

# Smart Sensor Network Organization: Sensor Data Fusion and Industrial Fault Traceability

Saud Altaf

A thesis submitted to  
Auckland University of Technology  
in fulfilment of the requirements for the degree of  
Doctor of Philosophy (PhD)

2015

School of Engineering, Computer and Mathematical Sciences

# Declaration

I, hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been accepted for the award of any other degree or diploma of a university or other institution of higher learning.

Auckland, 2015

---

Saud Altaf

# Abstract

The industrial environment usually contains multiple motors that are supplied through a common power bus. The power-line acts as a good conducting environment for signals to travel through the power-line network. In effect, this influences other motors with noisy signals that may indicate a fault condition. Further complexity arises when signals are generated by motors with different power ratings, a different slip speed and more than one source of fault signals. This sort of complexity and mixed signals from multiple sources makes them difficult to measure and precisely correlate to a given machine or fault.

Generally, an industrial power-line network consists of different sizes of induction motor from small to large, which together can have a considerable combined influence on the overall system's operation. The combined effect of all these induction motors can have a strong impact on power-line network permanence. In this thesis, the concept of cross evaluation of motor fault signals is considered to be signal propagation manifesting into healthy signal. Different concepts relating to propagation and manifestation of faults will be discussed and analysed.

Initially, a systematic technique was employed to analyse the influence of the fault electric current signals of different motors within a power-line network. Further analysis analysed the attenuation ratio of electrical signals that leads toward a technical framework which evaluates the strength of signal propagation over a power-line network. The diagnostic process was demonstrated at individual sensing points to estimate the strength of propagated signals and identify fault points. This proved very helpful in maximising the different independent observations.

A sample industrial distributed motor network was simulated, to observe the behaviour of a distributed power-line network in the presence of fault components. The multi-motor dynamic simulation model was developed, to compare the results with the test-bed practical results, to validate the acquired data. A number of case scenario experiments was done to verify the simulation results and validate the accuracy of these results.

In this research, analytical results present significant improvements in describing the interference of faulty signals amongst motors running parallel to the power-line network. Some shortcomings were observed while implementing the strategy of distributed fault diagnosis, including false identification of similar types of fault symptom in power-line network and failure of the diagnosis system due to interference from non-linear noisy signals travelling within multi motor network. Some of these complications are supposed to be solvable by using an efficient and proper knowledge-based numerical technique. Furthermore, the focus of this research was also to develop a wireless node-level feature extraction technique for data fusion, using MCSA at end node-level. Decision-level fusion was implemented at the node coordinator for efficient fault diagnosis.

In conclusion, this research does not claim to provide a complete solution to cover all types of fault diagnosis in electric drives. But it is a fitting attempt to provide a more reliable industry solution for motor fault diagnosis.

## **Publications**

[1] Saud Altaf, Adnan Al-Anbuky and Hamid GholamHosseini, “Fault Signal Propagation in a Network of Distributed Motors” 2014 IEEE 8<sup>th</sup> International Power Engineering and Optimization Conference (PEOCO) 24-25 Mar 2014, Langkawi, Malaysia.

[2] Saud Altaf, Adnan Al-Anbuky and Hamid GholamHosseini, “Fault diagnosis in a distributed motor network using Artificial Neural Network” 2014 22<sup>nd</sup> International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM 2014), 18-20 Jun 2014, Ischia, Italy.

# Acknowledgements

First and foremost, I would like to say my profoundest gratitude to Almighty ALLAH (الله), the most gracious and most merciful God in this world for his blessing, strength, patience and knowledge that he has given me throughout this long, challenging, exciting, inspiring and adventurous PhD journey. This journey wouldn't have been possible without the utmost guidance and assistance of many individuals who, in one way or another contributed and extended their helpful.

This research would not be possible without the professional guidance, patience and support from my supervisors throughout. Their continuous encouraging, guidance especially, keep me on the right track. My greatest gratitude to my respected Primary Supervisor *Professor Adnan Al-Anbuky*, Director of the Sensor Network and Smart Environment Research Centre (SeNSe). His constant motivation and valuable mentoring helped me sail smoothly through easy and tough times. The weekly progress meetings helped me to stay focused and gradually progress in my study. His professional thorough proof reading on thesis drafts with quick responses helps me to improve and do better throughout duration of study.

I would like to thanks my co-supervisor *Dr. Hamid GholamHosseini* whose valuable guidance and suggestions helped to improve my thesis documentation during different phases of study.

I would like to dedicate this thesis to my dear parents, *Muhammad Altafur-Rehman and Tabassum Tahseen*, who have always been there to support my long PhD journey in all possible ways. Without their encouragement and blessing, this thesis would have not been possible. My profoundest thanks also to my Parents-in-law (*Chaudhry Muhammad Shareef and Naseem Akhtar*), sisters (*Shumaila, Nida and Sarah*) and brothers, for their unending motivation in my studies.

Especial thanks and acknowledgment to my loving, caring and beautiful wife *Ayesha*. Without her sacrifices, patience, love and enthusiastic support over the duration, this thesis would have not been completed. To my dear, cute

daughter *Hadia*, thank you for being part of my life and making pleasant this journey.

The funding for this work was provided by the Ministry of Science and Innovation, New Zealand, as sponsored by AuCom Electronics, Christchurch, under the programme contract number AUCM1001. I would like to thank and acknowledge AuCom electronics and industry mentor *Dr. Roger McKenzie* for their continued guidance and support.

I would also like to acknowledge the postgraduate office of the AUT School of Engineering for their support and assistance related to the PhD fee funding and doctoral forms. Special acknowledgements to the Program administrators and technicians of the School of Engineering, Auckland University of Technology (AUT).

I would like to acknowledge my friends, especially *Mirza Sajid Mahmood* and others in Pakistan for their support and motivation throughout these years.

Last but not least, my PhD colleagues at SeNSE group and AUT friends; Waseem, Sivaramakrishnan, Craig, Hakilo, Izani and others who always motivated me to complete this thesis.

# Table of Contents

<b>Abstract</b> .....	<b>i</b>
<b>Publications</b> .....	<b>ii</b>
<b>Acknowledgements</b> .....	<b>iii</b>
<b>List of Figures</b> .....	<b>viii</b>
<b>List of Tables</b> .....	<b>xi</b>
<b>List of Abbreviations</b> .....	<b>xii</b>
<b>1. Introduction</b> .....	<b>1</b>
1.1 Industrial Motor Network Fault Diagnosis.....	1
1.2 Fault Diagnosis Using Motor Current Signature Analysis Method .....	2
1.3 Possible Induction Motor Faults.....	4
1.3.1 Rotor Bar Broken and End Ring Faults.....	6
1.3.2 Air-Gap Eccentricity Fault .....	7
1.3.3 Bearing Wear and Rolling Element & Cage Defect Faults.....	9
1.3.4 Stator Winding Open and Short Circuit Faults.....	10
1.4 Motivation and Problem Statement.....	10
1.5 Contribution and Research Objectives .....	13
1.6 Thesis Organization.....	14
<b>2. Research Background and Literature Review</b> .....	<b>16</b>
2.1 Introduction .....	16
2.2 Signal Processing Techniques .....	16
2.2.1 Feature Extraction Techniques .....	17
2.3 Fault Diagnostic Methods.....	22
2.3.1 Signal-Based Methods .....	23
2.3.2 Model-Based Methods.....	25
2.3.3 Knowledge-Based Methods .....	27
2.3.4 Hybrid Methods .....	29
2.4 Propagation of Fault Signal in Industrial Power-Line Network.....	31
2.5 Industrial Fault Diagnosis using Wireless Sensor Networks .....	322
2.6 Shortcomings in Existing Fault Diagnosis Research.....	33
2.7 Chapter Summary.....	35
<b>3. Distributed Motor Network Signature Analysis and Fault Type Diagnosis Concept Formulation</b> .....	<b>37</b>

3.1 Introduction .....	37
3.2 Distributed Motor Network under Study.....	37
3.3 Distributed Network Model and Key Influencing Factors that may Contribute to Confusion in Fault Diagnosis .....	39
3.3.1 Key Factors Influencing the Ambiguities in Fault Diagnosis.....	39
3.4 Fault Diagnosis System Solution .....	42
3.5 Fault diagnosis System at Motor-Level.....	43
3.5.1 Sampling and Classifying Abnormal Sidebands Components of Motor Waveforms .....	44
3.5.2 Example Signature Pattern Caused by the Fault Symptoms at Motor-Level.....	45
3.6 Neural Network Architecture for Fault Detection at Motor-Level .....	46
3.7 Technical Distributed Fault Diagnosis Solution at Network-Level .....	48
3.7.1 Multiple Sensing Points Measurement for Identify the Origin of Significant Components over Distributed Multi-Motors Network.....	50
3.8 The Role of Neural Network Architecture within a Distributed Multiple Motor Environment.....	57
3.9 Chapter Summary.....	58
<b>4. Motor Network Modelling And Simulation.....</b>	<b>59</b>
4.1 Introduction .....	59
4.2 Mathematical Modelling of a Network of Induction Motors .....	59
4.3 Multi-Motor Network Simulation Model.....	61
4.4 Fault Injection Model Block at Motor-Level .....	63
4.4 Rotor and Eccentricity Faults Formation Model.....	64
4.5 Injection and Propagation of Faulty Signal at Distributed-Level.....	65
4.6 Case Study 1: Noisy Fault Signal Propagation from a Single Motor and Abnormal Behaviour of Motors in a Distributed Environment.....	67
4.7 Case Study 2: BRB and Eccentricity Fault Frequencies Injection and Observe Influence on Different Measuring Points .....	71
4.8 Chapter Summary.....	77
<b>5. Artificial Neural Network Solution for Fault Diagnosis.....</b>	<b>78</b>
5.1 Introduction .....	78
5.2 Artificial Neural Network Fault Type Diagnosis Solution.....	78
5.3 Neural Network Architecture and Diagnosis Solution at Motor-level.....	79
5.3.1 Fault Type Diagnosis Methodology.....	81
5.3.2 Neural Network Simulation Model at Motor-Level.....	82

5.4 Neural Network Simulation Model for Distributed-level .....	84
5.5 Case Study 3: Fault Type Diagnosis in a Multi-Motor Network Modelling Environment .....	85
5.6 Chapter Summary .....	94
<b>6. Distributed Motors System Experimental Test-Bed Environment .....</b>	<b>95</b>
6.1 Introduction .....	95
6.2 Experimental Multi-Motors Test-Bed .....	95
6.2.1 Hardware Measurement Tools .....	96
6.2.2 Test-Bed Measuring Data .....	97
6.3 Wireless Sensors and Related Network .....	100
6.3.1 Motor Wireless Sensor Hardware .....	101
6.3.2 System Software Design .....	104
6.4 XBee PRO ZB (S2B) Module .....	105
6.4.1 Modes of Operation and configuration for XBee Module .....	106
6.5 Architecture of Wireless Sensor Network .....	108
6.5.1 Operation of the Arduino Sensor Node .....	110
6.6 Arduino Fault diagnosis Case Study: Experimental Results .....	112
6.7 Chapter Summary .....	115
<b>7. Results and Analysis .....</b>	<b>116</b>
7.1 Introduction .....	116
7.2 Case Study 4: Faulty Motor within Network with Single Fault Symptoms .....	116
7.3 Case Study 5: Dissimilar Faulty Motors with Multiple Faults in the Same Bus .....	126
7.4 Case Study 6: Dissimilar Faulty Motors with Multiple Faults in Different Power Buses .....	133
7.6 Chapter Summary .....	140
<b>8. Conclusions And Recommendations .....</b>	<b>141</b>
8.1 Introduction .....	141
8.2 Achievements of Research Tasks and their Future Scope .....	141
8.3 Recommendations and Future Research .....	144
<b>References .....</b>	<b>146</b>
<b>Appendix 1 .....</b>	<b>162</b>
<b>Appendix 2 .....</b>	<b>166</b>
<b>Appendix 3 .....</b>	<b>184</b>

# List of Figures

Figure 1.1: Fault diagnosis framework scheme using MCSA .....	3
Figure 1.2: A typical fault diagnosis process .....	4
Figure 1.3: Induction motor possible faults. Reproduced from Xue et al [21] .....	5
Figure 1.4: Summary of percentage of each of the failure mode.....	5
Figure 1.5: Examples of BRB fault in case of crack and broken bar.....	6
Figure 1.6: Graphical representation of eccentricity fault .....	8
Figure 1.7: Artificial created bearing fault symptoms .....	9
Figure 1.8: Stator winding faults: phase-to-phase, turn-to-turn, and phase-to-ground [40].....	10
Figure 2.1: Overview of common feature extraction techniques.....	17
Figure 2.2: Classification of different fault diagnosis methods.....	23
Figure 2.3: Architecture of the model based methods .....	26
Figure 3.1: Structure of a typical distributed industrial induction motor network model.....	38
Figure 3.2: Motor-level fault diagnosis framework .....	43
Figure 3.3: ANN architecture at individual motors level.....	48
Figure 3.4: Distributed generic WSN fault diagnosis framework for industrial motor network.....	49
Figure 3.5: Demonstration of multiple sensing points over network to identify the location of multiple motors .....	52
Figure 3.6: Propagation of faulty signal in induction motors network .....	54
Figure 3.7: Frequency propagation for a rotor fault.....	56
Figure 4.1: Per-phase equivalent circuit of induction motor.....	59
Figure 4.2: Network modeling with multiple motors at different power-line buses.....	61
Figure 4.3: Simulation multi-motor network model .....	62
Figure 4.4: Block diagram of simulink model of each motor with fault injection functionality .....	64
Figure 4.5: Fault signal propagation and injection model for two motor in distributed multi- motor networks.....	65
Figure 4.6: Propagation of faulty signal in a distributed motor network.....	67
Figure 4.7. Rotor current comparison of four motors .....	68
Figure 4.8: Rotor speed comparison of four motors .....	68
Figure 4.9: Rotor torque comparison of four motors .....	68
Figure 4.10: Frequency spectrum of targeted motors with no-load.....	69
Figure 4.11: Multi-frequency fault propagation by Motor 3 with full load.....	69
Figure 4.12: Propagation of faulty signal at significant frequencies points.....	69
Figure 4.13: Influence of multi-frequency faulty signal on other motors.....	71
Figure 4.14: Multiple sensing point's analysis over the network .....	72
Figure 4.15: Frequency spectrum measurement at targeted points with fault components .....	73
Figure 4.16: Influence of multi-frequency faulty signal on other motors.....	74

Figure 4.17: Frequencies spectrum results from targeted motors .....	76
Figure 5.1: ANN classification network architecture at motor-level .....	80
Figure 5.2: Neural network simulation model for fault type diagnosis at motor-level...	83
Figure 5.3: Internal architecture of each neural network block .....	84
Figure 5.4: Neural network simulation model for fault type diagnosis at distributed multi-motor network .....	85
Figure 5.5: Overview of the different ANN architecture chosen for fault diagnosis.....	89
Figure 5.6: Performance graphs of identified motors 1,3,6 and 9 using [6x10x4] neural network architecture.....	90
Figure 5.7: Confusion matrices of identified motors1, 3, 6 and 9 using targeted and output classes.....	92
Figure 5.8: Fault indicator proportional possibility calculation to show the presence of BRB fault in Motor 3 .....	93
Figure 5.9: Fault indicator proportional possibility calculation to show the presence of eccentricity fault in Motor 3 .....	93
Figure 6.1: Scaled down test-bed designed at AUT SeNSe laboratory to analyse the signature of motor faults .....	96
Figure 6.2 (a)-(i): Individual FFT spectrum of each electric Motors (1)-(9) without load.....	98
Figure 6.3 (a)-(i): Individual FFT spectrum analysis of each electric Motors (1)-(9) without load.....	100
Figure 6.4: Arduino experimental node .....	101
Figure 6.5: Circuit diagram of current sensor board.....	102
Figure 6.6: Arduino module connectivity using current sensor with motor sensing point .....	102
Figure 6.7: Speed sensor measuring circuit .....	103
Figure 6.8: Motor incremental motion indicator.....	103
Figure 6.9: Comparison between the reference motor speed and sensor measurement	104
Figure 6.10: Schematic layout of Arduino for testing in Proteus ISIS Professional ...	105
Figure 6.11: (a) XBee coordinator API configuration (b) End node device.....	106
Figure 6.12: End node transmission mode.....	107
Figure 6.13: System diagram .....	108
Figure 6.14: Role of coordinator and end node devices within network .....	110
Figure 6.17: Multiple sensing point's analysis using Arduino over the network .....	112
Figure 6.18: Terminal output of Motor 1 .....	113
Figure 7.1: Single faulty motor within network with single fault.....	116
Figure 7.2: Electric current spectrum of all motors in without any load and no fault condition.....	118
Figure 7.4: Multi-frequency fault propagation by Motor 3 with full load.....	120
Figure 7.5: Faulty signal at significant frequencies points .....	120
Figure 7.6: Performance graphs of all motors using [6x10x4] neural network architecture .....	123
Figure 7.7: Confusion matrices of all motors using targeted and output classes .....	124
Figure 7.8: Comparison of motors condition outputs.....	126
Figure 7.9: Multiple faulty motors within same bus with multiple faults.....	126

Figure 7.10: Current spectrum of all motors in with full load in BRB and ECE fault conditions .....	128
Figure 7.11: Multi-frequency fault propagation by Motor 3 and Motor 1 with full load to observe the BRB and Eccentricity faults influence at different Motors .....	128
Figure 7.12: Analysis and observation chart of multi-frequency fault influence by Motor 1 and Motor 3 .....	129
Figure 7.13: Measument of performance graphs of all motors using neural network architecture.....	131
Figure 7.14: Confusion matrices of all motors using tageted and output classes .....	131
Figure 7.15: Comparasion of motors condition outputs.....	133
Figure 7.16: Multiple faulty motors within different bus with multiple faults .....	134
Figure 7.17: Current spectrum of all motors in with full load in BRB and ECE fault conditions in multiple motors .....	135
Figure 7.18: Multi-frequency fault propagation by Motor 1, 5 and 9 with full load to observe the BRB and eccentricity faults influence at different motors according to their distance and size.....	136
Figure 7.19: Analysis and observation chart of multi-frequency fault influence by Motor 1, 5 and Motor 3 .....	136
Figure 7.20: Confusion matrices of all motors using tageted and output classes .....	138
Figure 7.21: Comparasion of motors condition outputs.....	139

# List of Tables

Table 2.1: Comparison between FFT, STFT and CWT.....	22
Table 2.2: Summary comparison of MCSA, vibration and others methods .....	25
Table 2.3: Comparison between artificial neural network and other techniques.....	30
Table 3.1: Possible sensing points within distributed power-line network.....	53
Table 4.1: Parameters used for motors modelling .....	62
Table 4.2: Significant fault frequencies measuring ranges and targeted points.....	70
Table 4.3: Significant fault frequencies at targeted sensing points.....	74
Table 5.1: Description of the implementation ANN.....	84
Table 5.2: Learning data set for training process for multiple motors.....	87
Table 5.3: Best architecture performances for classification.....	87
Table 6.1: Descriptions of measurement Tools.....	97
Table 6.2: Significant fault frequencies at targeted sensing points.....	114
Table 7.13: Significant fault sideband frequencies with amplitude values in all motors for case study 4 .....	121
Table 7.2: Learning data set for training process for multiple motors in case study 4 .	122
Table 7.3: Best performances for classification of all Motors for case study 4.....	125
Table 7.4: Learning data set for training process for multiple motors for case study 5	129
Table 7.5: Best performances for classification for case study 5.....	132
Table 7.6: Learning data set for training process for multiple motors for case study 6	137
Table 7.7: Best performances for classification for case study 6.....	138

# List of Abbreviations

ADC – Analog-to-Digital Converter  
AF – Axial Flow  
AI – Artificial Intelligence  
ANN – Artificial Neural Network  
AVR – Aboriginal Voices Radio  
BPA – Bayesian Pathway Analysis  
BPNN – Back Propagation Neural Network  
BRB - Broken Rotor Bar  
CF – Crest Factor  
CISC – Complex Instruction Set Architecture  
CWT – Continues Wavelet Transform  
DFT – Discrete Fourier Transform  
DS – Dempster Shafer  
ECE – Eccentricity  
EPROM – Erasable Programmable Read-Only Memory  
FFNN – Feed Forward Neural Network  
FFT - Fast Fourier Transform  
Hz – Hertz  
HP – Horse Power  
I – Electric current  
IEEE – Institute of Electrical and Electronics Engineers  
IWSN – Industrial Wireless Sensor Network  
LMA – Levenberg Marquardt Algorithm  
MCSA – Motor Current Signature Analysis  
MCU – Micro-Controller Unit  
MHz – Mega Hertz  
MIPS – Million Instructions per Second  
MLP – Multi-Layer Perceptron  
MSE – Mean Square Error

NN – Neural Network  
PDF – Probability Density Function  
PLC – Power-Line Communication  
RF – Radio Frequency  
RICS – Reduced Instruction Set Architecture  
RMS – Root Mean Square  
RPM – Revolutions per Minutes  
RPS – Reconstructed Phase Spaces  
SCADA – Supervisory Control and Data Acquisition  
SD – Secure Digital  
SSE – Sum of Square Error  
STFT – Short-Time Fourier Transform  
UART – Universal Asynchronous Receiver/Transmitter  
V- Voltage  
VAC – Volts of Alternating Current  
WSN – Wireless Sensor Network  
Z – Impedance

# CHAPTER 1

## INTRODUCTION

### 1.1 Industrial Motor Network Fault Diagnosis

During the last two decades, the field of fault diagnosis of induction motors has attracted great interest. This has helped improve the overall industrial system reliability. Generally, an industrial power-line network consists of different sizes of induction motors from small to large, which can have a considerable combined influence on the overall system's operation [1]. The combined effect of these diverse induction motors can have a strong impact on power-line network permanence.

Various types of induction motor fault are mostly associated with the rotating components and electric drives. Several techniques have been published in the literature [2]-[4] and some solutions are commercially available to perceive the behaviour of induction motors. But some industry engineers are still complaining about consistent, regular, unexpected motor faults. However, some faults are hard to detect at early stages and their symptoms appear only as they accelerate induction and component motor aging.

Various machine operation properties may be used for monitoring the health of a motor, for example, partial discharge, thermos graphic monitoring of hot-spots, chemical content, machine axial leakage flux, acoustics, torque, machine power efficiency, machine vibration signals, and motor current signatures [2]-[3]. Among these, the technique of analysing by machine stator electric current is most well-known, as Motor Current Signature Analysis (MCSA) is the best state-of-the-art technique [4] and the most used technique in industry because:

- It is much easier to simplify the electric current signal
- Remote monitoring is possible without motor access
- Cheap voltage and current sensors are also used for other monitoring purposes

However, apart from these advantages, fault diagnosis through electric current signals provide less reliability, compared with other diagnosis techniques such as vibration monitoring. Because of interference in the noise signals, normal operating conditions of the different motors in the neighbourhood and other sources of industrial site noise. These noises can easily be interrupt into a healthy signal and present as abnormal behaviour in the healthy signal. This may create confusion during the signature analysis process. Therefore, there is high demand from the industrial sector to provide robust solutions for motor fault diagnosis using electric current signal to avoid this type of confusion in decision making.

## **1.2 Fault Diagnosis using Motor Current Signature Analysis Method**

Voltage and electric current in the industrial motors network are mostly used parameters for fault symptoms indication. Electric current is a characteristic of the voltage operation in induction motors. However, the voltage of the motor terminal is a function of its supply voltage and represents the structure of power-line network. Therefore, voltage alone cannot be considered as indicative of the performance of an induction motor. Also, the supplied voltage is reliant on motor network architecture and its generated voltage. However, the combination of the voltage and current can be considered an effective method for diagnosis.

MCSA is a simpler and more successful method that is proposed in the literature [1–5] for induction motor faults diagnosis. This method utilises a pattern recognition strategy of electric current signal of induction motors to estimate the presence of pre-recorded faults signatures. Recently, a number of studies reported on industrial motor faults and claim MCSA as an efficient diagnosis technique. But the main challenge with MCSA is interference among the components of industrial power-line network. Some methodologies have been recommended to improve reliability and decrease the chances of uncertainty in decision making. However, the interference of neighbourhood motors may cause of frequency spectrum similar to the pattern of suspected faults. Furthermore, for most low-power components of electric motors, these approaches are too expensive. A review survey suggests that for many low, and medium, power electric motors, a single monitoring system is not feasible in terms of cost [5].

A top-down framework of fault diagnosis using MCSA that consists of different processes is shown in Figure 2.1.

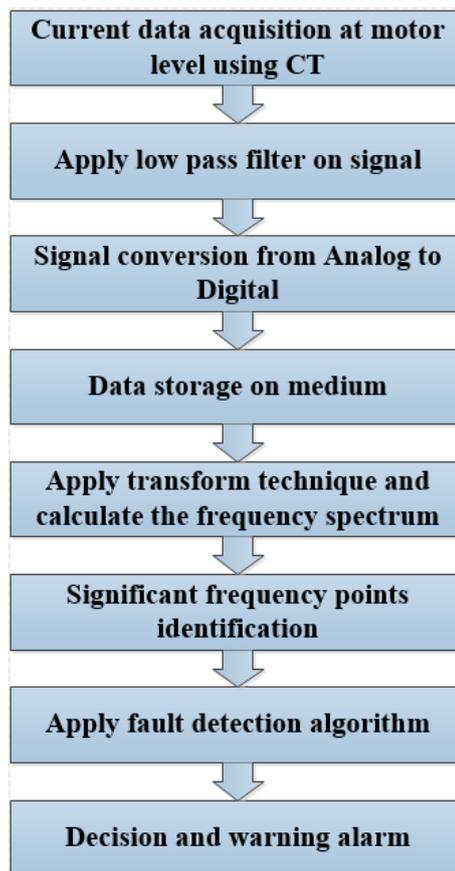


Figure 1.1: Fault diagnosis framework scheme using MCSA

As shown in Figure 1.1, different processes are involved in data acquisition, applying filters, signal conversion, applying transform techniques and fault diagnosis. For data acquisition, there are several components available. But mostly, the signal should be converted and sampled through a voltage transformer or electric current sensors. Then the 50 Hz frequency is considered to include the normal operation of induction motors. The next step is to eliminate the high frequency characteristics component of the electric signal that are not significant for fault diagnosis. After applying the low pass filter, one can convert the analog signal into a digital signal by calculating the frequency spectrum of waveform signal. Then a significant frequency point is categorised by using a data processing transformation approach. Finally, a fault diagnosis algorithm is needed to detect the fault and a post-processing system will be required to make a decision about the health of the motor generating the warning alarm.

A typical condition monitoring and fault diagnosis/prognosis process usually consists of four phases, that is, data acquisition, feature extraction, diagnosis/prognosis and decision making, as shown in Figure 1.2.

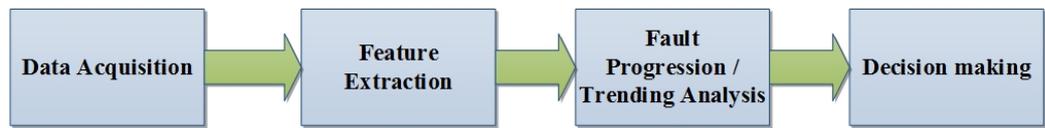


Figure 1.2: A typical fault diagnosis process

### 1.3 Possible Induction Motor Faults

Induction motor faults may have only minor symptoms, resulting in minor inefficiency and hence higher energy consumption, poorer performance and long term machinery shutdown. Even minor faults can also cause effects such as reduced efficiency, increasing temperature, which reduces insulation lifetime, and increasing vibration, which may reduce bearing lifetime [1]. These are caused by the operating environment condition and the machine's internal factors. Therefore, the diagnosis of induction motor problems is important and prevents expensive maintenance costs.

Many methods of induction motor fault diagnosis have been developed in the last few decades and many methods have been presented [2]-[5]. The most common technique used for fault diagnosis is motor current signature analysis (MCSA) [3]. Numerous induction motor faults identification and detection methods are based on Fast Fourier Transform (FFT) signature analysis [6],[7]. Additional techniques include temperature measurements, vibration analysis, vibration monitoring [10], spectral analysis of speed fluctuations [8],[9], state and parameters estimation [11], air-gap torque analysis [12], acoustic noise measurement, and magnetic field analysis [13],[14]. Currently, knowledge level techniques based on Artificial Intelligence (AI) have been utilised for diagnostic induction motor faults, such as fuzzy logic [15],[16], Neural Network [17], genetic algorithms [18], Bayesian classifiers and Dempster Shafer (DS) [19] in the three-phase stator electric current, with AI based on Reconstructed Phase Spaces (RPSs) and Gaussian mixture models [20].

Induction machines are generally symmetrical. Faults in the machine normally affect its symmetry and can produce the following possible symptoms: increased torque pulsation, unbalanced air gap voltages and line currents, higher losses and reduced efficiency, reduced shaft torque, and increased space harmonics [3]. There are four key branches of failure mode for induction motors in the industrial environment [21]. Figure 1.3 shows possible existing faults and their related symptoms:

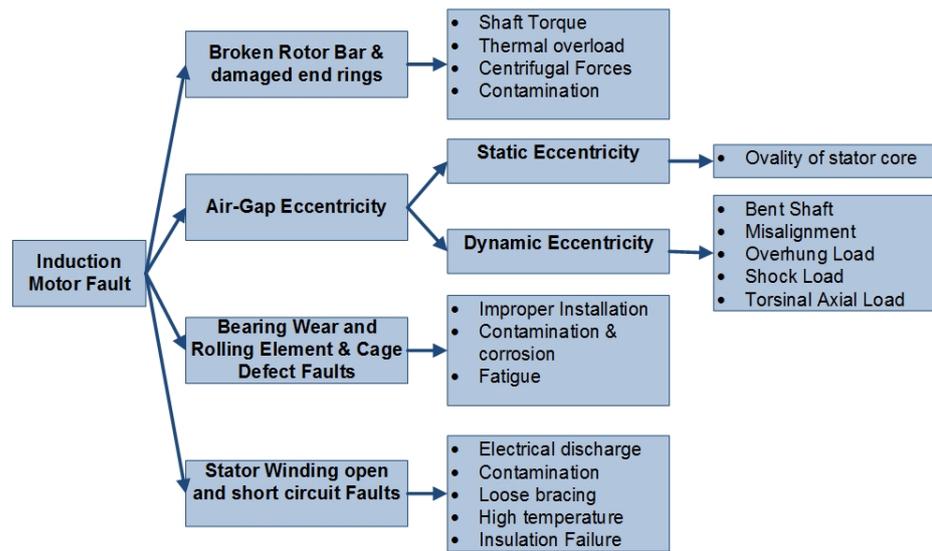


Figure 1.3: Induction motor possible faults.

Reproduced from Xue et al [21]

A detailed comparison survey is comprehensively presented in [1]. This comprehensive survey has been collected from several sources, including around 80 journal papers published in IEEE and other reputed electrical engineering journals. The Figure 1.4 presents a summary of the survey results.

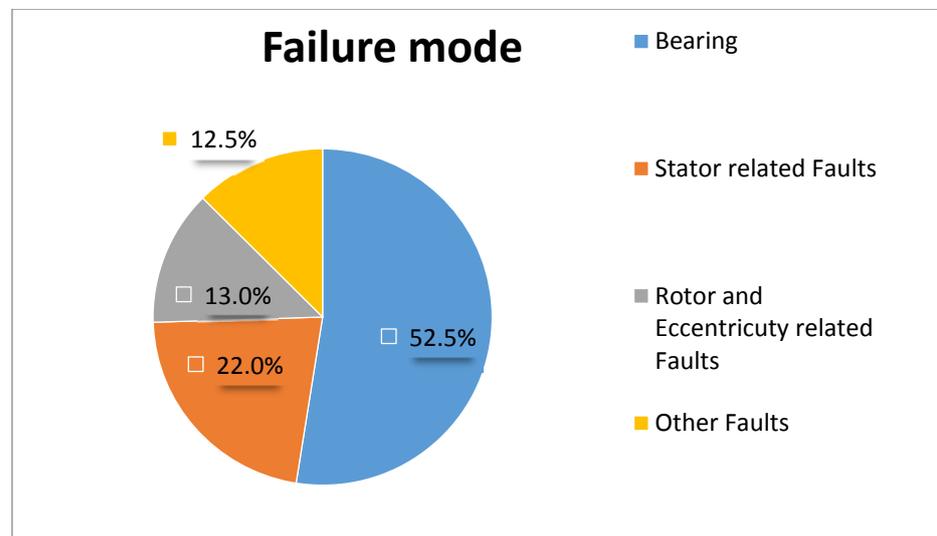


Figure 1.4: Summary of percentage of each of the failure modes

Most failure modes are due to bearing related faults (52.5%). If the bearing fault is combined with the stator faults, this together accounts for more than 87.5 percent of total existing faults. But bearing, related faults are normally due to mechanical and vibration problems. So this research focuses only on rotor and air gap eccentricity, and related faults because rotor related faults are the easiest to

detect in induction motors. This can be done through MCSA on the associated double slip frequency sidebands in the current spectrum of the fundamental supply frequency.

