, the analysis would focus on the Target Output features data that was used to train the ANFIS and that generated the FIS units (classifiers).

The proposed feature extraction method and the bearing classification system needs to be tested using different vibration data generated from different sources. This test would assess the generalisation ability of the proposed methods. A test using external data would be useful for the purpose of assessing the general performance of the proposed feature extraction method and the fault classification system.

There is scope to investigate additional fault types which are not studied here. These additional faults could be diagnosed and classified by the proposed bearing classification system. This could be achieved by introducing additional types of bearing fault cases to increase the diversity in applications of the classifier. However, until the mis-classifications can be reduced significantly, this work will remain secondary.

There is scope for investigating the reduction of the number of training and checking data points needed to train an ANFIS without losing the accurate representation and generalisation of the feature data. This would contribute to the reduction of feature data pre-processing time.

Future work could also aim to extend the application of the proposed methods to fault detection and classification in other rotating machinery systems such as gears and pumps.

References

Abbasion, S., Rafsanjani, A., Farshidianfar, A., and Irani, N. 2007, 'Rolling Element Bearings Multi-fault Classification Based on the Wavelet Denoising and Support Vector Machine', *Mechanical Systems and Signal Processing*, vol. 21, no. 7, pp. 2933-2945.

Abu-Mahfouz, I.A. 2005, 'A Comparative Study of Three Artificial Neural Networks for the Detection and Classification of Gear Faults', *International Journal of General Systems*, vol. 34, no. 3, pp. 261-277.

Al-Badour, F., Sunar, M., and Cheded, L. 2011, 'Vibration Analysis of Rotating Machinery Using Time–frequency Analysis and Wavelet Techniques', *Mechanical Systems and Signal Processing*, vol. 25, no. 6, pp. 2083-2101.

Alguindigue, I.E., Loskiewicz-Buczak, A., and Uhrig, R.E. 1993, 'Monitoring and Diagnosis of Rolling Element Bearings Using Artificial Neural Networks', *IEEE Transaction on Industrial Electronics*, vol. 40, no. 2, pp. 209-217.

Altmann, J., and Mathew, J. 2001, 'Multiple Band-pass Autoregressive Demodulation for Rolling-element Bearing Fault Diagnosis', *Mechanical Systems and Signal Processing*, vol. 15, no. 5, pp. 963-977.

Altug, S., Chow, M.-Y., and Trussell, H.J. 1999, 'Fuzzy Inference Systems Implemented on Neural Architectures for Motor Fault Detection and Diagnosis', *IEEE Transactions on Industrial Electronics*, vol. 46, no. 6, pp. 1069-1079.

Antonino-Daviu, J.A., Riera-Guasp, M., Folch, J.R., and Palomares, M.P.M. 2006, 'Validation of A New Method for the Diagnosis of Rotor Bar Failures via Wavelet Transform in Industrial Induction Machines', *IEEE Transactions on Industry Applications*, vol. 42, no. 4, pp. 990-996.

Baillie, D., and Mathew, J. 1994, 'Diagnosing Rolling Element Bearing Faults with Artificial Neural Networks', *Acoustics Australia*, vol. 22, no. 3, pp. 79-84.

Bay, O.F., and Bayir, R. 2005, 'Kohonen Network Based Fault Diagnosis and Condition Monitoring of Pre-engaged Starter Motors', *International Journal of Automotive Technology*, vol. 6, no. 4, pp. 341-350.

Bazovsky, I. 1961, *Reliability Theory and Practice*, Prentice-Hall, New York.

Bin, G.F., Gao, J.J., and Dhillon, B.S. 2012, 'Early Fault Diagnosis of Rotating Machinery Based on Wavelet Packets - Empirical Mode Decomposition Feature Extraction and Neural Network', *Mechanical Systems and Signal Processing*, vol. 27, pp. 696-711.

Braun, S., and Datner, B. 1979, 'Analysis of Roller / Ball Bearing Vibrations', *Journal of Mechanical Design, Transactions of ASME*, vol. 101, no. 1, pp. 118-125.

Brigham, E., 1988, *the Fast Fourier Transform and Its Applications*, Prentice-Hall, Englewood Cliffs, New Jersey.

Broomhead, D.S., and Lowe, D. 1988, 'Multivariable Function Interpolation and Adaptive Networks', *Complex System*, vol. 2, pp. 321-355.

Brotherton, T., Jahns, G., Jacobs, J., and Wroblewski, D. 2000, 'Prognosis of Faults in Gas Turbine Engines', *IEEE Aerospace Conference Proceedings 2000*, vol. 6, pp. 163-171.

Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., and Rosen, D.B. 1992, 'Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps', *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 698-713.

Castejon, C., Lara, O., and Garcia-Prada, J.C. 2010, 'Automated Diagnosis of Rolling Bearings Using MRA and Neural Networks', *Mechanical Systems and Signal Processing*, vol. 24, no. 1, pp. 289-299.

Celik, M.B., and Bayir, R. 2007, 'Fault Detection in Internal Combustion Engines Using Fuzzy Logic', *Proceedings of the Institution of Mechanical Engineers*, vol. 221, no. D5, pp. 579-587.

Chaturvedi, G.K., and Thomas, D.W. 1982, 'Bearing Fault-Detection Using Adaptive Noise Cancelling', *Journal of Mechanical Design*, vol. 104, no. 2, pp. 280-289.

Chen, Z., and Mechefske, C.K. 2000, 'Diagnosis of Machine Faults Based on Transient Vibration Signals', *INSIGHT – Non-Destructive Testing and Condition Monitoring*, vol. 42, no. 8, pp. 504-511.

Chen, Z., and Mechefske, C.K. 2002, 'Diagnosis of Machinery Fault Status Using Transient Vibration Signal Parameters', *Journal of Vibration and Control*, vol. 8, no. 3, pp. 321-335.

Chinnam, R.B., and Baruah, P. 2004, 'A Neuro-fuzzy Approach for Estimating Mean Residual Life in Condition-based Maintenance Systems', *International Journal of Materials and Product Technology*, vol. 20, no. 1-3, pp. 166-179.

Chui, C. K. 1992, *Wavelet Analysis and Its Applications Vol.1: Introduction to Wavelets*, Academic Press, Boston, Massachusetts.

Crandall, R.E., 1994, *Projects in Scientific Computation*, TELOS, Santa Clara, California.

Daadbin, A. 1991, 'Different Vibration Monitoring Techniques and Their Application to Rolling Element Bearings', *International Journal of Mechanical Engineering Education*, vol. 19, no. 4, pp. 295-304.

Dalpiaz, G., and Rivola, A. 1997, 'Condition Monitoring and Diagnostics in Automatic Machines: Comparison of Vibration Analysis Techniques', *Mechanical Systems and Signal Processing*, vol. 11, no. 1, pp. 53-73.

Dalpiaz, G., Rivola, A., and Rubini, R. 2000, 'Effectiveness and Sensitivity of Vibration Processing Techniques for Local Fault Detection in Gears', *Mechanical Systems and Signal Processing*, vol. 14, no. 3, pp. 387-412.

Daubechies, I. 1990, 'The Wavelet Transform, Time-frequency Localization and Signal Analysis', *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961-1005.

Demartines, P., and Herault, J. 1997, 'Curvilinear Component Analysis: A Self-organizing Neural Network for Nonlinear Mapping of Data Sets', *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 148-154.

Donat, W., Choi, K., An., W., Singh, S. and Pattipati, K. 2008, 'Data Visualization, Data Reduction and Classifier Fusion for Intelligent Fault Diagnosis in Gas Turbine Engines', *Journal of Engineering for Gas Turbines and Power (Transactions of the ASME)*, vol. 130, no. 4.

Du, Q., and Yang, S, 2007, 'Application of the EMD Method in the Vibration Analysis of Ball Bearings', *Mechanical Systems and Signal Processing*, vol. 21, no. 6, pp. 2634-2644.

Dyer, D., and Stewart, R.M. 1978, 'Detection of Rolling Bearing Element Damage by Statistical Vibration Analysis', *Journal of Mechanical Design*, vol. 100, no. 2, pp. 229-235.

Evsukoff, A., and Gentil, S. 2005, 'Recurrent Neuro-fuzzy System for Fault Detection and Isolation in Nuclear Reactors', *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 55-66.

Fukunaga, K. 1990, *Introduction to Statistical Pattern Recognition*, Academic Press, Boston, Massachusetts.

Gaing, Z.-L. 2004, 'Wavelet-based Neural Network for Power Disturbance Recognition and Classification', *IEEE Transactions on Power Delivery*, vol. 19, no. 4, pp. 1560-1568.

Galushkin, A.I. 2007, *Neural Networks Theory*, Springer, Heidelberg.

Gao, X.Z., and Ovaska, S.J. 2001, 'Soft Computing Methods in Motor Fault Diagnosis', *Applied Soft Computing*, vol. 1, no. 1, pp. 73-81.

Garga, A.K., McClintic, K.T., Campbell, R.L., Yang, C.-C., Lebold, M.S., Hay, T.A., and Byington, C.S. 2001, 'Hybrid Reasoning for Prognostic Learning in CBM Systems', *IEEE Aerospace Conference 2001 Proceedings*, vol. 1-7, pp. 2957-2969.

Gebraeel, N., Lawley, M., Liu, R., and Parmeshwaran, V. 2004, 'Residual Life Predictions from Vibration-Based Degradation Signals: A Neural Network Approach', *IEEE Transactions on Industrial Electronics*, vol. 51, no. 3, pp. 694-700.

Gelle, G., Colas, M., and Serviere, C. 2001, 'Blind Source Separation: A Tool for Rotating Machine Monitoring by Vibrations Analysis?', *Journal of Sound and Vibration*, vol. 248, no. 5, pp. 865-885.

Gertsbakh, I.B. 1976, *Models of Preventive Maintenance*, North-Holland Publication Co., Amsterdam.

Goddu, G., Li, B., Chow, M.-Y., and Hung, J.C. 1998, 'Motor Bearing Fault Diagnosis by A Fundamental Frequency Amplitude Based Fuzzy Decision System', *IEEE Annual Conference Proceedings of Industrial Electronics Society*, vol. 1-4, pp. 1961-1965.

Goode, P.V., and Chow, M.-Y. 1995, 'Using A Neural/Fuzzy System to Extract Heuristic Knowledge of Incipient Faults in Induction Motors: Part II-Application', *IEEE Transactions on Industrial Electronics*, vol. 42, no. 2, pp. 139-146.

Goswami, J.C., and Chan, A.K. 1999, *Fundamentals of Wavelets: Theory, Algorithms and Application*s, John Wiley & Sons, New York.

Goumas, S., Zervakis, M., Pouliezos, A., and Stavrakakis, G.S. 2001, 'Intelligent On-line Quality Control of Washing Machines Using Discrete Wavelet Analysis Features and Likelihood Classification', *Engineering Applications of Artificial Intelligence*, vol. 14, no. 5, pp. 655-666.

Graps, A. 1995, 'An Introduction to Wavelets', *IEEE Computational Science and Engineering*, vol. 2, no. 2, pp. 50-61.

Graupe, D. 1997, *Principles of Artificial Neural Networks*, World Scientific, Singapore.

Hauer, J.F., Demeure, C.J., and Scharf, L.L. 1990, 'Initial Results in Prony Analysis of Power System Response Signals', *IEEE Transactions on Power Systems*, vol. 5, no. 1, pp. 80-89.

Haykin S., 1999, *Neural Networks. A Comprehensive Foundation, Second Edition*, Pearson Prentice-Hall Publications, Ontario, Canada.

Heng, A., Zhang, S., Tan, A.C.C., and Mathew, J. 2009, 'Rotating Machinery Prognostics: State of the Art, Challenges and Opportunities', *Mechanical Systems and Signal Processing*, vol. 23, no. 3, pp. 724-739.

Heng, R.B.W., and Nor, M.J.M. 1998, 'Statistical Analysis of Sound and Vibration Signals for Monitoring Rolling Element Bearing Condition', *Applied Acoustics*, vol. 53, no. 1-3, pp. 211-226.

Ho, D., and Randall, R.B. 2000, 'Optimization of Bearing Diagnostic Techniques Using Simulated and Actual Bearing Fault Signals', *Mechanical Systems and Signal Processing*, vol. 14, no. 5, pp. 763-788.

Honarvar, F., and Martin, H.R. 1997, 'New Statistical Moments for Diagnostics of Rolling Element Bearings', *Journal of Manufacturing Science and Engineering*, vol. 119, no. 3, pp. 425-432.

Howard, I. 1994, *A Review of Rolling Element Bearing Vibration: "Detection, Diagnosis and Prognosis"*, DSTO Aeronautical and Maritime Research Laboratory, Melbourne, Australia.

Hu, N.-S., He, N.-N., and Hu, S. 2003, 'Fault Diagnosis of the Steam Turbine Condenser System Based on SOM Neural Network', *International Conference on Machine Learning and Cybernetics*, vol. 1-5, pp. 1222-1225.

Igarashi, T., and Hamada, H. 1982, 'Studies on the Vibration and Sound of Defective Rolling Bearings .1. Vibration of Ball-bearings with One Defect', *Bulletin of the Japan Society of Mechanical Engineers*, vol. 25, no. 204, pp. 994-1001.

Ilott, P.W., and Griffiths, A.J. 1997, 'Fault Diagnosis of Pumping Machinery Using Artificial Neural Networks', *Proceedings of the Institution of Mechanical Engineers*, vol. 211, no. E3, pp. 185-194.

Jack, L.B., and Nandi, A.K. 2001, 'Support Vector Machines for Detection and Characterization of Rolling Element Bearing Faults', *Proceedings of the Institution of Mechanical Engineers, Part C; Journal of Mechanical Engineering Science*, vol. 215, no. 9, pp. 1065-1074.

Jang, J.-S.R, Sun, C.-T., and Mizutani, E. 1997, *Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice-Hall, Upper Saddle River, New Jersey.

Jang, J.-S.R. 1993, 'ANFIS: Adaptive-Network-Based Fuzzy Inference System', *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 23 no. 3, pp. 665-685.

Jang, J.-S.R., and Sun, C.-T. 1995, 'Neuro-fuzzy Modeling and Control', *Proceedings of the IEEE*, vol. 83, no. 3, pp. 378-406.

Jardine, A.K.S. 1973, *Maintenance, Replacement and Reliability*, Pitman, London.

Jardine, A.K.S., Lin, D., and Banjevic, D. 2006, 'A Review on Machinery Diagnostics and Prognostics Implementing Condition-based Maintenance', *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483-1510.

Jayaswal, P., Verma, S.N., and Wadhwani, A.K. 2010. 'Application of ANN, Fuzzy Logic and Wavelet Transform in Machine Fault Diagnosis Using Vibration Signal Analysis', *Journal of Quality in Maintenance Engineering*, vol. 16, no. 2, pp. 190-213.

Jensen A., and La Cour-Harbo, A. 2001, *Ripples in Mathematics - The Discrete Wavelet Transform*, Springer-Verlag, New York.

Junsheng, C., Dejie, Y., and Yu, Y. 2005, 'Time-energy Density Analysis Based on Wavelet Transform', *NDT & E International*, vol. 38, no. 7, pp. 569-572.

Kankar, P.K., Sharma, S.C., and Harsha, S.P. 2011, 'Fault Diagnosis of Ball Bearings Using Continuous Wavelet Transform', *Applied Soft Computing*, vol. 11, no. 2, pp. 2300-2312.

Kohonen, T. 1997, *Self-Organizing Maps, Second Edition*, Springer, New York.

Kohonen, T., Oja, E., Simula, O., Visa, A., and Kangas, J. 1996, 'Engineering Applications of the Self-Organizing Map', *Proceedings of the IEEE*, vol. 84, no. 10, pp. 1358-1384.

Komgom, C.M., Mureithi, N.W., and Lakis, A.A. 2008,'Application of Time Synchronous Averaging, Spectral Kurtosis and Support Vector Machines for Bearing Fault Identification', *Proceedings of ASME Pressure Vessels and Piping Conference*, vol. 7, pp. 137-146.

Kothamasu, R., and Huang, S.H. 2007, 'Adaptive Mamdani Fuzzy Model for Condition-based Maintenance', *Fuzzy Sets and Systems*, vol. 158, no. 24, pp. 2715-2733.

Kowalski, C.T., and Orlowska-Kowalska, T. 2003, 'Neural Networks Application for Induction Motor Faults Diagnosis', *Mathematics and Computers in Simulation*, vol. 63, no. 3-5, pp. 435-448.

Kung, S.J., and Hwang, J.N. 1988, 'An Algebraic Projection Analysis for Optimal Hidden Units Size and Learning Rates in Back-propagation Learning', *IEEE International Conference on Neural Networks*, pp. 363-370.

Kuo, H.-C., Wu, L.-J., and Chen, J.-H. 2002, 'Neural-fuzzy Fault Diagnosis in A Marine Propulsion Shaft System', *Journal of Materials Processing Technology*, vol. 122, no. 1, pp. 12-22.

Latuny, J., and Entwistle, R.D. 2010, 'Bearing Fault Analyses through the Application of ANFIS and Vector Array Indicators Based on Statistical Parameters of Wavelet Transformation Components', *Proceedings of the 6th Australasian Congress on Applied Mechanics (ACAM 6) Conference*, pp. 552-559.

Lee, C.C. 1990, 'Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Parts I and II', *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 404-435.

Lee, S.K., and White, P.R. 1997, 'Higher-order Time-frequency Analysis and Its Application to Fault Detection in Rotating Machinery', *Mechanical Systems and Signal Processing*, vol. 11, no. 4, pp. 637-650.

Lei, Y., He, Z., Zi, Y., & Hu, Q. 2007, 'Fault Diagnosis of Rotating Machinery Based on Multiple ANFIS Combination with Gas', *Mechanical Systems and Signal Processing*, vol. 22, no. 5, pp. 2280-2294.