### 1.3.1 Rotor Bar Broken and End Ring Faults

The rotor bar problem has a high percentage share in unscheduled maintenance. In case of rotor bar faults, this problem is confirmed through vibrations and MCSA [19]. Broken rotor bars (BRB) or end-rings can be caused by frequent straight on-line motor starting (as the induction motor cage winding may not resist high electrical and mechanical stresses), by applying mechanical loads and by failure in the manufacturing procedure of the induction motor rotor cage [22]. BRB may not cause immediate failure of an induction motor. However, if the rotor bar pieces of the broken bars move and strike the stator end winding, then this can cause serious damage or injuries. The following, Figure 1.5, gives an example of cracked and broken bar faults from the literature [17]-[24].

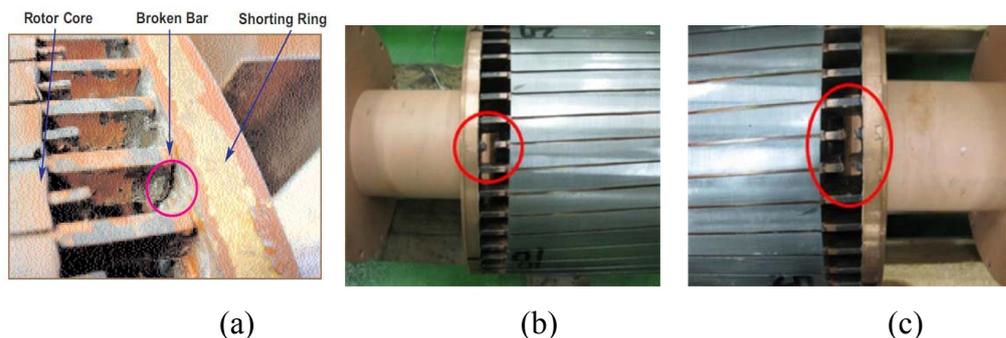


Figure 1.5: Examples of BRB fault in case of cracked and broken bars  
(a) Crack [17] (b) One BRB and (c) Two BRBs [24]

BRB have been at the centre of attention in most industries concerned with the multiple typical faults in induction motors, because the induction motor rotor itself is usually quite costly. According to induction motor philosophy, BRB fault increase the sidebands components values around the fundamental supply frequency with different amplitude values. Different authors in recent years, such as Krishna [25] Yong [26] Fernandez [27] Gyftakis [28] Bindu [29] and Eren [30], used main power-line network electric current spectrum analysis on isolated motors to detect BRB fault. They considered the sideband components nearest the fundamental supply frequency at

$$f_{brb} = f_1(1 \pm 2s)Hz \quad (1.1)$$

So, to calculate the slip, synchronised speed can be formulated from following formula:

$$Synchronise\ d\_Speed = S_{rpm} = \frac{120 \times f_1}{P} \quad (1.2)$$

$$Slip = S = \frac{n_s - n_r}{n_s} \times 100\% \quad (1.3)$$

Where  $f_1$  = supply frequency (Hz); and  $s$  = per unit slip;  $n_s$  = synchronous speed in revolutions per minute (RPM);  $n_r$  = motor speed in RPM;  $n$  = total number of samples;

Gyftakis et al [28] showed the BRB actually increases towards a sequence of multiple broken bar sideband components which can be formatted as:

$$f_{brb} = f_1(1 \pm 2ks)Hz \quad (1.4)$$

Where  $k = 1, 2, 3, \dots$  (harmonics) of number of broken bars.

The left side (lower sideband component) is distinctive due to the broken bars, and the right side (upper sideband component) is specific because of the resultant motor speed fluctuation [28].

In the case of rotor cracks that happen in the bars, the cracked rotor bars lead to overheating. Due to this change, the bars can be cracked or broken. Therefore, other adjacent bars will transfer a higher level of current and be subjected to a large mechanical and thermal stress on the motor [27]. This stress may cause the crack to start. So most of the electric current that flowed in the cracked/broken bar, will now transfer into the adjacent two rotor bars. This large mechanical/thermal stress may also damage a rotor and its lamination. It is also one of the causes of temperature increase in motors. The temperature distribution across the lamination can also change because of rotor asymmetry. The cracking of the bar can present at several locations, including the slot portion of the bars under consideration and the end rings of bar joints. The possibility of cracking in the end-rings of bar joints is greatest when the start-up time of the machine is long and when frequent starts are required [25].

### 1.3.2 Air-Gap Eccentricity Fault

An air-gap eccentricity fault is where an air-gap comes between the rotor and stator and causes imbalance. This fault produces vibration and noise in induction motors. When eccentricity increase in motors, the resulting unbalanced

radial force can develop contact with the rotor and stator and damage them [31]. Typically, the rotor is centrally aligned, with the stator bore in healthy condition. The rotation of the rotor is aligned in the center, the same as the linear position at the center of the stator [32]. When the rotor is not properly centrally aligned, the unbalanced rotor radial forces can produce a stator-to-rotor rub, which can damage them [33].

Figure 1.6 shows a typical air-gap eccentricity fault developing from a healthy, to static and dynamic condition [34].

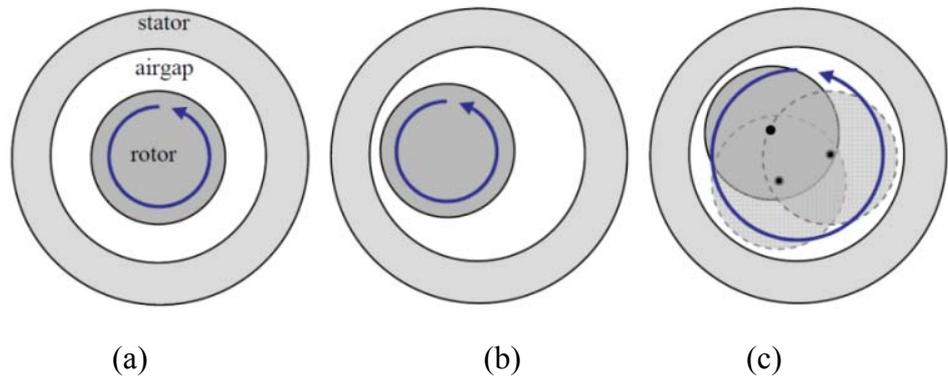


Figure 1.6: Graphical representation of eccentricity faults

(a) No eccentricity (b) Static eccentricity (c) Dynamic eccentricity [34]

Static air-gap eccentricity happens when the angular position of the rotor is fixed in a space with the lowest radial air-gap length. It does not change over time. It is only space dependent and can be caused by the elliptical stator core, or wrong position of the rotor and stator. In rotating electrical machines there is always an inherent level of static eccentricity due to manufacturing tolerances caused by the compound assembly of parts that each have their own tolerances.

Dynamic eccentricity happens when the angular position of the rotor is not fixed in space, with minimum radial air-gap length and change with time. It can be caused by a misalignment or bent shaft, thermal bowing of the rotor, bearing wear and movement, or a non-concentric rotor outer diameter.

The high frequency components of air-gap eccentricity are formulated by Thomson in [35]:

$$f_{ecc} = f_1 \left[ \left( \frac{R}{p} \right) (1-s) \pm k \right] \quad (1.5)$$

Where,  $R$  = number of rotor slots;  $p$  = number of poles-pairs;  $k = 1, 3, 5, \dots$  (harmonics).

And the low frequency components of air-gap eccentricity are formulated by Nandi in [36]:

$$f_{ece} = f_1 \pm kfr \quad (1.6)$$

Where  $f_r$  = rotational speed frequency

In a real motor system if the level of air-gap eccentricity is not retained within definite limits, it can significantly increase bearing wear and also increases noise levels.

### 1.3.3 Bearing Wear and Rolling Element & Cage Defect Faults

Bearing related faults are of practicable significance in the machinery industry. Fatigue failures can take place even under regular operating conditions with good alignment. These faults may increase noise and vibration levels [37]. Other than normal operational stresses, bearings can be affected by external interference such as contamination, improper lubrication and improper installation. Bearings consist of inner and outer rings that consist of a set of balls or roller elements. These balls are placed in raceways that between rotate these rings. Continuous stress on the bearings will lead to fatigue failures in the inner and outer rings. Shaft electric currents and voltages are also caused by bearing and rolling failures.

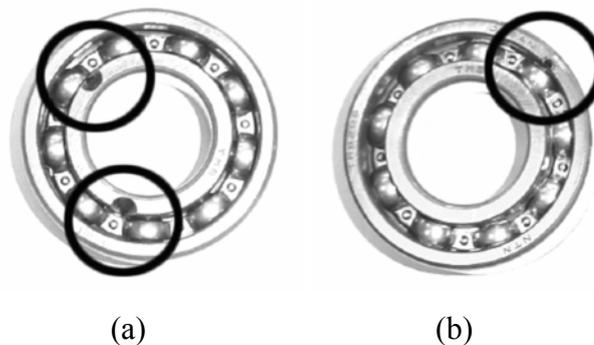


Figure 1.7: Artificially created bearing faults symptoms

(a) Inner-race defect (b) Outer-race defect [37].

Under the uniform functioning conditions of good alignment and balanced load, the fatigue failure usually begins with minor cracks or fissures located on the inner and outer bearing surfaces of the rolling elements and raceway. These cracks progressively increase on the surface and generate detectable vibrations or noise levels.

### 1.3.4 Stator Winding Open and Short Circuit Faults

Stator turn faults in balanced industrial machines causes a great propagating electric current in power-line network and then produces unnecessary heat, high voltage, vibration and other mechanical forces in the shorted turns. The major cause of stator faults is insulation failure due to stress. These stresses can perform individually or together to damage the insulation material. Many authors [38]-[39] have reviewed the literature on the root causes of stator related faults and failure modes in insulation systems and conducting components. Normally, stator winding-related failures can be categorised into three types: phase-to-phase, phase-to-ground and turn-to-turn. Figure (1.8) represents the stator windings of an induction motor and its possible types of stator fault.

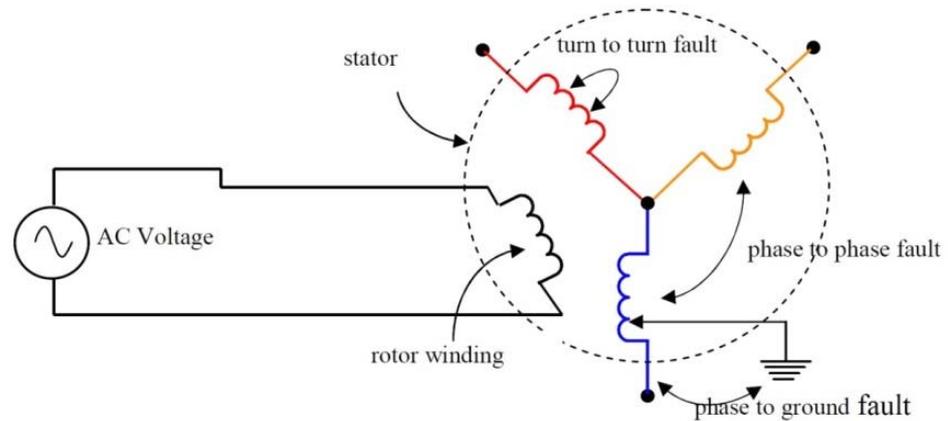


Figure 1.8: Stator winding faults: phase-to-phase, turn-to-turn, and phase-to-ground [40].

In above-mentioned three types, the turn-to-turn faults (or stator turn faults) are considered the utmost challenging task, since the other two types of stator fault are generally related to turn faults. Furthermore, turn-to-turn faults are very difficult to diagnose at early stages. To resolve the complexity and difficulty in detecting turn faults, numerous techniques have been proposed by different researchers using MCSA, which will be discussed in the next section.

## 1.4 Motivation and Problem Statement

Generally, an industrial power-line network consists of different sizes of induction motor from small to large, which can have a considerable combined influence on the overall system's operation. The combined effect from all these induction motors can have a strong influence impact on power-line network. In

the power system, induction motor load torque is considered part of the total power system load. An important concern for industrial power engineers is to observe the performance of the induction motors within a power-line network to ensure their smooth working and maintain the whole system's durability [1]. The problem is generally the difficulty in measuring and precisely correlating machine faults and the source of fault generation within the network of motors. However, if the motors present their feature characteristics dynamically, the combined effect on the power-line network performance can be more clearly visualised. This may assist in the measurement of performance.

Available motor fault detection approaches are effectively tested on equipment in separated systems. In a real industrial scenario, multiple induction motors are connected to same power bus and operate in groups for control of a given industrial operation. They normally share current and voltage from common terminal points and would easily influence other neighbouring motors. That means if a signal is picked up at any point in a distributed power-line network it would contain information about the motor itself and other motor components as well. But once the equipment is part of a system, fault signatures get corrupted due to interference and noise within the system. This happens through electrical and mechanical connectivity among the components that facilitate smooth paths for signals to travel throughout the system. Different equipment within the system can observe the unknown motor behaviour, which is operating individually outside the system network without sensors. This will result in a significant amount of interference between motors, hence a decrease in the chances of identification and localisation of the fault source within the power-line network system.

The purpose of this research is to develop a network-level data processing algorithm able to analyse motor faults at different test points within a power-line network. Every testing point demonstrates its views on the possible fault symptoms at different physical locations within the bus. These will then be managed to identify the fault type and physical location of the source of fault within the multi-motor network. Individual single sensing point decision-making normally causes serious failure in identification of fault type and localization of the origin of a fault signal within a distributed power-line network. But multiple measuring diagnosis points within a distributed power-line network can improve the accuracy of fault diagnosis, due to capturing more extensive data.

Understanding the way signals are influenced as they travel through the various types of media within a system may significantly help in easing this problem. Using different system parameters available within the industrial system, wide coverage for fault diagnosis could be managed, to identify equipment status and make a binary decision about faulty equipment. For that, a deep understanding is required of the infrastructure of distributed industrial power-line network. Any industrial site is a combination of different types of groups of equipment. These groups contain non-linear and static load with various sizes of electric equipment. Equipment in these groups are inter-connected to other equipment through cables and electrical connections. Also most motors are equipped with current and voltage measuring points. Faulty signals can travel through the main power-line from one bus to another. Therefore, faulty signals may be detected in the system wherever a physical connection generates a path between sources of fault signals and various measuring points.

Currently, most available sensors for machine fault diagnosis physically wired. These wires take electrical power from the main power-line and provide communication paths. However, in most applications, wired sensors are inconvenient or impractical. For example, sensors are difficult to mount on the surface of motors where the motors are producing vibration and high temperatures. Fault diagnosis techniques are continuously arriving at different solutions to make best use of the latest industrial scientific advances with portable data loggers and analysers of online wired monitoring systems [41]. SCADA is one popular solution for fault diagnosis in industrial machinery. In the industrial sector, online wired fault diagnosis and monitoring systems are currently successfully deployed to monitor critical machinery and process feature extraction. However, there are other noncritical machinery that is not monitored on regular basis. When combined, these two types of motor in a wired monitoring system, may considerably increase the cost of monitoring the system because of installation of extra cables. Sometimes the installation cost is much higher than the cost of the sensor itself, especially in the case of remote monitoring.

In this situation, the Wireless Sensor Network (WSN) provides a potential solution to deal with these challenges. Compared with industrial wired monitoring systems, WSN has many beneficial characteristics such as being easy to install, relatively low cost and easy to relocate. The WSN raises many issues that need to be addressed carefully with experimentation. The superiority of wireless

communication is based on experiment test-beds, data acquisition, modulations schemes and sensors. Wireless sensor reception may change with insignificant spatial displacement and vary over time. Most available sensor platforms use low-power radios with less frequency diversity to reject the multi-path elimination. Furthermore, when a signal transmits from the transmitter inside an induction motor, the motor operating noise may easily be inserted into the wireless signal and may change the behaviour of different characteristics. Therefore, this research focuses on developing a special, efficient filter to identify and extract unnecessary signals from real signals. Furthermore, this research is also focused on developing a wireless node level feature extraction technique for data fusion, using MCSA at the end node-level and decision-level fusion, implemented at the node coordinator for efficient fault diagnosis.

## **1.5 Contribution and Research Objectives**

This research takes advantage of multiple fields of study in industrial fault diagnosis to formulate a concept of fault signal propagation within power-line networks and manifest faulty signals into healthy signal while they are travelling in a scaled-down distributed power-line network. In this thesis, a systematic method has been employed to estimate the influence of faulty propagated signals (electric current signals) on main power-line network. It shows how it changes the behavioural characteristics of motor features in the electric current of in-network machines. Analysis estimating attenuation factors of the signals propagated was carried out to evaluate the possible route of different signals over a network. This analysis is presented here to estimate the source of faults on multiple path routes in a power-line and anticipated fault representation around the power-line network.

Different practical experiments have been carried out to take advantage of several point observations, so as to diagnose fault types and causes of fault signal. This activity supports the importance of multiple point observation and improves the accuracy of traditional methods in MCSA. Additionally, this practice provides a better and more efficient monitoring mechanism to observe the behaviour of electrical components at industrial sites. Further, the utilisation of signal processing technique was formulated for the transformation of spectrums and correlates with the targeted fault symptoms. A supervised knowledge-level approach is presented to identify multiple faults as well as the location of fault

events within an industrial motor network. Finally, a typical example of an industrial motor network was modelled and implemented in the AUT SeNSE electrical power laboratory to prove the concept of propagation of faulty signals over a power-line network.

The main objective of this research is to investigate the sensor network solution for the distributed motor signature analysis approach. This will be approached incrementally through the following activities:

- Development of a BRB and air-gap eccentricity motor faults library using existing fault frequency characteristics models
- Simulate the propagation of fault signals within electrical power-line network to prove the concept
- Simulation model of multiple motors connected as distributed motor network
- Development of an intelligent sensor-level data fusion method that acts as an environment for fault signatures acquisition, data fusion and fault diagnosis
- Developed and demonstrate a wireless end node network to improve the consistency and reliability of fault diagnosis in an industrial multi-motor network environment

## 1.6 Thesis Organization

The thesis is organised into the following eight chapters.

**Chapter 1** describes background, motivation, problem statements and research objectives.

**Chapter 2** presents a state-of-art existing literature review on available signal processing techniques, diagnosis methods, propagation of fault signals in an industrial power-line network, industrial fault diagnosis using wireless sensor networks, and shortcomings in existing fault diagnosis research.

**Chapter 3** describes the theoretical concept formulation of the distributed fault type diagnosis, signal propagation and data fusion approach developed in this thesis.

**Chapter 4** presents the simulation model of multiple motors connected as distributed motor network. Propagation of faulty signals over the network of

multiple motors have been represented to reflect the significant influence on healthy motors.

**Chapter 5** demonstrates an efficient simulation model of multiple fault diagnosis using neural network at each motor and distributed level modelling.

**Chapter 6** presents a typical experimental test-bed environment. The specific experimental setup, tools and resources offers good representation of real-time industrial case studies, to prove the actuality and draw conclusions from experience gained. WSN architecture has been presented to discuss the role of coordinator and end node within network and describe the step by step activities of data fusion, transformation, packet structuring and transmission to base station.

**Chapter 7** focuses on evaluation and implementation of the distributed diagnosis concepts based on simulation and experimental results. This chapter consists of different case studies and simulation results based on different fault indices, propagations of fault signal over power-line network and ANN to employing a smart diagnosis and decision making approaches in interpreting the possible fault indices.

**Chapter 8** outlines the findings summary, conclusions, and recommendations for further work in the area distributed fault diagnosis.

## RESEARCH BACKGROUND AND LITERATURE REVIEW

### 2.1 Introduction

This chapter presents a critical literature review analysis on machine-fault diagnosis and related topics. The review covers a wide range of recent literature in the problem domain and is classified into the following groups:

- Existing signal processing techniques
- Feature extraction methods
- Existing fault diagnosis methods
- Propagation of fault signals in industrial power-line network
- Fault type diagnosis using Wireless Sensor Networks (WSN) within industrial machinery networks
- Identified shortcomings in electric current fault diagnosis research

A significant related contribution, and development of these areas fault diagnosis and traceability within power-line networks will be discussed in detail throughout this chapter.

### 2.2 Signal Processing Techniques

In order to collect useful data from targeted physical assets, various fault diagnosis techniques are used in real environments. Machine condition monitoring data includes vibration, electric current, temperature, and pressure or environment data.

There are more studies on isolated machine fault diagnosis [42]-[47] than multiple motors signal fault diagnosis [48]-[49]. Raw data acquired from sensors were pre-processed before being used for further analysis. Errors caused by background noise, human factors and sensor faults need to be eliminated and appropriate features need to be calculated, selected and/or extracted for further fault diagnosis. Once a number of features are obtained, feature-selection methods

need to be employed to identify the most effective features to facilitate the fault diagnosis process.

### 2.2.1 Feature Extraction Techniques

For accurate Fault diagnosis, data must be turned into information before knowledge can be acquired. To turn waveform data into information, fault condition indicators (features) are extracted and/or selected from the acquired signals. Reliable features generally have the following characteristics [50]:

- Inexpensive computational measurement
- Understandable in physical terms
- Mathematically properly definable
- Insensitive to unnecessary variables
- Uncorrelated with other domain features

After acquiring the spectrum data, different types of signal processing methods have been utilised to extract useful feature information and interpret signal waveform data for further fault diagnosis purposes in motors. Most feature extraction techniques can be divided into three groups, as shown in Figure 2.1:

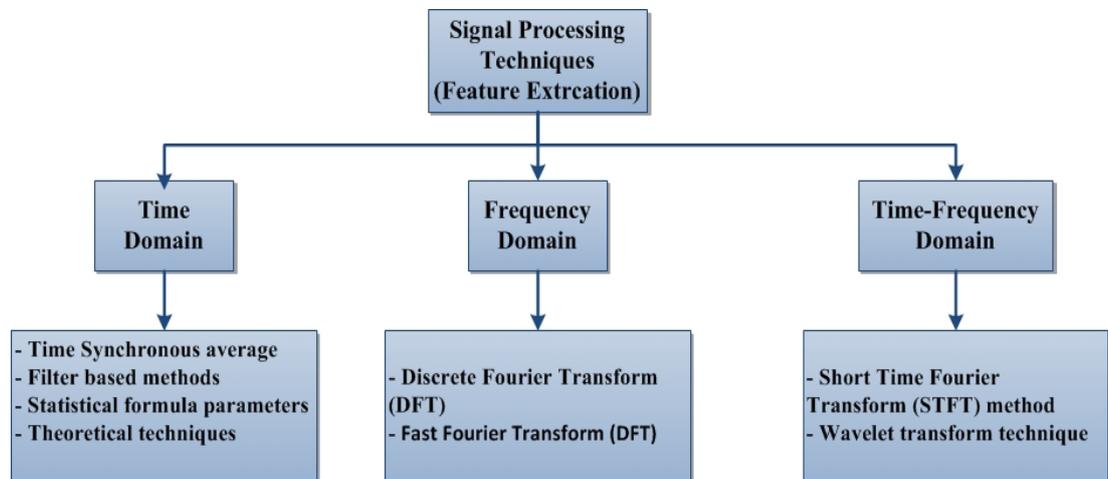


Figure 2.1: Overview of common feature extraction techniques

#### 2.2.1.1 Time-Domain Feature

Time-domain methods are based on the statistically characteristic behaviours of the waveform signal in time. The most prominent and simplest features in time-domain analysis are Root Mean Square (RMS) and Crest Factor (CF) of the signal. Other most frequently used features are variance, kurtosis, standard deviation and skewness. These features are based on the distribution of

signal samples with time series random variables also called moments or cumulate. In most constituent moments, the Probability Density Function (PDF) can be broken down into parts, because any change in the signal could alter the behaviour of the PDF and would change the cumulate. Therefore, observing this circumstance can provide useful diagnostic information.

Some other time-domain feature extraction techniques discussed in [51]-[53] include demodulation and adaptive noise cancelling, filter-based and stochastic techniques. One of the shortcomings of the time-domain feature extraction technique is a lack of visible symptoms of faults particularly when a fault is at an early phase. The technique may be useful when short-duration features are extracted from the signal.

RMS is one of the most significant time domain features and is very efficient in distinguishing any imbalance, related fault in industrial rotating equipment. However, it cannot normally identify explicit failing components. It is also not sensitive enough to detect incipient machinery fault [54]. RMS is the measure of the power content of a waveform and can be expressed as follows:

$$RMS = X_{rms} = \sqrt{\frac{\sum_{n=1}^N f^2 \cdot S(n)}{\sum_{n=1}^N S(n)}} \quad (2.1)$$

Where,  $f$ =frequency value; and  $S(n)$ =spectrum for  $n^{\text{th}}$  sample. i.e.  $n=1,2,3,\dots,n$ .

Crest Factor (CF) is expressed as a percentage of the peak-level of an input signal to RMS level. However, signal peaks in the time domain lead to change in the crest-factor value dynamically. CF can be useful for the detection of impulse vibration changes. CF is defined as:

$$CrestFactor = CF = \frac{Amp_{max}}{X_{rms}} \quad (2.2)$$

Where,  $Amp_{max}$  = Maximum value of amplitude in signal;

Other significant time-domain features can be calculated by the following equations:

$$Kurtosis = kur = \frac{\frac{1}{n}(X_{rms})^4}{X_{rms}} \quad (2.3)$$

$$\text{Variance} = \sigma^2 = \frac{1}{2} P(X_{rms})^2 \quad (2.4)$$

$$\text{Skewness} = ske = \frac{P(X_{rms})^3}{X_{rms}^3} \quad (2.5)$$

Where  $P$  indicates the expected value of the function.

An approach using NN has been developed and considers signal vibrations to be input features [55]. They use genetic algorithm [55] to extract the most considerable input features for fault diagnosis contexts. When doing this, six input features are selected from a large set of possible available features. The authors in [55] claim the major disadvantage of cost for vibration monitoring requires access to the machine.

### 2.2.1.2 Frequency-Domain Feature

Frequency-domain features are capable of overcoming the weaknesses of time-domain analysis. Frequency-domain methods are based on the information that a localised fault produces by a periodic waveform signal, along with distinctive frequency points and features.

When frequency-domain features are used for fault symptom detection, some changes in frequency-domain parameters may indicate the existence of faults, because diverse faults have different spectrums in the frequency-domain. Frequency-domain parameters can be also used for early detection of machine faults and failures [43]. Therefore, such indices can be used to perform fault diagnostics processes.

Fast Fourier Transform (FFT) is one of the most commonly used techniques in the frequency domain. The FFT, which is a fast algorithm for Discrete Fourier Transform (DFT), can easily transform a signal into the frequency domain. If it is difficult to analyse a signal in the time domain it is easier to transform and analyse it in the frequency domain.

To enhance the results of spectrum analysis, several types of frequency filter, side-band structure analysis, demodulation and descriptive representation methods are often used [36]. Different types of frequency spectra, such as power spectrum and high-order spectrum, have been developed. The most traditional way of producing a power spectrum is by using a DFT, but some additional methods can also be used, such as the maximum entropy technique. The following

parameters in the frequency domain are commonly used as fault indicators for diagnostics [56].

$$\text{sideband\_peakvalue} = Pv = \max|i(j)| \quad (2.6)$$

$$\text{Frequency\_Center} = FC = \frac{\sum_{n=1}^N f \cdot S(n)}{\sum_{n=1}^N S(n)} \quad (2.7)$$

$$\text{Mean Square Frequency} = \text{MSF} = \frac{\int_0^{+\infty} f^2 p(f) \partial f}{\int_0^{+\infty} p(f)} \quad (2.8)$$

$$\text{Root Mean Square Frequency} = \text{RMSF} = \sqrt{\frac{\int_0^{+\infty} f^2 p(f) \partial f}{\int_0^{+\infty} p(f)}} \quad (2.9)$$

$$\text{Standard deviation Frequency} = \text{SDF} = \sqrt{\frac{\sum_{n=1}^n (f - f_c)^2 S(n)}{\sum_{n=1}^n S(n)}} \quad (2.10)$$

Where,  $f$  represents frequency value in cycles per second (Hz);

$P$  = number of poles;

$n_s$  = synchronous speed in revolutions per minute (RPM);

$n_r$  = motor speed in RPM;

$n$  = total number of samples;

$Amp_{\max}$  = Maximum value of amplitude in signal;

$i(j)$  = series of signals for  $j = 1, 2, 3, \dots, N$  and  $N$  is the number of data points in the signal.

And  $s(n)$  = spectrum for  $n^{\text{th}}$  samples. i.e.  $n = 1, 2, 3, \dots, n$ .

Pineda et al. present the measurement of the instantaneous supply frequency for the diagnosis of two electric machines with rotor asymmetries [57]. The technique of instantaneous frequency is used based on the extraction of fault components (RMS, crest factor etc.) which associated with the frequency side bands and the assessment of the instantaneous supply frequency. Furthermore, in case of failure this technique, the neural network is described in [58]-[59] to solve the rotor asymmetries related faults.

Furthermore, Guasp et al. [60] extends the previously introduced methodology for the detection of rotor asymmetries and eccentricities for the detection of double-faults rotor asymmetry and eccentricity. They used slip and speed as frequency domain features on a single isolated motor for fault diagnosis. But in this research, other features are not considered that shows some sidebands related to other faults.

Moreover, some work introduced in [61] that are quite similar, one being a continuation of the other. They performed a thorough study of the temporal evolution of a lateral side-fault component on start-up, this being a pattern justified physically, including the evolution in amplitude value and frequency that ends with the proposition of a method for its simulation.

The technique proposed in [62] is structured on the extraction of the side band fault component, and its comparison with the simulated pattern computed in the previous study [60]. It also introduces a method for the quantification of the fault, dividing the spectrum of the fault component of the start-up signal. Liggins et al. [63], introduced a systematised methodology and extended theoretically to any type of induction machine fault, in which its fault components are a function of the slip, providing a practical guide for the application of the methodology.

### **2.2.1.3 Time Frequency Domain**

Time-frequency methods have the ability to describe machinery fault signatures in both time and frequency domains when the signal is non-stationary [64]. The traditional time-frequency technique uses the time and frequency distributions that signify the energy of the signal in two dimensions.

Short-Time Fourier Transform (STFT) is the most commonly used distribution technique when the signal is in a non-stationary state [65]. STFT is an enhanced form of Fourier Transform (FT). In this technique, the target signal is converted into small windows. After choosing the width of the window function, this is multiplied and shifted with the signal segment to produce concise non-stationary signals. Based on the same procedure, FT is then applied at each segment to obtain the STFT of the signal. This shows the changing behaviour of the frequency spectrum with time value. STFT gives a constant resolution at all necessary frequency points.

Another new time-frequency domain technique is wavelet transform, overcome the shortcomings of STFT. This technique is also used to analyse the signal in a non-stationary state with time values. Wavelet transform, provides a multi-resolution at different frequency levels.

A comparison of FFT, STFT and Continues Wavelet Transform (CWT) methods [66] - [69] is summarised in following Table 2.1 following:

Table 2.1: Comparison between FFT, STFT and CWT

<b>Techniques</b>	<b>Faults diagnosed</b>	<b>Advantages</b>	<b>Disadvantages</b>
<b>FFT</b> [66]	<ul style="list-style-type: none"> <li>• Broken rotor bar fault</li> <li>• Short winding fault</li> <li>• Air gap eccentricity</li> <li>• Bearing faults</li> </ul>	<ul style="list-style-type: none"> <li>• Suitable for high load conditions</li> <li>• Easy to implement</li> <li>• Good for visualisation fault symptoms</li> </ul>	<ul style="list-style-type: none"> <li>• Lost time information</li> <li>• Not effective in light load condition</li> </ul>
<b>STFT</b> [67]	<ul style="list-style-type: none"> <li>• Broken rotor bar Fault</li> <li>• Bearing faults</li> </ul>	<ul style="list-style-type: none"> <li>• Fast speed</li> <li>• Suitable for varying load conditions</li> </ul>	<ul style="list-style-type: none"> <li>• Analyse signal with fixed sized window</li> <li>• Poor frequency resolution</li> </ul>
<b>Wavelet Transform</b> [68][69]	<ul style="list-style-type: none"> <li>• Broken rotor bar fault</li> <li>• Short winding fault</li> <li>• Bearing faults</li> <li>• Load fault</li> </ul>	<ul style="list-style-type: none"> <li>• Fast speed</li> <li>• Suitable for varying load and light load conditions</li> <li>• Excellent low time and frequency resolution for Low-frequency sideband components</li> </ul>	<ul style="list-style-type: none"> <li>• Absence of phase information for a complex-value signal</li> <li>• Poor directionality</li> <li>• Shift sensitive for input-signal, causes an unpredictable change in transform coefficients in time.</li> </ul>

## 2.3 Fault Diagnostic Methods

Different fault diagnosis techniques have been applied for single and multiple fault diagnosis in industrial machinery systems. The four main types are signal-based, model-based, knowledge-based and hybrid methods [68]. Further classifications of these methods are presented in Figure 2.2.