Lei, Y., He, Z., and Zi, Y. 2009, 'Application of an Intelligent Classification Method to Mechanical Fault Diagnosis'', *Expert Systems with Applications*, vol. 36, no. 6, pp. 9941-9948.

Leonhardt, S., and Ayoubi, M. 1997, 'Methods of Fault Diagnosis'. *Control Engineering Practice*, vol. 5, no. 5, pp. 683-692.

Li, C.J., and Wu, S.M. 1989, 'On-line Detection of Localized Defects in Bearings by Pattern Recognition Analysis', *Transaction of ASME,Journal of Engineering for Industry*, vol. 101, no. 4, pp. 331-336.

Li, F., Meng, G., Ye, L., and Chen, P. 2008, 'Wavelet Transform-based Higher-order Statistics for Fault Diagnosis in Rolling Element Bearings', *Journal of Vibration and Control*, vol. 14, no. 11, pp. 1691-1709.

Li, H., Zhang, Y., and Zheng, H. 2009, 'Gear Fault Detection and Diagnosis Under Speed-up Condition Based on Order Cepstrum and Radial Basis Function Neural Network', *Journal of Mechanical Science and Technology*, vol. 23, no. 10, pp. 2780-2789.

Li, L., Qu, L., and Liao, X. 2007, 'Haar Wavelet for Machine Fault Diagnosis', *Mechanical Systems and Signal Processing*, vol. 21, no. 4, pp. 1773-1786.

Li, L.X., Mechefske, C.K., and Li, W.D. 2004, 'Electric Motor Faults Diagnosis Using Artificial Neural Networks', *INSIGHT*, vol. 46, no. 10, pp. 616-621.

Liao, G., Shi, T., and Xuan, J. 2005, 'Feature Selection and Condition Monitoring of Gearbox Using SOM', *IEEE International Joint Conference on Neural Networks (IJNN)*, vol. 1-5, pp. 2313-2318.

Lim, H.S., and Su, H. 2006, 'Motor Fault Detection Method for Vibration Signal Using FFT Residuals', *International Journal of Applied Electromagnetics and Mechanics*, vol. 24, no. 1, pp. 209-223.

Ling, J., and Qu, L. 2000, 'Feature Extraction Based on Morlet Wavelet and Its Application for Mechanical Fault Diagnosis', *Journal of Sound and Vibration*, vol. 234, no. 1, pp. 135-148.

Liu, B., Ling, S.-F., and Meng, Q. 1997, 'Machinery Diagnosis Based on Wavelet Packets', *Journal of Vibration and Control*, vol. 3, no. 1, pp. 5-17.

Liu, S., and Shi, W. 2001, 'Rough Set Based Intelligence Diagnostic System for Valves in Reciprocating Pumps', *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1-5, pp. 353-358.

Liu, T.I., and Mengel, J.M. 1992, 'Intelligent Monitoring of Ball Bearing Conditions', *Mechanical Systems and Signal Processing*, vol. 6, no. 5, pp. 419-431.

Liu, T.I., Singonahalli, J.H., and Iyer, N.R. 1996, 'Detection of Roller Bearing Defects Using Expert System and Fuzzy Logic', *Mechanical Systems and Signal Processing*, vol. 10, no. 5, pp. 595-614.

Lou, X., and Loparo, K.A. 2004, 'Bearing Fault Diagnosis Based on Wavelet Transform and Fuzzy Inference', *Mechanical Systems and Signal Processing*, vol. 18, no. 5, pp. 1077-1095.

Loutridis, S.J. 2008, 'Gear Failure Prediction Using Multiscale Local Statistics', *Engineering Structures*, vol. 30, no. 5, pp. 1214-1223.

Lu, C., Ma, N., and Wang, Z. 2011, 'Fault Detection for Hydraulic Pump Based on Chaotic Parallel RBF Network', *EURASIP Journal on Advances in Signal Processing*, vol. 49.

Luo, J., Yu, D., and Liang M. 2013, 'A Kurtosis-guided Adaptive Demodulation Technique for Bearing Fault Detection Based on Tunable-Q Wavelet Transform', *Measurement Science and Technology*, vol. 24, no. 5.

Mahamad, A.K., and Hiyama, T. 2011, 'Fault Classification Based Artificial Intelligent Methods of Induction Motor Bearing', *International Journal of Innovative Computing Information and Control*, vol. 7, no. 9, pp. 5477-5494.

Malhi, A., Yan, R., and Gao, R.X. 2011, 'Prognosis of Defect Propagation Based on Recurrent Neural Networks', *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 3, pp. 703-711.

Mallat, S.G. 1989, 'A Theory for Multiresolution Signal Decomposition: the Wavelet Representation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693.

Marichal, G.N., Artes, M., and Garcia-Prada, J.C. 2010, 'An Intelligent System for Faulty-bearing Detection Based on Vibration Spectra', *Journal of Vibration and Control*, vol. 17, no. 6, pp. 931-942.

Marichal, G.N., Artesa, M., Prada, G., and Casanova, O. 2011, 'Extraction of Rules for Faulty Bearing Classification by A Neuro-Fuzzy Approach', *Mechanical Systems and Signal Processing*, vol. 25, no. 6, pp. 2073-2082.

Mathew, J. and Alfredson, R.J. 1984, 'The Condition Monitoring of Rolling Element Bearings Using Vibration Analysis', *Journal of Vibration Acoustics Stress and Reliability in Design*, vol. 106, no. 3, pp. 447-453.

Mathworks Inc. 1997, *Wavelet Toolbox for Use With MATLAB - User's Guide*, the Mathworks, Inc., Natick, Massachusetts.

Mathworks Inc. 1998, *Fuzzy Logic Toolbox for Use With MATLAB - User's Guide*, the Mathworks Inc., Natick, Massachusetts.

McCormick, A.C., and Nandi, A.K. 1997, 'Real-time Classification of Rotating Shaft Loading Conditions Using Artificial Neural Networks', *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 748-757.

McFadden, P.D., and Smith, J.D. 1984, 'Vibration Monitoring of Rolling Element Bearings by the High Frequency Resonance Technique - A Review', *Tribology International*, vol. 17, no. 1, pp. 3-10.

McFadden, P.D., and Toozhy, M.M. 2000, 'Application of Synchronous Averaging to Vibration Monitoring of Rolling Element Bearings', *Mechanical Systems and Signal Processing*, vol. 14, no. 6, pp. 891-906.

Meesad, P., and Yen, G.G. 2000, 'Pattern Classification by A Neurofuzzy Network: Application to Vibration Monitoring', *ISA Transactions*, vol. 39, no. 3, pp. 293-308.

Meireles, M.R.G., Almeida, P.E.M., and Simoes, M.G. 2003, 'A Comprehensive Review for Industrial Applicability of Artificial Neural Networks', *IEEE Transactions on Industrial Electronics*, vol. 50, no. 3, pp. 585-601.

Mevel, L., Hermans, L., and Van der Auweraer, H. 2000, 'Application of A Subspace-based Fault Detection Method to Industrial Structures', *Mechanical Systems and Signal Processing*, vol. 13, no. 6, pp. 823-838.

Miguel, L.J. d., and Blazquez, L.F. 2005, 'Fuzzy Logic-based Decision-Making for Fault Diagnosis in A DC Motor', *Engineering Applications of Artificial Intelligence*, vol. 18, no. 4, pp. 423-450.

Mohammadi, R., Naderi, E., Khorasani, K., and Hashtrudi-Zad, S. 2011, 'Fault Diagnosis of Gas Turbine Engines by Using Dynamic Neural Networks',*Proceedings of the IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*.

Newland, D.E. 1994a, 'Wavelet Analysis of Vibration .1. Theory', *Transactions of The ASME - Journal of Vibration and Acoustics*, vol. 116, no. 4, pp. 409-416.

Newland, D.E. 1994b, 'Wavelet Analysis of Vibration .2. Wavelet Maps', *Transactions of The ASME - Journal of Vibration and Acoustics*, vol. 116, no. 4, pp. 417-425.

Nikolaou, N.G., and Antoniadis, I.A. 2002a, 'Rolling Element Bearing Fault Diagnosis Using Wavelet Packets', *NDT & E International*, vol. 35, no. 3, pp. 197-205.

Nikolaou, N.G., and Antoniadis, I.A. 2002b, 'Demodulation of Vibration Signals Generated by Defects in Rolling Element Bearing Using Complex Shifted Morlet Wavelet', *Mechanical Systems and Signal Processing*, vol. 16, no. 4, pp. 677-694.

Niu, X., Zhu, L., and Ding, H. 2005, 'New Statistical Moments for the Detection of Defects in Rolling Element Bearings', *International Journal of Advanced Manufacturing Technology*, vol. 26, no. 11-12, pp. 1268-1274.

Onel, I.Y., and Benbouzid, M.E.H. 2008, 'Induction Motors Bearing Failures Detection and Diagnosis Using a RBF ANN Park Pattern Based Method', *International Review of Electrical Engineering*, vol. 3, no. 1, pp. 159-165.

Pan, M.-C., and Sas, P. 1996, 'Transient Analysis on Machinery Condition Monitoring', *the 3rd International Conference on Signal Processing*, vol. 2, no. 1-2, pp. 1723-1726.

Patil, M.S., Mathew, J., and RajendraKumar, P.K. 2008, 'Bearing Signature Analysis as A Medium for Fault Detection: A Review', *Journal of Tribology, Transactions of ASME*, vol. 130, no. 1.

Patton, R.J., Chen, J., and Benkhedda, H. 2000, 'A Study on Neuro-fuzzy Systems for Fault Diagnosis', *International Journal of Systems Science*, vol. 31, no. 11, pp. 1441-1448.

Paya, B.A., and Esat, I.I. 1997, 'Artificial Neural Network Based Fault Diagnostics of Rotating Machinery Using Wavelet Transforms as A Processor', *Mechanical Systems and Signal Processing*, vol. 11, no. 5, pp. 751-765.

Peng, Y., Dong, M., and Zuo, M.J. 2010, 'Current Status of Machine Prognostics in Condition-based Maintenance: A Review', *The International Journal of Advanced Manufacturing Technology*, vol. 50, no. 1-4, pp. 297-313.

Peng, Z.K., and Chu, F.L. 2004, 'Application of the Wavelet Transform in Machine Condition Monitoring and Fault Diagnostics: A Review with Bibliography', *Mechanical Systems and Signal Processing*, vol. 18, no. 2, pp. 199-221.

Poyhonen, S., Jover, P., and Hyotyniemi, H. 2004, 'Signal Processing of Vibrations for Condition Monitoring of An Induction Motor', *ISCCSP: 2004 First*

International Symposium on Control, Communications and Signal Processing, pp. 499-502.

Prabhakar, S., Mohanty, A.R., and Sekhar, A.S. 2002, 'Application of Discrete Wavelet Transform for Detection of Ball Bearing Race Faults', *Tribology International*, vol. 35, no. 12, pp. 793-800.

Premrudeepreechacharn, S., Utthiyoung, T., Kruepengkul, K., and Puongkaew, W. 2002, 'Induction Motor Fault Detection and Diagnosis Using Supervised and Unsupervised Neural Networks', *IEEE International Conference on Industrial Technology*, vol. 1-2, pp. 93-96.

Prieto, M.D., Cirrincione, C., Espinosa, A.G., Ortega, J.A., and Henao, H. 2013, 'Bearing Fault Detection by A Novel Condition Monitoring Scheme Based on Statistical-Time Features and Neural Networks', *IEEE Transactions on Industrial Electronics*, vol. 60, no. 8, pp. 3398-3407.

Purushotham, V., Narayanan, S., and Prasad, S.A.N. 2005, 'Multi-fault Diagnosis of Rolling Bearing Elements Using Wavelet Analysis and Hidden Markov Model Based Fault Recognition', *NDT & E International*, vol. 38, no. 8, pp. 654-664.

Qiu, H., Lee, J., and Lin, J. 2006, 'Wavelet Filter-based Weak Signature Detection Method and Its Application on Roller Bearing Prognostics', *Journal of Sound and Vibration*, vol. 289, no. 4-5, pp. 1066-1090.

Rafiee, J., and Tse, P.W. 2009, 'Use of Autocorrelation of Wavelet Coefficients for Fault Diagnosis', *Mechanical Systems and Signal Processing*, vol. 23, no. 5, pp. 1554-1572.

Rafiee, J., Arvani, F., Harifi, A., and Sadeghi, M.H. 2007, 'Intelligent Condition Monitoring of A Gearbox Using Artificial Neural Network', *Mechanical Systems and Signal Processing*, vol. 21, no. 4, pp. 1746-1754.

Rafiee, J., Rafiee, M.A., and Tse, P.W. 2010, 'Application of Mother Wavelet Functions for Automatic Gear and Bearing Fault Diagnosis', *Expert Systems with Applications*, vol. 37, no. 6, pp. 4568-4579.

Rafiee, J., Rafiee, M.A., Prause, N., and Tse, P.W. 2009b, 'Application of Daubechies 44 in Machine Fault Diagnostics', *Proceedings of the 2nd International Conference on Computer, Control and Communication*, pp. 430-435.

Rai, V.K., and Mohanty, A.R., 2007, 'Bearing Fault Diagnosis Using FFT of Intrinsic Mode Functions in Hilbert–Huang Transform', *Mechanical Systems and Signal Processing*, vol. 21, no. 6, pp. 2607-2615.

Randall, R.B. 1987, *Frequency Analysis, Third Edition*, Bruel & Kjaer, Naerum, Denmark.

Randall, R.B. 2011, *Vibration-based Condition Monitoring: Industrial and Automotive Applications*, John Wiley & Sons, Ltd., West Sussex, United Kingdom.

Randall, R.B., and Antoni, J. 2011, 'Review: Rolling Element Bearing Diagnostics - A Tutorial', *Mechanical Systems and Signal Processing*, vol. 25, no. 2, pp. 485-520.

Randall, R.B., Antoni, J., and Chobsaard, C. 2001, 'The Relationship Between Spectral Correlation and Envelope Analysis in the Diagnostics of Bearing Faults and Other Cyclostationary Machine Signals', *Mechanical Systems and Signal Processing*, vol. 15, no. 5, pp. 945-962.

Rao, B.K.N., Pai, P.S., and Nagabhushana, T.N. 2012, 'Failure Diagnosis and Prognosis of Rolling Element Bearings Using Artificial Neural Networks: A Critical Overview', *the 25th International Congress on Condition Monitoring and Diagnostic Engineering (Comadem 2012)*, vol. 364.

Reda Taha, M.M., Noureldin, A., Lucero, J.L., and Baca, T.J. 2006, 'Wavelet Transform for Structural Health Monitoring: A Compendium of Uses and Features', *Structural Health Monitoring*, vol. 5, no. 3, pp. 267-295.

Reddy, M.J., and Mohanta, D.K. 2007, 'A Wavelet-neuro-fuzzy Combined Approach for Digital Relaying of Transmission Line Faults', *Electric Power Components and Systems*, vol. 35, no. 12, pp. 1385-1407.

Rong, J., Zhang, X.-W., Chen, X.-Y., Li, H., Liu, J., and Song, X.-F. 2009, 'Hydraulic Turbines Vibration Fault Diagnosis by RBF Neural Network Based On Particle Swarm Optimization', *Asia-Pacific Power and Energy Engineering Conference 2009*, vol. 1-7, pp. 961-964.

Rubini, R., and Meneghetti, U. 2001, 'Application of the Envelope and Wavelet Transform Analyses for the Diagnosis of Incipient Faults in Ball Bearings', *Mechanical Systems and Signal Processing*, vol. 15, no. 2, pp. 287-302.

Rumelhart, D.E., and McClelland, J.L. 1986, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press, Cambridge, Massachusetts.

Samanta, B., and Al-Balushi, K.R. 2003, 'Artificial Neural Network Based Fault Diagnostics of Rolling Element Bearings Using Time-Domain Feature', *Mechanical Systems and Signal Processing*, vol. 17, no. 2, pp. 317-328.

Saravanan, N., and Ramachandran, K.I. 2010, 'Incipient Gear Box Fault Diagnosis Using Discrete Wavelet Transform (DWT) for Feature Extraction and Classification Using Artificial Neural Network (ANN)', *Expert Systems with Applications*, vol. 37, no. 6, pp. 4168-4181.

Saravanan, N., Cholairajan, S., and Ramachandran, K.I. 2009, 'Vibration-based Fault Diagnosis of Spur Bevel Gear Box Using Fuzzy Technique', *Expert Systems with Applications*, vol. 36, no.2, pp. 3119-3135.

Satish, B., and Sarma, N.D.R. 2005, 'A Fuzzy BP Approach for Diagnosis and Prognosis of Bearing Faults in Induction Motors', *IEEE Power Engineering Society General Meeting*, pp. 2291-2294

Schwartz, D.G., Klir, G.J., Lewis III, H.W., and Ezawa. Y. 1994, 'Applications of Fuzzy Sets and Approximate Reasoning', *Proceedings of IEEE*, vol. 82, no. 4, pp. 482-498.

Seker, S., and Ayaz, E. 2003, 'FeatureExtraction Related to Bearing Damage in Electric Motors by Wavelet Analysis', *Journal of the Franklin Institute – Engineering and Applied Mathematics*, vol. 340, no. 2, pp. 125-134.

Selaimia, Y., Moussaoui, A., and Abbassi, H.A. 2006, 'Multi Neural Networks Based Approach for Fault Detection and Diagnosis of A DC-Motor', *Neural Network World*, vol. 16, no. 5, pp. 369-379.

Senguler, T., Karatoprak, E., and Seker, S. 2010, 'A New MLP Approach for the Detection of the Incipient Bearing Damage', *Advances In Electrical and Computer Engineering*, vol. 10, no. 3, pp. 34-39.

Serviere, C., and Fabry, P. 2004, 'Blind Source Separation of Noisy Harmonic Signals for Rotating Machine Diagnosis', *Journal of Sound and Vibration*, vol. 272, no. 1-2, pp. 317-339.

Seryasat, O.R., Shoorehdeli, M.A., Honarvar, F., and Rahmani, A. 2010, 'Multi-fault Diagnosis of Ball Bearing Using FFT, Wavelet Energy Entropy Mean and

Root Mean Square (RMS)', *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 4295-4299.

Shao, Y., and Nezu, K. 1999, 'Detection of Self-aligning Roller Bearing Fault by Asynchronous Adaptive Noise Cancelling Technology', *JSME International Series C: Dynamics, Control, Robotics, Design and Manufacturing*, vol. 42, no. 1, pp. 33-43.

Shi, D.F., Wang, W.J., and Qu, L.S. 2004, 'Defect Detection for Bearings Using Envelope Spectra of Wavelet Transform', *Journal of Vibration and Acoustics – Transactions of the ASME*, vol. 126, no. 4, pp. 567-573.

Shibata, K., Takahashi, A., and Shirai, T. 2000, 'Fault Diagnostics of Rotating Machinery through Visualisation of Sound Signals', *Mechanical Systems and Signal Processing*, vol. 14, no. 2, pp. 229-241.

Sreejith, B., Verma, A.K., and Srividya A. 2008, 'Fault Diagnosis of Rolling Element Bearing Using Time-domain Features and Neural Networks', *IEEE Region 10 Colloquium and the Third International Conference on Industrial and Information Systems*, vol. 1-2, pp. 619-624.

Stepanic, P., Latinovic, I.V., and Djurovic, Z. 2009, 'A New Approach to Detection of Defects in Rolling Element Bearings Based on Statistical Pattern Recognition', *International Journal of Advanced Manufacturing Technology*, vol. 45, no. 1-2, pp. 91-100.

Sugeno, M., and Kang, G.T. 1988, 'Structure Identification of Fuzzy Model', *Fuzzy Sets and Systems*, vo. 28, no. 1, pp. 15-33.

Sugumaran, V., and K.I. Ramachandran. 2011, 'Fault Diagnosis of Roller Bearing Using Fuzzy Classifier and Histogram Features with Focus on Automatic Rule Learning', *Expert Systems with Applications*, vol. 38 no. 5, pp. 4901-4907.

Sun, Q., and Tang, Y. 2002, 'Singularity Analysis Using Continuous Wavelet Transform for Bearing Fault Diagnosis', *Mechanical Systems and Signal Processing*, vol. 16, no. 6, pp. 1025-1041.

Sun, Q., Chen, P., Zhang, D., and Xi, F. 2004, 'Pattern Recognition for Automatic Machinery Fault Diagnosis', *Journal of Vibration and Acoustics*, vol. 126, no. 2, pp. 307-316.

Takagi, T., and Sugeno, M. 1985, 'Fuzzy Identification of Systems and Its Applications to Modelling and Control', *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116-132.

Tandon, N., and Choudhury, A. 1999, 'A Review of Vibration and Acoustic Measurement Methods for the Detection of Defects in Rolling Element Bearings', *Tribology International*, vol. 32, no. 8, pp. 469-480.

Theodoridis, S., and Koutroumbas, K. 2003, *Pattern Recognition, Second Edition*, Academic Press, San Diego, California.

Tsoukalas, L.H., and Uhrig, R.E. 1997, *Fuzzy and Neural Approaches in Engineering*, John Wiley & Sons, New York.

Tu, P.Y.L., Yam, R., Tse, P., and Sun, A.O.W. 2001, 'An Integrated Maintenance Management System for an Advanced Manufacturing Company', *International Journal of Advances in Manufacturing Technology*, vol. 17, no. 9, pp. 692-703.

Vasquez, D., Aradilla, G., Gruhn, R., and Minker, W. 2009, 'On Speeding Phoneme Recognition in A Hierarchical MLP Structure', *IEEE Workshop on Automatic Speech Recognition & Understanding (ARSU)*, pp. 345-348.

Vicente, S.A.d., Fujimoto, R.Y., and Padovese, L.R. 2001, 'Rolling Bearing Fault Diagnostic System Using Fuzzy Logic', *the 10th International Conference on Fuzzy System*, vol. 1-3, pp. 816-819.

Wang, C.-C., Kang, Y., Shen, P.-C., Chang, Y.-P., and Chung, Y.-L. 2010, 'Applications of Fault Diagnosis in Rotating Machinery by Using Time Series Analysis with Neural Network', *Expert Systems with Applications*, vol. 37, no. 2, pp. 1696-1702.

Wang, H., and Chen, P. 2011,'Intelligent Diagnosis Method for Rolling Element Bearing Faults Using Possibility Theory and Neural Network', *Computers & Industrial Engineering*, vol. 60, no. 4, pp. 511-518.

Wang, K., and Lei, B. 2001, 'Using B-spline Neural Network to Extract Fuzzy Rules for A Centrifugal Pump Monitoring', *Journal of Intelligent Manufacturing*, vol. 12, no. 1, pp. 5-11.

Wang, W. 2001, 'Early Detection of Gear Tooth Cracking Using the Resonance Demodulation Technique', *Mechanical Systems and Signal Processing*, vol. 15, no. 5, pp. 887-903.

Wang, W., Ismail, F., and Golnaraghi, F. 2004a, 'A Neuro-Fuzzy Approach to Gear System Monitoring', *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 5, pp. 710-723.

Wang, W.J., and McFadden, P.D. 1993, 'Early Detection of Gear Failure by Vibration Analysis – I. Calculation of the Time-Frequency Distribution', *Mechanical Systems and Signal Processing*, vol. 7, no. 3, pp. 193-203.

Wang, W.J., and McFadden, P.D. 1996, 'Application of Wavelets to Gearbox Vibration Signals for Fault Detection', *Journal of Sound and Vibration*, vol. 192, no. 5, pp. 816-828.

Wang, W.J., Chen, J., Wu, X.K., and Wu, X.T. 2001, 'The Application of Some Non-Linear Methods in Rotating Machinery Fault Diagnosis', *Mechanical Systems and Signal Processing*, vol. 15, no. 4, pp. 697-705.

Wang, W.Q., Golnaraghi, M.F., and Ismail, F. 2004b, 'Prognosis of Machine Health Condition Using Neuro-fuzzy Systems', *Mechanical Systems and Signal Processing*, vol. 18, no. 4, pp. 813-831.

Wang, X., Zi, Y., and He, Z. 2011, 'Multiwavelet Denoising with Improved Neighboring Coefficients for Application on Rolling Bearing Fault Diagnosis', *Mechanical Systems and Signal Processing*, vol. 25, no.1, pp. 285-304.

White, M.F. 1984, 'Simulation and Analysis of Machinery Fault Signals', *Journal of Sound and Vibration*, vol. 93, no. 1, pp. 95-116.

Widrow, B., and Lehr, M.A. 1960, 'Adaptive Switching Circuits', *IRE Western Electric Show and Convention Record*, part 4, pp. 96-104.

Widrow, B., and Lehr, M.A. 1990, '30 Years of Adaptive Neural Networks: Perceptron, Madaline, & Backpropagation', *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415-1442.

Williams, T., Ribadeneira, X., Billington, S., and Kurfess, T. 2001, 'Rolling Bearing Diagnostics in Run-to-failure Lifetime Testing', *Mechanical Systems and Signal Processing*, vol. 15, no. 5, pp. 979-993.

Wu, J.-D. and Kuo, J.-M. 2009, 'An Automotive Generator Fault Diagnosis System Using Discrete Wavelet Transform and Artificial Neural Network', *Expert Systems with Applications*, vol. 36, no. 6, pp. 9776-9783.

Wu, J.-D., and C.-H. Liu. 2008, 'Investigation of Engine Fault Diagnosis Using Discrete Wavelet Transform and Neural Network', *Expert Systems with Applications*, vol. 35, no. 3, pp. 1200-1213.

Wu, J.-D., and Chan, J.-J. 2009, 'Faulted Gear Identification of a Rotating Machinery Based on Wavelet Transform and Artificial Neural Network', *Expert Systems with Applications*, vol. 36, no. 5, pp. 8862-8875.

Wu, J.-D., and Hsu, C.-C. 2009, 'Fault Gear Identification Using Vibration Signal with Discrete Wavelet Transform Technique and Fuzzy–logic Inference', *Expert Systems with Applications*, vol. 36, no. 2, pp. 3785-3794.

Wu, J.-D., and Liu, C.-H. 2009, 'An Expert System for Fault Diagnosis in Internal Combustion Engines Using Wavelet Packet Transform and Neural Network', *Expert Systems with Applications*, vol. 36, no. 3, pp. 4278-4286.

Wu, J.-D., Hsu, C.-C., and Wu, G.-Z. 2009, 'Fault Gear Identification and Classification Using Discrete Wavelet Transform and Adaptive Neuro-fuzzy Inference', *Expert Systems with Applications*, vol. 36, no. 3, pp. 6244-6255.

Wu, J.-F., Hu, N.-S., Hu, S., and Zhao, Y. 2002, 'Application of SOM Neural Network in Fault Diagnosis of the Steam Turbine Regenerative System', *International Conference on Machine Learning and Cybernetics*, vol. 1-4, pp. 184-187.

Xi, F., Sun, Q., and Krishnappa, G. 2000, 'Bearing Diagnostics Based on Pattern Recognition of Statistical Parameters', *Journal of Vibration and Control*, vol. 6, no. 3, pp. 375-392.

Xu, Z., Xuan, J., Shi, T., Wu, B., and Hu, Y. 2009, 'A Novel Fault Diagnosis Method of Bearing Based on Improved Fuzzy ARTMAP and Modified Distance Discriminant Technique', *Expert Systems with Applications*, vol. 36, no. 9, pp. 11801-11807.

Yam, R.C.M., Tse, P.W., Li, L., and Tu, P. 2001, 'Intelligent Predictive Decision Support System for Condition-based Maintenance', *Advanced Manufacturing Technology*, vol. 17, no. 5, pp. 383-391.

Yan, R., and Gao, R.X. 2009, 'Energy-based Feature Extraction for Defect Diagnosis in Rotary Machines', *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 9, pp. 3130-3139.

Yang, B.S., Han, T., and An, J.L. 2004, 'ART-Kohonen Neural Network for Fault Diagnosis of Rotating Machinery', *Mechanical Systems and Signal Processing*, vol. 18, no. 3, pp. 645-657.

Yang, H., Mathew, J., and Ma, L. 2002, 'Intelligent Diagnosis of Rotating Machinery Faults - A Review', *the 3rd Asia-Pacific Conference on Systems Integrity and Maintenance, Cairns, Australia*.

Yang, H., Mathew, J., and Ma, L. 2003, 'Vibration Feature Extraction Techniques for Fault Diagnosis of Rotating Machinery: A Literature Survey', *Asia-Pacific Vibration Conference, Gold Coast, Australia*.

Yang, S., Li, W., and Wang, C. 2008, 'The Intelligent Fault Diagnosis of Wind Turbine Gearbox Based on Artificial Neural Network', *Proceedings of International Conference on Condition Monitoring and Diagnosis*, pp. 1327-1330.

Yang, W.-X., and Ren, X.-M. 2004, 'Detecting Impulses in Mechanical Signals by Wavelets', *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 8, pp. 1156-1162.

Yen, G.G., and Lin, K.-C. 2000, 'Wavelet Packet Feature Extraction for Vibration Monitoring', *IEEE Transactions on Industrial Electronics*, vol. 47, no. 3, pp. 650-667.

Zadeh, L.A. 1965, 'Fuzzy Sets', *Information and Control*, vol. 8, no. 1, pp. 338-353.

Zadeh, L.A. 1973, 'Outline of A New Approach to the Analysis of Complex Systems and Decision Processes', *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 1, pp. 28-44.

Zarei, J., Poshtan, J., and Poshtan, M. 2008, 'Bearing Fault Detection in Induction Motor Using Pattern Recognition Techniques', *IEEE 2nd International Power and Energy Conference*, vol. 1-3, pp. 749-753.

Zhan, Y., Makis, V., and Jardine, A.K.S. 2003, 'Adaptive Model for Vibration Monitoring of Rotating Machinery Subject to Random Deterioration', *Journal of Quality in Maintenance Engineering*, vol. 9, no. 4, pp. 351-375.

Zhang, J., and Morris, J. 1996, 'Process Modelling and Fault Diagnosis Using Fuzzy Neural Networks', *Fuzzy Sets and Systems*, vol. 79, no. 1, pp. 127-140.

Zhang, L., Xiong, G., Liu, H., Zou, H., and Guo, W. 2010, 'Bearing Fault Diagnosis Using Multi-scale Entropy and Adaptive Neuro-fuzzy Inference', *Expert Systems with Applications*, vol. 37, no. 8, pp. 6077-6085.

Zhang, S., and Ganesan, R. 1997, 'Multivariable Trend Analysis Using Neural Networks for Intelligent Diagnostics of Rotating Machinery', *Journal of Engineering for Gas Turbines and Power - Transactions of the ASME*, vol. 119, no. 2, pp. 378-384.

Zhao, F., Cheng, J., Guo, L., and Li, X. 2009, 'Neuro-fuzzy Based Condition Prediction of Bearing Health', *Journal of Vibration and Control*, vol. 15, no. 7, pp. 1079-1091.

Zhong, F., Shi, T., and He, T. 2005, 'Fault Diagnosis of Motor Bearing Using Self-Organizing Maps', *Proceedings of the Eighth International Conference on Electrical Machines and Systems*, vol. 1-3 pp. 2411-2414.

Zhu, K., Wong, Y.S., and Hong, G.S. 2009, 'Wavelet Analysis of Sensor Signals for Tool Condition Monitoring: A Review and Some New Results', *International Journal of Machine Tools and Manufacture*, vol. 49, no. 7-8, pp. 537-553.

Appendix 1 – Daubechies Wavelet Transform Implementation

In relation to the wavelet topic which is presented in Chapter 2 Section 2.4, a detail example implementation of Daubechies 4 transform is presented here. The mathematical formulation is adapted from Jensen and la Cour-Harbo (2001) which is based on the following Equations:

$$s^{(1)}[n] = S[2n] + \sqrt{3}S[2n+1], \tag{A1.1}$$

$$d^{(1)}[]n = S[2n+1] - 0.25\sqrt{3}s^{(1)}[n] - 0.25\left(\sqrt{3}-2\right)s^{(1)}[n-1], \tag{A1.2}$$

$$s^{(2)}[n] = s^{(1)}[n] - d^{(1)}[n+1], \tag{A1.3}$$

$$s[n] = \frac{\sqrt{3}-1}{\sqrt{2}}s^{(2)}[n], \tag{A1.4}$$

$$d[n] = \frac{\sqrt{3}+1}{\sqrt{2}}d^{(2)}[n], \tag{A1.5}$$

Implementation of Daubechies 4 using Equation A1.1 is carried out to calculate **s(1)[n]** for all values of **n**. In this case, for a signal of length **N**, the calculation can be carried out using iteration (loop) which is written in Matlab syntax as follows:

```
For n=1:N/2
      s1((n) = S(2*n-1) + sqrt(3)*S(*n);
end
```

Equation 1 can be interpreted as a vector equation, where **n** starts from **1** to **N/2**. It is illustrated as follows:

$$\begin{bmatrix} s^{(1)}[1] \\ s^{(1)}[2] \\ \mathrm{M} \\ s^{(1)}[N/2] \end{bmatrix} = \begin{bmatrix} S[1] \\ S[3] \\ \mathrm{M} \\ S[N-1] \end{bmatrix} = \sqrt{3}\begin{bmatrix} S[2] \\ S[4] \\ \mathrm{M} \\ S[N] \end{bmatrix}$$

The indices are related to indexing in Matlab and Matlab syntax of the vector equation is presented as follows:

```
s1 = S(1:2:N-1) + sqrt(3)*S(2:2:N);
```

In this implementation, periodization method is used since it provides unitary transformation. By using periodised signal, undefined elements are taken from the other end of the signal. Hence, it is defined that $s^{(1)}[-1] \equiv s^{(1)}[N/2]$, then Equation A1.2 is interpreted as follows:

$$\begin{bmatrix} d^{(1)}[1] \\ d^{(1)}[2] \\ \vdots \\ d^{(1)}[N/2] \end{bmatrix} = \begin{bmatrix} S[2] \\ S[4] \\ \vdots \\ S[N] \end{bmatrix} - \frac{\sqrt{3}}{4} \begin{bmatrix} s^{(1)}[1] \\ s^{(1)}[2] \\ \vdots \\ s^{(1)}[N/2] \end{bmatrix} - \frac{\sqrt{3}-2}{4} \begin{bmatrix} s^{(1)}[N/2] \\ s^{(1)}[1] \\ \vdots \\ s^{(1)}[N/2-1] \end{bmatrix}$$

Translation of Equation A1.2 into a Matlab syntax becomes

```
d1=S(2:2:N) – sqrt(3)/4*s1 – (sqrt(3)-2)/4*[s1[N/2] s1(1:N/2-1];
```

The change of a vector from **s1** to **[s1(N/2) s1(1:N/2-1)]** is referred as a *cyclic permutation* of the vector **s1**.