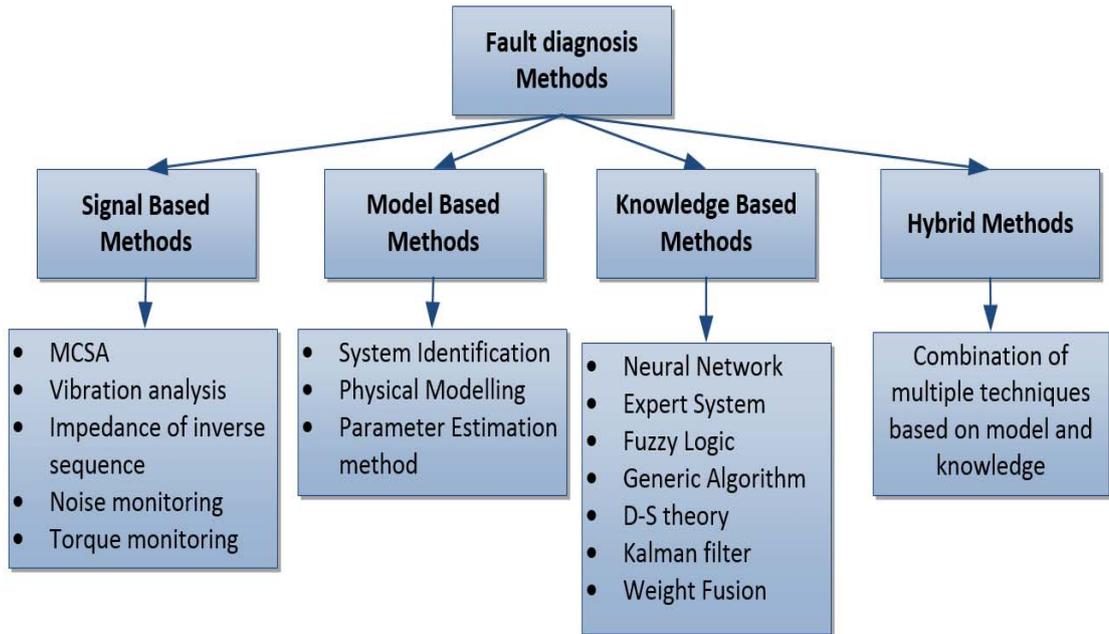


Figure 2.2: Classification of different fault diagnosis methods

### 2.3.1 Signal-Based Methods

Signal-based methods are largely dependent on signal processing methods for fault diagnosis. Usually, these techniques require pre-identified circumferences. Signals are dependent on features. Once the signal or features pass outside their boundaries, an abnormal situation may be happening [70]. There are many methods available that are based on signal analysis, such as vibration analysis, MCSA, Axial Flow (AF), torque analysis, noise monitoring and impedance of inverse sequences.

Most mechanical faults in high speed rotating machines lead to increase in vibration levels. The largest sources of vibration and noise in electric machines are the radial forces due to the air-gap field. Vibration monitoring is an effective and efficient approach to providing condition indicators for machine health management [71]. Vibration-based diagnostics is the best method for fault diagnosis, but needs expensive accelerometers and associated wiring. This limits its use in several applications, especially in small machines where cost plays a major factor in deciding the condition monitoring method. And this limitation becomes more complex when the diagnosis is based on multiple motors that are running in parallel with much noise.

Different authors in recent years, Iorgulescu [72] Gritli [73]-[74] Tsytkin [75] Raj [76] and others [77]-[78] discussed multi-motor faults detection using vibration analysis but isolated motors from the system. Different signal processing techniques were used for feature extraction. These studies compared different features in time and frequency domain using ANN. But they never observed the behaviour of multiple motors simultaneously, nor collective motoring of different motors.

In recent years, the stator current monitoring, well recognised as MCSA, has become the focus for many researchers in both academia and the industry. It can provide an indication of motor condition similar to the indication provided by other monitoring methods (e.g. vibration), without any need to access the motor [79]. In most electrical machine applications, the stator current is usually measured for motor protection. When the motor is being controlled by drive, measuring the current becomes integral to the drive apparatuses, which makes it available at no cost. There are three main methods through which captured current data can be analysed for fault detection using current signature analysis. These are: frequency spectral analysis; negative-positive and zero-sequence current components; and Park's vector representation of the three-phase electric current [80].

Different authors [81]-[85] in recent years have discussed multiple-motor faults detection using MCSA method, but they isolated motors from the system. Krishna [81] introduce, in a concise manner, MCSA for the diagnosis of abnormal mechanical and electrical conditions that specify, or may cause, a failure of multiple induction motors, but analyse through separation of the system. The MCSA utilises the results of signal analysis of the stator electric current for the detection of broken rotor bars, air-gap eccentricity and other component damage. Gheitasi [86] discussed fault diagnosis using MCSA on multiple motors simultaneously, but they diagnose a single fault and noise level in each motor. However, in this research, the authors did not focus much on uncertainty management due to the complexity of different faulty signal.

A comparison of MCSA, vibration and others methods [73]-[85] is summarised in Table 2.2 as follows:

Table 2.2: Summary comparison of MCSA, vibration and others methods

Methods	Required Measurement	Faults Diagnosis	Advantages	Disadvantages
<b>MCSA</b> [82][83][85]	Stator electric current	<ul style="list-style-type: none"> <li>• BRB</li> <li>• Air-gap eccentricity</li> <li>• Stator electric current fault</li> </ul>	<ul style="list-style-type: none"> <li>• Non-sensitive</li> <li>• Low cost</li> </ul>	<ul style="list-style-type: none"> <li>• Limited to some fault conditions at no-load</li> <li>• Frequency levels vary with motor size</li> </ul>
<b>Vibration analysis</b> [73][75][77]	Accelerometers and associated wiring	<ul style="list-style-type: none"> <li>• Bearing fault</li> <li>• Other Mechanical Faults</li> </ul>	<ul style="list-style-type: none"> <li>• Available in many configurations</li> <li>• integrate to Velocity output</li> </ul>	<ul style="list-style-type: none"> <li>• Sensitive to mounting techniques and surface conditions</li> <li>• One accelerometer does not fit all applications</li> </ul>
<b>Torque Harmonics Analysis</b> [76]	Two stator electric current and voltage	<ul style="list-style-type: none"> <li>• BRB</li> <li>• Mechanical faults in small load</li> <li>• Stator winding fault</li> </ul>	<ul style="list-style-type: none"> <li>• Non-sensitive</li> <li>• Good for mechanical faults detection</li> </ul>	Not accurate in short circuit-related faults diagnosis
<b>Axial Flow</b> [77]	Axial Flow	<ul style="list-style-type: none"> <li>• Air-gap Eccentricity</li> <li>• BRB</li> <li>• Stator current fault</li> </ul>	<ul style="list-style-type: none"> <li>• Cheap solution</li> </ul>	<ul style="list-style-type: none"> <li>• Non invasive</li> <li>• Not good for dynamic condition</li> </ul>
<b>Impedances of Inverse Sequence</b> [76] [79]	Two stator voltage and electric current values	<ul style="list-style-type: none"> <li>• Stator winding fault</li> </ul>	<ul style="list-style-type: none"> <li>• Initial faults detection</li> <li>• Non-intrusive</li> </ul>	<ul style="list-style-type: none"> <li>• Needs high measurement accuracy</li> </ul>

### 2.3.2 Model-Based Methods

Model-based fault diagnosis techniques are normally dependant on the dynamic system model. The model-based methods of an industrial system benefit from the actual system and model output. A comparison can be made between the simulation and actual data outputs and therefore through visualisation the

condition of a motor can be ascertained. Dynamic models can be developed using physical modelling, system identification and parameter estimation methods. The most significant problem with the model-based methods is that the accuracy of the developed model describes the behaviour of the diagnosis system [83]. Modelling uncertainty happens from the unfeasibility of obtaining knowledge from monitoring process when the system is running in a noisy environment.

Normally, model-based methods have also been used to collect the dynamic response of systems under normal and fault conditions, by different authors [84] - [85], but on motors isolated from systems. A general architecture of the model-based method is shown in following Figure 2.3 as follows.

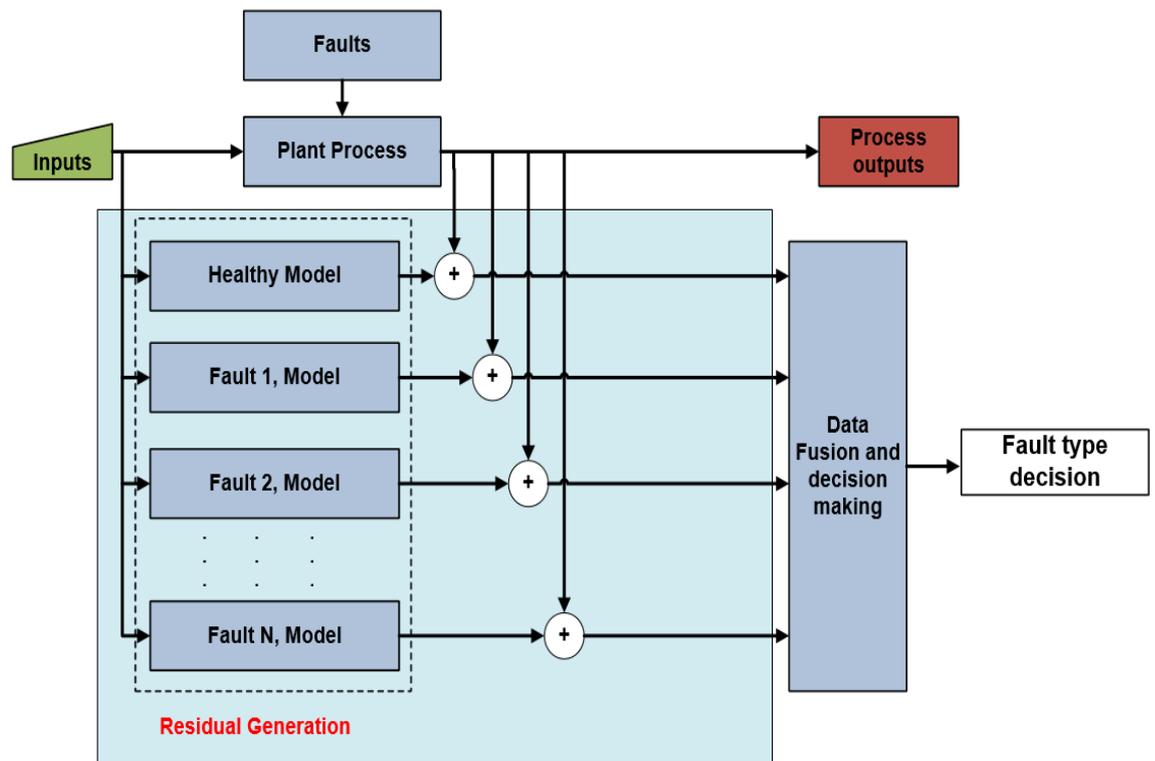


Figure 2.3: Architecture of model-based methods

Typically, model-based methods can be divided into two parts: residual generation and decision making. A fault diagnostics structure is presented in Figure 2.3. In the first portion of the diagram, process models in healthy and faulty conditions are compared with actual process measurements to produce continuation that describes the present condition of the development. In the second portion, the decision-making process is done based on the residual results. In both parts of fault diagnosis, it applies separate models that can be based on

data, knowledge-based, or a combination of both analytical models. The residual generation in the fault diagnostics system is normally based on model and pre-defined process outputs, but residuals can be generated through different methods where model parameter features are estimated from process measurements.

### **2.3.3 Knowledge-Based Methods**

Knowledge-based model strategies usually implement human brain-like knowledge of the process for machine fault diagnosis. In real-time fault diagnostic practices, the human professional expert could be an engineer who applies and operates the diagnosis process, having good knowledge about the strategies and methods of diagnosing multiple motor faults. The knowledge-based methods also work on expertise, like engineers, to diagnose the fault in a motor system when the signal is in a dynamic condition. These methods can be very useful to reduce the percentage of uncertainty when signals are in complex form.

Many studies have been presented in the research area of fault diagnosis using isolated induction motors based on different techniques [86]-[94]. The Artificial Neural Network (ANN) has been perhaps the most commonly used artificial intelligence technique in motor condition monitoring and fault diagnosis, due to its excellent pattern recognition ability and ability to recognise fuzzy and indefinite signals. ANN has the following special characteristics, enabling many applications in information fusion and fault diagnosis [94]:

- The neural network has the ability to gain new knowledge, similar to the way human beings acquire their knowledge. The learning process is implemented by continuous adjusting of the weight values among the neurons
- A neural network can be a Multi-Input and Multi-Output system (MIMO). This structure demonstrates that neural networks can handle complicated multiple object problems, like multiple faults in a machine
- The neural network processes the information in a parallel way, similar to the way humans process complicated information. This special feature indicates that neural networks can fuse information from different sources simultaneously and naturally

- The knowledge in a trained neural network is stored in a distributed way, by means of a set of weights. This also resembles the way the knowledge is stored in human memory
- A Neural network has good fault tolerance performance. This property mainly originates from its parallel structure and distributed information storage system

ANN is reported in the literature as being a knowledge-based technique for single/multiple motor fault diagnosis. These studies perform the diagnoses by mapping different fault symptoms in an isolated motor to produce a diagnosis decision. Eldin [91] presented a diagnosis system based on ANN on machines isolated from system, which applies the RMS measurements of electric current, voltage and speed to train the ANN in diagnosis of motor rotor faults. Voltage faults are only identified in a steady- state condition, not in a dynamic-load condition.

Another study was presented by Arabaci [92], based on the influence of the rotor fault on electric current in the frequency domain, using ANN in a steady-motor operating condition. This study demonstrated the possible symptoms of significant frequency components on the frequency spectrum related to a broken rotor bar fault. These symptoms are used as an input matrix using the supervised ANN architecture. The proposed technique concluded that the process of rotor fault diagnosis and discrimination between each fault occurred with reasonable accuracy.

Drira et al [93] presented a rotor fault model using Fast Fourier Transform (FFT) and the supervised ANN learning method. Significant features (RMS, crest factor, highest magnitude, etc.) were extracted from the electric current spectrum and all possible magnitude highest sideband components were observed using the Neural Network (NN). The Back Propagation Neural Network (BPNN) algorithm is applied for training to detect a rotor fault on a single motor. To introduce complexity to the model, they injected some noisy signals into the healthy spectrum to obtain a reliable and intelligent NN.

Hamdani et al [94] carried out the work on fault diagnosis using four layers of Feed Forward Neural Network (FFNN) for identification of broken rotor bar and eccentricity faults. They noted that the accuracy in fault detection was 86-92 percent, depending on NN architecture, classification method and number of

classified samples. The best NN structure [11×13×11×2] was proposed from different architectures, using the Levenberg Marquardt (LM) algorithm with 92.11 percent accuracy in classification.

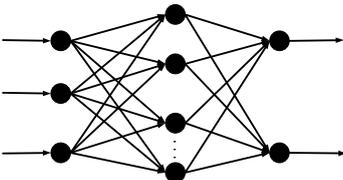
Most fault diagnosis studies based on ANN using isolated motors have been successful. But in the case of a distributed network, this may create confusion through multiple similar motor faults in a network, due to non-linear manipulation of the signal. This sort of complexity and mixture of signals from multiple sources makes it difficult to measure and precisely correlate the fault to a given machine or fault type. To overcome this confusion, a distributed ANN approach is used in this research to identify the fault type and location within a motor network on the basis of significant motor features. The details and methodology is discussed in Chapter 5. A summarised comparison between the neural network and other available knowledge-based techniques is as follows below in Table 2.3:

### **2.3.4 Hybrid Methods**

As each method for fault diagnosis has its own limitations, a combination of several approaches may become a good option. Several different authors have proposed combined techniques such as Neuro-fuzzy [100], Neural Network and Bayesian interface [101] and DS theory with expert system [102]. A hybrid system called Generic Integrated Intelligent System Architecture was proposed for equipment monitoring, fault diagnosis and maintenance [100]. The system integrated different AI techniques such as fuzzy logic and neural network.

A hybrid method [101] was developed that used neural networks to estimate an engine's internal health, and generic algorithms to detect and estimate sensor bias. The method had the advantage of a nonlinear approximation facility provided by neural networks, and advances the system robustness in measuring uncertainty through the combination of generic algorithms within the application.

Table 2.3: Comparison between artificial neural network and other techniques.

Methods	Expression	Advantages	Disadvantages
<b>Artificial Neural Network</b> [92]- [94]		<ul style="list-style-type: none"> <li>• Easy to implement</li> <li>• Non linear</li> <li>• Robust</li> <li>• Generalisation</li> <li>• Most common use in fault diagnosis</li> </ul>	<ul style="list-style-type: none"> <li>• Needs training to operate</li> <li>• Requires high processing time for large NN</li> </ul>
<b>Weighted fusion</b> [95]	$y(j) = \sum_{i=1}^n w_{ij} x(i, j)$	<ul style="list-style-type: none"> <li>• Easy and simple</li> <li>• Intuitive</li> <li>• Low computation load</li> </ul>	<ul style="list-style-type: none"> <li>• Accuracy percentage is low</li> <li>• Difficult to actuate weight</li> </ul>
<b>D-S evidence theory</b> [99][100]	$m(C) = \frac{\sum_{A_1 \cap A_2 \cap \dots \cap A_k = C} m_1(A_1) m_2(A_2) \dots m_k(A_k)}{1 - \sum_{A_1 \cap A_2 \cap \dots \cap A_k = \emptyset} m_1(A_1) m_2(A_2) \dots m_k(A_k)}$	<ul style="list-style-type: none"> <li>• Simple</li> <li>• Robust</li> <li>• Prior knowledge not necessary</li> <li>• No modelling restrictions</li> </ul>	<ul style="list-style-type: none"> <li>• Difficult to determine BPA</li> <li>• Assumes pieces of evidence</li> <li>• computational complexity</li> </ul>
<b>Bayesian inference</b> [96][97]	$P(f_1/O_y) = \frac{P(O_y/f_1) \times P(f_1)}{\sum_{x=1}^y P(O_y/f_x) \times P(f_x)}$	<ul style="list-style-type: none"> <li>• Prior information</li> <li>• Uncertainty management</li> <li>• Flexible</li> <li>• offers a framework</li> </ul>	<ul style="list-style-type: none"> <li>• More complex</li> <li>• Fewer off-the-shelf analyses</li> <li>• Ideological</li> </ul>
<b>Kalman Filter</b> [98]	$K_f = H_f x_f + w_f$	<ul style="list-style-type: none"> <li>• large uncertainty of the initialisation phase</li> <li>• Handles missing data</li> <li>• estimation accuracy</li> </ul>	<ul style="list-style-type: none"> <li>• Computationally complex</li> <li>• Requires conditional independence</li> <li>• Requires linear models for state dynamics</li> </ul>
<b>Fuzzy logic</b> [99]	$u(y) = \frac{\sum_{j=1}^n z^j \prod_{k=1}^m \mu_{jk}(y_k)}{\sum_{j=1}^n \prod_{k=1}^m \mu_{jk}(y_k)}$	<ul style="list-style-type: none"> <li>• Universal function approximations</li> <li>• Comprehensible</li> <li>• Parallel execution of rules</li> <li>• Explainable</li> <li>• Uncertain</li> </ul>	<ul style="list-style-type: none"> <li>• Computational cost</li> <li>• Define the complex rules</li> <li>• Optimisation</li> </ul>

## 2.4 Propagation of Fault Signal in Industrial Power-Line Network

In an industrial power-line network, when a faulty wave signal propagates within the main power-line, it shows a strong relationship between the electric current and voltage waves with certain impedance characteristics. The given input impedance of the multiple connected electric motors has been an interesting parameter, mainly in the closeness of the grid frequency (50/60 Hz) [103]. The importance of input impedance at a higher signal frequency level has gained attention due to the universal usage of available motor variable speed drives. The fast switching between the different phases of power semiconductors of the inverter injects different signals with high energy contents and a large frequency spectrum into the motor feeder cable. Due to the injection of the spectrum into the power-line network, it can generate electromagnetic emission, inverter problems, and damage the insulation winding of induction motors. The presence of these complications is related to the impedance discrepancy between the motor and industrial feeder cable. As discussed in section 2.1, the induction motor acts as a termination impedance when transmitting signals into an industrial low-voltage distribution network between the power-line and motor network. The high frequency signal characteristics of the induction motor may affect the influence of the high frequency characteristics of the main power-line path.

Induction motors within power-line network depend on several factors based on input impedance. In supplying the grid, different frequencies propagate close to the supply frequency and change the behaviour of different motor characteristics due to the injection of electric current signals into the induction motor terminals through stator winding. In this case, the input impedance may rely on the leakage inductance, magnetisation inductance of the stator coils resistance and mechanical torque load of the induction motor [102]. At high levels of frequency, the input impedance of induction motors may be affected due to capacitance and leakage inductance. Furthermore, due to the skin effect, all resistances of induction motors increase, that depending on frequency points.

There are a number of approaches that describe the propagation of fault signals throughout the power-line network. But Power-Line Communication (PLC) is a widely used strategy in the industrial power system to transfer transmission control messages through the power line network. PLC technology uses the power lines for signal propagation. Frequencies in the range of 30-500

kHz have been utilised in the industry for PLC communication [103]. These ranges of frequency are considered sufficient to be isolated from the normal operation of the power system. The available impedance characteristics of a power transmission-line are presented as the ratio between the electric current and voltage of the travelling waves with an infinite spectrum length [104].

The fault signal, may manifest within the different motors electric current through power-line network and it suffers from attenuation caused by the characteristics of the power-lines, and reflection at the junctions of power-lines due to the mismatch of impedances at the connections. The propagation delay between the multiple paths of the signal can result in disturbance of the signal. Therefore, attenuation is a very significant factor because it decides the signal strength as a function of distance; therefore, it plays an important role in validating the locations of faulty motors with in the power-line network. Continuous propagation of signal impedance in power-line, becomes a combination of the inductance, resistance and parallel capacitance of the transmission line [102].

Several studies have been presented [106]-[109] about implementation of the PLC concept in power-lines for fault diagnosis. The propagating frequency of PLC channels is higher than the fault signature frequency and is useful in estimating the attenuation level over transmission lines. However, the calculation method combines with the some technical approaches to develop a fault and attenuation pattern. For the authenticity of the fault signal within power-line network, different measuring points would be a better approach to diagnosing the origin of a fault generator.

## **2.5 Industrial Fault Diagnosis using Wireless Sensor Networks**

Several WSN solutions for industrial machinery have been developed and reported on from commercial organisations [110]-[111] or individual researchers [112]-[116]. Most of these solutions only use the WSN for data acquisition and transmitting signals. Feature extraction and data fusion tasks are then performed on a central computer. Upon sensor data acquisition, feature extraction and fault diagnosis is another tactic capable of diffusing raw data diffusion that can scale down the number of features and save node power. But most of these solutions are based on isolated motors. Industrial Wireless Sensor Network (IWSN) for motor fault diagnosis and condition monitoring needs to consider the high-power system requirements of industrial processes and the distinctive available

characteristics of motors [110]. Some industrial processes are very important, such as high sampling rate, quick data transmission rate and reliability of data. However, there are constraints in IWSN, such as computational ability, limited radio bandwidth and battery energy. Thus, limitations exist between the high system requirements of electrical machine fault diagnosis and the resource constraints feature characteristics of IWSN.

Some recent literature focuses on the application of IWSN in machine condition monitoring and fault diagnosis, pumping fault diagnosis, manufacturing machines, smart grids, power plants and structural health monitoring. Reference [111] presents electric current and vibration based data acquisition for monitoring rotating machinery in power plants. They present a sensor-level data fusion algorithm to diagnose the condition of isolated machines. A comparison result has been performed between available fused and healthy data by using wireless nodes. A time-series data judgment and task-level fusion algorithm was introduced to reduce power and bandwidth needs.

Hou et al, in [117]-[118], introduced a diagnosis solution using electric current and vibration signature data acquisition system for observing rotating machinery at power plants. The diagnosis system monitors the motor vibration and stator electric current signatures from two different motors. Node-level feature extraction techniques were implemented and a neural network classification method used for training and uncertainty management. Decision-level fusion was implemented at the node coordinator. The training was executed in offline mode in an efficient manner, and the BRB and eccentricity fault states have to be detected manually at two experimental motors by applying different load levels.

## **2.6 Shortcomings in Existing Fault Diagnosis Research**

The current harmonics present in the motor electric current are mainly created by machine asymmetries and vibrations from machine faults. The reliability of signal-processing techniques depends upon a good understanding of the electric and mechanical characteristics of the machine in both a healthy and faulted state under different loading conditions. The following shortcomings were identified on the basis of information discussed in the literature review:

- Most of research has been effectively tested on separate motors to diagnose the in condition and performance comparing healthy and faulty motors. Limited research has been done on distributed multi-motor signature analysis where all motors are part of the system and propagate a faulty signal over the network
- Some limitations have been perceived in implementing diagnosis in a distributed motor network, with confusion between different similar machine fault symptoms in the power-line network, and lack of accuracy in the analysis system due to the existence of non-linear interference from industrial noise signals
- There has been only limited research on the effect of load variations on the amplitudes of fault frequency components under healthy and faulty conditions. The majority of the studies only consider a full-load case, with limited research considering partial-load cases. To detect faults and estimate fault severity in machines, using characteristic fault frequencies, it is important to examine the variability in their amplitudes with other effects than load and fault severity. This area has had limited research on machine condition monitoring
- Few studies have focused on the detection of multiple faults, that is, the combination of broken rotor bars and eccentricity faults under varying loading conditions
- The majority of studies considered only the stator electric current as a diagnostic medium to detect different faults in induction motors. Only a few researchers have proposed using an instantaneous network power signal as a diagnostic medium to detect rotor-related faults under different load conditions. The use of instantaneous power to detect other major faults in the machines (eccentricity, shorted turn and misalignment) and multiple faults (combinations of different faults) under varying loading condition has not been reported in previous research
- Insensitivity to and independence of operating conditions in the power-line network system.
- Utilisation of the Neural Network technique on distributed industrial motor networks has not been reported in previous research where the signal propagation process could change the pattern behaviour of each

neighbouring motor and create a confusion in identifying the actual source of fault indices

- To date, distributed signature analysis using WSN with sensor-level data fusion, based on multiple motors that are propagating signals into the power-line network, is a relatively unexplored topic where all motors send their respective features data to central computer for fault diagnosis

Based on the above identified shortcomings, the focus of this research was to diagnose multiple faults when all motors operating are part of system and propagating a faulty signal over the network can create confusion between different similar motor faults symptoms in the power-line network. To overcome this confusion, ANN was utilised in identification of fault indices within a network when faulty signals manifest into healthy signals from other motors. Finally, wireless sensor-level data fusion was implemented using an Arduino development kit to improve efficiency and accuracy in decision making when all motors are operating in parallel.

## **2.7 Chapter Summary**

This chapter has covered a variety of different topics, as well as several particular techniques, algorithms, approaches and methods. The literature was mainly categorised into three major themes: fault propagation and diagnosis in power-line network; data fusion; and wireless sensor networks. The following conclusions were drawn from the literature review:

- Machinery fault diagnosis has been too reliant on single information sources of data, especially electric current or vibration data. The use of multiple information sources for fault diagnosis from multiple connected motors within the same power-line has not been well addressed and is an unexplored area
  - Correct feature extraction and selection increases the performance of a network and reduces the network input dimensions and training time
  - Consideration of multi-parameter data-fusion techniques can play a vital role in improving system performance, such as in accuracy, reliability and robustness

- Deployment of the WSN in industrial-machinery fault diagnosis can improve its efficiency and reliability, and reduce the chances of uncertainty in management of complex data.

Based on the above concluding points, the scope for this research was limited to the significant utilisation of available advanced techniques and approaches for feature extraction and motor fault diagnosis using WSN.

## CHAPTER 3

# DISTRIBUTED MOTOR NETWORK SIGNATURE ANALYSIS AND FAULT TYPE DIAGNOSIS CONCEPT FORMULATION

### 3.1 Introduction

This chapter focuses on concept development of distributed signature analysis and fault type diagnosis in the industrial multi-motor environment. An industrial distributed motor network model has been utilised for the development of a useable case study. A mathematical matrix equation of fault indices has been formulated to estimate the association of signals at each sensing point with known fault patterns. Attenuation influencing measurement of propagated signals within a power-line network has been expressed through a mathematical model to measure and identify the possible route path of different signals. The propagation of faulty signals in order to estimate possible faults over a power-line network at multiple sensing points has been discussed. A supervised multi-layer Artificial Neural Network (ANN) architecture has been discussed at motor level and provides reasoning for its role within those environments where multiple motors are running in parallel.

### 3.2 Distributed Motor Network under Study

A typical industrial multi motors power-line network layout structure is considered in this chapter to present the concept of fault signal propagation and identification through the main power-line. This, in turn, acts as a good conducting environment for signals to travel through the network, and influence other motor behavioural signals according to their distance from each other [119]. Different aspects have been considered for a configuration of connected induction motors via an interconnected supply bus. Multiple measuring points are considered, to observe the behaviours of induction motors. For the development

of this concept, a multi-motor model is considered here that consists of a main power-line, sub-segmented bus bars and connected motors, as shown in Figure 3.1.

In order to evaluate the observations at each measuring point and analyse the effectiveness of the diagnosis, a different number of measuring points has been assumed at certain points, to observe the behaviours of each individual motor within the same bus.

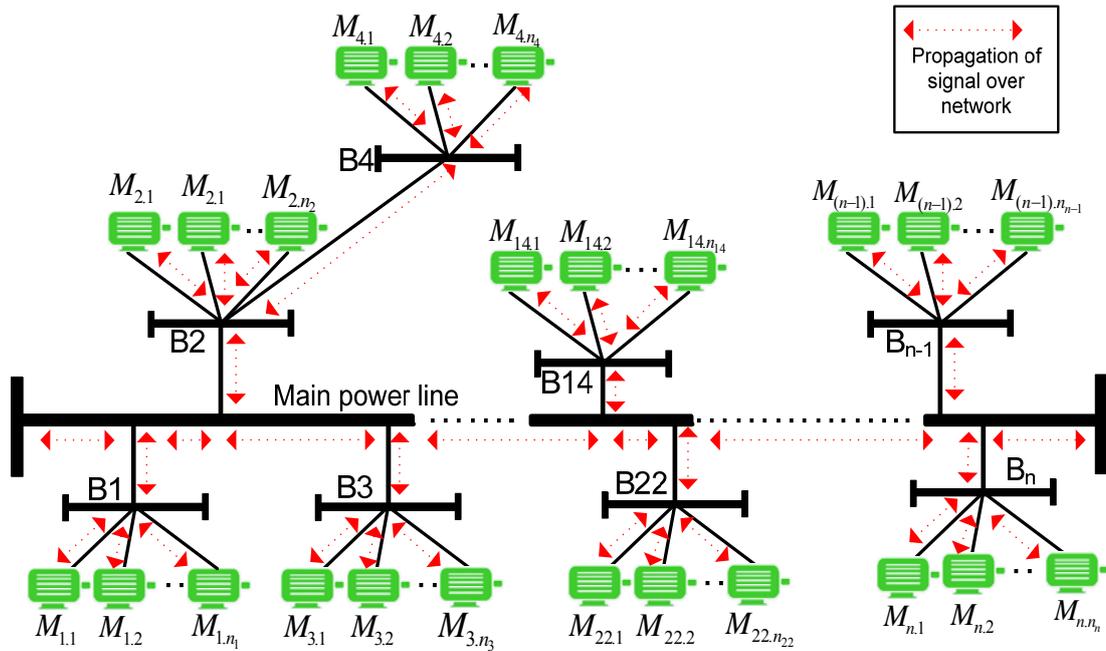


Figure 3.1: Structure of a typical distributed industrial induction motor network model

This work is not an attempt to develop a comprehensive solution from all available diagnosis methods, instead emphasising analysis using the MCSA method. MCSA theory is implemented to examine the condition of motors and development of fault patterns. From these patterns, significant values of feature characteristics were chosen from each motor electric current signal, based on the different frequency sideband points and associated these sideband amplitude values of each fault type [119]. These features are very valuable in presenting the characteristics of motor behaviour within motor networks and are used in the data fusion technique [107].

Individual and combined analysis of each fault pattern has been discussed using a low and high-power induction motor network [120]. It reflects fault signal propagation over the power-line network. It also emphasises the significant influence of faulty signals on neighbouring motors within the same bus, as well

as other buses. This signal propagation process could change the pattern behaviour of each neighbouring motor and create confusion in identifying the actual source of fault indices. When the faulty signal manifests into a neighbouring motor signal, it is necessary to cancel the environmental motor network noise and filter the fault signal of the targeted motor. Thus, the actual condition of the signal can easily be observed. The pattern recognition scheme has been extended to all available sensing points over the network using a distributed signature analysis diagnosis strategy.

### **3.3 Distributed Network Model and Key Influencing Factors that may Contribute to Confusion in Fault Diagnosis**

With reference to Figure 3.1, a power-line system consists of many different load nodes. These nodes are physically linked via electrical connections to each other with a range of different attenuation coefficients [121]. Different sizes of motor have a considerable combined influence on an overall system's operation. The fault signal may affect overall network performance while travelling over power-line network, and can cause a drop in the value of voltage on an electrical bus, that gives rise to imitative electric current [107]. Therefore, to identify the faulty signature within a power-line network, a full understanding of industrial motors and correct modelling is required.

#### **3.3.1 Key Factors Influencing the Ambiguities in Fault Diagnosis**

In order to analyse the different characteristics of faulty signals in power-line network and diagnose faults correctly, different significant aspects were carefully considered including noise level, multiple motor sizes in a distributed network, measurement of attenuation factors and similarity between different fault symptoms. These factors have a significant influence on electric current signal and may contribute to confusion in fault diagnosis.