The complete Matlab codes for implementing Daubechies 4 wavelet transform is presented as below.

```
s1 = S(1:2:N-1) + sqrt(3)*S(2:2:N);
d1 = S(2:2:N) – sqrt(3)/4*s1 – (sqrt(3)-2)/4*[s1(N/2) s1(1:N/2-1)];
s2 = s1 – [d1(2:N/2) d1(1)];
s = (sqrt(3)-1)/sqrt(2) * s2;
d = (sqrt(3)+1/sqrt(2) * d1;
```

A practical implementation of the complete codes of Daubechies 4 transform is presented using a simple sinusoidal signal which was generated in Matlab using the following codes:

```
t = 0.0009765625:0.0009765625:1; % 1024 samples
signal = 20*(t.^2).*((1-t).^4).*cos(12*pi.*t);
```

The original sinusoidal signal is shown in Figure A1.1(a). The numbers of data points of the signal is 1024. The Daubechies 4 (db4) wavelet transform result is shown in Figure A1.1(b).

The original signal is transformed using the complete code that represents 1 level of signal decomposition using Daubechies 4 algorithm.

The transform result is presented as a combined plot of Approximation (A) and Details (D) coefficients. In Figure A1.1(b). The first half of the plot (the first 512 data points) is the Approximation (A) results of the transformation, and the rest half of the plot (512 data points) is the Details (D) results of the transformation.

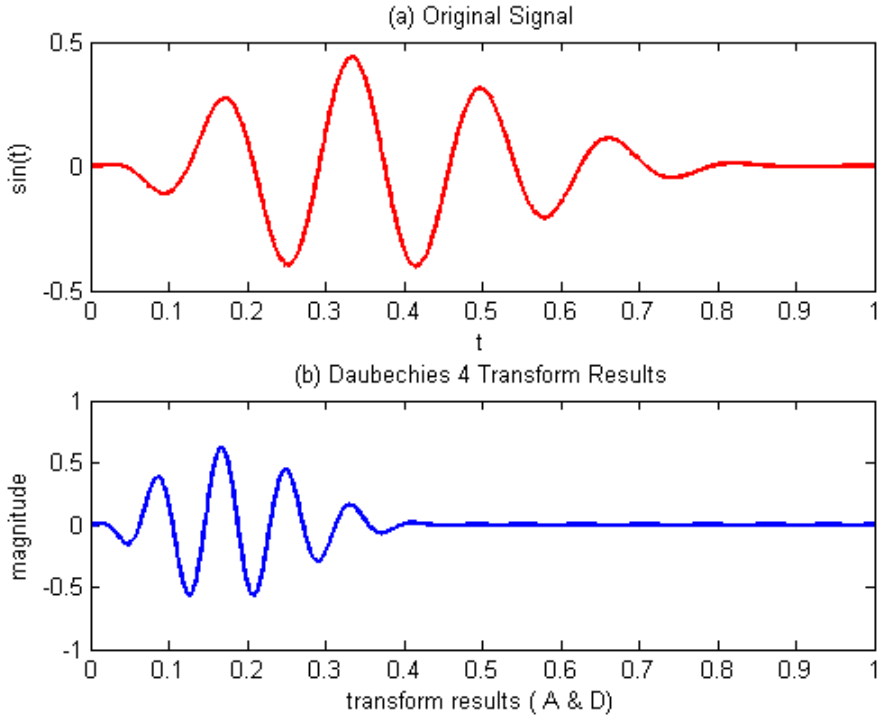

Figure A1.1. Original signal and Daubechies 4 transform results

The separated plots which show Approximation (A) and Details (D) results are presented in Figure A1.2 and Figure A1.3. The first half 512 data points that

represent Approximation (A) results are plotted in Figure A1.2. The rest of 512 data points which represent Details (D) results are plotted in Figure A1.3.

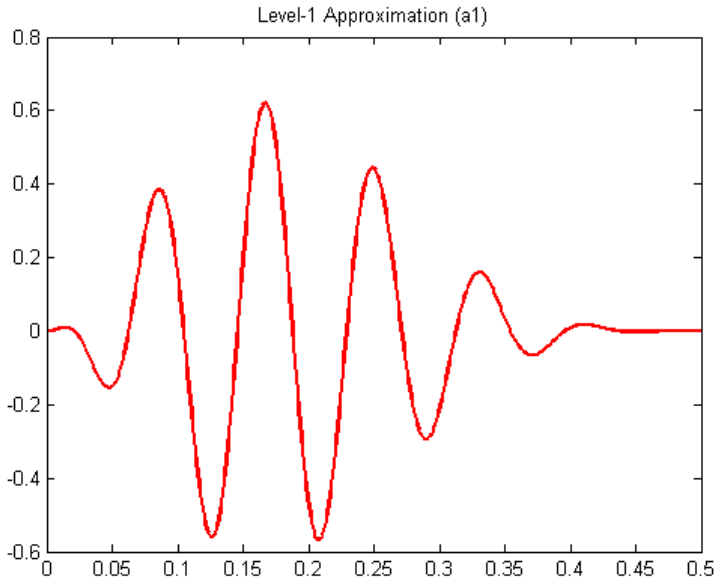

Figure A1.2. Approximation (A) part of the transform result

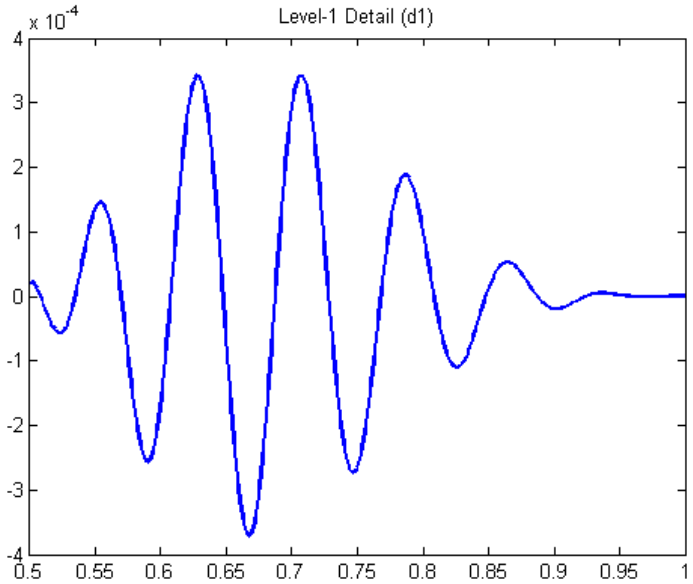

Figure A1.3. Details (D) part of the transform result

It shown that 1 level wavelet transformation of the signal produce two parts of results which are labelled Approximation (A) part and Detail (D) part in accordance with the wavelet transform scheme presented in Chapter 2 Section 2.4.

The reconstruction of the original signal (inverse transform) by using the transformation results of A and D parts can be carried out using the following codes,

```
d1 = d * ((sqrt(3)-1)/sqrt(2));
s2 = s * ((sqrt(3)+1)/sqrt(2));
s1 = s2 + circshift(d1,-1);
S(2:2:N) = d1 + sqrt(3)/4*s1 + (sqrt(3)-2)/4*circshift(s1,1);
S(1:2:N-1) = s1 - sqrt(3)*S(2:2:N);
```

Figure 4 shows the comparison between the original signal and the un-scaled reconstructed signal. There is amplitude difference between these two signals. The correction to equalize the amplitude of both signals is carried out using the following codes:

```
rec_signal = S; % reconstructed signal
scl_factor = sum(rec_signal) / sum(signal); % get scaling factor
rec_signal = S./scl_factor; % scaling
```

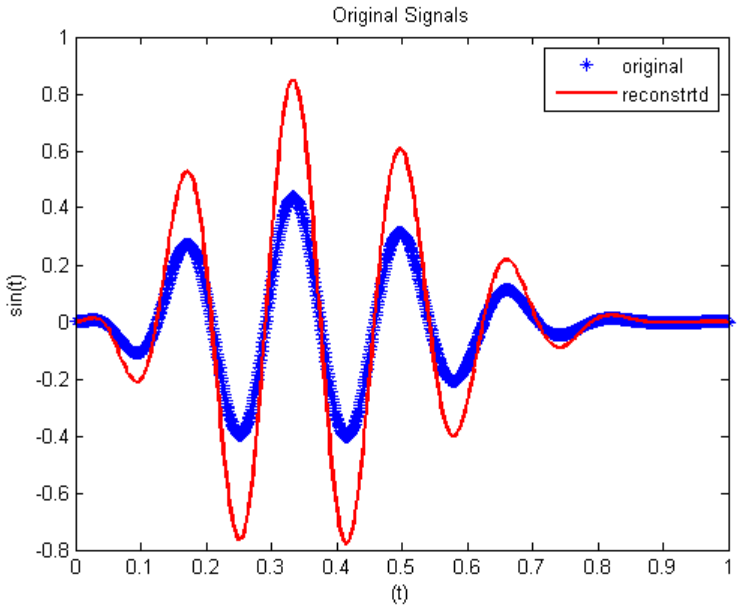

Figure A1.4. Original signal and un-scaled reconstructed signal

The corrected reconstructed signal that has been scaled to match the amplitude of the original signal is presented in Figure A1.5.

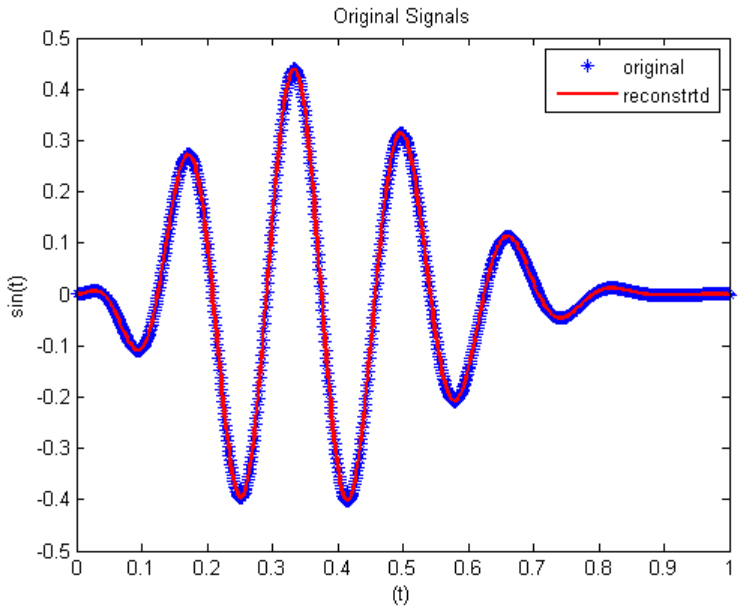

Figure A1.5. Original signal and scaled reconstructed signal

Appendix 2 – Fuzzy Sets Theory *

A.2.1. Basic Definitions and Terminology

Let assume that *X* be a space of objects and *x* be a regular element of *X* then a conventional set *A* in which $A \subset X$, is defined as a collection of elements $x \in X$, with the condition that each *x* can either be included or not be included to the set *A* by stating a characteristic function for each element *x* in *X*. In this case, the conventional set A can be represented by a set of ordered pairs (*x*, 0) or (*x*, 1), which indicates $x \notin A$ or $x \in A$, respectively.

In contrast to the conventional set, a fuzzy set expresses the degree to which an element is included to a set. Hence the characteristic function of a fuzzy set is allowed to have values between interval 0 and 1. In this context, the characteristic function denotes the degree of membership of an element in a given set.

A.2.2. Fuzzy Sets and the Membership Functions

If *X* is a collection of objects denoted by *x*, then a fuzzy set *A* in *X* is defined as a set of ordered pairs:

$$A = \left\{ \left(x, \mu_A(x) \right) \middle| x \in X \right\} \qquad \text{(A2.1)}$$

where $\mu_A(x)$ is called the membership function (MF for short) for the fuzzy set *A* and *X* is referred to the universe of discourse. The MF maps each element of *X* to a membership grade (or membership value) between 0 and 1. It is clear that the definition of a fuzzy set is a simple definition or generalization of a conventional set in which the characteristic function is allowed/permitted to have any values between 0 and 1. If the value of membership function $\mu_A(x)$ is reduced to either 0 or 1, then the fuzzy set A becomes conventional set with membership function 0 and 1.

---

* Jang *et al*. 1997, Neuro-Fuzzy and Soft Computing, Prentice-Hall, New York

The universe of discourse may consist of discrete objects or continuous space. It is shown in the following example.

Example A2.1. Fuzzy sets with a discrete universe

Let X = {10, 15, 20, 25, 30, 35, 40} be the set of comfortable room temperature in C for human. Then the fuzzy set A for "comfortable room temperature" may be explained as follows:

A = {(10,0.1), (15,0.3), (20,0.7), (25,1), (30,0.7), (35,0.3), (40,0.1)}.

The MF for fuzzy set A is shown is shown in Figure A2.1(a). The membership grades of this fuzzy set are clearly based on subjective measures.

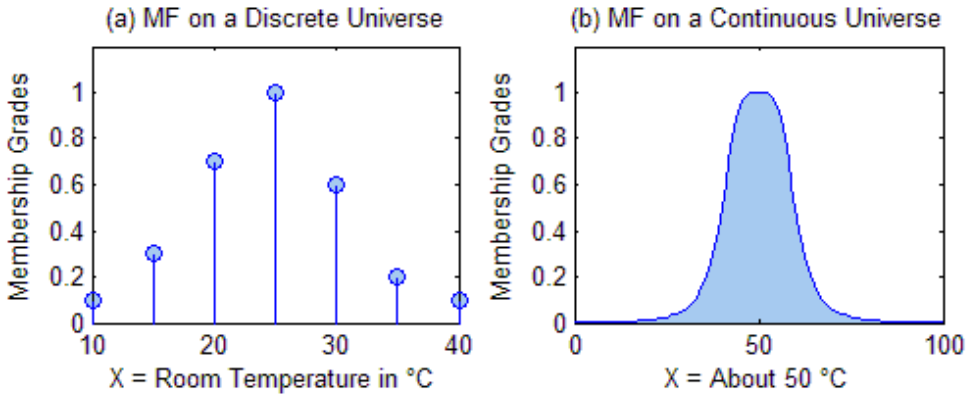

Figure A2.1. (a) A = "comfortable room temperature"; (b) B = "about 50°C".

Example 2. Fuzzy sets with a continuous universe

Let assume that $X = R$ be the set of temperature range from 0 °C to 100 °C. Then the fuzzy set B for "about 50 °C" is expressed as:

$$B = \left\{ \left(x, \mu_B(x) \right) \middle| x \in X \right\} \qquad \text{(A2.2)}$$

where

$$\mu_B(x) = \frac{1}{1 + \left(\frac{x - 50}{10} \right)^4}$$

It is illustrated in Figure A2.1(b).

For simplicity of notation, fuzzy set A can be denoted as follows:

$$A = \sum_{x_i \in X} \mu(x_i)/x_i, \text{ if X is a collection of discrete objects} \quad \text{(A2.3a)}$$

$$A = \int_X \mu_A(x)/x, \text{ if X is a continuous space in } \mathbf{R} \quad \text{(A2.3b)}$$

Fuzzy sets in Examples A2.1 and A2.2 can be rewritten based on the standard notation from Equation A2.3a and A2.3b, respectively as

A = 0.1/10 + 0.3/15 + 0.7/20 + 1.0/25 + 0.7/30 + 0.3/35 + 0.1/40,

and

$$B = \int_R \frac{1}{1+\left(\frac{x-50}{10}\right)^4} \Big/ x$$

The summation and integration sign in the above equations stand for the union of $(x, \mu_A(x))$ pairs. The signs do not indicate summation or integration. Similarly, "/" is only used as a marker and does not imply division.

A2.3. Fuzzy Sets Partition

Fuzzy sets are defined by partitioning a continuous space or real line **R** universe of discourse *X*. The partition of *X* into several fuzzy sets which MFs cover *X* in a more or less uniform manner.

The fuzzy set in the partitions usually use names that are taken from the adjective words of human linguistics such as "low", "medium", or "high." The adjective words that are used to name fuzzy sets in *X* partition are called linguistic values or linguistic labels.

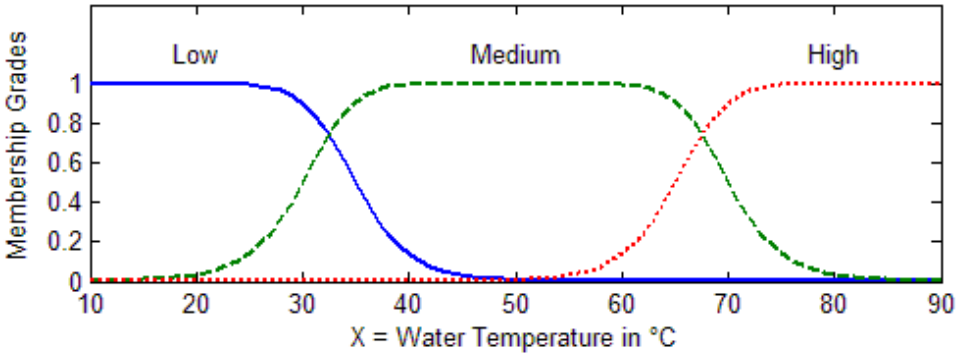

Figure A2.2. Typical MFs of linguistic variables "low", "medium" and "high".

Figure A2.2 shows the universe of discourse of water temperature in °C. There are three fuzzy sets, *low*, *medium* and *high*. These fuzzy sets are characterized by MFs $\mu_{low}(x), \mu_{medium}(x), \text{and } \mu_{high}(x)$. As a real variable can assume various values, similarly a linguistic variable "Temperature" can assume different linguistic values, such as "low", "medium" and "high". If the values of "temperature" is in the fuzzy set "high", then the expression "temperature is high" is applied, and so forth for other values.

A2.4. Properties of Fuzzy Sets

Core of fuzzy set. A core are elements which have membership degree = 1 as shown Figure A2.3.

$$core(\mathrm{A}) = \left\{x \middle| x \in X, \mu_A(x) = 0\right\}$$

Support of fuzzy set. A support of a subset of the universe X is collection of elements of which each has a membership degree bigger than zero (Figure A2.3).

$$support\ (\mathrm{A}) = \left\{x \middle| x \in X, \mu_A(x) > 0\right\}$$

For example, the support of the fuzzy set *A* in Figure A2.3 is the interval (3, 7).

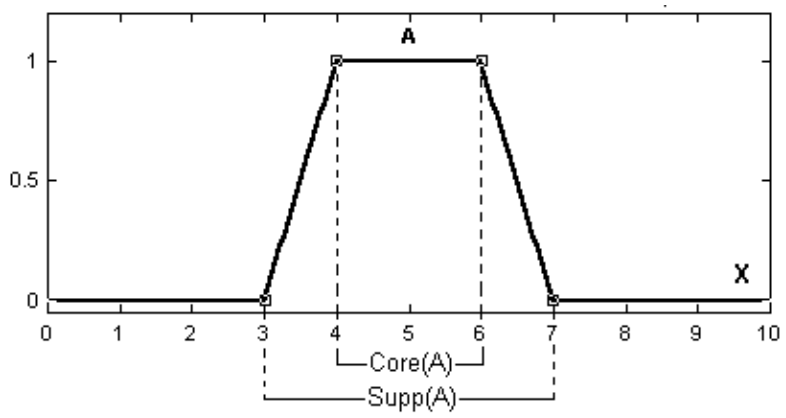

Figure A2.3. Support and Core of Fuzzy Set A

A2.5. Operations on Fuzzy Sets

Operations on Fuzzy sets are based on operations of conventional Boolean logic with several extensions to their fuzzy equivalents. The set operations are AND (intersection), OR (union) and NOT (complement). Basic classical set operations of union, intersection, and complements are shown in Table A2.1.

Table A2.1. Logic Operations of Classical Set

A	B	A & B Conj	A U B Disj.	Not Comp.
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Definition A2.1. Union (disjunction)

The union of two fuzzy sets A and B is a fuzzy set C, written as C = A $\cup$ B or C = A OR B, whose MF is related to those of A and B by

$$\mu_{(c)} = \max\left(\mu_A(x), \mu_B(x)\right) = \mu_A(x) \vee \mu_B(x)$$

Definition A2.2. Intersection (conjunction)

The intersection of two fuzzy sets A and B is a fuzzy set C, written as $C = A \cap B$ or C = A AND B, whose MF is related to those of A and B by

$$\mu_{(c)} = \min\left(\mu_A(x), \mu_B(x)\right) = \mu_A(x) \wedge \mu_B(x)$$

Definition A2.3. Complement (negation)

The complement of fuzzy set A, denoted by $\overline{A}$ or ($\neg A$, NOT A) and it is defined as $\mu_{\overline{A}} = 1 - \mu_A(x)$

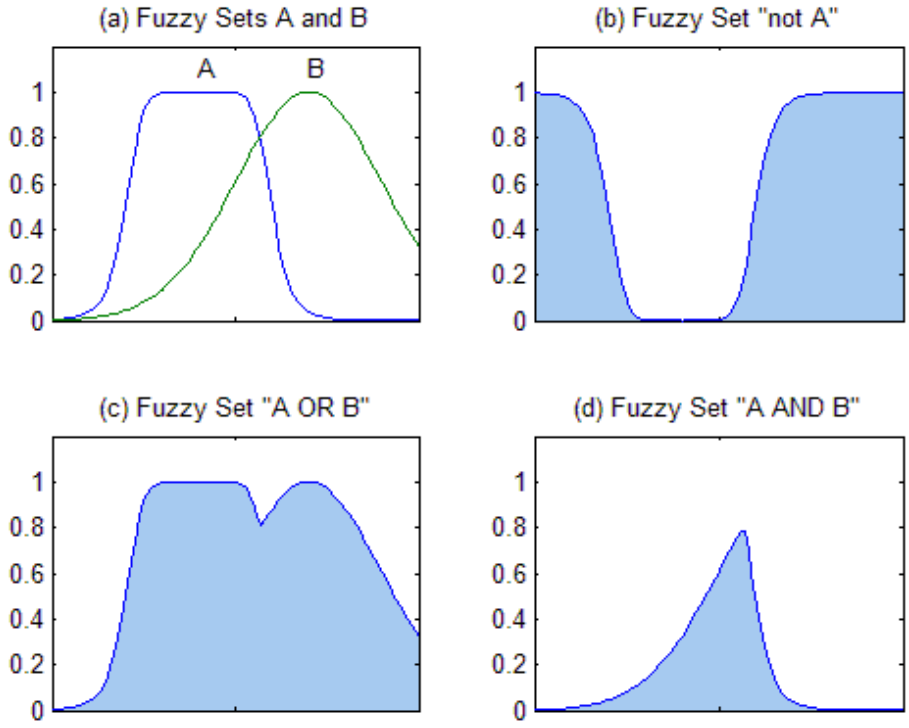

Figure A2.4.*Operations on fuzzy sets*: a) *two fuzzy sets*A and B; (b) $\overline{A}$; (c) $A \cup B$; (d) $A \cap B$

Figure A2.4 shows these three basic operations. In the detail, Figure A2.4(a) illustrates two fuzzy sets A and B; Figure A2.4(b) is the complement of A; Figure A2.4(c) is the union of A and B; Figure A2.4(d) is the intersection of A and B.

A2.6. MF Formulation and Parameterization

This section explains several MFs that are commonly used to characterize fuzzy sets in its universe of discourse. It is convenient and concise to express a MF as a mathematical formula as in Example A2.2.

Definition A2.4.*Triangular MFs*

A **triangular MF** is specified by three parameters {a, b, c} as follows:

$$\text{Triangle } (x;\, a, b, c) = \begin{cases} 0, x \le a \\ \dfrac{x-a}{b-a}, a \le x \le b \\ \dfrac{c-x}{c-b}, b \le x \le c \\ 0, c \le x \end{cases}$$

The parameters $\{a, b, c\}$ with $(a < b < c)$ determine the x coordinates of the three corners of the underlying triangular MF. Figure A2.5(a) shows a triangular MF define by a triangle (*x;* 20, 50, 80).

Definition A2.5.*Trapezoidal MFs*

A **trapezoidal MF** is specified by four parameters $\{a, b, c, d\}$ as follows:

$$\text{Trapezoid } (x;\, a, b, c, d) = \begin{cases} 0, x \le a \\ \dfrac{x-a}{b-a}, a \le x \le b \\ 1, b \le x \le c. \\ \dfrac{d-x}{d-c}, c \le x \le d \\ 0, d \le x \end{cases}$$

The parameters $\{x; a, b, c, d\}$(with $a < b < c < d$) determine the x coordinates of the four corners of the underlying trapezoidal MF.

Figure A2.5(b) shows a trapezoidal MF defined by trapezoid (*x*; 10, 20, 60, 95). Note that a trapezoidal MF with parameter $\{a, b, c, d\}$ reduces to a triangular MF when b is equal to c.

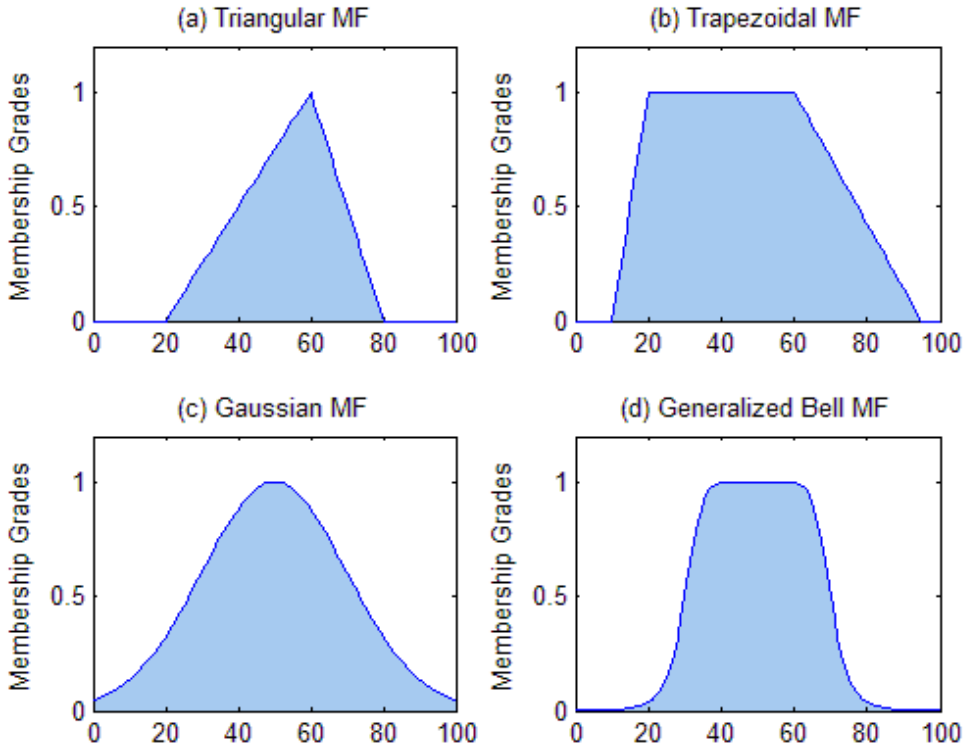

Figure A2.5. Examples of four classes of parameterised MFs:
(a) triangle (x; 20, 50, 80); (b) trapezoid (x; 10, 20, 60, 95);
(c) gaussian (x; 50, 20); (d) bell (x; 20, 4, 50)

Both triangular MFs and trapezoidal MFs have been used extensively, especially in real-time implementations, due to their simple formulas and computational efficiency. However, since the MFs are composed of straight-line segments, these segments are not smooth at the corner points specified by the parameters. The next two MFs definition are other types of MFs that have smooth curve and these curves are defined by non-linear functions.

Definition A2.6.*Gaussian MFs*

A Gaussian MF is specified by two parameters {g, σ}:

gaussian (x; g, σ) = $e^{-\frac{1}{2}\left(\frac{x-g}{\sigma}\right)^2}$

A Gaussian MF is determined completely by parameters g and σ; g represents the MF centre and σ determines the MFs width. Figure A2.5(c) plots a Gaussian MF defined by gaussian (*x*; 50, 20).

Definition A2.7. Generalised bell MFs

A **generalised bell MF** (or **bell MF**) is specified by three parameters $\{d, e, f\}$:

$$\text{Bell}(x; d, e, f) = \frac{1}{1 + \left|\frac{x - f}{d}\right|^{2e}}$$

where the parameter e is usually positive. (If e is negative, the shape of bell MF becomesan upside down bell.) Figure A2.5(d) shows a generalized bell MF defined by bell (x; 20, 4, 50).