- **Motor Size**
  - The level of electric current is not equally required for different multiple sized motors in a network when they are running in parallel at an industrial site. In networks, high-powered electric motors may generate a strong faulty signal strength when compared with low-power electric motors with a similar fault signal, while all motors are

running in parallel. In this situation, observing only the amplitude value of an acquired signal is not enough to discern the source of the fault signal and decide the condition of motors

- If the high-power motor propagates a faulty signal, it has a significant influence on low-power motors compared with same-power motors. The strength of a fault's influence may reduce on other motors that are operating in adjacent buses, depending on the distance between buses, because of the attenuation factor. On the other hand, in case of low-power motors, the influence of a faulty signal on high-power motors may be weaker in the same bus and may be not observable in other buses as a mirror signal. In this situation, weaker symptoms in high-power motors may create uncertain identification of the noise/attenuation level at different sensing points and the type of fault

- **Noise Level**

- A significant level of noise may be expected in the industrial environment due to the normal running process of distributed networks, including multiple motors, transformers and other electrical equipment. These kinds of noise contain a certain range of frequencies and these may be eliminated superficially by applying different filtrations in an isolated equipment situation. Distributed power-line network transients, including motor start-up, electric current propagation on the power-line, and steady-state of running of nonlinear motors may be a source of different electrical signals that present similarly to some types of fault indicators at multiple frequency points. These noisy signals may easily manifest into other motor signals, at certain frequency ranges that are connected with the main power-line and is difficult to discern the original signal
- The strength of noise coming into the signal depends upon the motor sizes and the distance between them. As with any electric current signal into a main power-line, noise and fault indicators can easily travel between different buses on a network and lead to a wrong interpretation of fault type
- The majority of noises can be eliminated from the spectrum by improving the quality of data acquisition and correct data processing

techniques. These techniques are not usually very effective where the motor size is relatively small or has noise symptoms in the same frequency range. But this procedure will be suitable for large motors that normally are not very successful in an environment where small electrical motors are running, and can also interpret major noise causes

- **Distance between Multiple Motors/Attenuation Factor**

- Several load nodes can be physically connected to each other through electrical links with a set of attenuation factors. While travelling as a signal on the power-line, a high attenuation value in an electric current signal may decrease the voltage level. Most fault traceability strategies research is based on circumstances where fault signal attenuation in distributed power-line systems is related to the distance between different measuring points [107]
- With reference to Figure 3.1; in order to estimate the electric current signal attenuation value in buses, a fault and its related spectrum takes place in the same bus and its neighbouring motors. The fault signal then propagates to other buses on the network. The existence of a faulty signal can be observable in neighbouring motors and other parts of the network as well. The important concern is, the same bus motors neighbouring a faulty motor have a high-strength fault signal than the other buses motors
- As long as, a signal propagates in a power-line and reaches other motors in different buses, the strength of the fault signal should be lower than in the originating bus. High-power motors can propagate a strong signal for longer distances than low-power motors
- Normally, the distance between the motors is known in an industrial motor network and you can easily distinguish the location in different zones. But due to manifestation of a faulty signal from different sources into multiple motors electric current spectrum, serious types of ambiguity exist in identifying the origin of a fault. As a result, analysis of attenuation factors of the propagated signals should be carried out to evaluate the possible route path of different signals over the network. This analysis will help to estimate the source of a fault on

multiple path routes in the power-line network and anticipated the fault representation around the power-line

- **Mirror Effect (Similar Faults of Neighbouring Motors)**
  - Propagation of motor fault signals in a main power bus, leaves influences on the whole power-line network due to voltage drops in multiple buses. The fault signal then propagates signals to other buses of the network and manifests. The mirror signal is observable in neighbouring motors and other parts of the network. The important concern is, the motors of faulty motor in the same bus have a stronger fault signal than other buses-motors. As long as, signals propagate in a power-line and reach other motors in different buses, the strength of the fault signal should be lower than in the faulty bus. Importantly, if any motor is under the influence of faulty signals from other motor, but at the same time it has also been producing the same fault at the same frequency point, a clash can occur between the symptoms for identification. Then it will be very difficult to discern which signal belongs to one motor and which is manifesting from on others

While sometimes, with deployment of an efficient fault diagnosis system, it is hard to identify the fault source motor in a network and its fault type. Sometimes, faults have the same fault symptom patterns at different frequency ranges in the same motor. This research is an attempt to reduce this confusion in decision-making from these key factors to an acceptable level and focus on different aspects involved the propagation of signals in-network.

### **3.4 Fault Diagnosis System Solution**

A fault condition is expected to damage symmetrical properties where a faulty motor operation induces an abnormal harmonic modulation in the motor electric current signal due to its inherent electro-magnetic coupling. According to the MCSA theory, electrical and mechanical faults have an inimitable influence on the electric current spectrum, and change the behaviour of healthy spectrum characteristics. Therefore, some pattern recognition techniques are required to detect motor faults [107].

In consideration of above key factors that create ambiguities in fault diagnosis, a generic framework solution has been presented, based on a knowledge-based wireless sensor network to estimate predicted faults in a multi-motor network. For that, two levels of a diagnosis system have been proposed at motor and system level to reduce ambiguities. A motor-level diagnosis solution would offer fault detection at each node level and gain some knowledge about neighbouring motors if they listen to some disturbance in the surroundings. Distributed in-network diagnosis solution deals with improving the dependability and reliability of fault detection over the network when a direct diagnosis system is available, and provides a preliminary diagnosis result when no direct measuring method is available. The details for both diagnosis levels are discussed in detail in subsequent sections.

### 3.5 Fault Diagnosis System at Motor Level

A systematic diagnostic framework has been formulated here to present the activities involved in fault diagnosis at each motor level as shown in Figure 3.2:

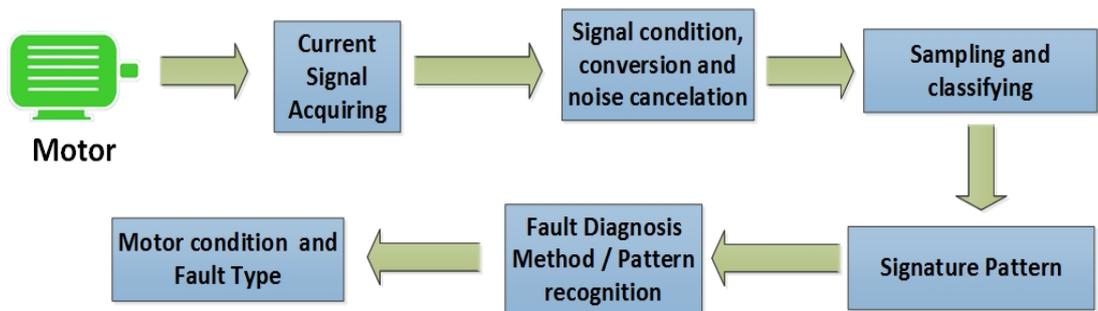


Figure 3.2: Motor-level fault diagnosis framework

In Figure 3.2, different activities are presented for complete diagnosis at each motor level. Many data acquisition components are required to capture and store as spectrum. With all sizes of motor, initially the signal is collected via a voltage or electric current transducer or transformer. Then the fundamental frequency (50Hz) is considered to observe the normal running operation of the motor.

The next phase is to exclude the high-level frequency components of the signal that are not involved in contributing to a diagnosis. To remove unnecessary noise from the signal, a threshold level is used to exclude noise from the signal. The threshold level is subjected to the level of noise on site, the structure of the motor network and the required accuracy. An analog signal then transforms the sampled data and converts it into a digital signal using an analog-to-digital convertor. The processes of digitisation are followed by estimating the frequency spectrum of the waveform signal. Then significant components of the waveform signal are categorised by using a data processing technique and making the spectrum signature patterns. Most of the signatures and their fault patterns have been formed by using frequency spectrums.

Other signal processing techniques are involved in additional efforts to generate harmonious patterns for fault diagnosis. Therefore, this research utilised the FFT for the compilation of fault patterns. Finally, a fault diagnosis and detection system was necessary to form the fault indicators. An efficient post-processing system was required to identify fault types and generate a report on motor condition.

### 3.5.1 Sampling and Classifying Abnormal Sidebands Components of Motor Waveforms

After acquiring the signal, the first step in diagnosing a fault is to classify the abnormal sideband peaks of waveforms in an appropriate form for further analysis. Fault features are difficult to recognise in the time-domain, but it is easier to analyse the features transformation in the frequency-domain [107]. This research employed FFT to analyse the electric current signal of motors from their dynamic behaviour in the power-line network and then separate any weak or noisy signal results. In order to extend the concept of fault identification from a motor signal to a network of motors, an  $n$ -dimensional matrix of fault indices at any measuring point for any suspected fault frequencies within a multi-motor network ( $MM_{s\_fault}$ ) is as follows [107]:

$$MM_{s\_fault} = \begin{bmatrix} F_{x,1,1} & F_{x,1,2} & F_{x,1,3} & \cdots & F_{x,1,z} \\ F_{x,2,1} & F_{x,2,2} & F_{x,2,3} & \cdots & F_{x,2,z} \\ F_{x,3,1} & F_{x,3,2} & F_{x,3,3} & \cdots & F_{x,3,z} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ F_{x,y,1} & F_{x,y,2} & F_{x,y,3} & \cdots & F_{x,y,z} \end{bmatrix} \quad (3.1)$$

Where,  $F_{x,y,z}$  = any other fault indices within network;  $x$  = fault type (each matrix row relates to a fault type);  $y$  = speed group of same speed motors;  $z$  = Number of measuring points within network.

### 3.5.2 Example Signature Pattern Caused by the Fault Symptoms at Motor Level

The broken rotor bar and the eccentricity fault resulted in amplitude modulation in the electric current signal at characteristic frequencies. The broken rotor bar signature is observable twice at slip frequency away from fundamental frequency and eccentricity. The eccentricity signature is observable at rotor frequency, away from the fundamental frequency [107]. In order to diagnose a fault and analyse the significant sideband points in the spectrum, motor slips have been transposed with different fault names to estimate the related speed ( $S$ ) at each significant possible frequency point.

$$Slip_{brb} = \frac{T - f_1}{2f_1^2} \quad (3.2)$$

$$Slip_{ece} = \frac{\frac{T_p}{2} - f_1 k_p \frac{p}{2}}{f_1 k} \quad (3.3)$$

Where  $T$  is the frequency of suspected sideband points;

By using the above equations (3.2)-(3.3); the suspected motor speed can be estimated as shown in Equation 3.4.

$$S = (1 - slip) n_s \quad (3.4)$$

Placements of significant frequency components are dependent on the speed of the drive and its variations. Therefore, by looking at the rotor speed and any deviation from the nominal speed, the frequency spectrum pattern is captured in several significant frequency bands. As an example, significant status bands for an induction motor with speed variation is normally 1400 to 1450RPM with a nominal frequency of 50Hz [107]. This variation of speed may create discrimination uncertainty in frequency amplitude. So, to justify the variation of speed, a pattern of frequencies set is required to identify the severity of the fault. A signature pattern for a fault is a set of frequency components that can be represented by following formula 3.5:

$$SP = F_1, F_2, F_3, \dots, F_n \quad (3.5)$$

Where,  $SP$  is the signature pattern;  $F_n$  is the frequency sidebands components.

The difference between the amplitude value of these frequency sideband components and the fundamental frequency shows the degree of seriousness of potential faults and the motor's nominal electric current value.

$$A_i = CE(A_1, A_2, A_3, \dots, A_n) \quad (3.6)$$

Where,  $A_i$  = Amplitude values of frequency sideband components;  $C$  = nominal electric current of motor;  $E$  = index values of severity of potential faults;  $f_i$  = frequency sidebands components associated with fault  $i$ .

Fault indices are normally constant values associated with faults, but motor nominal electric current and index values of seriousness of potential faults differs for different sizes of induction motor [123]. It also depends on the severity of the faults and the possible intensity of any frequency sideband components of signatures of faults.

However, the set of sideband frequencies with associated amplitude values of a given fault signature is proportional to the maximum strength of the fault amplitude, as given below. Therefore, in order to simulate suspected faults in motors, a set of spectrum frequencies has been formulated with modified magnitude and proportional for calculating the strength of fault in following equation (3.7) [107]:

$$Fault_i = \begin{bmatrix} SP = F_1, F_2, F_3, \dots, F_n \\ A_i = CE(A_1, A_2, A_3, \dots, A_n) \end{bmatrix} \quad (3.7)$$

### 3.6 Neural Network Architecture for Fault Detection at Motor-Level

Due to the complexity of motor network modelling in industry, a fixed number of hidden layer nodes is not efficient and might take a long time for training and classification. The neural network has been utilised to recognise the abnormal representation of a voltage signal, due to the inherent capacity of the classification and generalisation process efficiently [128]. In this section, no attempt is made to describe all particulars related to typical neural networks in detail. The objective here is rather than providing a general overview of the decisions one might face when designing a Multi-Layer Perceptron (MLP), the

focus is only on NN architectures and methodology for the targeted fault type's diagnosis.

All inputs consist of dynamic feature values in multiple motors, depends on their power. Each network has one hidden layer that consists of the maximum number of neurons according to the number of inputs values. The number of nodes in the hidden layer has a direct influence on the performance of ANN. I tested multiple architecture for the selection of hidden layer nodes for better performance reported in Chapter 5. If the number of nodes is small, the network may not be trained adequately. On the other hand, if the number of neurons is too large, the network will take a long time to train. Therefore, the selection of the dimensions and number of hidden layer nodes has been carefully selected for greater efficiency.

Here, I demonstrate a NN architecture that consist of input, hidden and output layers. Dynamic hidden layers are associated with every bus in the power-line network, which deals with respective bus motors and analyses the condition of the motors, as shown in Figure 3.3.

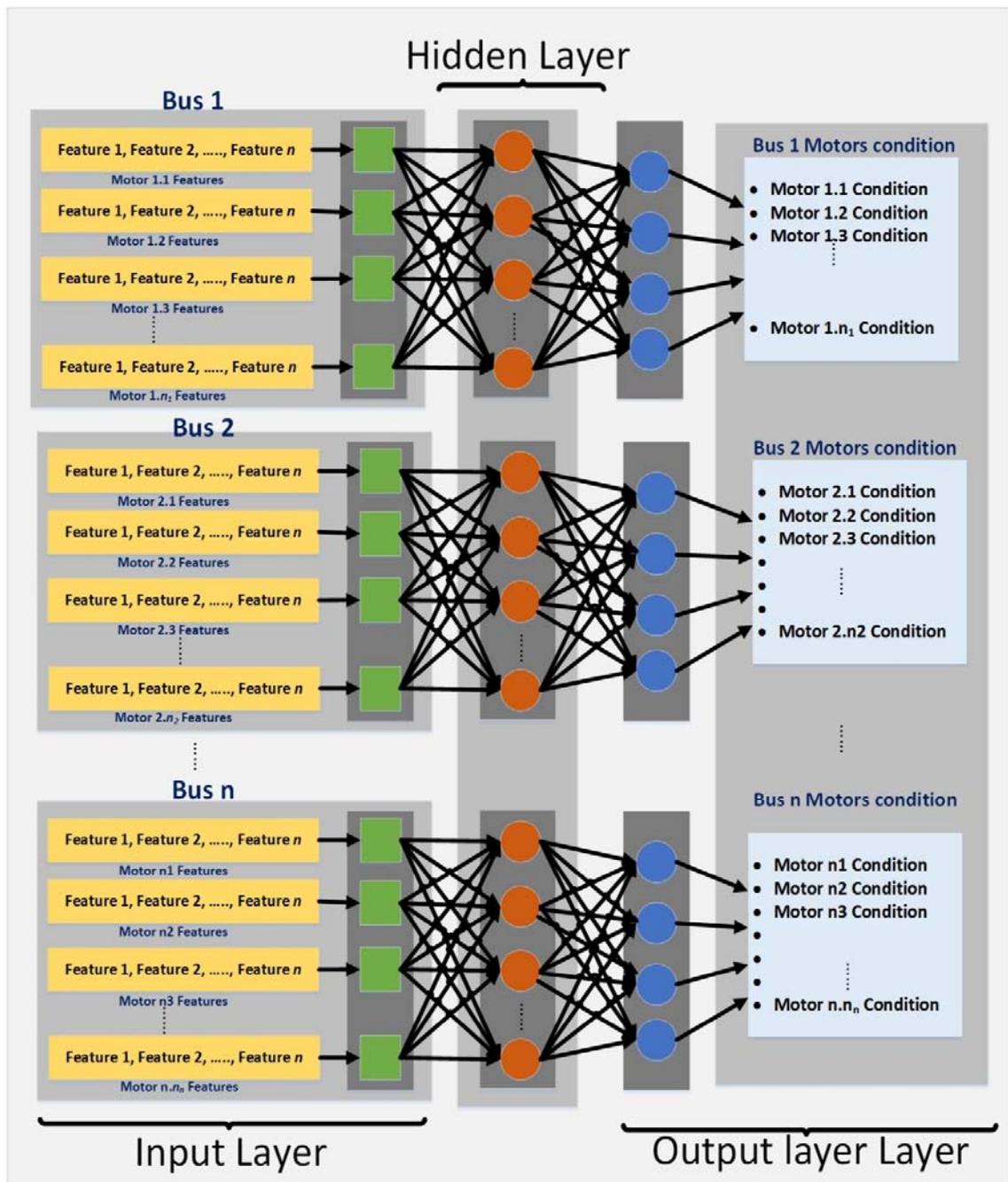


Figure 3.3: ANN architecture at individual motor level

### 3.7 Technical Distributed Fault Diagnosis Solution at Network Level

Individual single point decision-making often may cause serious failure in identification of fault type and localises the origin of a fault signal within a distributed power-line network. However, multiple measuring diagnosis points can improve the accuracy of fault diagnosis due to capturing dependent data. This serves to distinguish the fault origin and isolate individual motor faults with satisfactory accuracy.

The focus of this section is to highlight the possible accuracy and success in identifying the possible existence of fault signals through revealing the associated spectrum at node level. Identification of possible diagnostic confusion due to noise, fault signal propagation, available possible travelling paths on power-line network, causes for live hearing of neighbouring nodes and multiple similar faults at multiple nodes is necessary. Then follows the need for fault location identification in the power-line network.

This research has utilised a distributed fault detection-framework for multiple-motor architecture that is based on WSN connectivity for fault diagnosis, as shown in Figure 3.4. The test bed development and discussion of the distributed architecture will be demonstrated in Chapter 6.

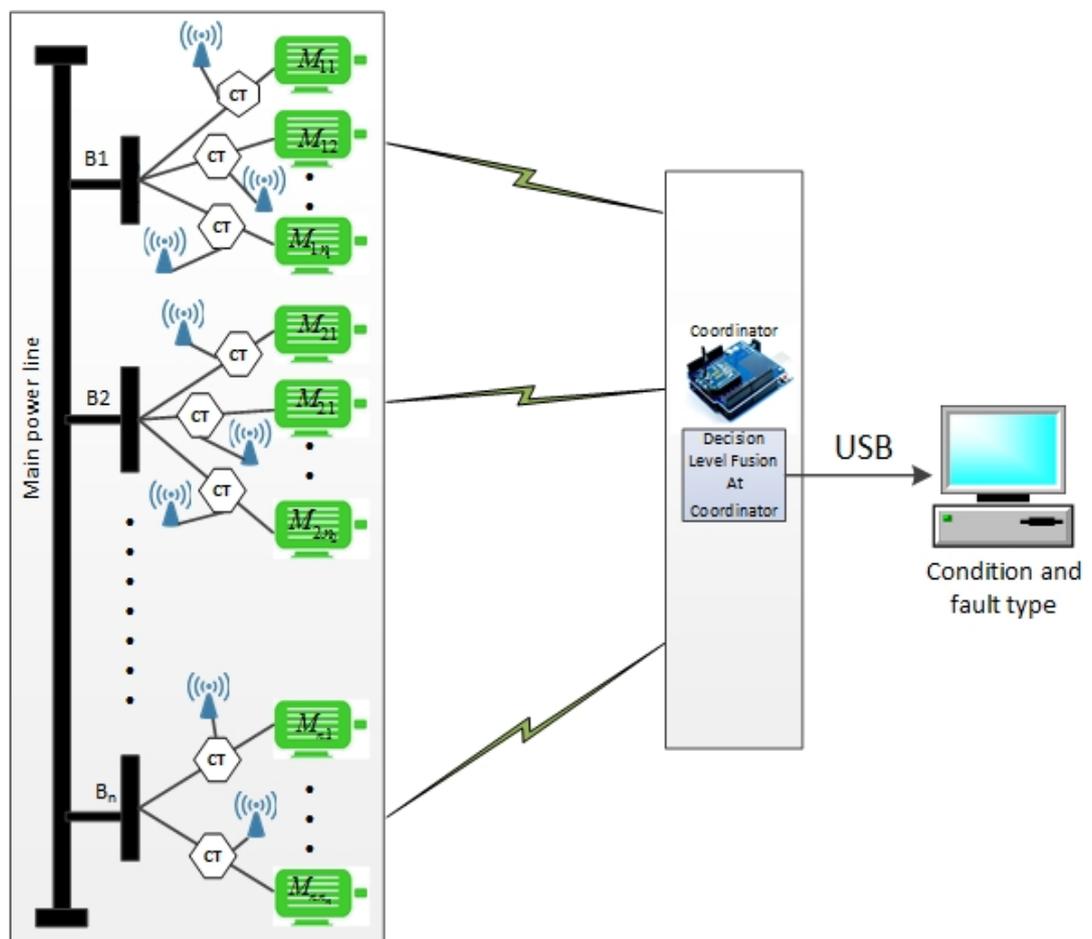


Figure 3.4: Distributed generic WSN fault diagnosis framework for industrial multi-motor network

Figure 3.4, demonstrates the hypothesis of distributed signature analysis, with consideration of multiple sensing points to observe the behaviour of the

electric current signal. This distributed signatures analysis may offer better results for fault diagnosis, as it takes advantages of the maximum potential accuracy of direct analysis and diagnosis whenever multiple points are available. It may also provide a better framework to further clarify the fault symptoms manifest from propagated noise over the network. The focus of discussion here is on the approach to more precisely identifying the faulty motors within power-line network. This is also helpful in estimating the influence of fault signals in the electric current of in-network motors, using a signal processing approach.

Further investigation is based on signal attenuation of electric current signals that leads to a practical framework, which estimates the propagation level of fault signals in motor networks and identification of different path routes for signal propagation. Finally, the framework shows the potential and technical role of the WSN node in distributed signature analysis. Wireless nodes would be responsible for data collection, the creation of possible fault signatures, identification of noise levels into signals, diagnosis of fault error symptoms, real-time hearing of neighbouring node behaviour, and generating an alert to coordinator node in case of suspected situations. This distributed diagnosis solution is anticipated to allow more effective and reliable results in motor networks with a direct measuring method, and provide better early indications of fault occurrence for in-network electric motors where no direct measuring method is available.

### **3.7.1 Multiple Sensing Points Measurement to Identify the Origin of Significant Components over Distributed Multi-Motors Network**

The dynamic behaviour of motors in distributed power-line network and the necessity for eliminating any suspicious results recommends the use of a time-variant methodological solution. In order to analyse the multiple-motors spectrum in each bus, the following steps need to be considered:

- First, collect the data at different sensing points within the power-line network. Then significant sideband frequency points and amplitude values are considered as prospective fault signals caused by faults in multiple motors
- The next step is to gather the significant frequencies sideband values at multiple sensing points within the respective bus related to the targeted faults. A threshold is defined at all sensing points and

collected data. All relevant frequency sideband components that are not matched with significant frequency band patterns will be detached from the signal, and remaining signal will be classified as per the reference fault pattern

- Subsequently, all significant frequency sideband amplitude values should be observed at different sensing points. If the amplitude values differ strongly enough from the defined healthy range, it can be assumed that the fault is coming from its own motor or coming from a neighbouring motor within the same bus. Otherwise, it is suspected that the signal is manifesting from other buses
- The strength and influence of the faulty sidebands at each motor depend on the distance between the different buses and motors. In the same bus, if a high-power motor propagated the faulty signal, then similar powered motors would have the same influence, with a minor amplitude value difference, but a low-power motor would have a higher influence. This shows that they have some type of fault. In fact, this frequency has been manifesting from other motors. In the same way, if a low-power motor is propagating a faulty signal, then a high-power motor in the same bus would have less influence due to its higher Horse-Power (HP). This creates some uncertainty that the suspected sideband components are generating from their own motor or manifesting from others. This sort of confusion may create an inability to identify the actual source of the fault indices
- In order to identify the faulty motor in a network, after examining multiple sensing points, select the suspected bus zone. Then examine all motor signatures in the same bus and identify the ambiguously behaving motor. But due to uncertainty existing in the faulty sideband, the actual speed of the suspected motor needs to be measured and compare with the synchronised speed based on their mechanical properties. If the speed is in the normal range, then it can be assumed that this sideband frequency has manifested from other motor. The speed factor is not sufficient to take a decision whether the motor is faulty or not. Therefore, significant points will be classified as an indication for some suspected faults and generate a fault index based on multiple frequency points. A representation of multiple sensing

points over network to discover about the actual location of a motor follows in Figure 3.5.

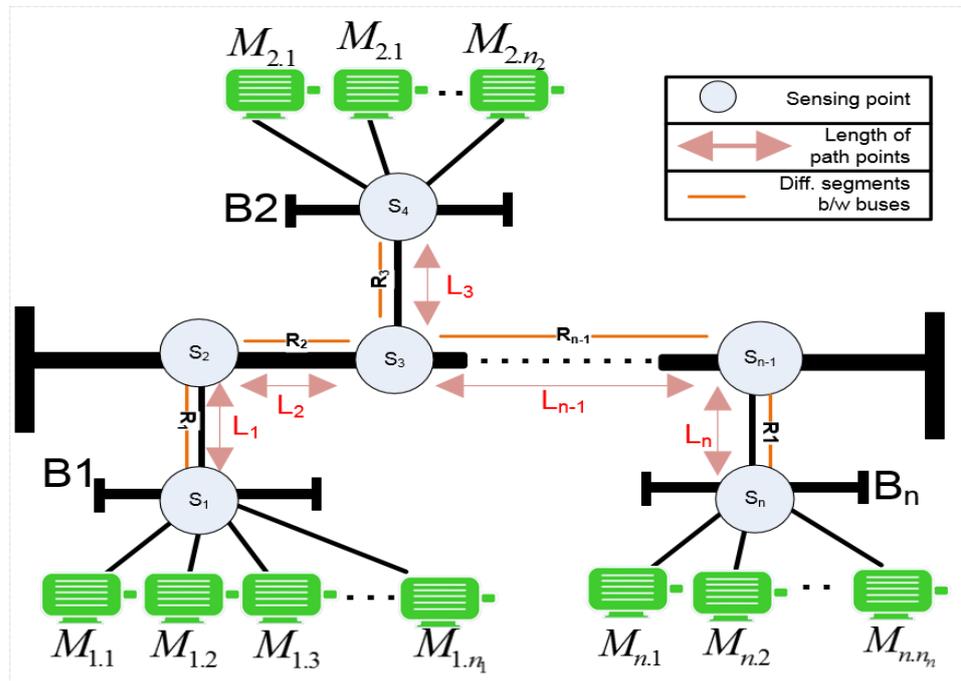


Figure 3.5: Demonstration of multiple sensing points over a network to identify the location of multiple motors

Multi-path propagation of the signal and sensing points is shown in Figure 3.5, which consists of multiple sensing points ( $S_1, S_2, \dots, S_n$ ) over a network and relates to the respective sub-bus. Each link has four branches and consists of different route paths ( $R_1, R_2, \dots, R_N$ ) of length between buses ( $L_1, L_2, \dots, L_N$ ) and shows the characteristic impedance of each bus length at each sensing point and others. Different combinations between sensing points can be predicted, to analyse the strength of fault at each point, if the distance between motors is known at industrial plant. It would also be helpful in identifying the direction of propagated signal origin. In the case of unknown distance between motors and buses, sensing only the value at each point would be not sufficient to discriminate between fault signals. The attenuation factor would ease this concern and have an imperative role in substantiating the locations of faulty motors within the network, and will be discussed in following sections.

This fault index indicates the strength of each relative fault signal at any measuring point. A fault index can be presented in the form of a table to show multiple sensing points between actual locations of different motors as follows Table 3.1:

Table 3.1: Possible sensing points within distributed power-line network

Connection between buses	Route Segment	Distance between Buses	Possible combination of sensing points
$B1 \rightarrow B2$	$R1 \rightarrow R2 \rightarrow R3$	$L_1 + L_2 + L_3$	$S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4$ $S_2 \rightarrow S_1$ $S_2 \rightarrow S_3 \rightarrow S_4$ $S_3 \rightarrow S_4$ $S_3 \rightarrow S_2 \rightarrow S_1$
.....	.....	.....	.....
$B1 \rightarrow B2 \rightarrow B3 \dots \rightarrow B_n$	$R1 \rightarrow R2 \rightarrow R3 \rightarrow$ $\dots \rightarrow R_{n-1}$	$L_1 + L_2 + \dots + L_n$	$S_1 \rightarrow S_2 \rightarrow S_3 \dots S_n$

Due to a fault in the motor, other behavioural characteristic features could also change. Thus, other characteristic features also need to be examined along with the healthy features, for validity of decision making using the through Neural Network.

### 3.7.1.1 Fault Signal Propagation between Buses on Multiple Paths

The total route impedance may be estimated by the following equations, measuring the frequency-dependent impedance of motors and the connection between them as follows:

$$Z_{M_{fc}} = \frac{\partial V}{\partial I} \quad (3.8)$$

$$Z_{J_{fc}} = \frac{V_s - V_e}{I_i} \quad (3.9)$$

Where  $Z_{J_{fc}}$  = Total Impedance value at Junction with carrier frequency;  $V_s$  &  $V_e$  = Voltage value at starting and ending measuring points.

As described earlier in Chapter 2, connections and motors are considered to be inductive in industrial networks. The propagation of fault signals in multi-conductor transmission lines is always a linear combination of dissimilar propagation modes at different propagation speeds. Therefore, their impedance angle is not necessarily the same at each point. The fault-point phase voltage can be expressed as follows, in terms of fault distance and phase [121]:

$$\begin{bmatrix} V_{a_s} \\ V_{b_s} \\ V_{c_s} \end{bmatrix} = \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} - F_D \begin{bmatrix} Zt_{aa} & Zt_{ab} & Zt_{ac} \\ Zt_{ba} & Zt_{bb} & Zt_{bc} \\ Zt_{ca} & Zt_{cb} & Zt_{cc} \end{bmatrix} \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} \quad (3.10)$$

Where,  $V_a$ ,  $V_b$  and  $V_c$  is the voltage at A, B & C phase respectively.

### 3.7.1.2 Faulty Signal Influence on Electric Current Spectrum

To formulate a mathematical model for propagation and influence on faulty signals, Figure 3.6 is offered as reference. In this figure, one Motor ( $M_{1,3}$ ) is assumed to be a faulty motor that is propagating a rotor fault signal and may cause a drop in the voltage of the system.

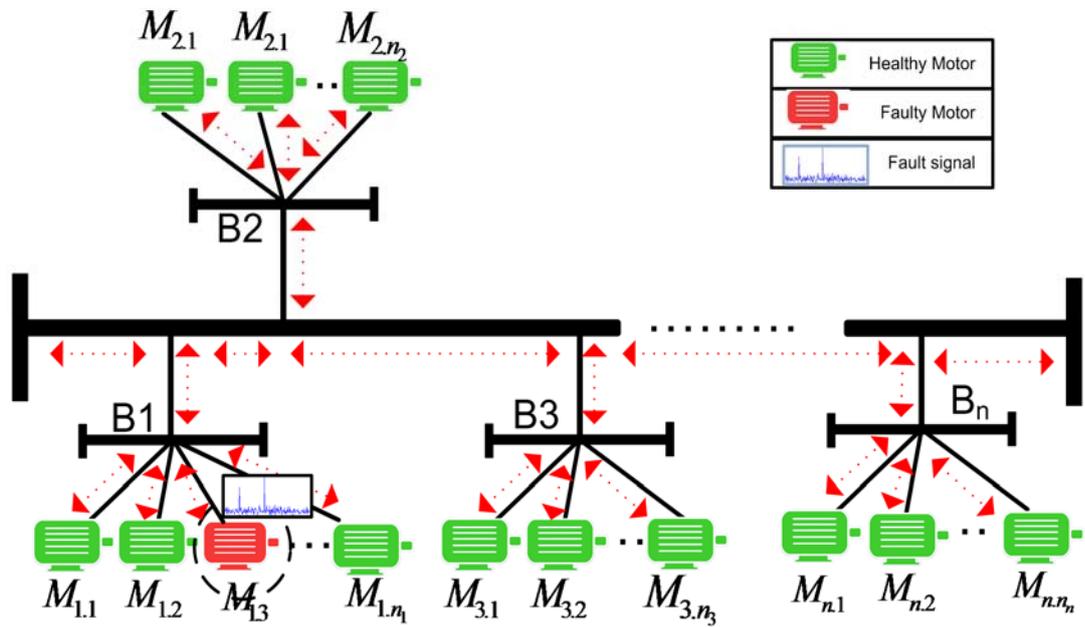


Figure 3.6: Propagation of faulty signal in induction motor network

The voltage-drop phasor  $V_{drop}$  for a section of line with impedance  $Z$  and carrying electric current  $I$  is given by:

$$V_{drop} = I.Z \quad (3.11)$$

In distribution systems, it is the arithmetic difference between the sending and receiving end voltage which is the more functional voltage value [119]. In Figure 3.6, the resultant voltage at bus 1 for a nominal frequency caused by faulty Motor 1.3 can be calculated by a specific motor, as follows, within the main power bus [120]:

$$V_{main} = \{V_{B1}, V_{B2}, V_{B3}, \dots, V_{Bn}\} \quad (3.12)$$

$$V_{main} = \sum_{i=1}^n V_{main \rightarrow i} \quad (3.13)$$

Where,  $V_{main}$  is the resultant voltage of the distribution power-line network and  $i$  is the index for the  $i^{th}$  motor within each of the sub-buses.