Appendix 3 – Integrated ANFIS Training Code

```
% Integrated Script for ANFIS Training.
% Created by Jonny Latuny - Perth - December 19, 2011

clear all;
close all;
clc;

win_env = 0; mac_env = 0;
override_mat_files_check = 0;

if exist('c:\Users\2103900\works2009','dir')
   target_dir='c:\Users\2103900\works2009\text_data\';
   source_dir='c:\Users\2103900\works2009\text_data\';
   target_dir_fis='c:\Users\2103900\works2009\fis_data\';
%res_code = 'ch1wxf';
   win_env = 1;
elseif exist('c:\works2009','dir')
   target_dir='c:\works2009\text_data\';
   source_dir='c:\works2009\text_data\';
   target_dir_fis='c:\works2009\fis_data\';
   override_mat_files_check = 1;
%res_code = 'ch1wxt';
   win_env = 1;
elseif exist('d:\works2009','dir')
   target_dir='d:\works2009\text_data\';
   source_dir='d:\works2009\text_data\';
   target_dir_fis='d:\works2009\fis_data\';
   override_mat_files_check = 1;
%res_code = 'ch1wxq';
   win_env = 1;
elseif exist('e:\works2009','dir')
   target_dir='e:\works2009\text_data\';
   source_dir='e:\works2009\text_data\';
   target_dir_fis='e:\works2009\fis_data\';
   override_mat_files_check = 1;
%res_code = 'ch1wx';
   win_env = 1;
else
   disp('Assigning target dir for Mac OS X');
   target_dir='/Users/jonnylatuny/works2009/text_data/';
   source_dir='/Users/jonnylatuny/works2009/text_data/';
   target_dir_fis='/Users/jonnylatuny/works2009/fis_data/';
   override_mat_files_check = 1;
%res_code = 'ch1mx';
   mac_env = 1;
end

res_code = 'ch1all';
xdecomp_level = 10;

xdata_code = input('Enter data code identifier: ');
xdata_id = num2str(xdata_code);
[brs,klm] = size(xdata_id);
if klm >= 4
    xdata_code = xdata_id;
```

```
elseif klm == 3
    xdata_code = ['0',num2str(xdata_code)];%add string 0data code
else
    disp('Data code identifier number not match');
end

xdb_init = 4;
for db_idx = 1:5 % db4,db8,db12,db22&db44
if db_idx == 1
      xdb_type = xdb_init;
end
if db_idx == 2
      xdb_type = xdb_init * 2;
end
if db_idx == 3
      xdb_type = db_idx * xdb_init;
end
if db_idx == 4
      xdb_type = 22;
end
if db_idx == 5
      xdb_type = xdb_init * 11;
end

    filedata1 = ['ch12_',num2str(xdata_code),'_','1.mat'];
    filedata2 = ['ch12_',num2str(xdata_code),'_','984.mat'];
    chk_mat1 =  exist(filedata1,'file');
    chk_mat2 =  exist(filedata2,'file');

if override_mat_files_check == 1
        chk_mat1 = 2;
        chk_mat2 = 2;
end

    filename1 = [target_dir,'multires10_db',num2str(xdb_type),...
'_ch1d_mvar_',num2str(xdata_code),'_984.txt'];
    filename2 = [target_dir,'multires10_db',num2str(xdb_type),...
'_ch2d_mvar_',num2str(xdata_code),'_984.txt'];
    chk_file1 =  exist(filename1,'file');
    chk_file2 =  exist(filename2,'file');

if chk_mat1 == 0 && chk_mat2 == 0
        disp('MAT files Not exist: calculation aborted');

elseif chk_file1 == 2 && chk_file2 == 2
        disp(['File hasil: ',filename1,' exists']);
        disp(['File hasil: ',filename2,' exists']);
        disp('Plotting existing Ch1 and Ch2 results ....');
        result1_avr = [];
        result2_avr = [];
% ====== load data existing data and do plotting ======
        data_ch1 = load(filename1);
        data_ch2 = load(filename2);

for i=1:xdecomp_level %level 1-10 wavelet decomposition
            data_plot_ch1 = eval(['data_ch1(:,',num2str(i),')']);
            data_plot_ch2 = eval(['data_ch2(:,',num2str(i),')']);
            data_ch1_avr = (sum(data_plot_ch1)) / 984;
            data_ch2_avr = (sum(data_plot_ch2)) / 984;
```

```
            eval(['data_ch1_avr_all_' num2str(i) ' = ...
data_ch1_avr;']);
            eval(['data_ch2_avr_all_' num2str(i) ' = ...
data_ch2_avr;']);

            result1_avr = [result1_avr; ...
eval(['data_ch1_avr_all_'num2str(i)])];
            result2_avr = [result2_avr; ...
eval(['data_ch2_avr_all_'num2str(i)])];

            chk_ith = mod(i,2);
if chk_ith == 1
                  attrib = 'r-';
else
                  attrib = 'b-';
end
            disp(['plotting cA data db',num2str(xdb_type),'-', ...
                num2str(i),'-th energy level of ch1 & ch2 data']);

            figure(1)
            subplot(2,1,1)
            plot(data_plot_ch1, attrib);
            title(['(a) Ch1 cA energy: db',...
num2str(xdb_type),'-',num2str(i), ...
' ith level - data code: ', ...
num2str(xdata_code)]);
            subplot(2,1,2)
            plot(data_plot_ch2, attrib);
            title(['(b) Ch2 cA energy: db',...
num2str(xdb_type),'-', ...
num2str(i), ' ith level - data code: ', ...
 num2str(xdata_code)]);
pause(0.5);
end

        result1_avr = result1_avr(1:9,:);
        result2_avr = result2_avr(1:9,:);

        figure(2)
        subplot(2,1,1);
        bar(result1_avr,'r');
        title(['(a) Average Ch1 cA En - ',num2str(xdata_code),...
' max values - db',num2str(xdb_type),...
' level 1 - 10']);
[ye,exj] = max(result1_avr);

        subplot(2,1,2);
        bar(result2_avr);
        title(['(b) Average Ch2 cA En - ',num2str(xdata_code),...
' max values - db',num2str(xdb_type),...
' level 1 - 10']);
[ye2,exj2] = max(result2_avr);

else

       xdata_awal = 1;
       xdata_akhir = 984;

       results1 =[];
       results2 =[];
```

```
% Channel 1 Signal processing
for j=xdata_awal:xdata_akhir

% Read signal in sequence
       signal_in = ...
['ch12_',num2str(xdata_code),'_',num2str(j),'.mat'];

       disp(['Processing Ch1 data no: ',...
signal_in, ' - db type: ', num2str(xdb_type)]);

%data_in = load(signal_in); %for txt data file
       eval('load(signal_in)'); %for mat data file

       signal_c1 = data_all(:,1);

       fctr = nextpow2(length(signal_c1));

       [h,g,rh,rg] = daub(xdb_type); % specify Daubechies filter
       mrx = multires(signal_c1,h,rh,g,rg,xdecomp_level); %do MRA

for v=1:xdecomp_level
if v==1
               d1 = mrx(1,:);
               a1 = mrx(2,:);
               a1_en = 1/2^(fctr-1) * sum(a1.^2);
               a1_rms = sqrt( ((sum(a1.^2))/length(a1)));
               a1_kurto = kurtosis(a1);
               [a1_y_fft, a1_x_fft] = max(abs(fft(a1)));
               a1_std = std(a1);
               a1_var = var(a1);
end
if v==2
               d2 = mrx(2,:);
               a2 = mrx(3,:);
               a2_en = 1/2^(fctr-2) * sum(a2.^2);
               a2_rms = sqrt( ((sum(a2.^2))/length(a2)));
               a2_kurto = kurtosis(a2);
               [a2_y_fft, a2_x_fft] = max(abs(fft(a2)));
               a2_std = std(a2);
               a2_var = var(a2);
end
if v==3
               d3 = mrx(3,:);
               a3 = mrx(4,:);
               a3_en = 1/2^(fctr-3) * sum(a3.^2);
               a3_rms = sqrt( ((sum(a3.^2))/length(a3)));
               a3_kurto = kurtosis(a3);
               [a3_y_fft, a3_x_fft] = max(abs(fft(a3)));
               a3_std = std(a3);
               a3_var = var(a3);
end
if v==4
               d4 = mrx(4,:);
               a4 = mrx(5,:);
               a4_en = 1/2^(fctr-4) * sum(a4.^2);
               a4_rms = sqrt( ((sum(a4.^2))/length(a4)));
               a4_kurto = kurtosis(a4);
               [a4_y_fft, a4_x_fft] = max(abs(fft(a4)));
               a4_std = std(a4);
               a4_var = var(a4);
```

```
end
if v==5
                d5 = mrx(5,:);
                a5 = mrx(6,:);
                a5_en = 1/2^(fctr-5) * sum(a5.^2);
                a5_rms = sqrt( ((sum(a5.^2))/length(a5)));
                a5_kurto = kurtosis(a5);
                [a5_y_fft, a5_x_fft] = max(abs(fft(a5)));
                a5_std = std(a5);
                a5_var = var(a5);
end
if v==6
                d6 = mrx(6,:);
                a6 = mrx(7,:);
                a6_en = 1/2^(fctr-6) * sum(a6.^2);
                a6_rms = sqrt( ((sum(a6.^2))/length(a6)));
                a6_kurto = kurtosis(a6);
                [a6_y_fft, a6_x_fft] = max(abs(fft(a6)));
                a6_std = std(a6);
                a6_var = var(a6);
end
if v==7
                d7 = mrx(7,:);
                a7 = mrx(8,:);
                a7_en = 1/2^(fctr-7) * sum(a7.^2);
                a7_rms = sqrt( ((sum(a7.^2))/length(a7)));
                a7_kurto = kurtosis(a7);
                [a7_y_fft, a7_x_fft] = max(abs(fft(a7)));
                a7_std = std(a7);
                a7_var = var(a7);
end
if v==8
                d8 = mrx(8,:);
                a8 = mrx(9,:);
                a8_en = 1/2^(fctr-8) * sum(a8.^2);
                a8_rms = sqrt( ((sum(a8.^2))/length(a8)));
                a8_kurto = kurtosis(a8);
                [a8_y_fft, a8_x_fft] = max(abs(fft(a8)));
                a8_std = std(a8);
                a8_var = var(a8);
end
if v==9
                d9 = mrx(9,:);
                a9 = mrx(10,:);
                a9_en = 1/2^(fctr-9) * sum(a9.^2);
                a9_rms = sqrt( ((sum(a9.^2))/length(a9)));
                a9_kurto = kurtosis(a9);
                [a9_y_fft, a9_x_fft] = max(abs(fft(a9)));
                a9_std = std(a9);
                a9_var = var(a9);
end
if v==10
                d10 = mrx(10,:);
                a10 = mrx(11,:);
                a10_en = 1/2^(fctr-10) * sum(a10.^2);
                a10_rms = sqrt( ((sum(a10.^2))/length(a10)));
                a10_kurto = kurtosis(a10);
                [a10_y_fft, a10_x_fft] = max(abs(fft(a10)));
                a10_std = std(a10);
                a10_var = var(a10);
end
```

```
end

        energy_A = [a1_en a2_en a3_en a4_en a5_en a6_en ...
            a7_en a8_en a9_en a10_en];
        rms_A = [a1_rms a2_rms a3_rms a4_rms a5_rms a6_rms ...
            a7_rms a8_rms a9_rms a10_rms];
        kurto_A = [a1_kurto a2_kurto a3_kurto a4_kurto ...
            a5_kurto a6_kurto a7_kurto a8_kurto a9_kurto a10_kurto];
        fft_a_x = [a1_x_fft a2_x_fft a3_x_fft a4_x_fft ...
            a5_x_fft a6_x_fft a7_x_fft a8_x_fft a9_x_fft a10_x_fft];
        fft_a_y = [a1_y_fft a2_y_fft a3_y_fft a4_y_fft ...
            a5_y_fft a6_y_fft a7_y_fft a8_y_fft a9_y_fft a10_y_fft];
        std_A = [a1_std a2_std a3_std a4_std a5_std a6_std ...
            a7_std a8_std a9_std a10_std];
        var_A = [a1_var a2_var a3_var a4_var a5_var a6_var ...
            a7_var a8_var a9_var a10_var];

        hasil_all1 = [energy_A rms_A kurto_A ...
fft_a_x fft_a_y std_A var_A];
        eval(['hasil_final1_' num2str(j) ' = hasil_all1;']);
        results1 = [results1; eval(['hasil_final1_' num2str(j)])];

        clear hasil_final1*;
        clear d*; clear a*;

end

    eval(['save ', filename1 ,' results1 -ASCII'])
    disp('======== Done Processing Ch1 Data ...');

% Channel 2 Signal processing
for j=xdata_awal:xdata_akhir

% Read signal in sequence
        signal_in =
['ch12_',num2str(xdata_code),'_',num2str(j),'.mat'];

%data_in = load(signal_in); % for txt data file
        eval('load(signal_in)'); % for mat data file
disp(['Processing Ch2 data no: ',...
signal_in, ' - db type: ', num2str(xdb_type)]);

        signal_c2 = data_all(:,2);
        fctr = nextpow2(length(signal_c2));

        [h,g,rh,rg] = daub(xdb_type);
        mrx = multires(signal_c2,h,rh,g,rg,xdecomp_level);

for v=1:xdecomp_level
if v==1
            d1 = mrx(1,:);
            a1 = mrx(2,:);
            a1_en = 1/2^(fctr-1) * sum(a1.^2);
            a1_rms = sqrt( ((sum(a1.^2))/length(a1)));
            a1_kurto = kurtosis(a1);
            [a1_y_fft, a1_x_fft] = max(abs(fft(a1)));
            a1_std = std(a1);
            a1_var = var(a1);
```

```
end
if v==2
            d2 = mrx(2,:);
            a2 = mrx(3,:);
            a2_en = 1/2^(fctr-2) * sum(a2.^2);
            a2_rms = sqrt( ((sum(a2.^2))/length(a2)));
            a2_kurto = kurtosis(a2);
            [a2_y_fft, a2_x_fft] = max(abs(fft(a2)));
            a2_std = std(a2);
            a2_var = var(a2);
end
if v==3
            d3 = mrx(3,:);
            a3 = mrx(4,:);
            a3_en = 1/2^(fctr-3) * sum(a3.^2);
            a3_rms = sqrt( ((sum(a3.^2))/length(a3)));
            a3_kurto = kurtosis(a3);
            [a3_y_fft, a3_x_fft] = max(abs(fft(a3)));
            a3_std = std(a3);
            a3_var = var(a3);
end
if v==4
            d4 = mrx(4,:);
            a4 = mrx(5,:);
            a4_en = 1/2^(fctr-4) * sum(a4.^2);
            a4_rms = sqrt( ((sum(a4.^2))/length(a4)));
            a4_kurto = kurtosis(a4);
            [a4_y_fft, a4_x_fft] = max(abs(fft(a4)));
            a4_std = std(a4);
            a4_var = var(a4);
end
if v==5
            d5 = mrx(5,:);
            a5 = mrx(6,:);
            a5_en = 1/2^(fctr-5) * sum(a5.^2);
            a5_rms = sqrt( ((sum(a5.^2))/length(a5)));
            a5_kurto = kurtosis(a5);
            [a5_y_fft, a5_x_fft] = max(abs(fft(a5)));
            a5_std = std(a5);
            a5_var = var(a5);
end
if v==6
            d6 = mrx(6,:);
            a6 = mrx(7,:);
            a6_en = 1/2^(fctr-6) * sum(a6.^2);
            a6_rms = sqrt( ((sum(a6.^2))/length(a6)));
            a6_kurto = kurtosis(a6);
            [a6_y_fft, a6_x_fft] = max(abs(fft(a6)));
            a6_std = std(a6);
            a6_var = var(a6);
end
if v==7
            d7 = mrx(7,:);
            a7 = mrx(8,:);
            a7_en = 1/2^(fctr-7) * sum(a7.^2);
            a7_rms = sqrt( ((sum(a7.^2))/length(a7)));
            a7_kurto = kurtosis(a7);
            [a7_y_fft, a7_x_fft] = max(abs(fft(a7)));
            a7_std = std(a7);
            a7_var = var(a7);
end
```

```
if v==8
            d8 = mrx(8,:);
            a8 = mrx(9,:);
            a8_en = 1/2^(fctr-8) * sum(a8.^2);
            a8_rms = sqrt( ((sum(a8.^2))/length(a8)));
            a8_kurto = kurtosis(a8);
            [a8_y_fft, a8_x_fft] = max(abs(fft(a8)));
            a8_std = std(a8);
            a8_var = var(a8);
end
if v==9
            d9 = mrx(9,:);
            a9 = mrx(10,:);
            a9_en = 1/2^(fctr-9) * sum(a9.^2);
            a9_rms = sqrt( ((sum(a9.^2))/length(a9)));
            a9_kurto = kurtosis(a9);
            [a9_y_fft, a9_x_fft] = max(abs(fft(a9)));
            a9_std = std(a9);
            a9_var = var(a9);
end
if v==10
            d10 = mrx(10,:);
            a10 = mrx(11,:);
            a10_en = 1/2^(fctr-10) * sum(a10.^2);
            a10_rms = sqrt( ((sum(a10.^2))/length(a10)));
            a10_kurto = kurtosis(a10);
            [a10_y_fft, a10_x_fft] = max(abs(fft(a10)));
            a10_std = std(a10);
            a10_var = var(a10);
end
end

        energy_A = [a1_en a2_en a3_en a4_en a5_en a6_en ...
            a7_en a8_en a9_en a10_en];
        rms_A = [a1_rms a2_rms a3_rms a4_rms a5_rms a6_rms ...
            a7_rms a8_rms a9_rms a10_rms];
        kurto_A = [a1_kurto a2_kurto a3_kurto a4_kurto ...
            a5_kurto a6_kurto a7_kurto a8_kurto a9_kurto a10_kurto];
        fft_a_x = [a1_x_fft a2_x_fft a3_x_fft a4_x_fft ...
            a5_x_fft a6_x_fft a7_x_fft a8_x_fft a9_x_fft a10_x_fft];
        fft_a_y = [a1_y_fft a2_y_fft a3_y_fft a4_y_fft ...
            a5_y_fft a6_y_fft a7_y_fft a8_y_fft a9_y_fft a10_y_fft];
        std_A = [a1_std a2_std a3_std a4_std a5_std a6_std ...
            a7_std a8_std a9_std a10_std];
        var_A = [a1_var a2_var a3_var a4_var a5_var a6_var ...
            a7_var a8_var a9_var a10_var];

        hasil_all2 = [energy_A rms_A kurto_A fft_a_x ...
fft_a_y std_A var_A];

        eval(['hasil_final2_' num2str(j) ' = hasil_all2;']);
        results2 = [results2; eval(['hasil_final2_' num2str(j)])];

        clear hasil_final2*;
        clear d*; clear a*;

end

    eval(['save ', filename1 ,' results1 -ASCII'])
    eval(['save ', filename2 ,' results2 -ASCII'])
```

```
    disp('======== Done Processing Ch2 Data ...');

    data_ch1 = results1;
    data_ch2 = results2;
    result1_avr = [];
    result2_avr = [];

for i=1:xdecomp_level

            data_en_ch1 = eval(['data_ch1(:,',num2str(i),')']);
            data_en_ch2 = eval(['data_ch2(:,',num2str(i),')']);
            data_ch1_avr = (sum(data_en_ch1)) / 984;
            data_ch2_avr = (sum(data_en_ch2)) / 984;

            eval(['data_ch1_avr_all_'...
num2str(i) ' = data_ch1_avr;']);
            eval(['data_ch2_avr_all_'...
num2str(i) ' = data_ch2_avr;']);

            result1_avr = [result1_avr; ...
eval(['data_ch1_avr_all_' num2str(i)])];
            result2_avr = [result2_avr; ...
eval(['data_ch2_avr_all_' num2str(i)])];

            chk_ith = mod(i,2);
if chk_ith == 1
                  attrib = 'r-';
else
                  attrib = 'b-';
end
            disp(['plotting data cA db',num2str(xdb_type),'-', ...
                num2str(i),'-th energy level of ch1 & ch2 data']);

            figure(3)
            subplot(2,1,1)
            plot(data_en_ch1, attrib);
            title(['(a) Ch1 cA energy: db',num2str(xdb_type),'-',...
num2str(i), ' ith level - data code:...
',num2str(xdata_code)]);

            subplot(2,1,2)
            plot(data_en_ch2, attrib);
            title(['(b) Ch2 cA energy: db', ...
                num2str(xdb_type),'-',num2str(i), ...
' ith level - data code: ',num2str(xdata_code)]);

            pause(0.5);

end

        result1_avr9 = result1_avr(1:9,:);
        result2_avr9 = result2_avr(1:9,:);
        [ye,exj] = max(result1_avr9);
        [ye2,exj2] = max(result2_avr9);

end

    beep2(800,0.3);
```

```
if chk_mat1 == 0 || chk_mat2 == 0
        exj = 0; % set exj (max level) to zero
break;
end

    disp('------------------------------------------------');
    disp(['Max energy magnitude of ch1 data at level: ', ...
        num2str(exj)]); % show max level info

% ================== training & checking ====================
level_start = exj; % data level group to be used
    level_end = exj; % data level group to be used
    perms_start = 1; % number of permutations combination
    perms_end = 721; % max value 721
    swap_start = 1; % number of swap index
    swap_end = 7; % max 7

% Set string of filename of data files needed for training
    filename1 = [source_dir,'multires10_db',...
num2str(xdb_type),'_ch1d_mvar_',...
num2str(xdata_code),'_984.txt'];
    filename2 = [source_dir,'multires10_db',...
num2str(xdb_type),'_ch2d_mvar_',...
num2str(xdata_code),'_984.txt'];
    chk_file1 =  exist(filename1,'file');
    chk_file2 =  exist(filename2,'file');

if chk_file1 == 0 && chk_file2 == 0
          disp('Results files Not exist - process cancelled');
else

%Load data existing data from TXT result files
       data_ch1 = load(filename1);
       data_ch2 = load(filename2);

% Preparing empty container for 10 level of data
       data_level1 = []; data_level2 = [];
       data_level3 = []; data_level4 = [];
       data_level5 = []; data_level6 = [];
       data_level7 = []; data_level8 = [];
       data_level9 = []; data_level10 = [];

% Group data into data level 1 - 10
       klm = 1;
for i=1:7
          datax = data_ch1(:,klm);
          data_level1 = [data_level1 datax];
          klm = klm + 10;
end

       klm = 2;
for i=1:7
          datax = data_ch1(:,klm);
          data_level2 = [data_level2 datax];
          klm = klm + 10;
end

       klm = 3;
```

```
for i=1:7
            datax = data_ch1(:,klm);
            data_level3 = [data_level3 datax];
            klm = klm + 10;
end

        klm = 4;
for i=1:7
            datax = data_ch1(:,klm);
            data_level4 = [data_level4 datax];
            klm = klm + 10;
end

        klm = 5;
for i=1:7
            datax = data_ch1(:,klm);
            data_level5 = [data_level5 datax];
            klm = klm + 10;
end

        klm = 6;
for i=1:7
            datax = data_ch1(:,klm);
            data_level6 = [data_level6 datax];
            klm = klm + 10;
end

        klm = 7;
for i=1:7
            datax = data_ch1(:,klm);
            data_level7 = [data_level7 datax];
            klm = klm + 10;
end

        klm = 8;
for i=1:7
            datax = data_ch1(:,klm);
            data_level8 = [data_level8 datax];
            klm = klm + 10;
end

        klm = 9;
for i=1:7
            datax = data_ch1(:,klm);
            data_level9 = [data_level9 datax];
            klm = klm + 10;
end

        klm = 10;
for i=1:7
            datax = data_ch1(:,klm);
            data_level10 = [data_level10 datax];
            klm = klm + 10;
end

        disp('Ten levels of data processed successfully');

for level = level_start:level_end
```

```
%Preparing data for processing
       data_levelx = eval(['data_level',num2str(level)]);

chk_dir = exist('target_dir','var');
if chk_dir == 0
disp('Error: Target Dir not exist')
else

for swap_idx = swap_start:swap_end

          all_results = [];
          fis_record = [];
          fis_recordx = [];

if win_env == 1
            filehasil = [target_dir,'trn_mvar_v9db',...
num2str(xdb_type),'_',res_code,'_',...
num2str(xdata_code),'_res',...
num2str(level),num2str(swap_idx),'.txt'];
            file_fis = [target_dir,'fis_rec_',...
num2str(xdata_code),'_db',num2str(xdb_type),'_L',...
num2str(level),'_S',num2str(swap_idx),'.txt'];
else
            filehasil = [target_dir,'trn_mvar_v9db',...
num2str(xdb_type),'_',res_code,'_',...
num2str(xdata_code),'_res',num2str(level),...
num2str(swap_idx),'.txt'];
            file_fis = [target_dir,'fis_rec_',...
num2str(xdata_code),'_db',num2str(xdb_type),'_L',...
num2str(level),'_S',num2str(swap_idx),'.txt'];
end

          chk_file_result = exist(filehasil,'file');

if chk_file_result == 2
            msg_res = ['File hasil for ',num2str(xdata_code),...
' at level ',num2str(level),' swap ',...
                num2str(swap_idx),...
' exist - training process skipped'];
            disp(msg_res);

else
% Swapping data columns according to loop value
            data_levelx(:,[swap_idx 7]) = ...
data_levelx(:,[7 swap_idx]);
            data_output = data_levelx(:,7);

% New inputs data arrangements
if swap_idx == 1
              input_name = char('Variance','RMS',...
'Kurtosis','FFTx','FFTy','StdDev');
              out_name = '1Energy';
              input_idex = [7; 2; 3; 4; 5; 6];
elseif swap_idx == 2
              input_name = char('Energy','Variance','Kurtosis',...
'FFTx','FFTy','StdDev');
              out_name = '2RMS';
              input_idex = [1; 7; 3; 4; 5; 6];
elseif swap_idx == 3
              input_name = char('Energy','RMS','Variance',...
```

```
'FFTx','FFTy','StdDev');
                out_name = '3Kurtosis';
                input_idex = [1; 2; 7; 4; 5; 6];
elseif swap_idx == 4
                input_name = char('Energy','RMS','Kurtosis',...
'Variance','FFTy','StdDev');
                out_name = '4FFTx';
                input_idex = [1; 2; 3; 7; 5; 6];
elseif swap_idx == 5
                input_name = char('Energy','RMS','Kurtosis',...
'FFTx','Variance','StdDev');
                out_name = '5FFTy';
                input_idex = [1; 2; 3; 4; 7; 6];
elseif swap_idx == 6
                input_name = char('Energy','RMS','Kurtosis',...
'FFTx','FFTy','Variance');
                out_name = '6StdDev';
                input_idex = [1; 2; 3; 4; 5; 7];
elseif swap_idx == 7
                input_name = char('Energy','RMS','Kurtosis',...
'FFTx','FFTy','StdDev');
                out_name = '7Variance';
                input_idex = [1; 2; 3; 4; 5; 6];
end

            input_idx_no = 1:1:6;
            all_index = perms(input_idx_no);
            perms_akhir = [1 2 3 4 5 6];
            all_index = [all_index; perms_akhir];

% ============ permutations loop section ==================
for perms_idx = perms_start:perms_end %
trn_input_idx = all_index(perms_idx,:);

% assign proper input names
                input_name_nu = char([ eval(['input_name('...
                    num2str(all_index(perms_idx,1)) ',:)' ]); ...
eval(['input_name('...
num2str(all_index(perms_idx,2)) ',:)' ]); ...
eval(['input_name('...
                    num2str(all_index(perms_idx,3)) ',:)' ]); ...
eval(['input_name('...
num2str(all_index(perms_idx,4)) ',:)' ]); ...
eval(['input_name('...
num2str(all_index(perms_idx,5)) ',:)' ]); ...
eval(['input_name('...
                    num2str(all_index(perms_idx,6)) ',:)' ]) ]);

% getting original index of data index (1 - 7)
                input_idex_nu = ([ ...
eval(['input_idex('...
num2str(all_index(perms_idx,1)) ',:)' ]) ...
eval(['input_idex('...
num2str(all_index(perms_idx,2)) ',:)' ]) ...
eval(['input_idex('...
num2str(all_index(perms_idx,3)) ',:)' ]) ...
eval(['input_idex('...
                    num2str(all_index(perms_idx,4)) ',:)' ]) ...
eval(['input_idex('...
num2str(all_index(perms_idx,5)) ',:)' ]) ...
eval(['input_idex('...
```

```
num2str(all_index(perms_idx,6)) ',:)' ]) ]);

% arranging input data using permutation index
                data_input = [ ...
data_levelx(:,trn_input_idx(:,1)) ...
data_levelx(:,trn_input_idx(:,2)) ...
data_levelx(:,trn_input_idx(:,3)) ...
data_levelx(:,trn_input_idx(:,4)) ...
data_levelx(:,trn_input_idx(:,5)) ...
data_levelx(:,trn_input_idx(:,6))];
                data_in_out = [data_input data_output];
% Preparing training and checking data
                trn_data = data_in_out(1:2:end,:);%odd rows
                chk_data = data_in_out(2:2:end,:);%even rows
%=======================================
                input_index = exhsrch(2, trn_data, chk_data,...
                    input_name_nu);
                input_index_x = input_index;
% =======================================

                new_trn_data = trn_data...
                    (:,[input_index_x, size(trn_data,2)]);
                new_chk_data = chk_data...
                    (:,[input_index_x, size(chk_data,2)]);
input_idx1 = input_index_x(1,1);
                input_idx1 = input_idex_nu(:,input_idx1);
                input_idx2 = input_index_x(1,2);
                input_idx2 = input_idex_nu(:,input_idx2);
                input_name1 = eval(['input_name_nu('...
                    num2str(input_index_x(1,1)) ',:)']);
                input_name2 = eval(['input_name_nu('...
                    num2str(input_index_x(1,2)) ',:)']);
% ===========================================
                chk_var1 =  exist('new_trn_data','var');
                chk_var2 =  exist('new_chk_data','var');
if chk_var1 == 0 && chk_var2 == 0
                    disp('TRN & CHK Data Not exist');
else
trn_data = new_trn_data;
chk_data = new_chk_data;
testdat = new_chk_data;
output_name = char(out_name);
output_index = 1;
% ========== Training Parameters ==========
mf_n = 4;
mf_type = 'gbellmf';
method = 1; % hybrid
epoch_n = 100;
err_tol = nan;
ss = 0.01; % default
disp_opt = nan;
ss_dec_rate = 0.5; % default = 0.9
ss_inc_rate = 1.5; % default = 1.1
in_fismat = genfis1(trn_data, mf_n, mf_type);
% ========= Training Method ==============
[trn_out_fismat trnError step_size ...
                        chk_out_fismat chkError] = ...
                        anfis(trn_data, in_fismat, ...
[epoch_n err_tol ss ss_dec_rate ss_inc_rate],...
disp_opt, chk_data, method);
[RMSE, Epoch] = min(chkError);
```

```
disp(['Minimum Training RMSE = ' num2str(RMSE)]);
disp(['Epoch of Minimum Training RMSE = '...
num2str(Epoch)]);
% ==== Assign names to Inputs and Output ====
for i_name=1:length(input_index);
trn_out_fismat = setfis(trn_out_fismat, ...
'input', i_name, 'name', deblank...
(input_name_nu(input_index(i_name), :)));
 chk_out_fismat = setfis(chk_out_fismat, ...
'input', i_name, 'name', deblank...
(input_name_nu(input_index(i_name), :)));
end

                    trn_out_fismat = setfis(trn_out_fismat, ...
'output',1, 'name', deblank(output_name...
                        (size(output_name, 1), :)));
                    chk_out_fismat = setfis(chk_out_fismat, ...
'output',1, 'name', deblank(output_name...
                        (size(output_name, 1), :)));

% Test the Generated FIS
                    testdat = testdat(:,1:2);
                    fuzzy_out = evalfis(testdat, trn_out_fismat);

end

disp(['Selected inputs: ',input_name1,' & ', input_name2]);
[eR,Pe] = corrcoef(fuzzy_out, chk_data(:,3));
corr_r = eR(1,2);
corr_p = Pe(1,2);
disp(['Correlation coeff R fuzzy_out & chk_data_c3 = ', ...
num2str(corr_p)]);
disp(['Correlation coeff P fuzzy_out & chk_data_c3 = ', ...
num2str(corr_r)]);

hasil_all = [input_idx1 input_idx2 RMSE ...
                Epoch corr_r corr_p trn_input_idx swap_idx...
input_idex_nu];
eval(['hasil_calc_' num2str(perms_idx) ' = hasil_all;']);
all_results = [all_results; ...
eval(['hasil_calc_' num2str(perms_idx)])];
clear hasil_calc_*;
filename_trn = [target_dir_fis,'trn_out_',...
num2str(xdata_code),'_db_',...
num2str(xdb_type),'_L',...
num2str(level),'_S',num2str(swap_idx),'_E',...
num2str(Epoch),'_a'];
filename_chk = [target_dir_fis,'chk_out_',...
num2str(xdata_code),'_db_',...
num2str(xdb_type),'_L',...
num2str(level),'_S',num2str(swap_idx),'_E',...
num2str(Epoch),'_a'];
filename_trns = ['trn_out_',...
num2str(xdata_code),...
'_db_',num2str(xdb_type),'_L',...
num2str(level),'_S',num2str(swap_idx),'_E',...
num2str(Epoch),'_a'];
if RMSE < 10^-3 && Epoch >=5 && Epoch < 75 &&...
corr_r >= 0.9;
```

```
                eval(['fis_record_chk_'...
                    num2str(perms_idx) ' = filename_trns;']);
                fis_record = [fis_record; ...
{eval(['fis_record_chk_'...
                    num2str(perms_idx)])}];
                file_fis_a = ['trn_out_',...
num2str(xdata_code),'_db_', ...
(xdb_type),'_L',num2str(level),'_S',...
num2str(swap_idx),...
'_E',num2str(Epoch),'_a.fis'];
                chk_fis_a =  exist(file_fis_a,'file');
if chk_fis_a == 0
                   writefis(trn_out_fismat,filename_trn);
end
                fis_rec = fis_record;
                clear fis_record_chk_*;
else
                disp(' ===***=== FIS not saved ===***=== ');
end
% ================================================
            disp(['Calculating ',res_code,' at level: ',...
                num2str(level),' swap index: ',...
                num2str(swap_idx),' perms combination: ', ...
num2str(perms_idx), ' db type ',...
num2str(xdb_type)]);

end

% Saving all results data to disk
          eval(['save ', filehasil ,' all_results -ASCII'])
          [brs_fis, klm_fis] = size(fis_record);
if brs_fis > 0

% save to text file
              fid = fopen(file_fis, 'wt');
              fprintf(fid, '%s\n', fis_rec{:});
              fclose(fid);
end
clear input_index;
          clear all_results;
end
end
end
end
    clear data_levelx;
end
end
```

Appendix 4 – Analogue Low-Pass Filter (LPF)

Schematic of the analogue low-pass filter (LPF) used for data acquisition process is shown in Figure 1. The LPF design was based on Sallen-Key LPF circuit (2-pole Butterworth filter).

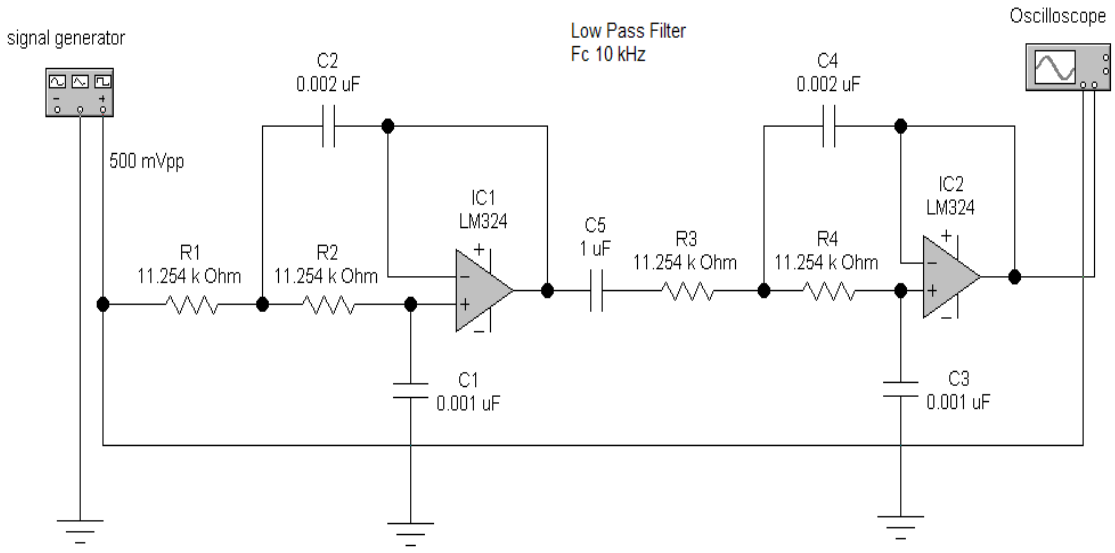

Figure 1. LPF Circuit

Design procedures (Carter, 2001) ♣

- Choose a cut-off frequency fo (Hz).
- In this research fo = 10 kHz.
- Select C1 value between 100 pF and 0.1 uF.
- C1 was chosen to be 1000 pF.
- Make C2 = 2 x C1
- C2 = 2 x C1 = 2000 pF
- Calculate R1 = R2 = 0.707 / (2 · π · fo · C1)
- R1 = R2 = 0.707 / (2 · π · 10kHz · 1000 pF) = 11.2 K Ohms

♣Carter, B. 2001, Filter Design in Thirty Seconds - Application Report SLOA093, Texas Instruments

Appendix 5 – Matlab Code for NI-DAQ USB 6251

```
% Script to run data acquisition session with NI USB-6251 device
% Created by Jonny Latuny

tic
run_speed = 35;
loop_no = 984;

%helpdoc datestr for more info about date string
tanggal = datestr(now);
tgl_date = datestr(tanggal,7);
tgl_month = datestr(tanggal,5);
tgl_year = datestr(tanggal,10);
tgl_time = datestr(tanggal,13);
tgl_ampm = datestr(tanggal, 15);
hour = tgl_ampm(:,1:2);
min = tgl_ampm(:,4:5);
sec = tgl_time(:,7:8);
format_tgl = [tgl_date tgl_month tgl_year '_' hour min sec];
nama_dir=['c:\mat_data\mat_data_',tgl_date,tgl_month,tgl_year,'_',
...
    num2str(run_speed),'\'];
if exist(nama_dir,'dir')
% do nothing, directory exist
   disp('Directory exist');
else
   mkdir(nama_dir); % create directory;
end;

plot_graph = 1;
simpan_data = 1;

for i=1:loop_no

AI = analoginput('nidaq','Dev1');
set(AI,'InputType','SingleEnded')
addchannel(AI,0:1); % For NI and MCC

Fs = 48192;
sample_rate = Fs;
blocksize = sample_rate * 1; % 1 or 10 second(s) acquisition

set(AI,'SampleRate', sample_rate);
set(AI,'SamplesPerTrigger', blocksize);

start(AI);
wait(AI,10); %wait for AI stopped before clearing workspace
data = getdata(AI);
data_daq = data;
data_ch1 = data(:,1);
data_ch2 = data(:,2);

if plot_graph == 1
```

```
    figure(1)
    subplot(2,1,1)
    plot(data_ch1,'r-');
    xlabel('Samples');
    ylabel('Signal (Volts)');
    title('Acceleration data: Channel 1');

    subplot(2,1,2)
    plot(data_ch2,'b-');
    xlabel('Samples');
    ylabel('Signal (Volts)');
    title('Acceleration data: Channel 2');

    fft_data_ch1 = abs(fft(data_ch1))/length(data_ch1);
    [ymax1,xmax1] = max(fft_data_ch1);

    fft_data_ch2 = abs(fft(data_ch2))/length(data_ch2);
    [ymax2,xmax2] = max(fft_data_ch2);

    t = 1:1:length(fft_data_ch1);
    max_freq1 = num2str(xmax1);
    max_freq2 = num2str(xmax2);
    [freq1_db,mag1_db] = daqdocfft(data_ch1,Fs,blocksize);
    [freq2_db,mag2_db] = daqdocfft(data_ch2,Fs,blocksize);

    [ymax1_db,maxindex1_db]= max(mag1_db);
    maxfreq1_db = num2str(maxindex1_db);

    [ymax2_db,maxindex2_db]= max(mag2_db);
    maxfreq2_db = num2str(maxindex2_db);

    figure(3)
    subplot(2,1,1)
    plot(freq1_db,mag1_db,'r-',maxindex1_db,ymax1_db,'bo')
    grid on
    ylabel('Magnitude (dB)')
    xlabel('Frequency (Hz)')
    judul_daq1 = ['Frequency Components of Ch1 - Max Freq: ', ...
        maxfreq1_db, ' Hz'];
    title(judul_daq1);

    subplot(2,1,2)
    plot(freq2_db,mag2_db,'b-',maxindex2_db,ymax2_db,'ro')
    grid on
    ylabel('Magnitude (dB)')
    xlabel('Frequency (Hz)')
    judul_daq2 = ['Frequency Components of Ch2 - Max Freq: ', ...
        maxfreq2_db, ' Hz'];
    title(judul_daq2);
end

delete(AI);
clear AI;

extension = '.mat'; % ekstension utk nama file
namafile = [nama_dir,'ch12_',tgl_date,tgl_month,'_',...
num2str(i),extension];
```

```
data_all = [data_ch1 data_ch2];
if simpan_data == 1
   eval(['save ', namafile ,' data_all']);
end

pesan = ['Processing File No: ',num2str(i)];
disp(pesan);
pause(1);

end

toc
```