The resultant voltage in bus1 caused by Motor 1.3 is as follows:

$$V_{B1 \rightarrow M_{1.3}} = I_{M_{1.3}} \cdot Z_{M_{1.3}} \quad (3.14)$$

Then the influence of the Motor 3 electric current signal on neighbouring motors (Motor 1.1 & Motor 1.2) within the same bus region is identical to that of Motor 1.3.

For that electric current, mirror theory is used for observing the electric current in different motors [120].

$$I_{M_{1.3} \Rightarrow M_{1.2} \rightarrow M_{1.1}} = \frac{V_{B1 \rightarrow M_{1.3}}}{Z_{M_{1.1}} + Z_{M_{1.2}}} \quad (3.15)$$

As described earlier, the attenuation of the signal in sub-bus 1 is directly proportional length of the cable within the main distribution power-line network, and then the influence of a faulty signal on bus 2 & 3 is less than bus 1. The resultant influence on the main power bus and sub-bus 1 can be derived from Equation (3.12).

$$V_{B1 \rightarrow main} = V_{B1} + I_{main} \cdot Z_{main} \quad (3.16)$$

$$V_{B1 \Rightarrow main \rightarrow B2 \rightarrow B3} = V_{B1} + (I_{main} \cdot Z_{main}) | V_{B1} + (I_{B2} \cdot Z_{B2}) | V_{B1} + (I_{B3} \cdot Z_{B3}) \quad (3.17)$$

In the same mechanism, the resultant influence on the main power bus and other sub-buses can be calculated easily. The influence of this faulty signal on other motors can sometimes be the cause of corruption in the signal behaviour. This then manifests an indication of the fault features. Consequently, it may change the behaviour of different characteristics, resulting in dropping the voltage within the same bus or the whole power-line network [123]. Through the electric current mirror theory, a faulty signal can easily be propagated to other neighbouring motors throughout the power-line network at the given supply frequency. Estimation of the static impedance of motors at the given supply frequency can be formulated through the following equation [121]:

$$Z_{B_n M_n} = \sqrt{\frac{V_{sf}}{P_{sf}}} \quad (3.18)$$

Where  $V_{sf}$  &  $P_{sf}$  are defined as constants for filtering the frequency.

The presence of a rotor fault will lead to imbalance of the rotor electric current and produce a reverse magnetic field frequency. This frequency relates to

an inverse electric current sequence component at frequency  $-sf_1$ . This inverse electric current sequence will be reflected on the stator side as well, and produce a left side frequency  $(1-2s)f_1$  as shown in following Figure 3.7.

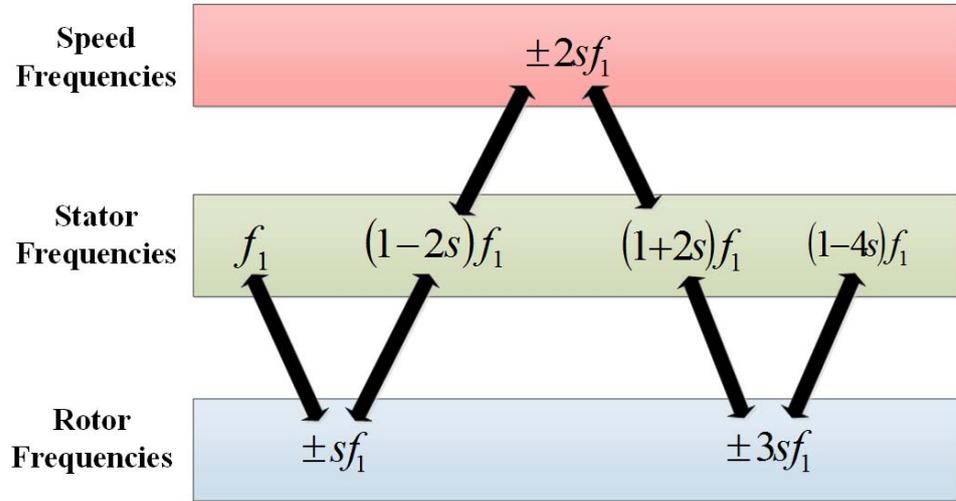


Figure 3.7: Frequency propagation for a rotor fault.

### 3.7.1.3 Attenuation Influence Measurement of Propagated Signal in a Power-Line Network

Fault signal can travel within the power-line network; and may suffers from attenuation caused by the characteristics of the power-lines, and reflection at the joint connections of power-lines, due to the mismatch of impedances at the connections. The propagation delay between the multiple paths of the signal can result in disturbance of the signal. Thus, attenuation is a very significant factor because it decides the signal strength as a function of distance, therefore, it plays an important role in validating the locations of faulty motors with in the network [107].

The attenuation of signals transmitted through the power-lines is closely related to the length of the dedicated path it takes to reach the other motors. The relationship between signal attenuation, length and carrier frequency is given by [127]:

$$A[f_c, F_D] = e^{-[(p_0 + p_1) \cdot f_c^k] \cdot F_D} \quad (3.19)$$

Where,  $f_c$  is the supply carrier frequency;  $F_D$  = Total distance of dedicated path points between buses;  $p_0$  &  $p_1$  = Attenuation parameters;  $k$  = exponent of the attenuation value factor.

In Equation 3.19, the parameter values of  $k$ ,  $p_0$  and  $p_1$  are related to the features of the power-line network, which can be obtained from the calculated propagated faulty signal within the network.

With reference to Figure 3.5, the total impedance is calculated from junction  $S_1$  to  $S_n$  of the main power line network as follows [128]:

$$Z_{j1} = Z_{L_1} + Z_{L_2} + \dots + Z_{L_N} \quad (3.20)$$

The signals propagated in all power-line network need to be efficiently dissipated. With reference to Equation 3.11, the signal generated within the individual bus can cause a voltage drop, and total impedance of Bus 1 can be calculated as:

$$V_{B1} = I_1 \times Z_{B1 \cdot f_c} \quad (3.21)$$

$$Z_{B_N \cdot f_c} = \frac{V_{B_N}}{I_{B_N}} \quad (3.22)$$

### **3.8 The Role of Neural Network Architecture within a Distributed Multiple Motor Environment**

The role of an artificial neural network at the distributed motor level is to simplify complex situations, and is capable of cancelling the suspected network bias of interfering due to the normal running process procedures of electric motors, or the intervention of environmental noises. It would be helpful to discriminate between the different but similar fault symptoms at multiple sensing points, and diagnose the correct fault type. It is very difficult to visualise just the signal at multiple points. This research has demonstrated a numerical features-based ANN solution to utilise available fault indices values from multiple sensing points, and diagnose fault locations in different situations.

Based on this distributed ANN architecture, a number of significant inputs are considered in this research. A network of different architectures of hidden layers has been assumed according to the dynamic behaviour of each motor and their feature characteristics. Output layers consist of two parts; one would consider decision at the motor level and the second would validate the results from each motor level and the overall power-line network analysis, to identify fault generating buses by motor and fault type. Development and simulation of ANN in a distributed environment will be discussed in detail in chapter 5.

### 3.9 Chapter Summary

This chapter has formulated different areas relating to the concept formulated around distributed motor network theory and fault-type diagnosis, as follows:

- A distributed industrial motor network model was presented to show the interference of faulty signals with other network's motors. This model further simulates this using Simulink in chapter 4 to prove the concept
- Key factors discussed have a significant influence on electric current signals and may contribute to confusion in fault diagnosis
- Sampling and classifying abnormal sideband components of motor waveforms was formulated to analyse the motor electric current signal to identify the characteristics of each targeted faults
- Multiple sensing point's measurement steps have been discussed in detail to identify the origin of significant components over a distributed motors network
- Demonstrating a framework of NN architecture at the motor level that consists of dynamic hidden layers associated with every bus in a power-line network, which deals with respective bus motors-condition and analyses the condition of the overall network. The role of NN in a distributed motor environment then eases out any detected network suspected bias

## MOTOR NETWORK MODELLING AND SIMULATION

### 4.1 Introduction

This chapter discusses a simulation model of multiple motors connected in a distributed motor network. Propagation of faulty signals over the network of multiple motors has been represented to reflect their significant influence on healthy motors. The relevant mathematical modelling is discussed to support the mechanism of motor model development. Finally, two case studies are presented on the injection of faults into a power-line network and the behaviour change of different characteristics of multiple motors under the influence of noisy and rotor fault signals.

### 4.2 Mathematical Modelling of a Network of Induction Motors

Figure 4.1 represents a typical circuit model for an induction motor, as suggested in reference [107]. The mathematical representation of this model is expressed by Equations 4.1 for the stator, 4.3 for the rotor and 4.5 for the mechanical portion [107].

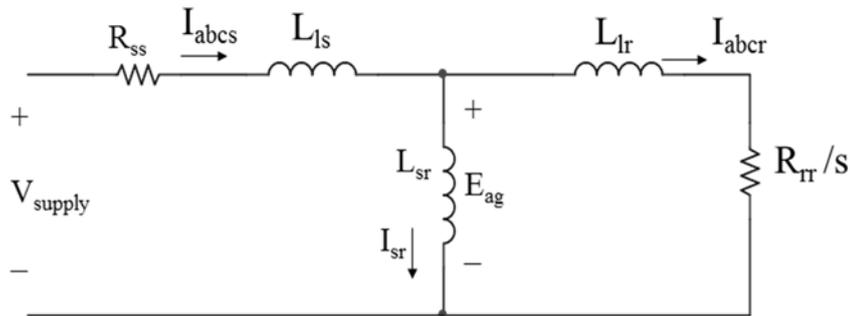


Figure 4.1: Per-Phase Equivalent Circuit of an Induction Motor

$$V_{abcs} = R_{ss} I_{abcs} + \frac{d(\lambda_{abcs})}{dt} \quad (4.1)$$

$$V_{abcs} = \begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix} \quad i_{abcs} = \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} \quad \lambda_{abcs} = \begin{bmatrix} \lambda_{as} \\ \lambda_{bs} \\ \lambda_{cs} \end{bmatrix} \quad (4.2)$$

$$V_{abcr} = R_{rr} I_{abcr} + \frac{d(\lambda_{abcr})}{dt} \quad (4.3)$$

$$V_{abcr} = \begin{bmatrix} v_{ar} \\ v_{br} \\ v_{cr} \end{bmatrix} \quad i_{abcr} = \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} \quad \lambda_{abcr} = \begin{bmatrix} \lambda_{ar} \\ \lambda_{br} \\ \lambda_{cr} \end{bmatrix} \quad (4.4)$$

$$\frac{d\omega_{rs}}{dt} = \frac{P}{K} (T_{em} - T_{Ld}) \quad (4.5)$$

$$\omega_{rs} = \frac{d\theta_{ra}}{dt} \quad (4.6)$$

$$T_{em} = I_{abcs} \left( \frac{dM_{sr}}{d\theta_{ra}} \right) I_{abcr} \quad (4.7)$$

Where,

$V_{abcs} = [V_{as}, V_{bs}, V_{cs}]$  stator voltage matrix (vector);  $V_{abcr} = [V_{ar}, V_{br}, V_{cr}]$  rotor voltage matrix (vector);  $R_{rs}$  = Rotor and Stator resistance matrix;  $I_{abcs}$  = Stator electric current matrix;  $I_{abcr}$  = Rotor electric current vector;  $\omega_{rs}$  = Rotor mechanical speed;  $K$  = Rotor moment of inertia;  $T_{em}$  = Electromagnetic Torque;  $T_{Ld}$  = Load Torque;  $\theta_{ra}$  = Rotor angle position;  $M_{sr}$  = Mutual Stator Rotor Inductance;  $\lambda_{abcs}$  flux that links stator windings ;  $\lambda_{abcr}$  flux that links rotor windings;  $R_{ss}$  = Stator winding resistance;  $R_{rr}$  = Rotor winding resistance;  $L_{ls}$  = Stator leakage inductance;  $L_{lr}$  = Rotor leakage inductance;  $L_{sr}$  = Mutual inductance;  $s$  = Slip.

The above Equations 4.1 - 4.7 relate to healthy and faulty induction motors. Rotor-related faults influence the resistivity of rotor winding resistance  $R_{rr}$ , rotor leakage inductance  $L_{lr}$  and mutual inductance  $L_{sr}$ . These parameter values depend on the rotation of the rotor (angular rotor position) as well.

Existing approaches describe common features for combined induction motors by using a load-power averaging technique from their respective circuit characteristics in the same power bus [107]. An efficient modelling of multiple motors is presented by Kataoka et al [120], which consists of the parameters of the per-phase equivalent circuit of two induction motors on the same power bus and the same load level. Based on the same procedure of multi-motor model and the case study discussed in this chapter, the aggregation of multiple induction motors on the same power bus line is presented in Figure 4.2.

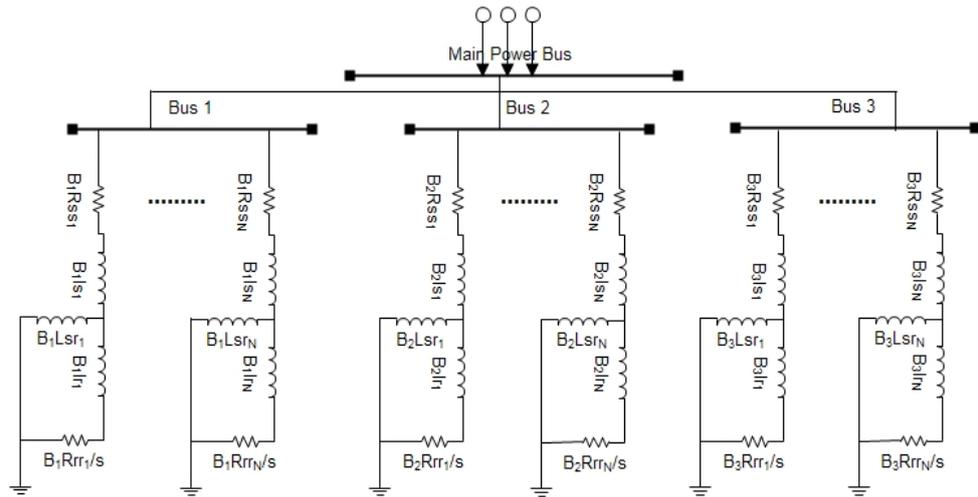


Figure 4.2: Network modelling with multiple motors on different power buses

Network impedances cause deviation of the main power terminal voltages of the machines from their rated values. In order to obtain an accurate specification of overall network voltage performance, those of each individual machine must be modified according to the variations of their terminal voltages [126]. At the rated frequency, the slip, which affects the input impedance of each machine, is a function of the machine's terminal voltage. As a result, the network admittance matrix is voltage-dependent and the system's nodal matrix equation is non-linear. However, Newton-Raphson's iterative method can be used to solve the non-linear network matrix equation for the voltages at the bus bars [127]. To form the network admittance matrix, the input admittance of each machine must first be found.

### 4.3 Multi-Motor Network Simulation Model

This research used the MATLAB SimPowerSystem for development of a distributed motor network and targeted faults so that frequency domain analysis could be performed. For validation of this concept with test bed, we have modelled a similar characteristics case study in Simulink discusses in Chapter 3. The model consists of a main power-line and has three sub-segmented bus bars directly connected with the main power bus. A total of nine motor models are connected with each sub bus, as shown by Figure 4.3. Single power motors are used for simulation at 50 Hz supply frequency, impedance (27.5) and voltage (220) values. All other individual motor parameters used for modelling are described in Table 4.1.

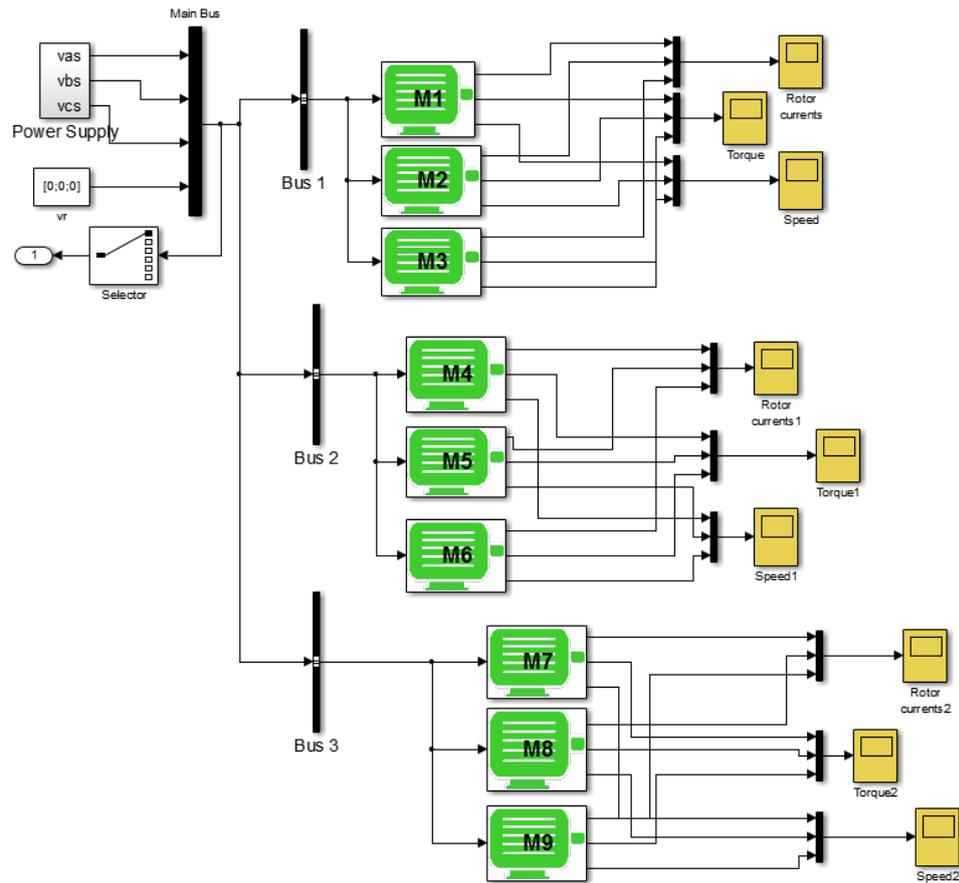


Figure 4.3: Simulation multi-motor network model

Table 4.1: Parameters used for motor modelling

	<b>Induction Motors Specifications</b>								
	<b>Bus1</b>			<b>Bus2</b>			<b>Bus3</b>		
	<b>M1</b>	<b>M2</b>	<b>M3</b>	<b>M4</b>	<b>M5</b>	<b>M6</b>	<b>M7</b>	<b>M8</b>	<b>M9</b>
Output [w]	15w	15w	25w	25w	15w	15w	15w	15w	25w
Current[Amps]	0.26-0.33	0.26-0.33	0.54	0.54	0.26-0.33	0.26-0.33	0.26-0.33	0.26-0.33	0.54
Speed [rpm]	1200 - 1500	1200 - 1500	1250 - 1550	1250 - 1550	1200 - 1500	1200 - 1500	1200 - 1500	1250 - 1550	1200 - 1500
Stator winding resistance $R_{ss}$	0.68 37	0.68 33	0.63 37	0.682 0	0.676 0	0.655 5	0.667 0	0.683 7	0.683 3
Rotor winding resistance $R_{rr}$	0.45 1	0.45 5	0.44 1	0.444	0.460	0.451	0.451	0.423	0.433
No. of poles	4	4	4	4	4	4	4	4	4
No-load	1.0	1.0	1.70	1.70	1.0	0.0	1.0	1.0	1.70
Full-load	1.25	1.25	2.0	2.0	1.25	1.25	1.25	1.25	2.0

A network of motors connected to different bus bars with all sub-buses connected with the main power bus is shown in Figure 4.3. The network impedances cause a deviation of the main power terminal voltages of the machines from their rated values. Impedance of each motor to the main power bus is set to resistance ( $1\Omega$ ) and impedance of each sub bus is set to  $0.6\Omega$  (inductive) and  $0.6\Omega$  (resistive).

In order to obtain an accurate specification of the overall network performance, those of the individual machines must be modified according to the variations of their terminal voltages. At the rated frequency, the slip which affects the input impedance of each machine is a function of the machine's terminal voltage. As a result, the network admittance matrix is voltage dependent and the system's nodal matrix equation is non-linear. For each single motor among all the bus bars, it is presumed that all individual motor parameters are known and based on the two operating states. These are at no-load and full-load locked rotor. At the first operating condition, we presume that the value of the slips is zero for all induction motors connected to different buses.

#### **4.4 Fault Injection Model Block at Motor Level**

A block diagram given in Figure 4.4 was developed using Simulink with the flexible dynamic induction motor model through the Simulink-Matlab function (*s-function*). This is based on the absolute reference frame derived from a standard induction motor model through standards equations. This block diagram contain several sub blocks that consist of stator current, rotor current, torque, speed and leakage characteristics properties. The values of these blocks can be associated with the parameters file and can be changed easily according to requirements. A fault block is used to inject the fault components into the rotor electric current. This fault component contains the fault features through the representation of the spectrum relevant to that fault.

The fault-related block in Figure 4.4 contains fault events that are modelled by multiple frequency sets at different values of amplitude and damping measurements. To validate the motor model, waveforms of multiple motors in a transient state were measured, and all motors assumed to run at no-load to full-load operating conditions at different speeds, based on data given in Table 4.1. These parameters were collected from the developed test bed motor environments data sheet, and will be discussed next Chapter 5.

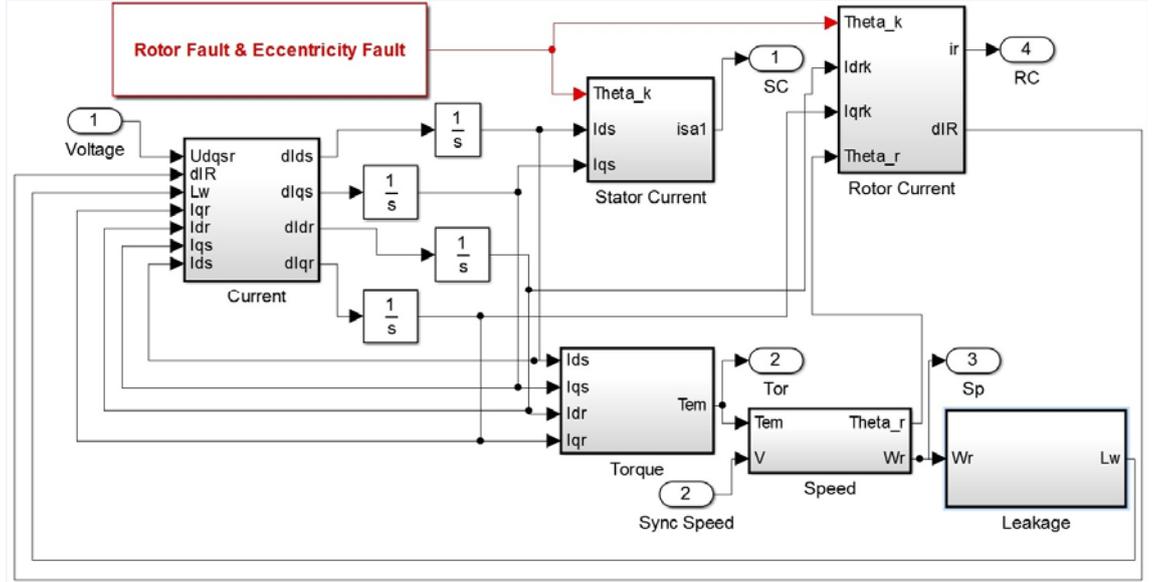


Figure 4.4: Block diagram of Simulink model of each motor with fault injection functionality

#### 4.4 Rotor and Eccentricity Faults Formation Model

Few methods have been reported in the literature that dealt with simulation of rotor fault. These are:

- Change of  $R_{rr}$  in the fault loop [128]
- Change of  $L_{lr}$  in the fault loop [123]
- And Combination of changing value of  $R_{rr}$  and  $L_{lr}$  values [124]

This research has employed the method change of  $R_{rr}$  in the fault loop. The following matrix is derived from the literature to show the system rotor loop for targeted faults [128].

$$R_{rr} = \begin{bmatrix} R_0 & -R_b & 0 & \cdots & -R_b & -R_e \\ -R_b & R_0 & -R_b & \cdots & 0 & -R_e \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ -R_b & 0 & 0 & \cdots & R_0 & -N_r R_b \\ -R_e & -R_e & -R_e & \cdots & -R_e & -N_r R_e \end{bmatrix} \quad (4.8)$$

Where,  $R_b$  = rotor bar and end ring resistance;  $R_e$  = air gap eccentricity resistance;  $N_r$  = number of bars.

The reason for choosing this method is the possibility of change in the fracture bar resistance. It permits simulating a motor with multiple partially break bars through changing parameters. If we consider breaks of rotor bars, the

resistance is higher, not equal to infinity ( $\infty$ ). The rotor bar electric current flows are modelled through a laminated rotor core. In the rotor fault model, the -broken bar resistance  $R_b$  value is greater than the healthy bar resistance. Change of the rotor resistance is caused by a fraction between the bar that has an influence on the voltage decrease and vibrations. Air-gap eccentricity fault can easily be created in the model by changing the air-gap eccentricity resistance  $R_e$  value in file “*parameters.m*” in *Appendix 1*.

#### **4.5 Injection and Propagation of Faulty Signal at Distributed Level**

With reference to Figure 4.4; initially, the model was employed to inject rotor and eccentricity faults into the motor level to observe the behavioural characteristics of each motor in a faulty condition. In this section, propagation of a fault over the network and its manifestation into other motor electric current has been carried out. For this purpose, we used a Simulink *s-function* block for the signal propagation through changing motor parameters and attached to the faulty block (Figure 4.5). By implementing this method, a faulty signal can easily be injected into other motor rotor electric currents and the necessary frequency sidebands at certain points. This method has been utilized throughout the development of this distributed motor model. The advantage of this method is the ability to change the configuration parameters according to the required scenarios. A MCSA fault has been demonstrated as a set of frequency-amplitude pair values with a series of impedances, to check and observe fault signals.

Electric current of the targeted motors has been logged using 25000 samples over three seconds. The first two seconds is considered start-up transient and hence was removed. The remaining data samples are compatible with the defined resolution and accuracy. The data acquisition tools block does not provide the choice to reduce the signal length resolution. Therefore, all measurements were taken with the same sampling rate to get the sidebands round the fundamental frequency. The same sampling rate would be considered in real-time test bed data acquisition.

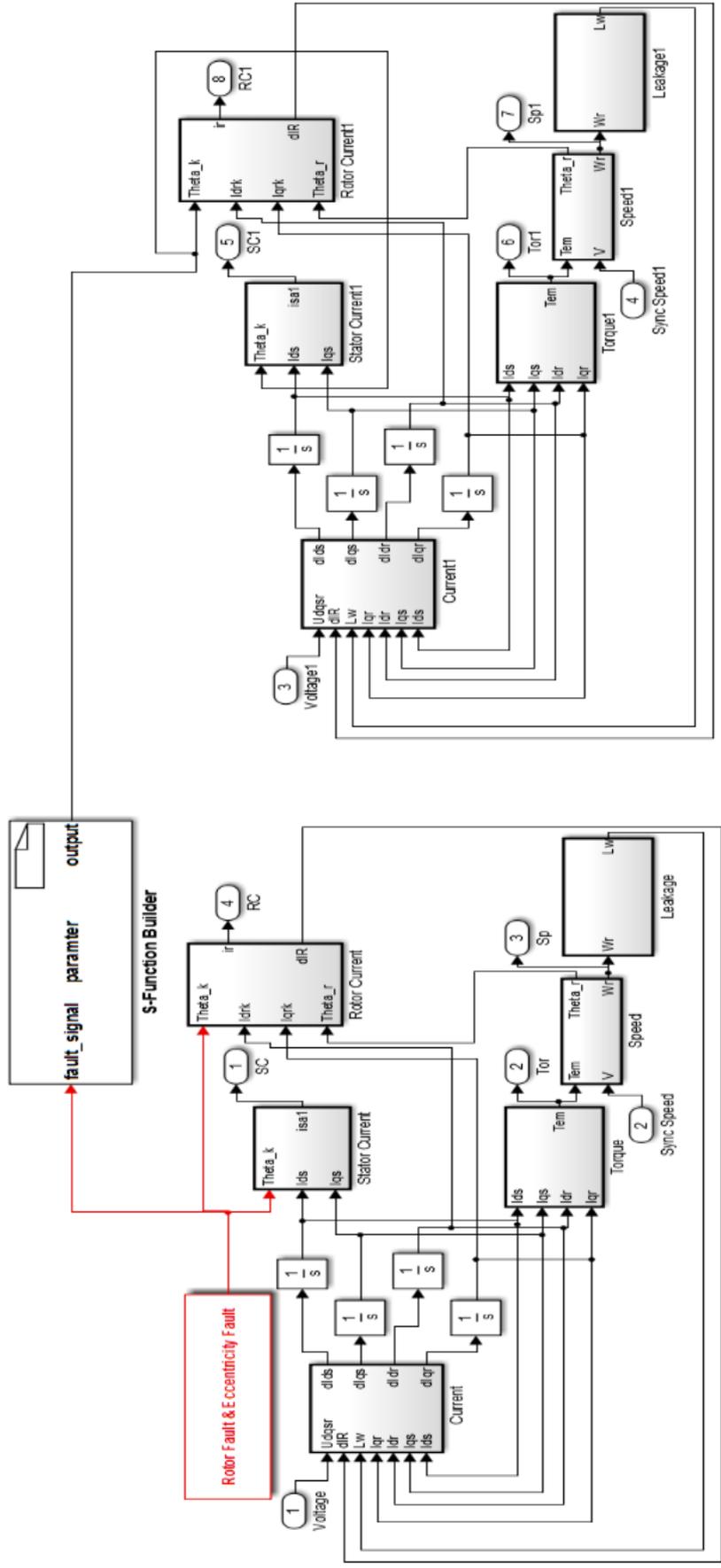


Figure 4.5: Fault signal propagation and injection model for two motors in a distributed multi-motor network

## 4.6 Case Study 1: Simulation on Noisy Signal Propagation from a Single Motor and Injected into Other Motors in a Distributed Environment

This case study demonstrates a noisy signal propagation and its influence on neighbouring motors' characteristics, according to their distance due to present the influence of noisy signal on healthy motors signal. Hence Motors 1, 3, 6 and 9 were chosen from the model (Figure 4.3) to develop the fault propagation scenario as shown in Figure 4.6 as follows:

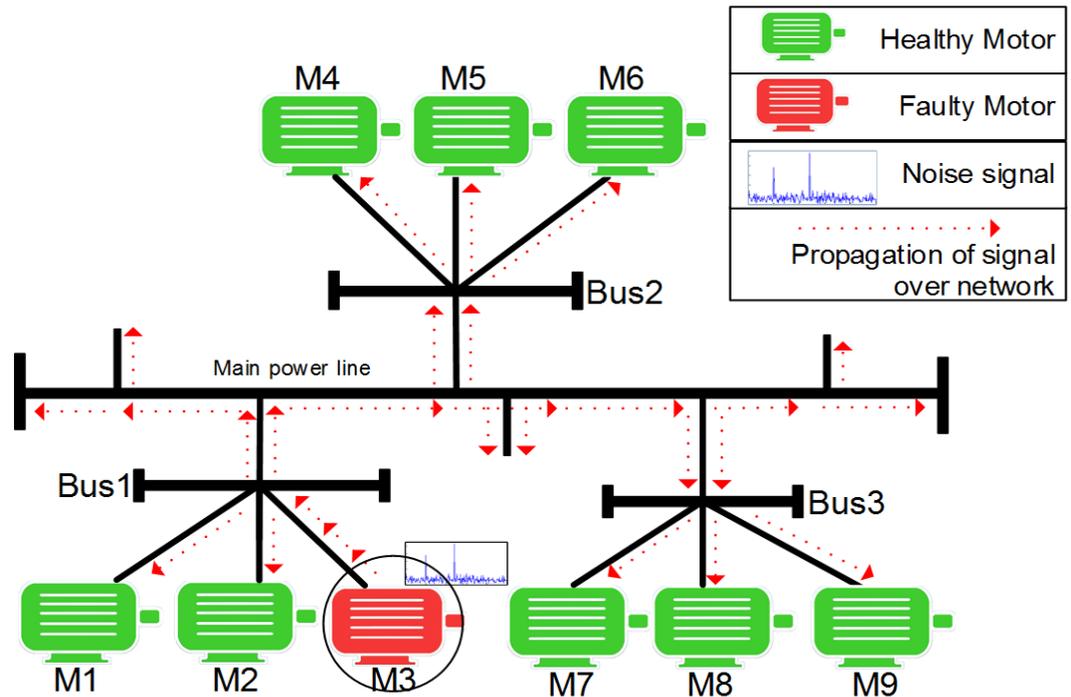


Figure 4.6: Propagation of faulty signal in a distributed motor network

In Figure 4.6, a noisy fault model (Figure 4.4) is injected into Motor 3. Figure 4.7 (a) presents the identified motor's rotor electric current response. It can be clearly observed that after a certain time-frame, the rotor electric current at no-load was running smoothly and linearly. On applying the full load on (a locked rotor) Motor 3, the rotor electric current behaviour was affected and resulted in the injection of this abnormal influence on other motor's electric current characteristics as well (see Figure 4.7 (b)). Due to the abnormal behaviour of Motor 3 during running, other motor's characteristic feature values of speed and torque also show some changes and may cause an increase in the speed and variation in the torque level curve in some cases, as shown in Figures 4.8 - 4.9 respectively.