Appendix 6 – Fault Classifier Testing Code

```
% Integrated Script for Loading data and Testing FIS units
% This script is used to get input1 & input2 indexes
% and use the information to load appropriate FIS unit
% Created by Jonny Latuny.
% Last Updated: April 9, 2013. Version 3

clear all;
close all;
clc;

xdecomp_level = 10;
win_env = 0; mac_env = 0; linux_env = 0;
show_plot = 0;

if exist('c:\Users\2103900\','dir')
   target_dir='c:\works2009\text_data\';
   source_dir='c:\works2009\text_data\';
   target_dir_fis='c:\works2009\fis_data\';
win_env = 1;
elseif exist('c:\works2009','dir')
   target_dir='c:\works2009\text_data\';
   source_dir='c:\works2009\text_data\';
   target_dir_fis='c:\works2009\fis_data\';

   win_env = 1;
elseif exist('d:\works2009','dir')
   target_dir='d:\works2009\text_data\';
   source_dir='d:\works2009\text_data\';
   target_dir_fis='d:\works2009\fis_data\';
win_env = 1;
elseif exist('e:\works2009','dir')
   target_dir='e:\works2009\text_data\';
   source_dir='e:\works2009\text_data\';
   target_dir_fis='e:\works2009\fis_data\';
win_env = 1;
elseif exist('/home/latunyj/','dir')
   disp('Working in Ubuntu Linux Environment');
   target_dir='/media/LatunyJ500G/works2009/text_data/';
   source_dir='/media/LatunyJ500G/works2009/text_data/';
   target_dir_fis='/media/LatunyJ500G/works2009/fis_data/';
linux_env = 1;
else
   disp('Assigning target dir for Mac OS X');
   target_dir='/Users/jonnylatuny/works2009/text_data/';
   source_dir='/Users/jonnylatuny/works2009/text_data/';
   target_dir_fis='/Users/jonnylatuny/works2009/fis_data/';
mac_env = 1;
end

res_code = 'ch1all';

chk_dir = exist('target_dir','var');
if chk_dir == 0
    disp('Error: Target Dir not exist!.m')
else
```

```
    results_corrx_all = [];
    xdb_init = 4;
for db_idx = 1:5 % cycle between db4,8,12,22, & 44
if db_idx == 1
      xdb_type = xdb_init;
end
if db_idx == 2
      xdb_type = xdb_init * 2;
end
if db_idx == 3
      xdb_type = db_idx * xdb_init;
end
if db_idx == 4
      xdb_type = 22;
end
if db_idx == 5
      xdb_type = xdb_init * 11;
end

%data_id = [1812 1912 2012 2112 0301 0401 0701 0801 0901 1001 1101
% 1201 1401 1501 1601 1701 1801 2101 2301]; % BF .7mm data DEFAULT
%data_id = [2211 2311 2611 2711 2811 2911 3011 0312 0412 0512 0612
% 0712 1412 1712]; % IRF .7mm
%data_id = [511 611 911 1511 1911 2011 2111]; % ORF .7mm
data_id = [1402 1502 1802 2002 2202 2602]; % Normal bearing
[brs_id, klm_id] = size(data_id);

corr_container = [];

for jex = 1:klm_id
    xdata_code = data_id(1,jex); % first column (default)
xdata_id = num2str(xdata_code);
[brs,klm] = size(xdata_id);
if klm >= 4
%xdata_code = xdata_code % do nothing
xdata_code = xdata_id;
elseif klm == 3
xdata_code = ['0',num2str(xdata_code)]; % adding 0 string
else
disp('Data code identifier number not match');
end

filename1 = [target_dir,'multires10_db',...
num2str(xdb_type),'_ch1d_mvar_',...
num2str(xdata_code),'_984.txt']; % ch1 data
filename2 = [target_dir,'multires10_db',...
num2str(xdb_type),'_ch2d_mvar_',...
num2str(xdata_code),'_984.txt']; % ch2 data
        chk_file1 =  exist(filename1,'file');
        chk_file2 =  exist(filename2,'file');

        filedata1 = ['ch12_',num2str(xdata_code),'_','1.mat'];
filedata2 = ['ch12_',num2str(xdata_code),'_','984.mat'];
chk_mat1 =  exist(filedata1,'file');
        chk_mat2 =  exist(filedata2,'file');

if chk_mat1 == 0 && chk_mat2 == 0
disp('MAT files Not exist');

elseif chk_file1 == 2 && chk_file2 == 2 %
```

```
disp_msg1 = ['File hasil: ',filename1,' exists'];
disp_msg2 = ['File hasil: ',filename2,' exists'];
disp(disp_msg1);
disp(disp_msg2);
disp('Plotting existing Ch1 and Ch2 results ....');
result1_avr = [];
result2_avr = [];

% load data existing data and do plotting
data_ch1 = load(filename1); % for txt data file
data_ch2 = load(filename2); % for txt data file

for i=1:xdecomp_level
data_plot_ch1 = eval(['data_ch1(:,',num2str(i),')']);
data_plot_ch2 = eval(['data_ch2(:,',num2str(i),')']);
data_ch1_avr = (sum(data_plot_ch1)) / 984;
data_ch2_avr = (sum(data_plot_ch2)) / 984;

eval(['data_ch1_avr_all_' num2str(i) ' = data_ch1_avr;']);
eval(['data_ch2_avr_all_' num2str(i) ' = data_ch2_avr;']);

result1_avr = [result1_avr; ...
eval(['data_ch1_avr_all_' num2str(i)])];
          result2_avr = [result2_avr;...
eval(['data_ch2_avr_all_' num2str(i)])];
chk_ith = mod(i,2);
if chk_ith == 1
attrib = 'r-';
else
attrib = 'b-';
end
pesan = ['plotting cA data db',num2str(xdb_type),'-',...
num2str(i),'-th energy level of ch1 & ch2 data'];
disp(pesan);

if show_plot == 1
figure(1)
subplot(2,1,1)
plot(data_plot_ch1, attrib);
judul1 = ['(a) Ch1 cA energy: db',...
num2str(xdb_type),'-',...
num2str(i), ' ith level - data code: ',...
num2str(xdata_code)];
title(judul1);

subplot(2,1,2)
plot(data_plot_ch2, attrib);
judul2 = ['(b) Ch2 cA energy: db',...
num2str(xdb_type),'-',...
num2str(i), ' ith level - data code: ',...
num2str(xdata_code)];
title(judul2);
end
end

result1_avr = result1_avr(1:9,:);
result2_avr = result2_avr(1:9,:);

if show_plot == 1
figure(2)
```

```
subplot(2,1,1);
bar(result1_avr,'r');
judul1 = ['(a) Average Ch1 cA En - ',...
num2str(xdata_code),' max values - db',...
num2str(xdb_type),' level 1 - 9'];
title(judul1)

subplot(2,1,2);
bar(result2_avr);
judul2 = ['(b) Average Ch2 cA En - ',...
num2str(xdata_code),' max values - db',...
num2str(xdb_type),' level 1 - 9'];
title(judul2)
end
[ye1,ex1] = max(result1_avr); % get the max avr level
[ye2,ex2] = max(result2_avr); % get the max avr level
end


message1 = ['Max energy magnitude of ch1 data occurs at level: ',
num2str(ex1)];
disp('------------------------------------------------------');
disp(message1);

level_start = ex1; % data level group to be used
level_end = ex1; % max value 10
swap_start = 1; % number of swap index
swap_end = 7; % max 7

% Data / Result files for ANFIS training
filename1 = [source_dir,'multires10_db',...
num2str(xdb_type),'_ch1d_mvar_',...
num2str(xdata_code),'_984.txt'];
filename2 = [source_dir,'multires10_db',...
num2str(xdb_type),'_ch2d_mvar_',...
num2str(xdata_code),'_984.txt'];
chk_file1 =  exist(filename1,'file');
chk_file2 =  exist(filename2,'file');

if chk_file1 == 0 && chk_file2 == 0
disp('TXT files of result Not exist - process cancelled');
else%if TXT file exist, continue

%Load data existing data from TXT result files
data_ch1 = load(filename1); % for txt data file
data_ch2 = load(filename2); % for txt data file

data_level1 = []; data_level2 = [];
       data_level3 = []; data_level4 = [];
data_level5 = []; data_level6 = [];
data_level7 = []; data_level8 = [];
data_level9 = []; data_level10 = [];

% Group data into data level 1 - 10
klm = 1; %1st level
for i=1:7 % 7 parameters
datax = data_ch1(:,klm);
data_level1 = [data_level1 datax];
klm = klm + 10;
end
```

```
klm = 2;
for i=1:7 % 7 parameters
datax = data_ch1(:,klm);
data_level2 = [data_level2 datax];
klm = klm + 10;
end

klm = 3;
for i=1:7 % 7 parameters
datax = data_ch1(:,klm);
data_level3 = [data_level3 datax];
klm = klm + 10;
end

klm = 4;
for i=1:7 % 7 parameters
datax = data_ch1(:,klm);
data_level4 = [data_level4 datax];
klm = klm + 10;
end

klm = 5;
for i=1:7 % 7 parameters
datax = data_ch1(:,klm);
data_level5 = [data_level5 datax];
klm = klm + 10;
end

klm = 6;
for i=1:7 % 7 parameters
datax = data_ch1(:,klm);
data_level6 = [data_level6 datax];
klm = klm + 10;
end

klm = 7;
for i=1:7 % 7 parameters
datax = data_ch1(:,klm);
data_level7 = [data_level7 datax];
klm = klm + 10;
end

klm = 8;
for i=1:7 % 7 parameters
datax = data_ch1(:,klm);
data_level8 = [data_level8 datax];
klm = klm + 10;
end

klm = 9;
for i=1:7 % 7 parameters
datax = data_ch1(:,klm);
data_level9 = [data_level9 datax];
klm = klm + 10;
end

klm = 10; % 10th level
for i=1:7 % 7 parameters
datax = data_ch1(:,klm);
data_level10 = [data_level10 datax];
```

```
klm = klm + 10;
end

disp('Ten levels data processed successfully');

for level = level_start:level_end
data_levelx = eval(['data_level',num2str(level)]); %
end% iterating 10 times

end% check whether file txt exist

name_idx = 1;
input_name =...
char('Energy','RMS','Kurtosis','cA_x','cA_y',...
'StdDev','Variance');
        input_idex = [1; 2; 3; 4; 5; 6; 7];
        all_index = [1 2 3 4 5 6 7]; % clmn 7 as target output

trn_input_idx = all_index(1,:);
input_name_nu = char([ eval(['input_name('...
num2str(all_index(name_idx,1)) ',:)' ]); ...
eval(['input_name(' num2str(all_index(name_idx,2)) ',:)' ]);...
eval(['input_name(' num2str(all_index(name_idx,3)) ',:)' ]);...
eval(['input_name(' num2str(all_index(name_idx,4)) ',:)' ]);...
eval(['input_name(' num2str(all_index(name_idx,5)) ',:)' ]);...
eval(['input_name(' num2str(all_index(name_idx,6)) ',:)' ]);...
eval(['input_name(' num2str(all_index(name_idx,7)) ',:)' ]) ]);

% getting index of data index (1 - 7)
input_idex_nu = ([ ...
eval(['input_idex(' num2str(all_index(name_idx,1)) ',:)' ])...
eval(['input_idex(' num2str(all_index(name_idx,2)) ',:)' ])...
eval(['input_idex(' num2str(all_index(name_idx,3)) ',:)' ])...
eval(['input_idex(' num2str(all_index(name_idx,4)) ',:)' ])...
eval(['input_idex(' num2str(all_index(name_idx,5)) ',:)' ])...
eval(['input_idex(' num2str(all_index(name_idx,6)) ',:)' ])...
eval(['input_idex(' num2str(all_index(name_idx,7)) ',:)' ]) ]);

% arranging input data
data_input = [ ...
data_levelx(:,trn_input_idx(:,1)) ...
data_levelx(:,trn_input_idx(:,2)) ...
data_levelx(:,trn_input_idx(:,3)) ...
data_levelx(:,trn_input_idx(:,4)) ...
data_levelx(:,trn_input_idx(:,5)) ...
data_levelx(:,trn_input_idx(:,6)) ...
data_levelx(:,trn_input_idx(:,7))];

data_in_out = data_input; %joining 7th column data
trn_data = data_in_out(1:2:end,:); %training data
chk_data = data_in_out(2:2:end,:); %checking data
new_trn_data = data_in_out(1:2:end,:);%training data
new_chk_data = data_in_out(2:2:end,:);%checking data

trn_data = new_trn_data; % 7 columns data
chk_data = new_chk_data; % 7 columns data
testdat = new_chk_data; % 7 columns data

% ============== Load and Test FIS ==================
```

```
tested_fis = readfis('trn_out_2301_db_44_L7_S2_E14_a.fis');

%testdat = [chk_data(:,6) chk_data(:,2)]; %Inp1 & Inp2
testdat = [chk_data(:,6) chk_data(:,7)]; %Inp1 & Inp2

fuzzy_out = evalfis(testdat(:,1:2), tested_fis); %test FIS
trn_data_out = testdat(:,1); % define Target Output

if show_plot == 1
figure(3)
subplot(3,1,1)
plot(fuzzy_out,'r-');
xlabel(['chk data']);
ylabel('mag');

subplot(3,1,2)
plot(trn_data_out,'b-');
xlabel(['chk data']);
ylabel('mag');

zeros_data = zeros(1,length(fuzzy_out));
zeros_idx = 1:1:length(fuzzy_out);

subplot(3,1,3)
plot(zeros_idx, zeros_data,'c-');
hold on;
plot((fuzzy_out-trn_data_out),'k-');
title(['(c) error of chk data vs FIS Out']);
xlabel(['data points']);
ylabel('error');
hold off;
end

[eR,Pe] = corrcoef(fuzzy_out, trn_data_out);
corr_r1 = eR(1,1);
corr_r2 = eR(1,2);
disp(['Correlation coeff (R) fuzzy_out vs trn_data_out: ', ...
num2str(corr_r1), ' vs ',num2str(corr_r2)]);

eval(['corrx_' num2str(jex) ' = corr_r2;']);
corr_container = [corr_container; ...
eval(['corrx_' num2str(jex)])];
clear corrx_*;

end

   eval(['hasil_corr_all' num2str(db_idx) ' = corr_container;']);
   results_corrx_all = [results_corrx_all ...
eval(['hasil_corr_all' num2str(db_idx)])];
   clear hasil_corr_all*;

end
end
```

Addressing Matter Suggested by an Examiner

Background to the Usage of the Seven Features

The research aimed to integrate all of the features / parameters related to the physical faults of a bearing which includes time-domain, frequency-domain and statistical parameters into a set of independent features for the purpose of training an intelligent system (i.e., ANFIS).

The motivation for the usage of the seven selected parameters as features in the research was the finding that, in practice, features or indicators that related to the physical condition of rotating machine components (e.g., bearings and gears) of interest are needed for the purpose of condition monitoring. This aims to reduce the amount of vibration data and offers an efficient representation of a convenient method for identifying trends or patterns that relate to the operating condition developments of rotating machine components (Wang and McFadden, 1993). The selection of statistical parameters was also supported by their wider application in bearing fault detection applications and their advantages when used for the purposes of tracking bearing damage from an early stage of its development, when the results that are not influenced by the variations of both load and speed (Martin and Honarvar, 1995)

Furthermore, the inclusion of all related features (i.e., the seven features) that have relationship to the faults aimed to improve the accuracy of the fault detection system designed and implemented in the research. The application of the seven features also aimed to extend the coverage to which the fault detection system (i.e., ANFIS) was applicable since it was developed with a wider range of features than other systems. Hence it could extend the scope of information which was useful for the purposes of the learning and identification processes utilized in the research.

In addition, the literature shows that statistical parameters such as RMS, kurtosis (Boulahbal *et al*. 1999), (Wang and Wong, 2002), standard deviation (Mathew and Alfredson, 1984), (Xu *et al*. 2009), variance (Loutridis, 2008), (Rafiee *et al*. 2009)

are among the most frequently used features that related to the physical condition of a bearing. As an example, the use of the kurtosis parameter was based on its popularity in bearing fault detection and its characteristics which are useful in the detection of early stages of bearing deterioration. In this case, the variation of the vibrations signal's kurtosis spikiness could be used to detect an incipient fault in a bearing (Xi *et al*. 2000). This parameter has shown its usefulness and has been widely used (Dyer and Stewart, 1978), (Howard, 1994), (Lee and White, 1997), (Xi *et al*. 2000), (Raj and Murali, 2013).

The two features labelled *cA_x* and *cA_y* were extracted from the cA parts of wavelet transform with MRA were partitioned into several sub-bands section (Qiu *et al*. 2006, Zhu *et al*. 2009, Wu and Kuo, 2009, Wu and Hsu, 2009). Each cA part has a dominant frequency and this dominant frequency (*cA_x*) has a corresponding magnitude (*cA_y*). Therefore, these two parameters have a unique pattern that is a candidate for use as features in bearing fault detection. The use of these two features may also be seen as an investigation to the possibility of using the results of MRA (multi-resolution analysis) of the wavelet transform.

The use of energy level of the wavelet transform results was encouraged by the application of this parameter by Goswami and Chan (1999). However, the formulation for the calculation of energy level was modified as presented in Latuny and Entwistle (2010). The energy level was included as one of the seven features since it has the advantage that it shows up an early sign of bearing deterioration condition. Therefore, it was also included as one of the seven features that aimed to enhance the detection of low-level energy signatures that corresponding to early deterioration of a bearing.

Additional References

Boulahbal, D., Golnaraghi, M.F., and Ismail, F., 1999, 'Amplitude and Phase Wavelet Maps for the Detection of Crack in Geared Systems', *Mechanical Systems and Signal Processing*, vol. 13, no. 3, pp. 423-436.

Raj, A.S., and Murali, N., 2013. 'Early Classification of Bearing Faults Using Morphological Operators and Fuzzy Inference.' *IEEE Transactions on Industrial Electronics*, vol. 60, no. 2, pp. 567-574.

Wang, W.Y., and Wong, A.K., 2002, 'Autoregressive Model-based Gear Fault Diagnosis', *Journal of Vibration and Accoustics – Transactions of the ASME*, vol. 124, no. 2, pp. 172-179.