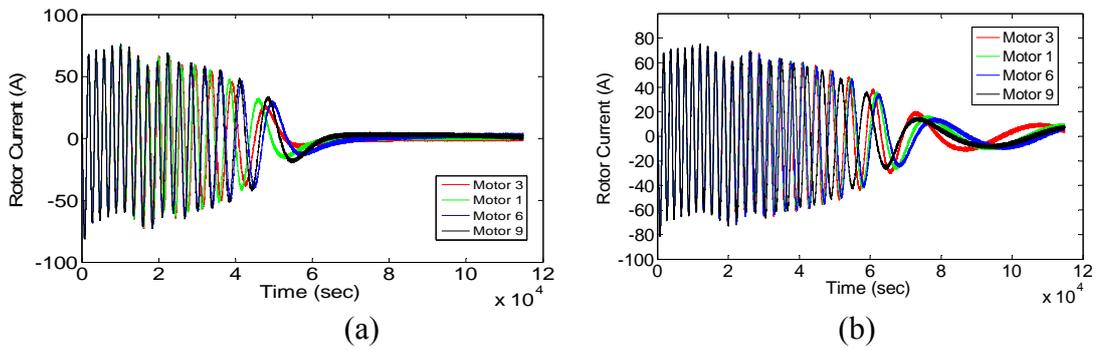


Figure 4.7. Rotor electric current comparison of four motors at (a) No-load (b) Rotor of motor 3 locked at full-load

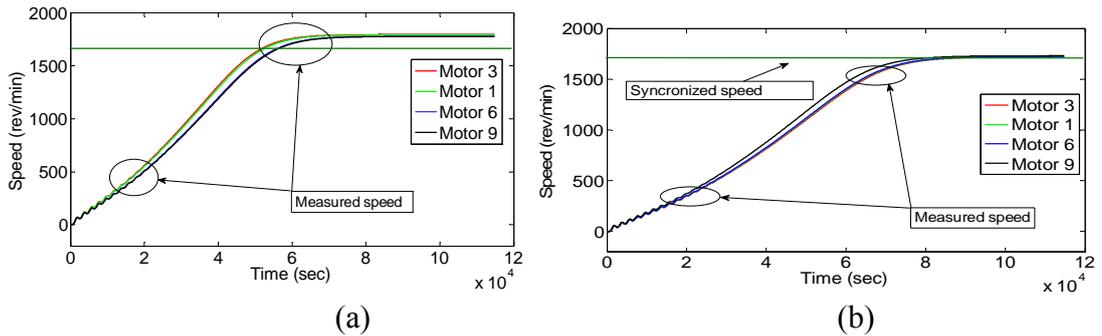


Figure 4.8: Rotor speed comparison of four motors at (a) No-load (b) Rotor of motor 3 locked at full-load

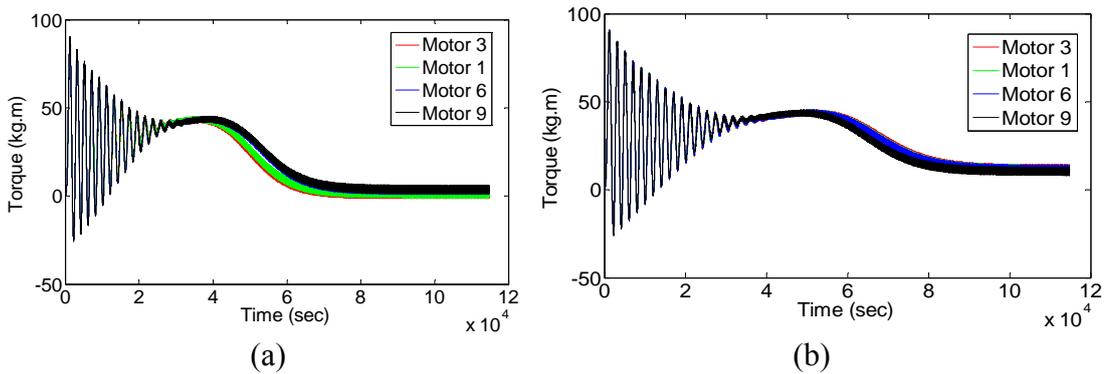


Figure 4.9: Rotor torque comparison of four motors at (a) No-load (b) Rotor of motor 3 locked at full-load

As shown in Figure 4.10, after applying Fast Fourier Transform (FFT) on the electric current signal of all motors identified, at no-load level we observed that all motor signals are represented in a normal condition and no sidebands appeared. At full- load, some significant sideband frequency points were injected at 28 Hz, 68 Hz, 76 Hz, 129 Hz and 226 Hz respectively, around the supply frequency 50 Hz (see Figure 4.11) with different amplitude values, to show their influence on multiple motors at different rates. These frequencies are supposed to show the manifestation of noise signals into healthy signals at the applied sampling frequency (25000 Hz).

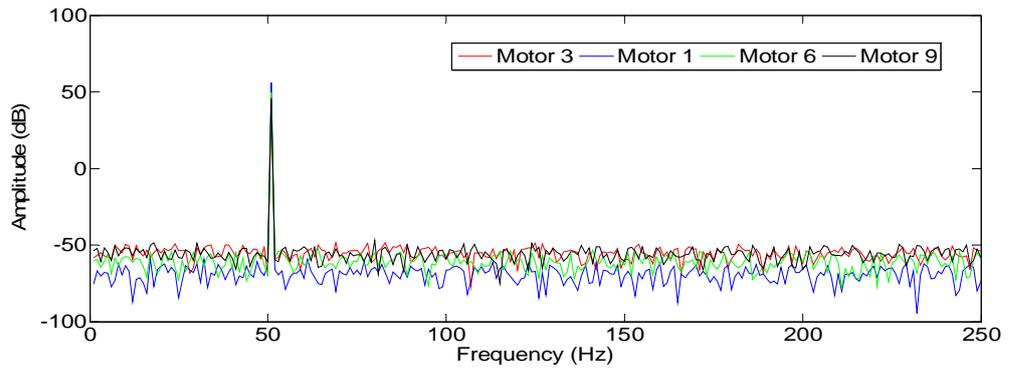


Figure 4.10: Frequency spectrum of targeted motors with no-load

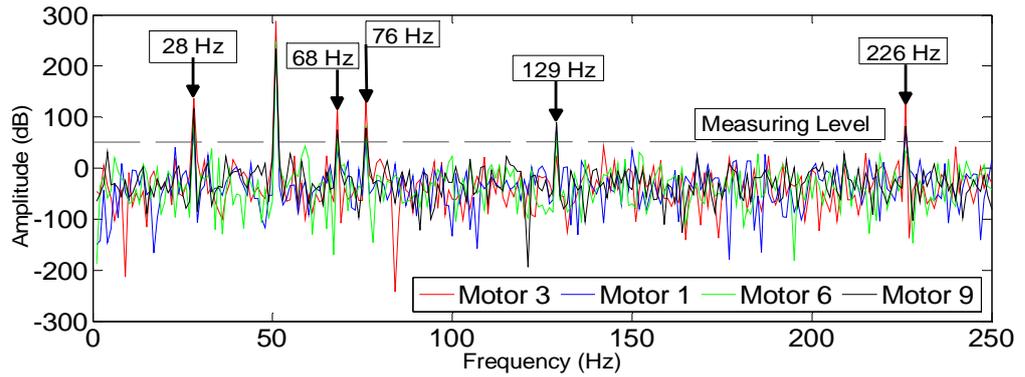


Figure 4.11: Multi-Frequency noise propagation by Motor 3 with full-load

A magnified closer view of the abnormal frequencies at 28 Hz, 68 Hz and 76 Hz, 129 Hz and 226 Hz shows the shared influence of the faulty motor on targeted other motors and a change in behaviour of the healthy signal in the Figure 4.12.

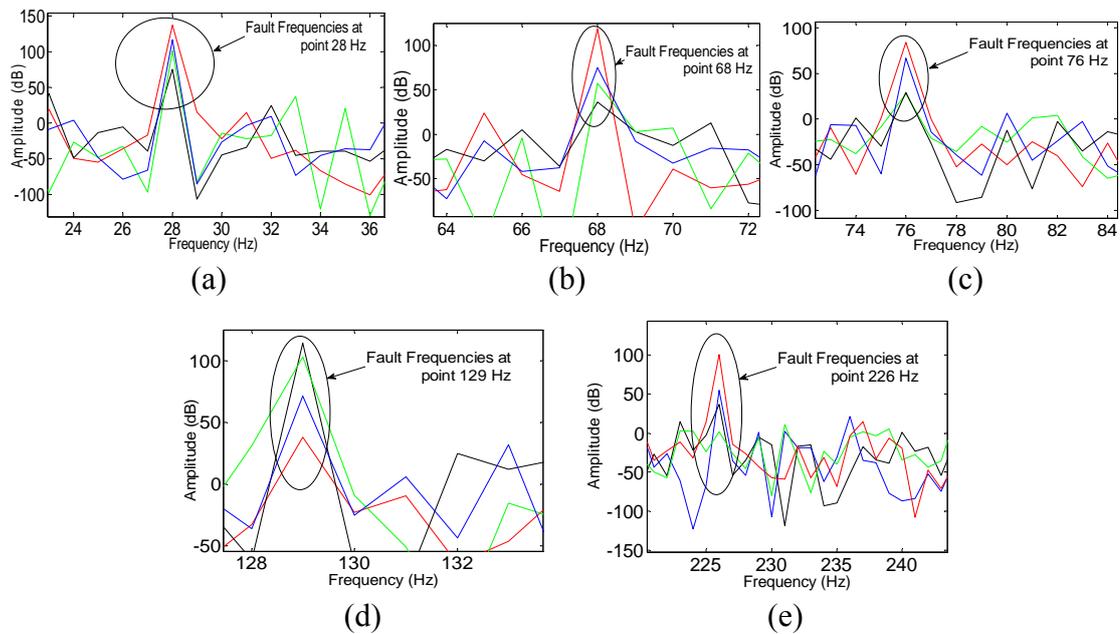


Figure 4.12: Propagation of faulty signal at significant frequency points

(a) 28 Hz (b) 68 Hz (c) 76 Hz (d) 129 Hz and (e) 226 Hz

At frequency points 28 Hz, 68 Hz and 76 Hz, it can be clearly observed that Motor 1 signal is the most affected waveform compared with the other two motors, due to their closeness to faulty Motor 3. Both Motors 1 and 3 belong to the same sub Bus 1. But at frequency point 129 Hz, we perceive that the rotor fault frequency peak propagated by the Motor 9 and Motor 6 signal was more affected with the other two motors, due to short distance compared to the other targeted motors. This is because Motor 9 was running in Bus 3 and Motor 6 in Bus 2. Even so, Motor 3 was also affected by this faulty signal and shows the fault frequency peak. Finally, at frequency point 226 Hz, Motor 3 was again responsible for propagating a faulty signal and having an influence on other buses, consistent with their distance. The difference between the amplitude ranges is high, at 129 Hz, because Motor 9 is higher powered than Motor 3. Therefore, it also propagates a signal with more strength and causes an increase in the attenuation coefficient variance within the network. These changes in multiple signals show some of the fault indices in Motors 1 and 6, but actually has no fault itself and shows the rotor fault frequency symptoms in healthy signal.

The next stage is to identify the actual source of the fault event at all frequency points mentioned, to overcome this confusion in identifying the source of the fault. From Table 4.2 and Figure 4.13, it can be determined that Motor 3 is responsible for causing the faulty signal and having an influence on other signals at all targeted frequency points, except 129 Hz, because this faulty signal was propagated by Motor 9 over the power-line network.

Table 4.2: Significant fault frequencies measuring ranges and targeted points

Motor Number	Significant Fault Frequencies Measuring Range				
	1 Hz – 50 Hz	51 Hz – 100 Hz		101 Hz – 150 Hz	151 Hz – 250 Hz
	Amp at FP 28	Amp at FP 68	Amp at FP 76	Amp at FP 129	Amp at FP 226
<b>Motor 3</b>	135	110	85	38	100
<b>Motor 1</b>	125	75	68	74	55
<b>Motor 6</b>	105	55	33	105	30
<b>Motor 9</b>	85	40	31	116	25

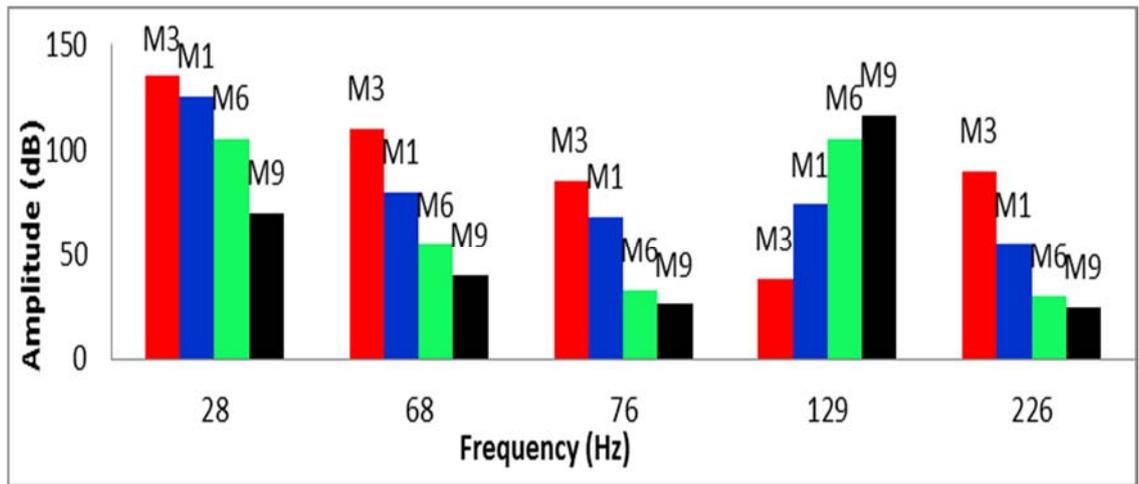


Figure 4.13: Influence of multi-frequency faulty signals on other motors

By using the propagation of a faulty signal mechanism in a distributed motor network model, we can calculate the threshold level between all possible targeted level points as a healthy point. From this threshold level, we perceive that the faulty signal is generated by the same group of motors by observing the strength of the sideband; otherwise, other point's strengths would be considered to be from other sub buses connected with the main power bus.

#### 4.7 Case Study 2: BRB and Eccentricity Fault Frequencies Injection and Observe Influence on Different Measuring Points

As discussed, faults inside the motor are related with a set of frequency sideband components. Therefore, faults are demonstrated here by a set of frequency points with different amplitude ratios. Initially, multiple fault frequency signals were injected into the model (see Figure 4.4) at multiple frequency points, with the rotor full-load level in Motor 3 through changing value  $R_b$  and  $R_e$ . To extend case study 1, the following Figure 4.14 demonstrates the concept of multiple measuring points to observe behaviour at multiple locations in-network.

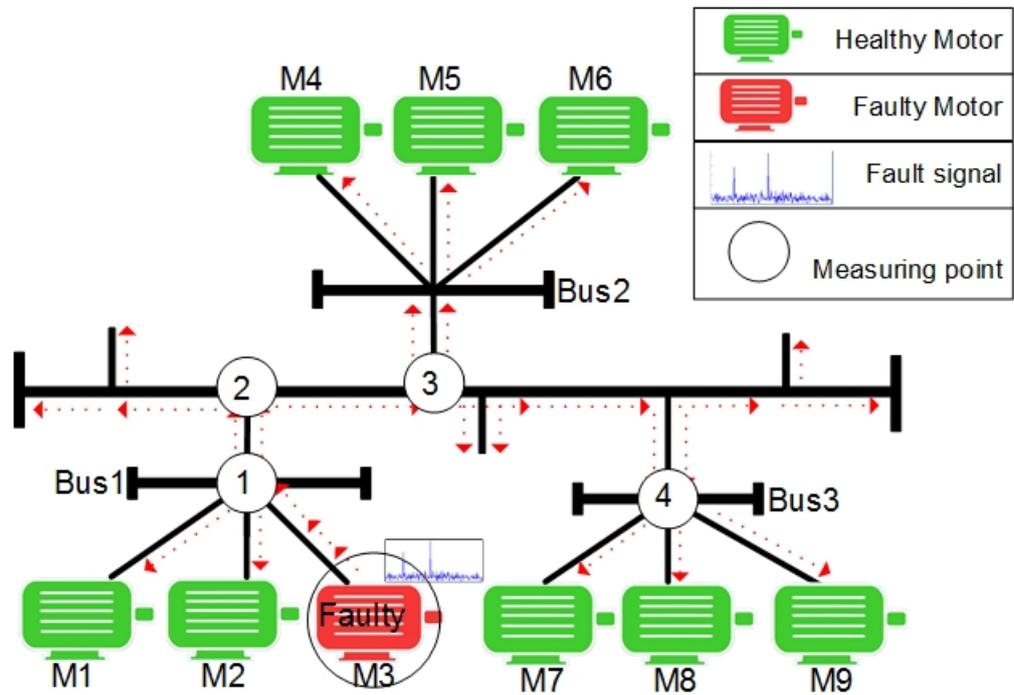


Figure 4.14: Multiple sensing point's analysis over a network

Initially, a total four measuring points were considered for case scenario development. Here, targeted fault frequencies were injected into electric current of Motor 3 in Bus 1. As in Equations 1.5 and 1.7, the rotor and air-gap eccentricity fault symptoms can be calculated at certain frequency points. By using fault formulas, in Figure 4.5, air-gap eccentricity fault-related frequencies are injected at point 4 Hz and 96 Hz in a combination of frequencies around the supply frequency. The BRB fault-associated frequencies are 46 Hz and 54 Hz, with significant amplitude values. To create complexity in visualisation and identify the fault symptoms, a noise was injected at sensing point 4 and the influence of other motor spectrums was accordingly observed. This process was very helpful for showing the validity of the model developed and to compare between real-time and simulated data in Chapter 7.

As in Equations 1.5 and 1.7, the rotor and eccentricity faults symptoms can be calculated at certain frequency points by using the relevant equation. In Figure 4.15, the fault symptoms sideband components had a significant observable impact on the electric current spectrum of other motors. This impact is due to the propagation of a fault signal through main power-line.

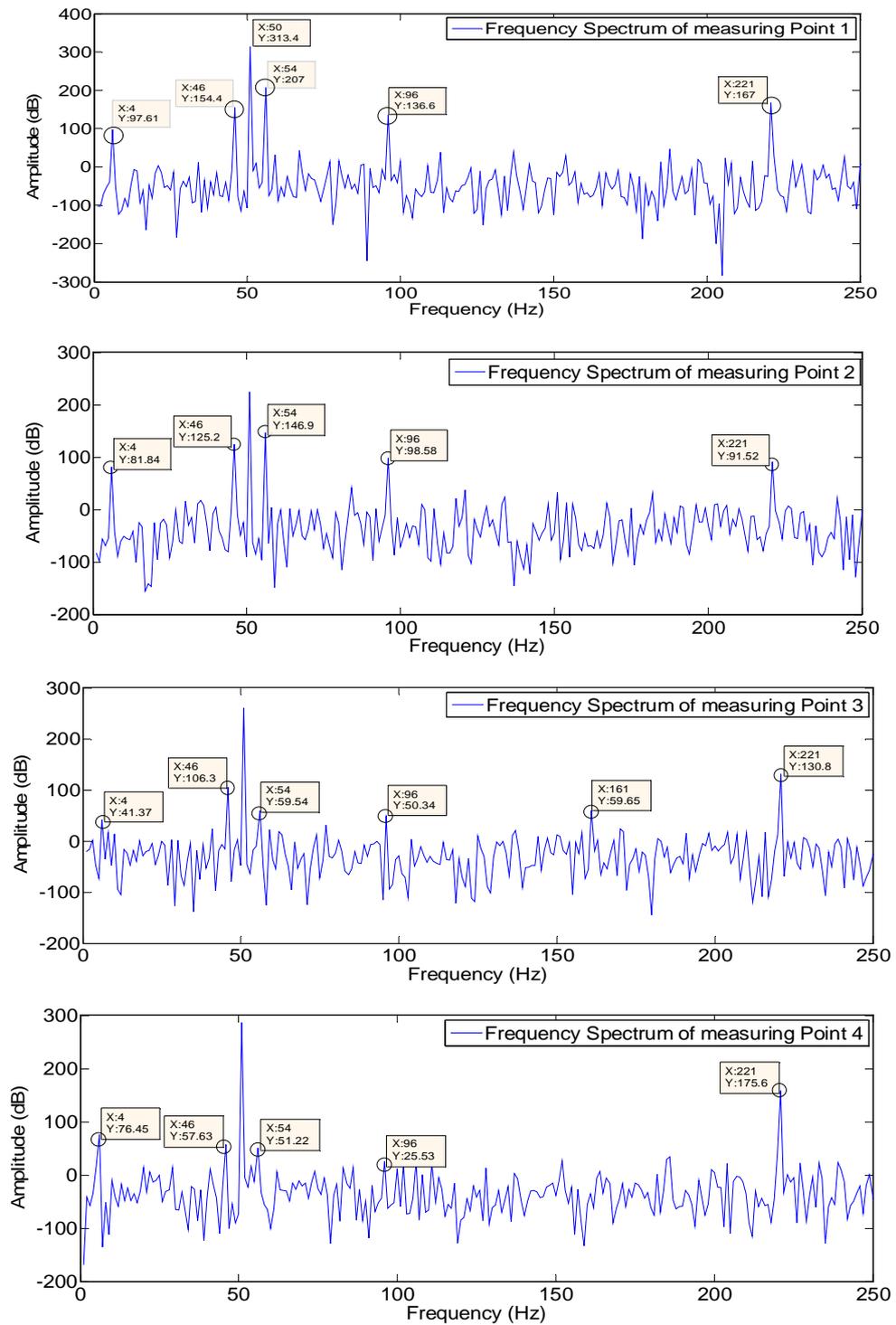


Figure 4.15: Frequency Spectrum measurement at targeted Points with fault components

The amplitude of the fault signal is shown in Figure 4.15 at each measuring point. Figure 4.16 provides a comparative information at each measuring point about the injected faults and provides a better picture for estimating the degree of relationship of each measuring point with the injected faults (see in Table 4.3).

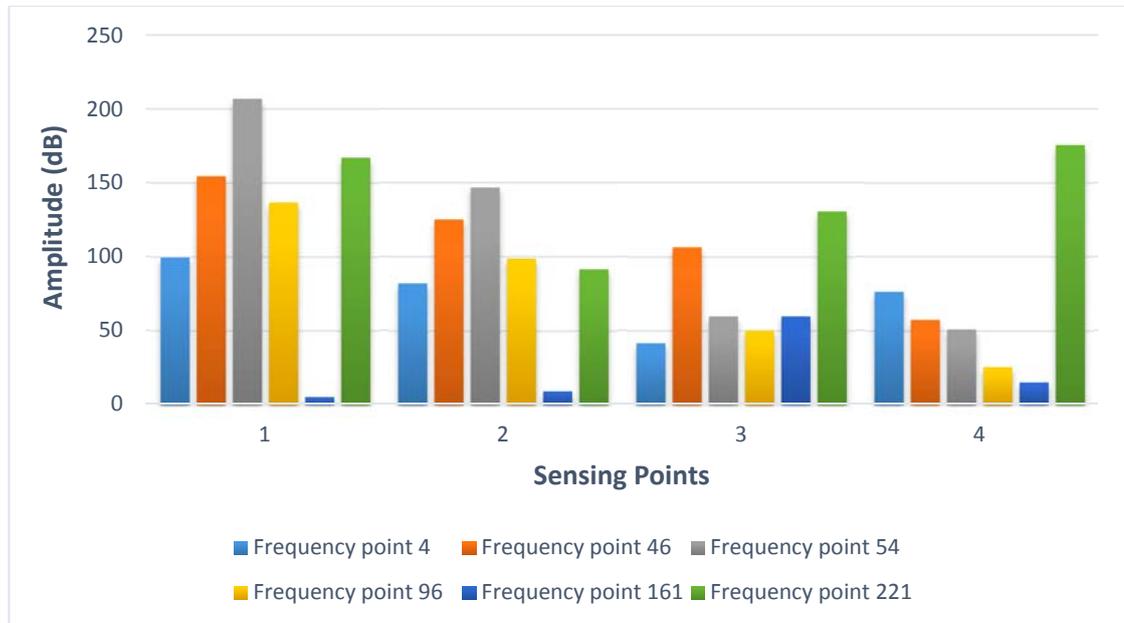


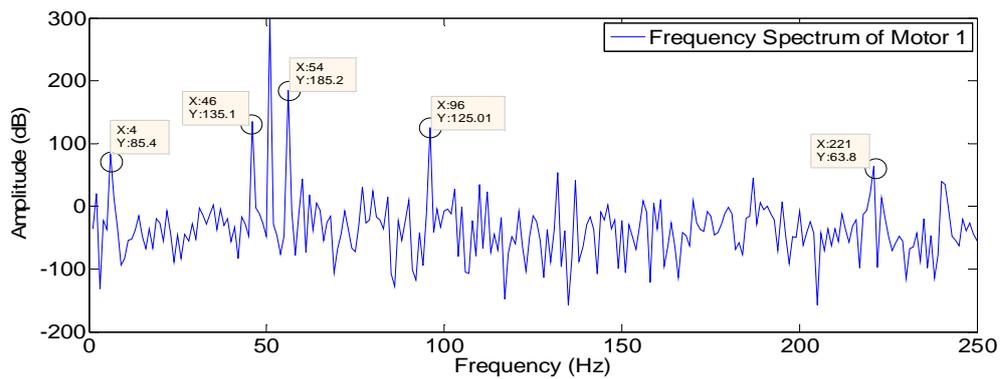
Figure 4.16: Influence of multi-frequency faulty signals on other motors

Table 4.3: Significant fault frequencies at targeted sensing points

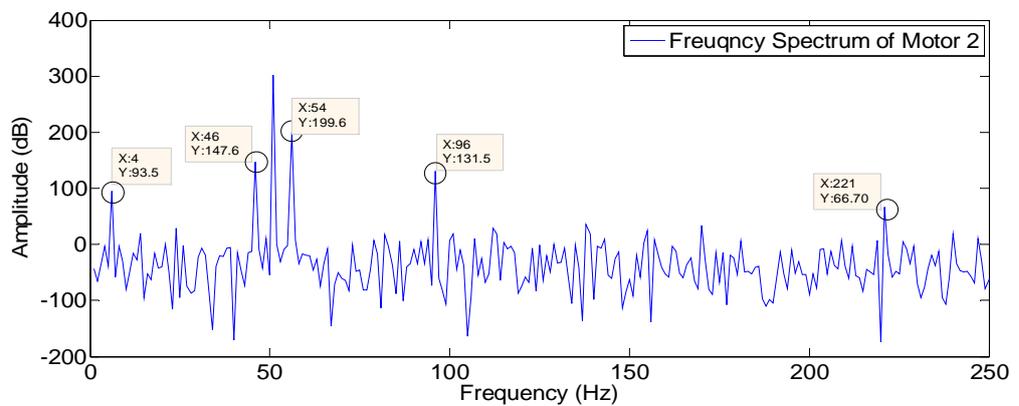
	Frequency point 4	Frequency point 46	Frequency point 54	Frequency point 96	Frequency point 116	Frequency point 221
<b>Sensing point 1</b>	99.61	154.4	207	136.6	5	167
<b>Sensing point 2</b>	81.84	125.2	146.9	98.58	9	91.52
<b>Sensing point 3</b>	41.37	106.3	59.54	50.34	59.65	130.8
<b>Sensing point 4</b>	76.45	57.63	51.22	25.53	15	175.6

As we can see in Table 4.3, at sensing point 1, the trend of signal preciously lowers at frequency points 4, 46 and 96. But at point 116, the amplitude value is too low because some noise manifests from other motors near sensing point 3, and due to low intensity, this influence can be ignored. The value at point 221 is much higher than at sensing points 2 and 3. This can be anticipated as due to the existence of own fault near sensing point 1, the abnormal behaviour of the signal has been occurring and creating a confusion at both peaks generating near sensing point 1, or travelling from sensing point 4. At frequency 116, the peak amplitude value is not very much higher but is within the range of defined threshold level. Therefore, we can assume that some fault exists near sensing point 3, or just an environmental noise travelling over the main power-line network. Frequency point 221 showed a higher value near sensing point 4, but had a significant influence on sensing point 1 when compared with the other two sensing

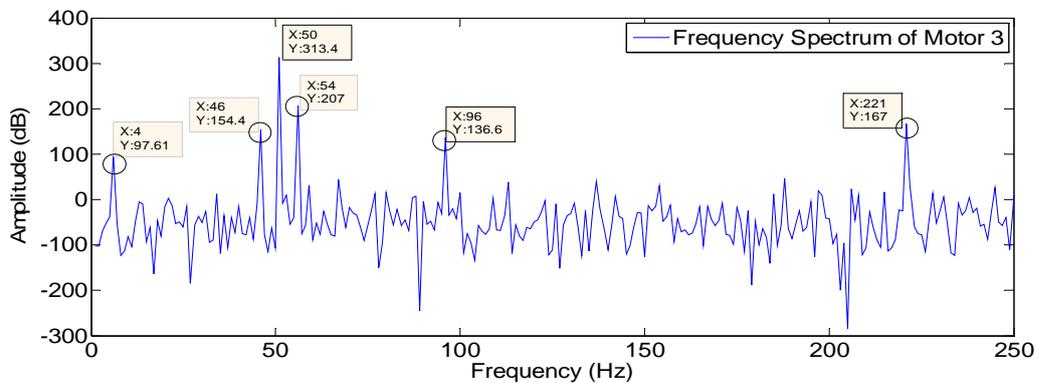
points (2 & 3). This type of confusion can create doubt to identifying the correct path of propagated signal and its source. The following Figure 4.17 shows the spectrum of multiple motors within a network. Due to some suspect behaviour near sensing points 1 and 4, motors 1, 2 & 3 were selected from Bus 1 to observe behaviour under the influence of a faulty signal. Motor 6 was targeted, from bus 2, to observe the behaviour in the middle but actually, there was nothing wrong near sensing point 3. Similarly, Motors 7 and 9 were chosen from bus 3, as a faulty signal was propagated near sensing point 4.



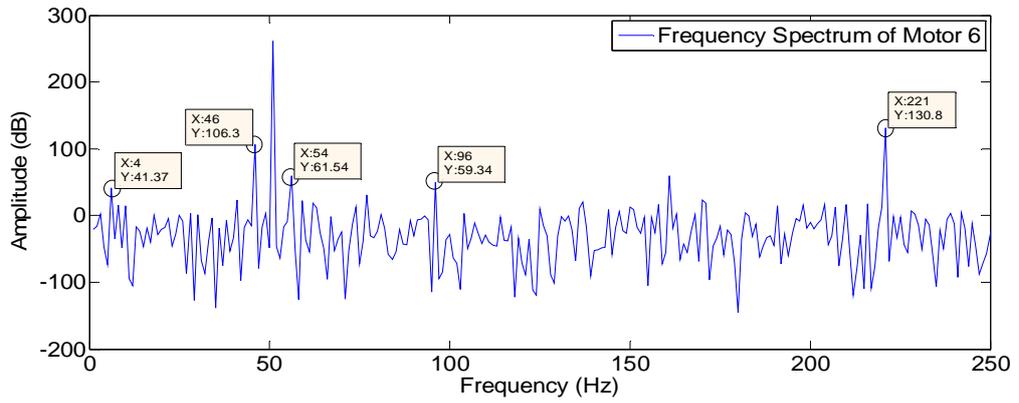
(a)



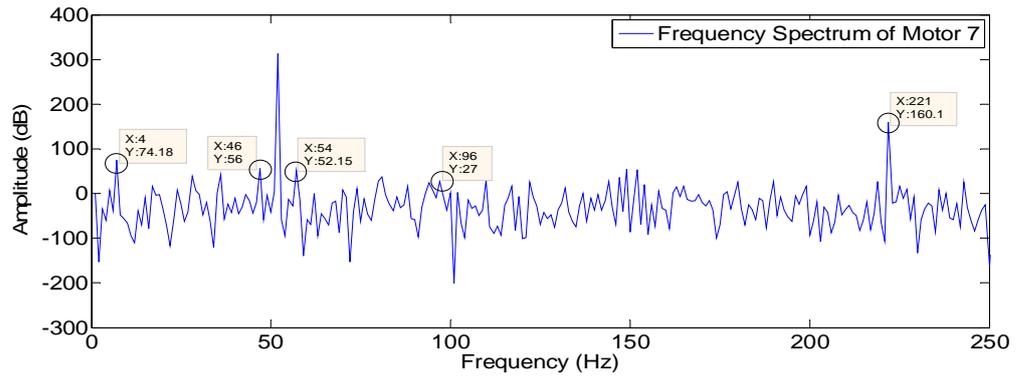
(b)



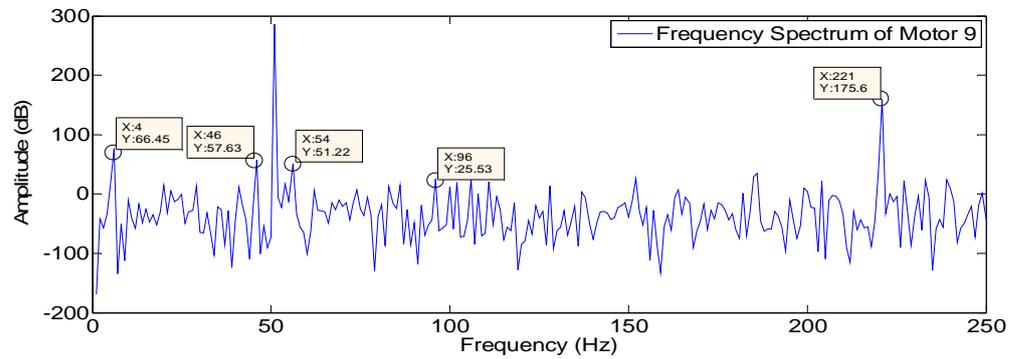
(c)



(d)



(e)



(f)

Figure 4.17 (a)-(f): Frequency spectrum results from targeted motors

From the above Figure 4.17, it can be clearly observed that in Bus 1, Motor 3 held higher values of sideband components at the targeted frequency points compared with the other two adjacent motors. The sidebands at frequencies 46 Hz and 56 Hz show strong symptoms for a BRB fault. Similarly, frequency points 4 and 96 show strong evidence of an air-gap eccentricity fault symptom. But frequency point 221 shows an irregular sideband component and, after analysis from the overall network, it can be assumed that this noisy component has become manifest through the main power-line as a signal. The influence of

that strength of faulty signal was dramatically be decreased in other motors in Buses 2 and 3. Therefore, it can be anticipated that Motor 3 in bus 1 has a BRB and eccentricity fault, along with some noise signal that was injected into this signal from another motor from Bus 3.

As discussed in Chapter 3, the strength and influence of the faulty sidebands on each motor depend on the distance between the different buses, and motor sizes as well. But due to the uncertainty existing about faulty sidebands in different motors, mere visualisation of a motor signal is not enough to decide which one the faulty motor within a network. For that, different other features also need to be considered to take a final decision about motor health by using this diagnosis method. Chapter 5 will discuss other significant features that have an important role in diagnosing the correct fault type and the source of the fault origin.

In order to further validate the simulation model and analyse the different characteristics of faulty signals in power-line network, different significant aspects are considered later including noise level, multiple motor sizes in distributed networks, measurement of attenuation factors and similarity between different faults symptoms, in Chapter 7.

## **4.8 Chapter Summary**

This chapter has demonstrated a Simulink model of multiple motors connected with a main power-line that can have a considerable combined influence on an overall system's operation. A fault-injection model block at motor level was presented to inject BRB and eccentricity faults into any motor. This model was then extended to the mechanism of propagation of faults from one motor to others. Two case studies demonstrated the concept of injection of faults at multiple measuring points, to observe behaviour at multiple locations in-network and identify faulty motors.

## ARTIFICIAL NEURAL NETWORK SOLUTION FOR FAULT DIAGNOSIS

### 5.1 Introduction

This chapter demonstrates an efficient simulation model of multiple fault diagnosis using the neural network at each motor, and distributed-level modelling. This approach has been scaled down into multiple architectures to achieve the required objectives. A set of significant fault features, such as synchronised speed, slip, significant frequency sideband peak points, amplitude values of sidebands, Root Mean Square (RMS) and Crest Factor (CF) value of electric current is used to train the ANN using a Back Propagation (BP) algorithm. A case study is presented to demonstrate that the proposed technique is able to identify fault types and location of events within a multi-motor network.

### 5.2 Artificial Neural Network Fault Type Diagnosis Solution

In a multi-frequency signal, it may be difficult to implement a general mathematical technique to identify significant features and fault location in distributed modelling. However, Feed Forward Neural Network (FFNN) methodology allows input/output mapping between nodes in a non-linear relationship of significant features [129]. This method has been utilised to recognise the abnormal representation of a signal due to the inherent capacity of the classification and generalisation process [130]. This methodology gives good performance, especially when the response time and sensitivity of the actual process presents a repetition of fault sets; and creates uncertainty in fault identification in the power-line at multiple sensing points.

As discussed in Chapter 3, two levels of a diagnosis system have been proposed at both motor and system levels to reduce ambiguities in decision making. A motor-level ANN diagnosis solution would offer fault detection at each node level. A distributed solution improve the dependability and reliability of a

fault detection solution over a network whenever no direct measuring method is available. In the following sections, first we will discuss the role of NN at the motor level and then over a network when all motors are running in parallel.

### **5.3 Neural Network Architecture and Diagnosis Solution at Motor level**

Selection of appropriate signal features is the main challenge in the selection of ANN input parameters and this may complicate the network structure. Correct feature selection increases the performance of the network and reduces the network input dimensions and training time [131].

In this research, we have chosen six significant features in diagnosing the targeted faults at each motor level. The synchronised speed of each motor, the rotor slip value, the amplitude value of each fault frequency component, the frequency Root Mean Square (RMS) and the crest factor value (of electric current signals) are preferred. The reason for considering the speed and slip of each motor as a feature is because these two parameters are associated with BRB and eccentricity faults. Without these values, target faults are difficult to diagnose in multi-motor networks. Motor slip has been presented with different notations for each fault, to estimate the associated synchronised speed at each individual significant frequency sideband point. The amplitude values of each fault frequency component show the severity of suspected frequency points for BRB and eccentricity faults. Finally, the RMS and crest factor features were chosen to identify the noise signals or other unknown fault symptoms (e.g. bearing faults and load effects that are not targeted in this research).

Location of a fault symptom within a network usually has a close association with motor speed, where the amplitude of the fault signal is contingent on the type and motor characteristics and is related to the severity of the fault [132].

Referring to equations 1.1, 1.2, 1.3, 1.5, 2.1 and 2.2 in Chapters 1 and 2; these mathematical equations are used to calculate these features and the proposed network consists of six input data values. Matlab was used to extract different feature values through a script.

For simulation, an array of notations was expressed for these features with  $\{x_1, x_2, \dots, x_6\}$  respectively. All input consists of dynamic feature values in

multiple motors that can be easily changed in the Matlab file “*parameters.m*”. Each network has one hidden layer that consists of a number of neurons according to the inserted input values. The number of nodes in the hidden layer has a direct influence on the performance of ANN. If the number of nodes is small, the network may not be trained adequately. On the other hand, if the number of neurons is too large, the network will take a long time to train. Therefore, the selection of dimensions and number of hidden layers depends on NN accuracy in initial tests. The *sigmoid* activation function *tansig* is used for training in hidden layers. The output layer provides the condition of the motor. It consists of four neuron nodes and the activation function *logsig* is used for each output. For the target output, a vector of classes is formulated as follows:

- [1;0;0;0]: for healthy condition
- [0;1;0;0]: for BRB fault
- [0;0;1;0]: for air-gap eccentricity fault
- [0;0;0;1]: for unknown fault

A multi-layer FFNN method is used in this research for type identification of faults within a network. The proposed architecture of ANN for a single motor is presented in Figure 5.1.

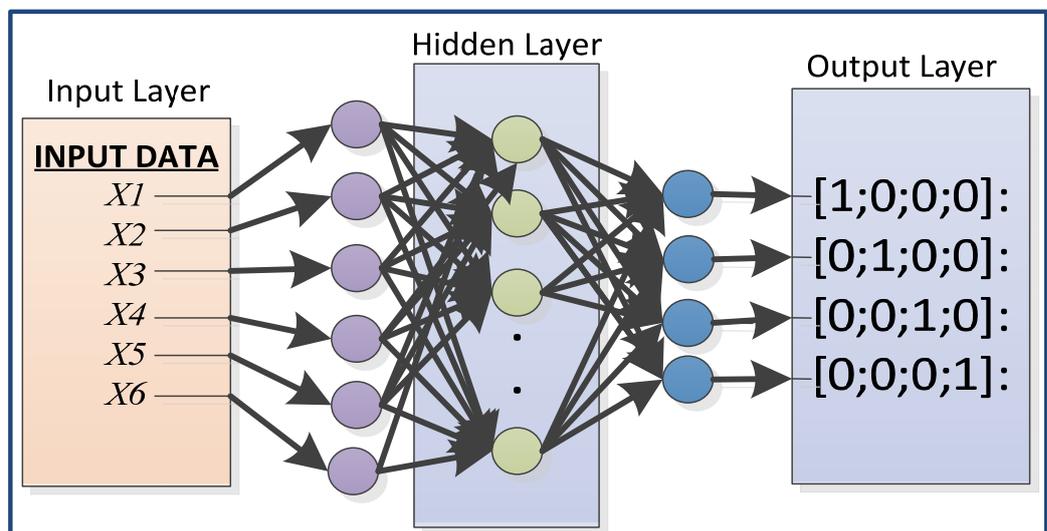


Figure 5.1: ANN Classification Network Architecture at Motor Level

On the basis of the same training mechanism, we used the same ANN classification network architecture for multiple motors. Each motor feature’s vector class was used in the input data and classified through the same network architecture.

Once the network learning process started, the neural network adjusted the weights and threshold levels automatically over all the layers, to decrease the Mean Square Error (MSE) between the targeted and actual output. For this, the Levenberg Marquardt algorithm function *trainlm* performed fast training, with intrinsic regularisation properties [91]. The misclassification and data-error rate were calculated through the Sum of Square Error (SSE). Therefore, the SSE was used to train each network using Equation 5.1 until the SSE goal was met.

$$SSE = \sqrt{\frac{1}{N} \sum_{l=1}^N \sum_{m=1}^E (X_{lm} - d_{lm})^2} \quad (5.1)$$

Where N represents the number of training Data;  $E$ = output layer units;  $X_{lm}$ = actual output; and  $d_{lm}$ = predicted output.

The performance of the resulting NN can be expressed as a target ratio, defined as the percentage of correctly identified conditions of the total, shown in the following Equation 5.2.

$$\text{Target ratio} = \frac{\text{Number of correct predictions on the data set}}{\text{Total number of points in the data set}} \times 100 \quad (5.2)$$

### 5.3.1 Fault Type Diagnosis Methodology

For the analysis of motor spectrums, initially possibly significant sideband points were considered as potential fault symptoms caused by any fault. Then each relevant motor's speed was measured and compared with the defined range in the acceptable healthy range. After that, from this measured speed, rotor slip could easily be calculated and compared with the defined set of features range. All features then were be compared with the defined ranges in the NN architecture and the condition of the motor estimated.

To decrease the level of uncertainty within the network of motors, it is mandatory to identify that whether these faulty symptoms are generated by the motor itself or coming from other motors embedded in-network. The system will take the final decision about any motor fault, not merely on the basis of a single feature's abnormal characteristic. It also compares the other feature values with the targeted values. If the other feature values show much difference from the targeted values, the motor can be considered as faulty but otherwise healthy or having an unknown fault.

In order to calculate the features from each motor's data, different range rules have been defined to calculate the training data patterns for every target vector class. In the case of features  $x_3$  and  $x_4$ , if an amplitude value varies from the actual and targeted sideband amplitude values, the system will assume the existence of a BRB or air-gap eccentricity faults symptom. But to decide the type of fault, the NN system will also compared the other features with the targeted class. Therefore, if other feature values differ more than the defined feature ranges, the motor may be considered faulty. Otherwise, the system will suspect these changes to be a signal fault, propagated through the main power-line from other motor and influencing the targeted motor. On the other hand, in the case of features  $x_5$  and  $x_6$ , if the measured value is different from the set values of the range, the system assumes these changes exist due to a suspected signal that comes from a neighbour motor.

The next step is to decide the type of fault. Therefore, spectrum sideband components will be compared with the fault symptoms range. These set ranges data contain the left-and-right peak frequency points and their amplitude values, to decide the type of fault. The amplitude values difference between the sideband and fundamental frequency will compute, but if it is higher than the set range, the system will store this value into a Matlab file and repeat this procedure with all identified sidebands. When all significant values are identified then system will compare these values with the healthy range to decide the type of targeted fault (BRB or ECE). However, if the features  $x_1$ ,  $x_2$ ,  $x_3$  and  $x_4$  values come within the set defined range and  $x_5$ ,  $x_6$  present their values as faulty, the system will assume this variation is an unknown fault.

Taking into account the speed of a faulty generating motor and the strength of its influence on other motors are the interim variables of the diagnosis process. Therefore, different architectures for training have been considered to evaluate the concept of fault diagnosis.

### **5.3.2 Neural Network Simulation Model at Motor-Level**

Matlab and Simulink were used for the development of the proposed ANN model to diagnose fault types and the location of faulty motors within motor network. Base on motor features data, the following neural network model was developed using Simulink, as shown in following Figure 5.2. For all the motors,

the same type of neural network architecture has been used at an individual-level to observe the condition of each motor.

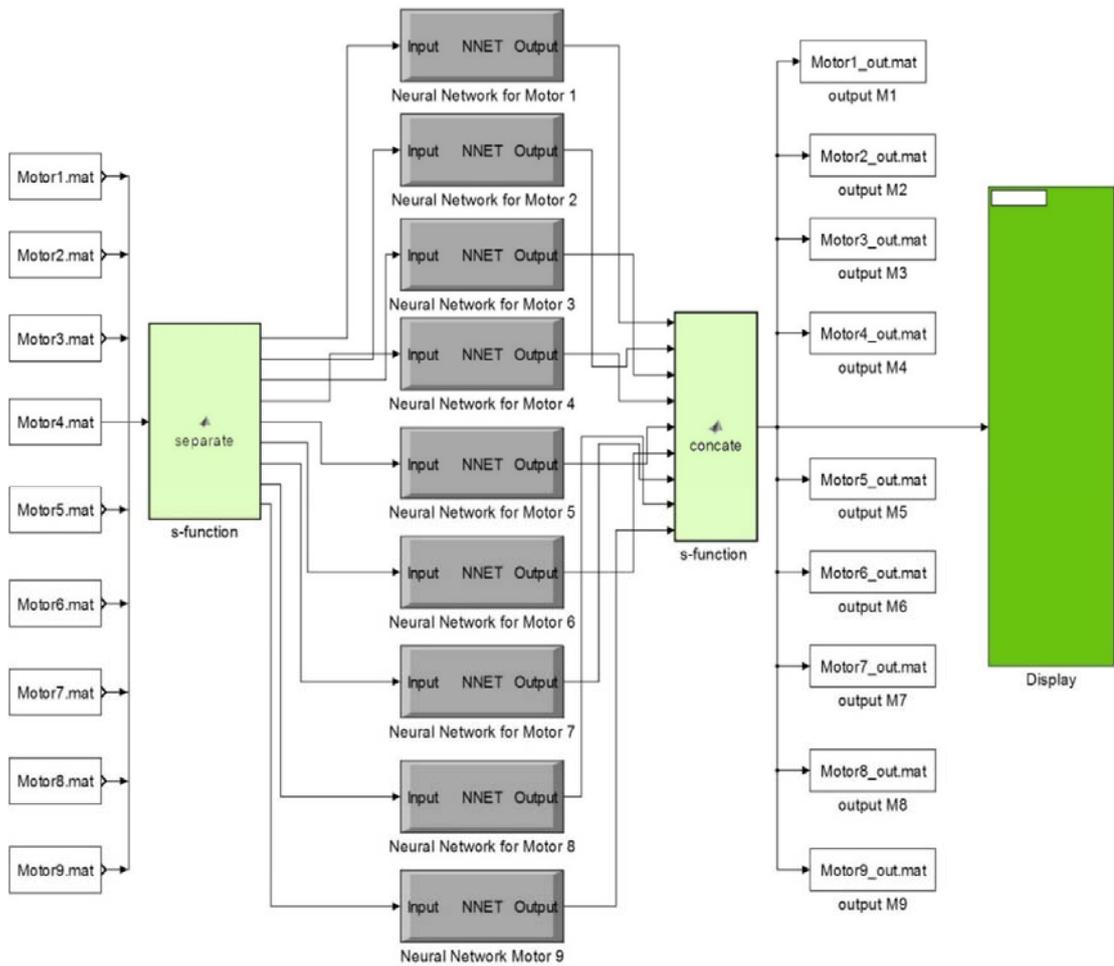


Figure 5.2: Neural Network Simulation Model for Fault Type Diagnosis at Motor Level

All feature values were stored in *Mat* files and assigned these values with the each motor. The Matlab scripts were used to merge all the features and generate the entire set of training data range for the validation and testing process in both fault and non-fault cases. The following Figure 5.3 presents the inside architecture of each neural network for each motor.

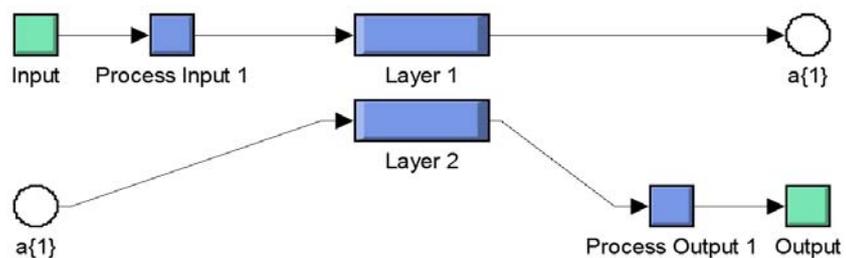


Figure 5.3: Internal Architecture of each Neural Network block

### 5.3.2.1 Model Setup and Training

Once the use of the neural network for the non-linear modelling of a system has been established, certain neural network particulars have to be considered and decided upon, such as these precedents:

- Total number of hidden layers and their neurons
- Transfer functions in neurons
- Error criteria
- Training algorithm
- Stop criteria when output achieved
- Initial values of the weights within all layers.

No attempt is made to describe all the above particulars in detail. The objective is rather to provide a general overview of the decisions one might face when designing an MLP. The brief configuration details of the neural network have been outlined as follows:

Table 5.1: Description of the implemented ANN

<b>Tasks</b>	<b>Configuration of the implemented ANN</b>
<b>Network Type</b>	Feed Forward Neural Network
<b>Learning Method</b>	Back Propagation
<b>Training goal</b>	0.0100
<b>Input data of each motor for each experiment</b>	The input is the one dimensional matrix for the ANN of 6 inputs where all data in each sensing point near motor are in a fault index
<b>Number of hidden layer neurons</b>	Different architectures are applied with different neurons. For example, [6x3x4], [6x10x4] and [6x15x4] (see Figure 5.5).
<b>Vector of classes for the target outputs</b>	Numerical matrices refer to the fault with value 1 or 0. For example, [0, 1, 0, 0] represent BRB fault and [0, 0, 1, 0] for Eccentricity.

### 5.4 Neural Network Simulation Model for Distributed level

In distributed modelling, all attached motors propagate their signals on a main power-line and create a complex form of transform signal in power-line network. From that signal, it is difficult to discern the fault component source. The role of the artificial neural network in this study at distributed motor level is to be capable of cancelling the suspected network bias that is interfering, due to the normal running-process procedure of electric motors or the intervention of environmental noises. It would be helpful to discriminate between different but similar fault symptoms at multiple sensing points and diagnose the correct fault

type. It is very difficult to simply visualise the signal at the motor-level. The following Figure 5.4 shows the simulation model to deal with a distributed motor model as follows:

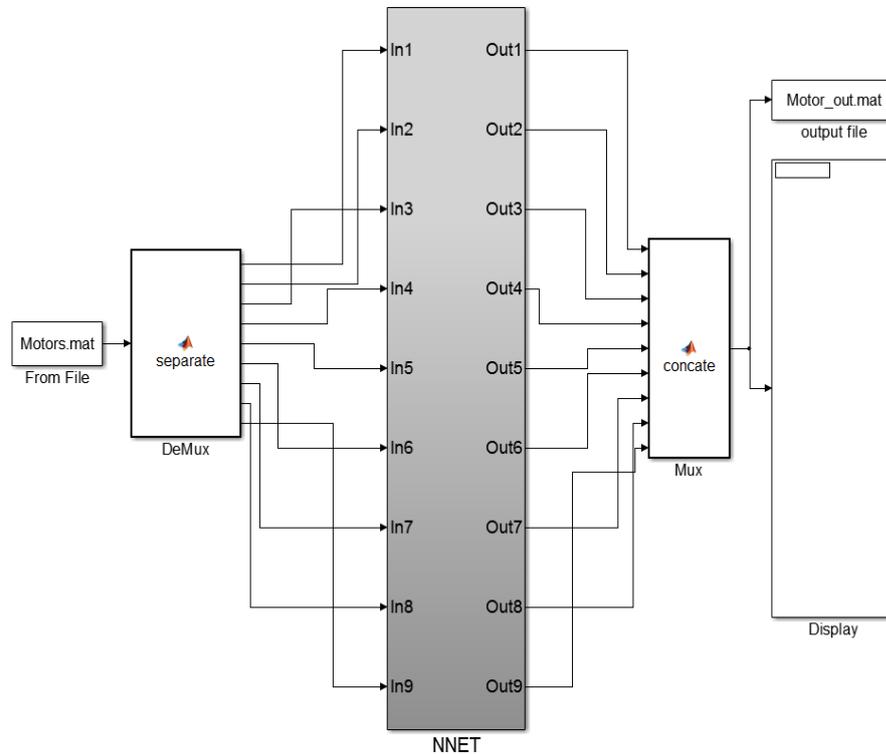


Figure 5.4: Neural network Simulation model for fault type Diagnosis at distributed multi-motor network

In Figure 5.4, all motor features data are stored in signal files and those features are assigned a motor number. All motor's data will then be trained by a single network of hidden layers, and generate an entire set of training data for the validation and testing process in both fault and non-fault cases for all motors. All the motor's data are trained by features in sequence through the network. This will help to identify the uncertainty existing between the feature's abnormal variations from multiple motors. In this process, the selection of dimensions and numbers of hidden layers depends on NN accuracy, increases the performance of the network and reduces training time when input features create some doubt in decision-making due to uncertain conditions.

### 5.5 Case Study 3: Fault Type Diagnosis in a Multi-Motor Network Modelling Environment

With reference to case study 1 discussed in Chapter 4, the significant features input was extracted using the Matlab scripts to generate an entire set of

training data from each motor from Figure 4.3, as shown in Table 5.2. A number of ranges were defined, associated with each feature. On the basis of the feature values, motor condition could easily be observed. Against each feature data input, we set a healthy target feature range in Table 1 for the identification of motor health. If the feature value exist within the normal range, the feature is considered healthy, but otherwise, faulty. For example, if the motor features speed, slip, RMS and crest factor values lie within the specified range limits we can consider the feature value as healthy.

As we can deduce from Table 5.2, Motor 3 is affected by both targeted faults, because all fault values come into the defined range of validation checks. Due to the propagation of a fault signal through the power-line, all other motors showed the influence of a fault frequency and indicate some fault condition. Motor 1 and 2 are the most the affected motors, due to being on the same sub-bus. However, because of the distance between motors, it does not show a strong value for both faults. In this situation, the system considered these changes an unknown fault. Through the proposed ANN methodology, the system can easily analyse the condition of each motor within a network and identify the fault type through any change in behavioural characteristics.

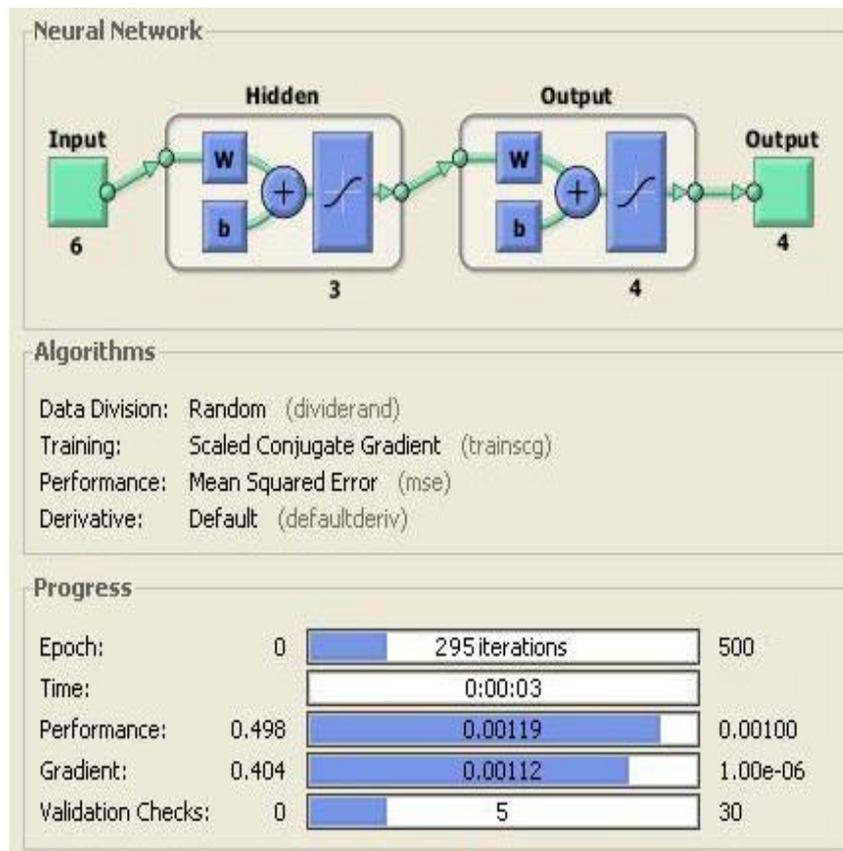
The decision on the number of nodes in a hidden layer is a complicated step in NN [45]. In spite of calculating the performance impact of the neural network structure on each motor fault data classification, multiple architectures were tested for the hidden layer design by changing the number of neuron nodes. The Levenberg Marquardt algorithm was implemented, using a Matlab script, to perform the learning phase. For this purpose, classification performance accuracy is defined by the pre-determined ratio between the total number of classified samples and tested samples. The training ratio was set at 80-90 percent with 500 epochs. To check the validity of data, 30 different cross-validation checks were applied. The initial values of offsets and weights were randomly assigned. The weights of the hidden layer were repeatedly adjusted during the training phase, until calculated outputs reached the targeted output performance. Three types of neural network architecture were tested for training, to adjust the weights of the hidden layer until the targeted output was achieved, as shown in Table 5.3.

Table 5.2: Learning data set for training process for multiple motors

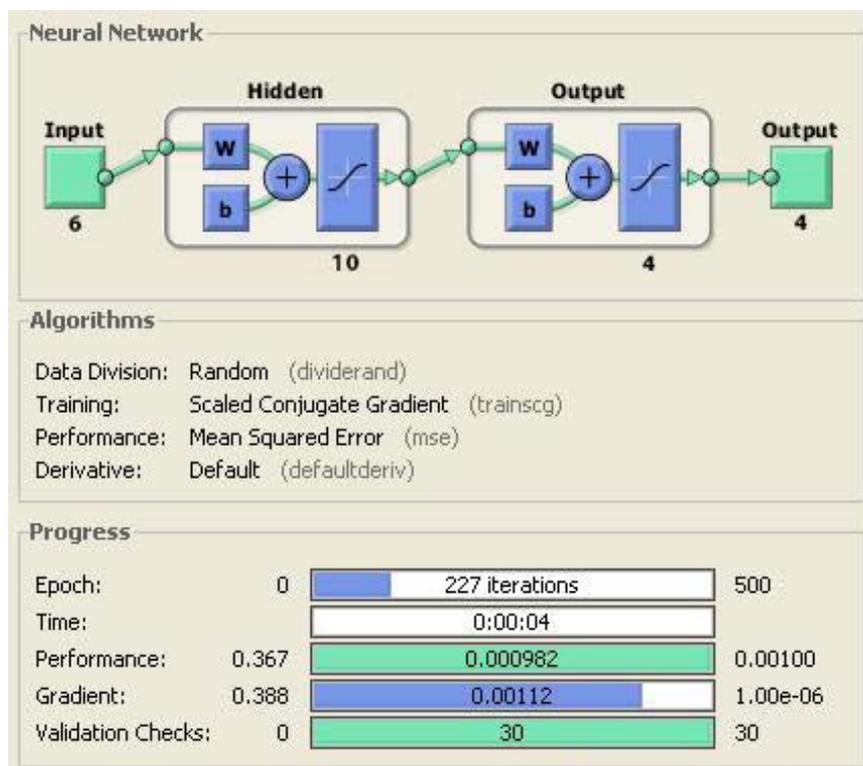
Feature No.	Motors			
	M1	M3	M6	M9
$x_1$	1260	1360	1280	1300
$x_2$	0.013	0.006	0.02	0.039
$x_3$	[4;46;54;96;221]	[4;46;54;96;221]	[4;46;54;96;116;221]	[4;46;54;96;221]
$x_4$	[81.8;125.2;146.9;98.5;91.5]	[99.6;154;207;136;167]	[41.3;106;59.5;50.3;59.65;130.8]	[76.4;57.6;51.2;25.5;175.6]
$x_5$	0.248	0.271	0.247	0.25
$x_6$	0.21	0.2	0.212	0.219

Table 5.3: Best architecture performances for classification

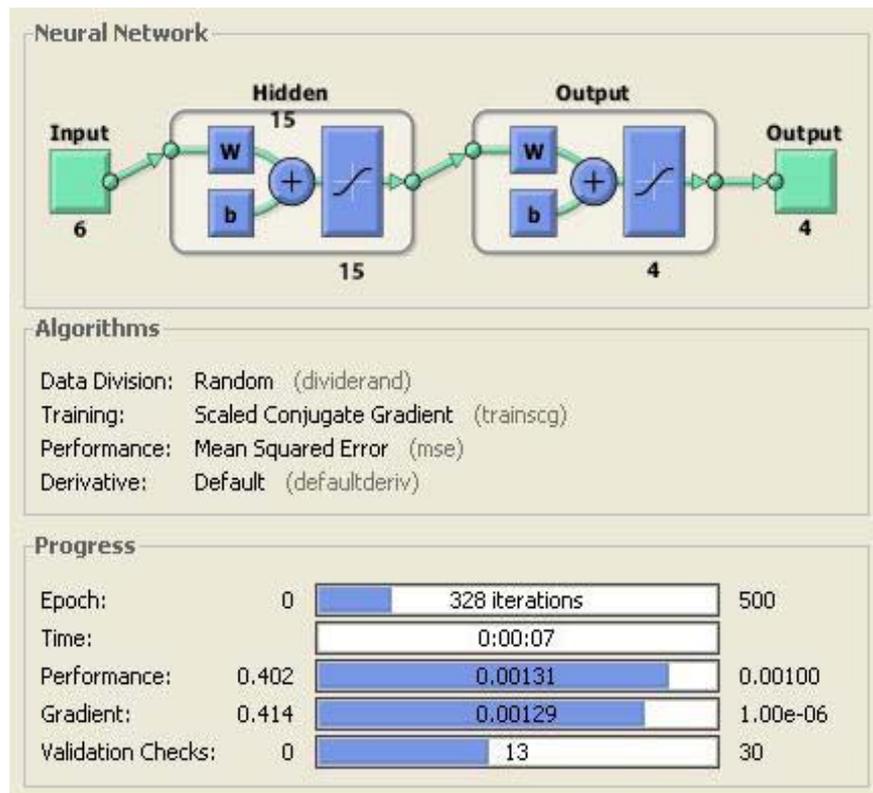
Architecture	Motors	MSE Performance	No. of Epochs	Accuracy (%)	Classification error
[6x3x4]	M1	$6.69 \times 10^{-3}$	284	78.1	21.9
	M3	$6.19 \times 10^{-3}$	290	78.4	21.6
	M6	$7.37 \times 10^{-3}$	299	79.5	20.5
	M9	$8.01 \times 10^{-3}$	310	77.8	22.2
[6x10x4]	M1	$8.49 \times 10^{-3}$	221	84.6	14.1
	M3	$8.59 \times 10^{-3}$	218	85.9	15.4
	M6	$8.72 \times 10^{-3}$	232	85.9	14.1
	M9	$8.99 \times 10^{-3}$	185	86.0	14.0
[6x15x4]	M1	$8.01 \times 10^{-3}$	329	84.1	15.9
	M3	$6.23 \times 10^{-3}$	327	83.9	16.1
	M6	$7.85 \times 10^{-3}$	346	83.2	16.8
	M9	$8.56 \times 10^{-3}$	387	78.5	21.5



(a)



(b)



(c)

Figure 5.5: Overview of the different ANN architectures chosen for fault diagnosis.

(a) [6x3x4] (b) [6x10x4] (c) [6x15x4]

Table 5.3 presents the performance of different architectures [6x3x4], [6x10x4], [6x15x4] respectively. It shows snapshots of different trained architectures of ANN with the [6x3x4], [6x10x4] and [6x15x4] configurations. It is noted that the number of iterations with [6x3x4] architecture is 295 after training the 500 epochs. It can be observed that the MSE performance in fault diagnosis has not been achieved by the end of the training process, that is, the MSE value is 0.00119 which is greater than the set target value (0.00100). And the number of validation check fails was 5 out of 30 wrongly classified during training process. On the other hand, in the [6x10x4] architecture, the MSE performance in fault diagnosis achieved by the end of the training process i.e. 0.000982, which is much closer to the set target value. All validation checks were successfully classified during training process and achieved the required results with fewer than the [6x3x4] architecture epochs. The third architecture [6x15x4] has more hidden layer neurons and trained all the iterations in more time than other two architectures. A total 13 validation checks was wrongly classified with MSE performance 0.00131, which is greater than the targeted value. It was proven

through the case study that the number of nodes in the hidden layer has a direct influence on the performance of ANN. If the number of nodes is small, the network may not be trained adequately. On the other hand, if the number of neurons is too large, the network will take a long time to train. Therefore, the selection of dimensions and number of hidden layers depends on NN accuracy in initial tests.

It can be seen that architecture [6x10x4] (6 input layer neurons, 10 hidden layer neurons and 4 output layer neurons) has the best MSE performance of the architectures. The fewest epochs were used during the training period to achieve the required accuracy, which shows the best performance in the targeted architecture, along with the smallest error percentage. Figure 5.6 shows a training performance graph of the neural network [6x10x4], which attained the best result of the NN structures.

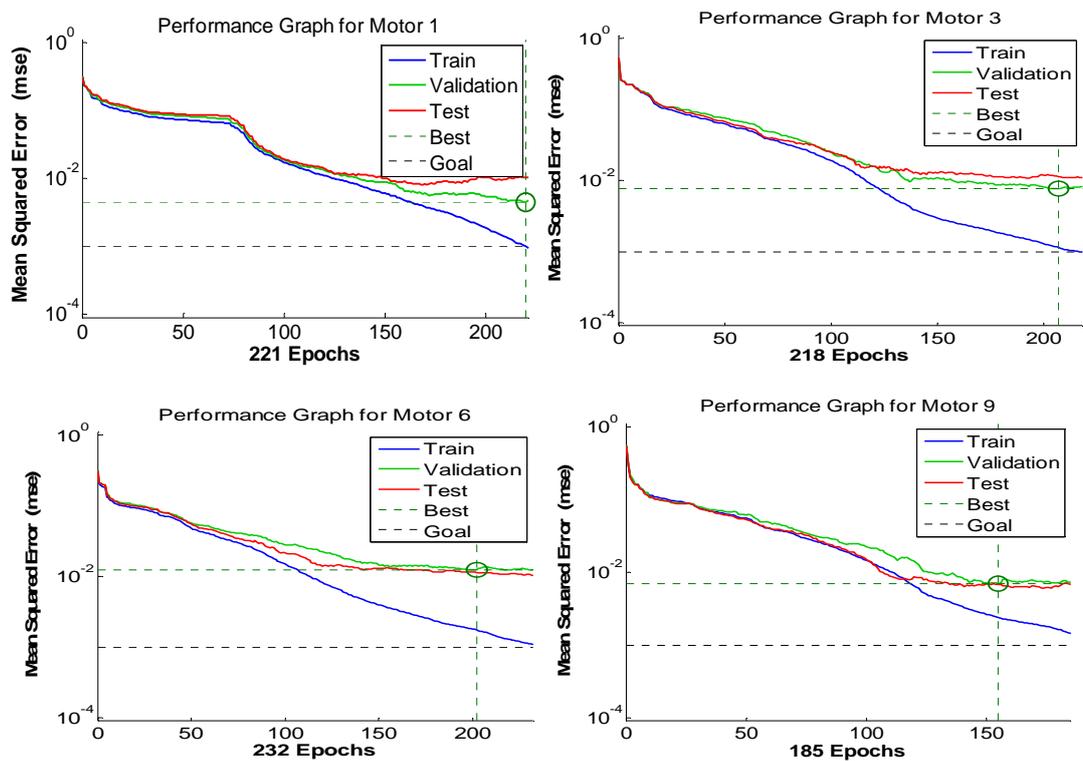


Figure 5.6: Performance graphs of identified motors 1, 3, 6 and 9 using [6x10x4] neural network architecture

After measuring the performance of each motor, the second means of testing the performance of the neural network was to measure the classification confusion matrices for the various types of error that occurred during the training process. To obtain the confusion matrix, data is fed into the neural network model. Basically, the confusion matrix holds the information about the comparison

between targeted and predicted classification classes [134]. Figure 5.7 presents the confusion matrices for the three process phases of training, testing and validation of each targeted Motor's 1, 3, 6, and 9 respectively. A total of four targeted and predicted classes (horizontal and vertical) were defined to compare each motor feature data set. In the case of successful classification of a targeted class trial, the diagonal cells are shown in green. Each diagonal cell indicates the number of cases that have been classified correctly by the neural network, to identify feature condition, whether healthy or faulty for each motor. The non-diagonal cells in red indicate the number of cases that have been wrongly classified by the ANN, or where the condition of features was not identified. The last cell, in blue, in each matrix indicates the total percentage of cases that were classified correctly in green and the vice-versa in red.

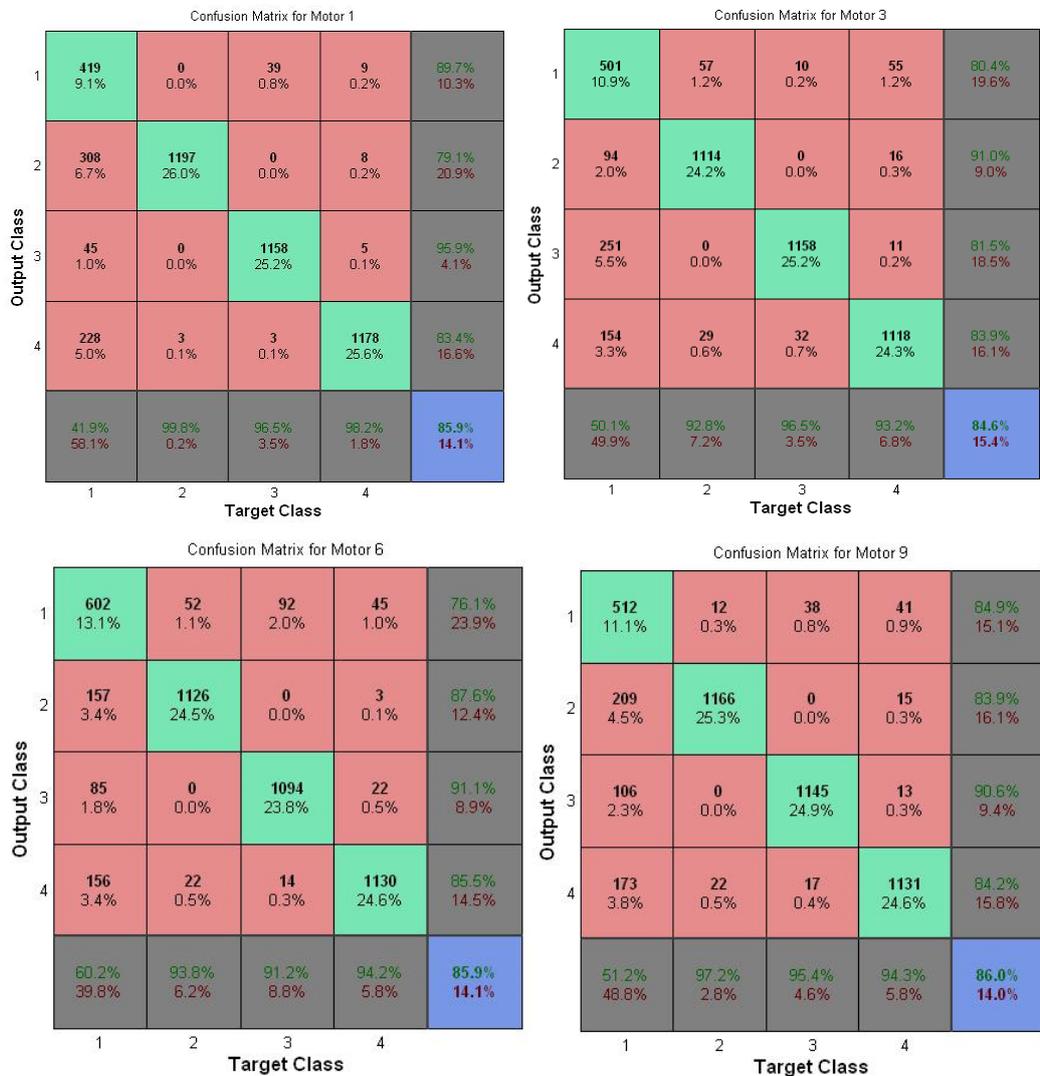


Figure 5.7: Confusion matrices of identified motors 1, 3, 6 and 9 using targeted and output classes

In case of M1, the above diagram shows each class had maximum of 1200 testing trials. Classes 1 to 4 represent the motor in a healthy condition, a BRB fault, air-gap eccentricity and an unknown fault respectively. Reading vertically, 419 trials were successfully classified as class 1, and none is wrongly classified. A total of 308 trials were wrongly classified as class 2, and 45 trials were wrongly classified in class 3. In class 4, a total of 228 trials were wrongly classified due to the complexity and mixing of features. When the confusion matrix is read horizontally, 39 trials of class 3 and 9 trials of class 4 were wrongly classified. The last row (vertical and horizontal) presents the successful classification rate of each class. The total of 3952 testing trials was classified and the final performance rate of success was 85.9 percent. Only a 14.1 per cent error-rate occurred, which is quite a reasonable rate in a multi-motor network. The same process will be applied to the other targeted motor's confusion matrices. More detailed results will be presented in the following chapters through different case scenarios.

It can be seen that through the chosen neural network architecture [6x10x4], a satisfactory accuracy was achieved in fault detection in the feature vector, ranging from 82 to 87 percent. This reflects the performance efficiency of the ANN algorithm, in reducing the level of uncertainty in decision making. To compare the confusion matrix performance of the proposed methodology and architecture, some work is needed on motor faults. However, a comparable work, albeit on a single motor was done by the Hamdani et al. [97]. They achieved a 92.11 percent classification accuracy in successfully testing the trial cases in the case of isolated motors. However, this proposed model achieved 86.6 per cent classification accuracy rate, due to the existence of complexity and mixing of multiple signals in a multi-motor power-line network. This percentage reflects the efficient performance of the proposed ANN architecture, and reduces the level of uncertainty in decision making.

As we can observe from Figure 5.7, the percentage of the testing trials failure in Motor 3 is higher than in the other three motors selected. In order to estimate and observe the behaviour of the system performance in case study 2 in Chapter 4, the proportional possibility of targeted motors can be utilised to verify the event of fault indices occurrences of similar faults in Motor 3. The following Figures 5.8-5.9 show the percentage of proportional possibility of detecting BRB

and eccentricity faults at multiple sensing points by the using neural network back propagation iteration method as follows:

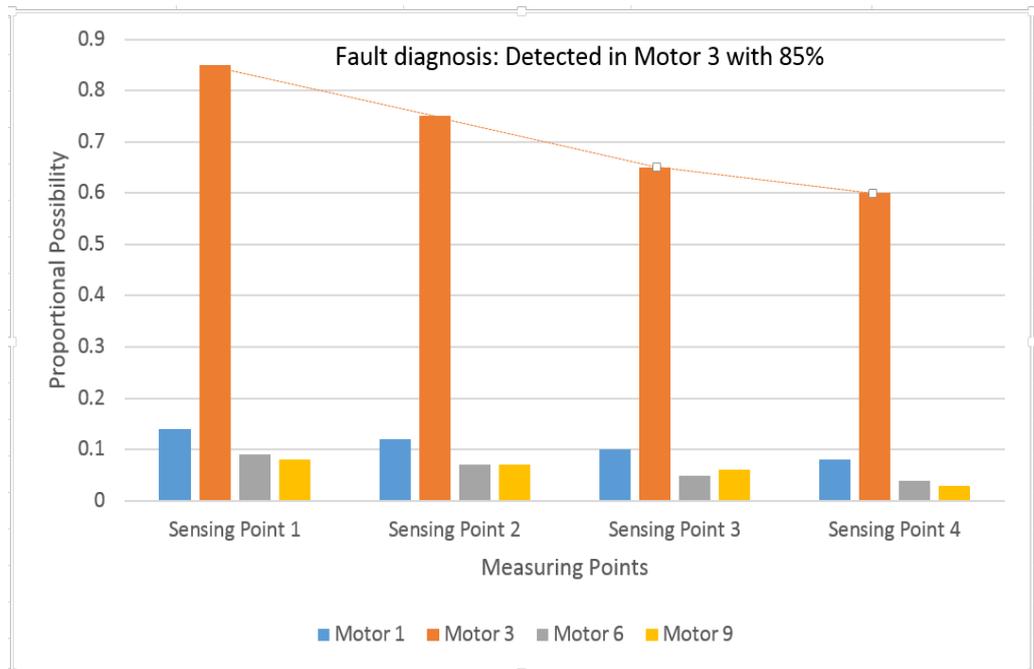


Figure 5.8: Fault indicator proportional possibility calculation to show the presence of a BRB fault in Motor 3

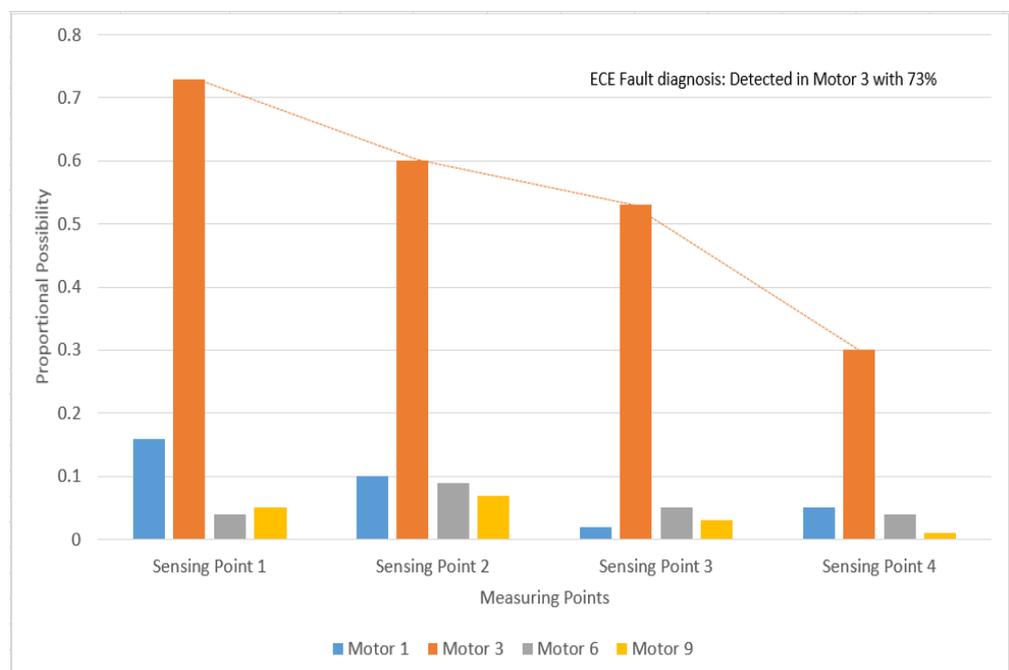


Figure 5.9: Fault indicator proportional possibility calculation to show the presence of an eccentricity fault in Motor 3

As shown in Figures 5.8 and 5.9, Motor 3 was detected as a faulty motor with a high proportional possibility percentage in terms of a BRB fault, and the

possibilities of a BRB fault decreased in other motors according to their distance. The severity of the BRB fault is greater than the air-gap eccentricity in faulty Motor 3, due to the high resistance of a voltage drop. Thus, it can be observed that the diagnosis model of target faults using NN provided an accurate and efficient answer. Therefore, the proposed network architecture and methodology will be considered, based on real-time data experimental data, in Chapter 6 and 7, with more complex and comparative data to test the robustness of the diagnostic system.

## **5.6 Chapter Summary**

The suitability of multilayer FFNN architecture for the identification of motor faults within an industrial motor network system was presented. The relationship of features with the faults is also discussed. To improve the MSE accuracy rate, various types of neural network architecture were tested in training to adjust the weights of the hidden layer until the targeted output was achieved. Accuracy in identifying motor condition depends on the neural network architecture. The results indicate that accuracy in fault detection in the feature vector has been achieved through a simulated model, and classification performance is satisfactory in discerning healthy and faulty conditions, including fault type, against each motor level.

## DISTRIBUTED MOTORS SYSTEM EXPERIMENTAL TEST-BED ENVIRONMENT

### 6.1 Introduction

This chapter introduces a typical experimental test-bed environment. The specific experimental setup, tools and resources offers good representation of real-time industrial case studies, to prove their actuality, and draw conclusions from the experience gained. A WSN architecture was presented to discuss the role of coordinator and end nodes within the network and describe the step-by-step activities of data fusion, transformation, packet structuring and transmission to a base station. Different results were presented to show the comparison between manual data capturing and sensor nodes capturing to validate the relevancy and accuracy of the results.

### 6.2 Experimental Multi-Motors Test-Bed

In order to prove the propagation of faulty signals within a power-line network and the detection of fault types, a typical example of an experimental motor network was modelled and implemented at AUT SeNSe laboratory. Two sizes of single-phase induction motor were connected within the same main power-line. A total of nine induction motors were used to model the multi-motor network. These motors were distributed on three sub-buses, as shown by Figure 6.1.

Each bus contained three induction motors of two sizes. These are two 15w (Model S7I15GE-S12) and one 25w motor (Model S8I25GE-S12). The reason for using different sizes of induction motor was to analyse the influence of high power-rated motors on low power-rated motors. With reference to Table 4.1, chapter 4, the same motor specifications were used in the test-bed modelling. Each motor had its own measuring point to capture the data. Load torque was applied through the brakes to create vibration in a motor. The sampling rate for the

measurement was 25000 samples per second, and each measurement took a second to store into a flash drive. Faults were artificially created, by applying vibration through the brakes and through rotor misalignment.

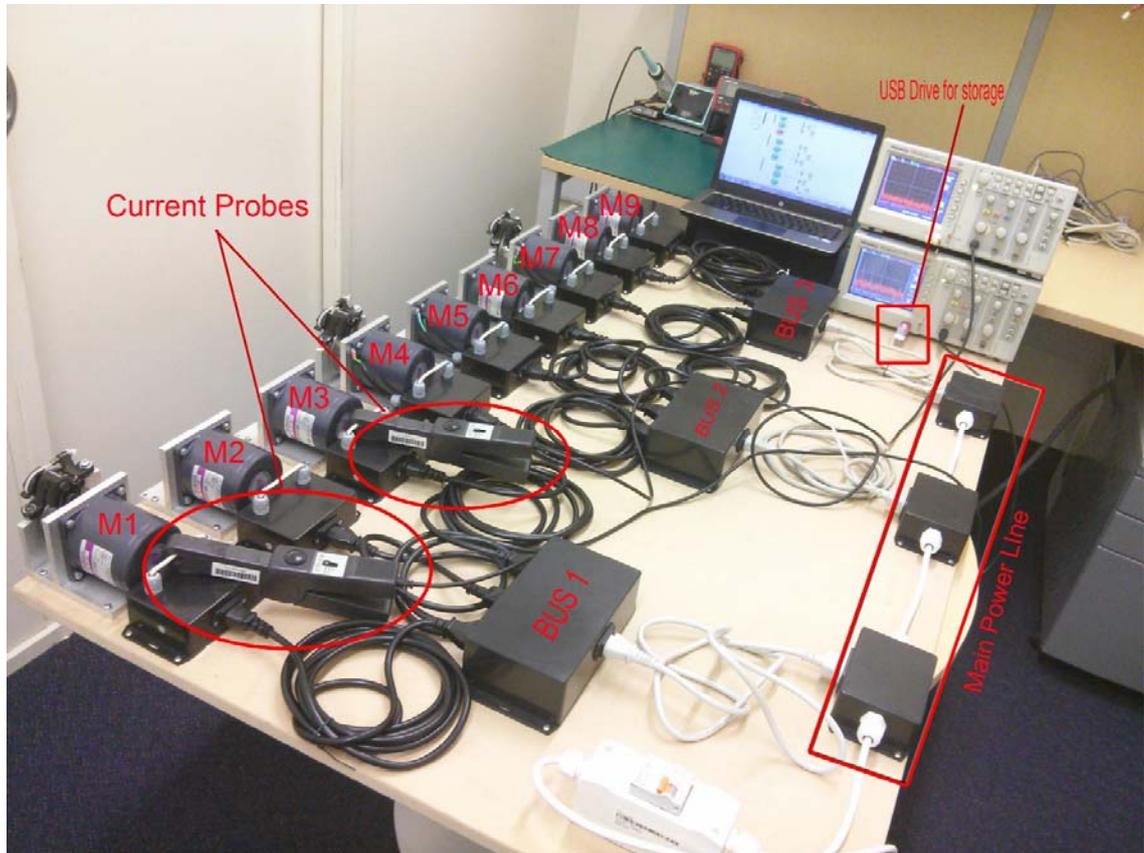


Figure 6.1: Scaled down test-bed designed at AUT SeNSE laboratory to analyse the signatures of motor faults

### 6.2.1 Hardware Measurement Tools

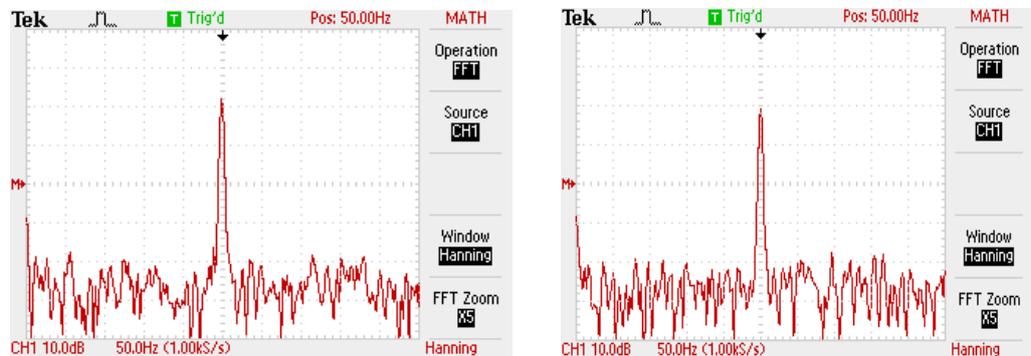
Different practical experiments were done during the experimental work. To store the measurement data into a flash drive, Tektronix storage oscilloscope (TDS2012B) was employed. Current probes Tektronix (A622) was used to capture the electric current data at the same time as manual data capturing. To measure the speed of each motor, a digital laser tachometer (Standard ST-6234B) was used. All the measurement data was stored into a separate Excel spreadsheet file for each motor. Information about the measuring tools used to capture the data is formulated in Table 6.1 following.

Table 6.1: Descriptions of measurement Tools

Tektronix oscilloscopes		Current probes Tektronix A622	
Parameter	Measuring Values	Characteristic	Value
Record Length in points	1024	Frequency Range	DC to 100 kHz
Sample Interval	0.2466067	Max Input Current	100 A peak
Vertical Units	dB	Output	10/100 mV/A
Horizontal Units	Hz	<b>Standard ST-6234B Tachometer</b>	
Source	MATH	Range RPM	2.5 to 99,999
Operation	FFT	Max RPM Resolution	0.1RPM
Window	Hanning	Basic Accuracy	+0.05%+1d

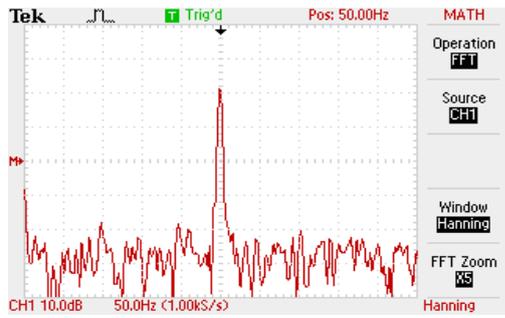
### 6.2.2 Test-Bed Measuring Data

In the initial test of test visualisation, data was captured through a hand-held current probe by using an oscilloscope. For that, all electric motors electric current data was acquired using two oscilloscopes and current probes (See Figure 6.1). Hence, this data was not collected at the same capture time. Due to the limitations of the measuring equipment, the captured data was taken at different instants of time from different motors and may regulate the quality of the sidebands associated with the electric current signal. However, this change does not necessarily reduce the quality of the measured data, because its variation rate is reasonably likely to decelerate in diagnosis of internal faults motor. The electric current waveform was captured by probe and then the built-in FFT function was applied from oscilloscope to observe the sidebands. Figure 6.2 (a)-(i) shows the real-time data capture from all targeted motors without any load, with a horizontal axis of frequency (Hz) and vertical amplitude (dB) as follows:

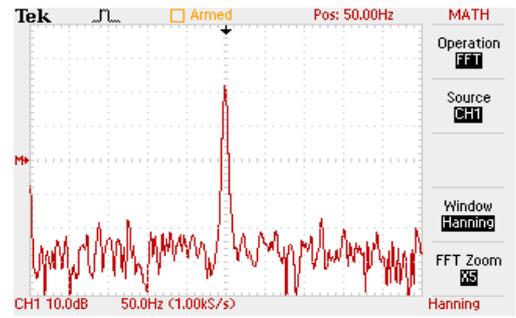


(a)

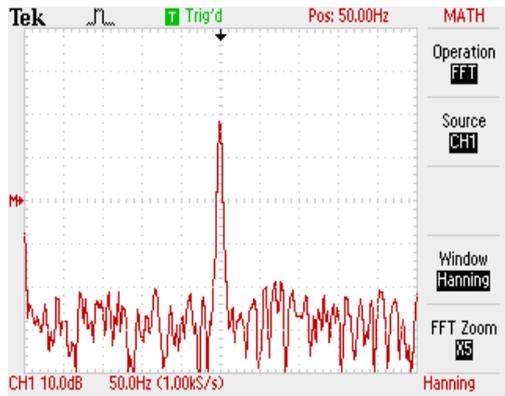
(b)



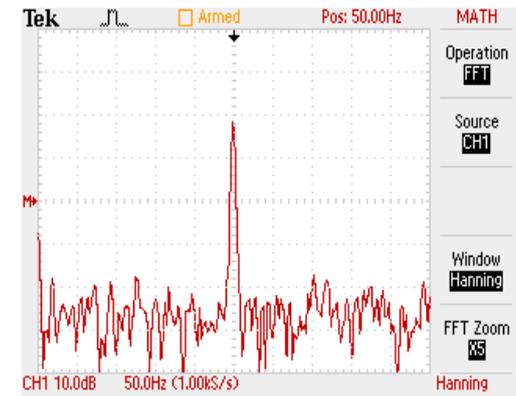
(c)



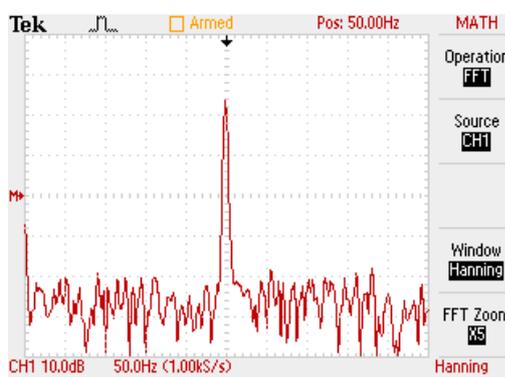
(d)



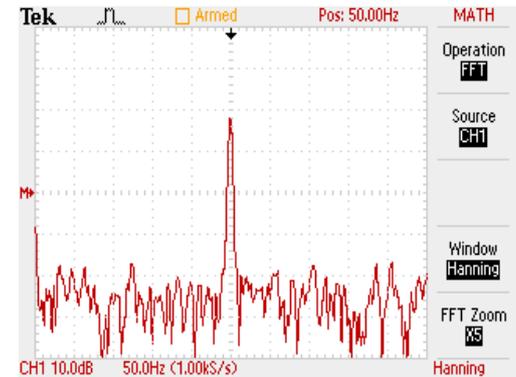
(e)



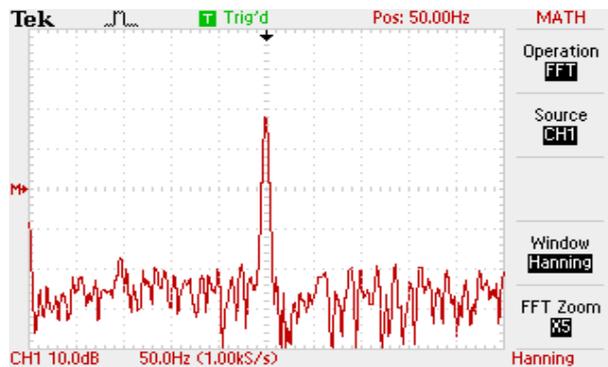
(f)



(g)



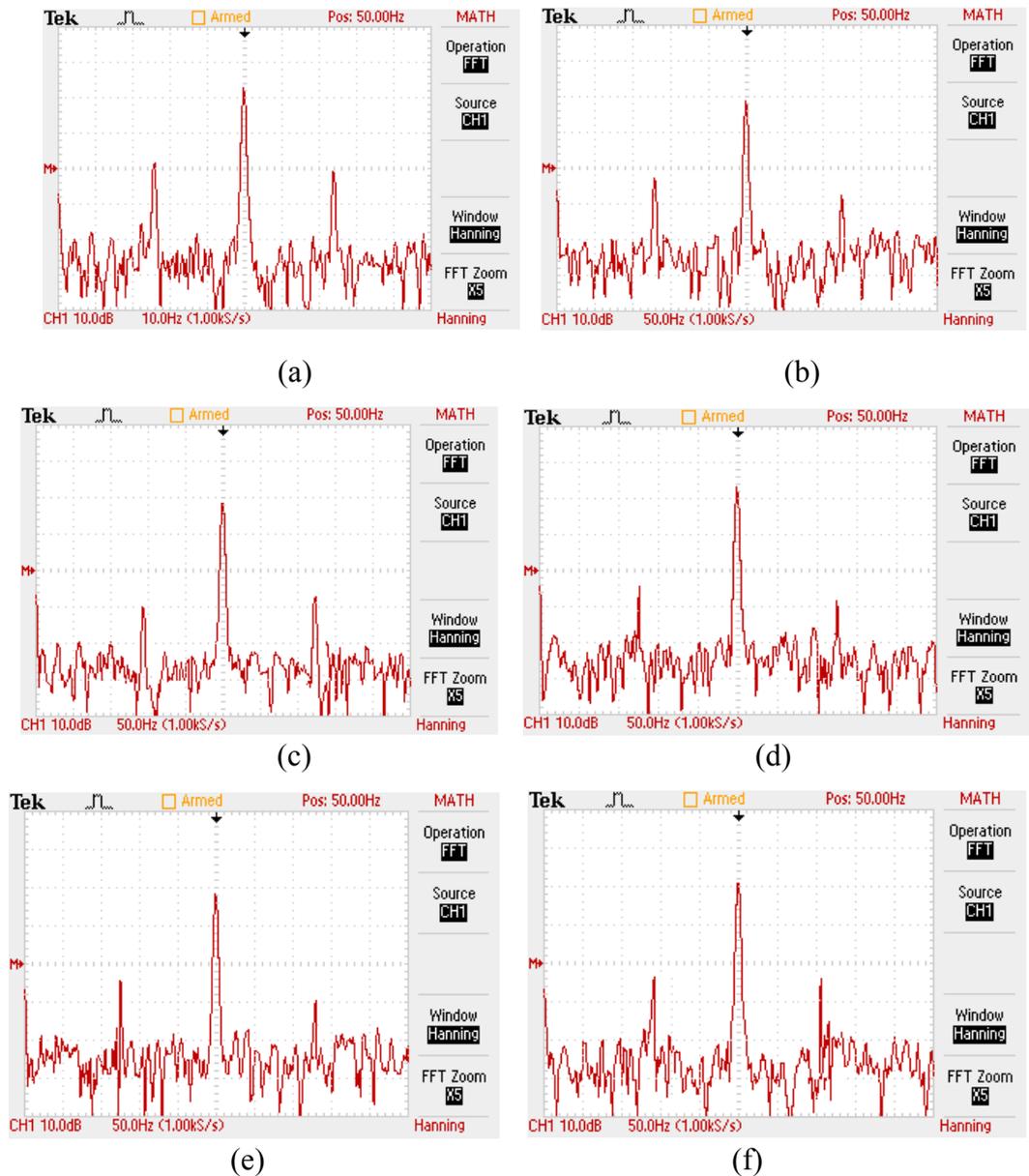
(h)



(i)

Figure 6.2 (a)-(i): Individual FFT spectrum of each electric Motor (1)-(9) without load

As we can observe from Figure 6.2, no significant sideband appears in any electric motor, as all data was captured in an operational state without any load torque. As discussed in Chapter 3, without any load torque, the appearance of a faults sideband is not possible. Therefore, faults were created artificially in Motor 1. Figure 6.3 represents the influence of a faulty signal from Motor 1 on the others via the main power-line as follows:



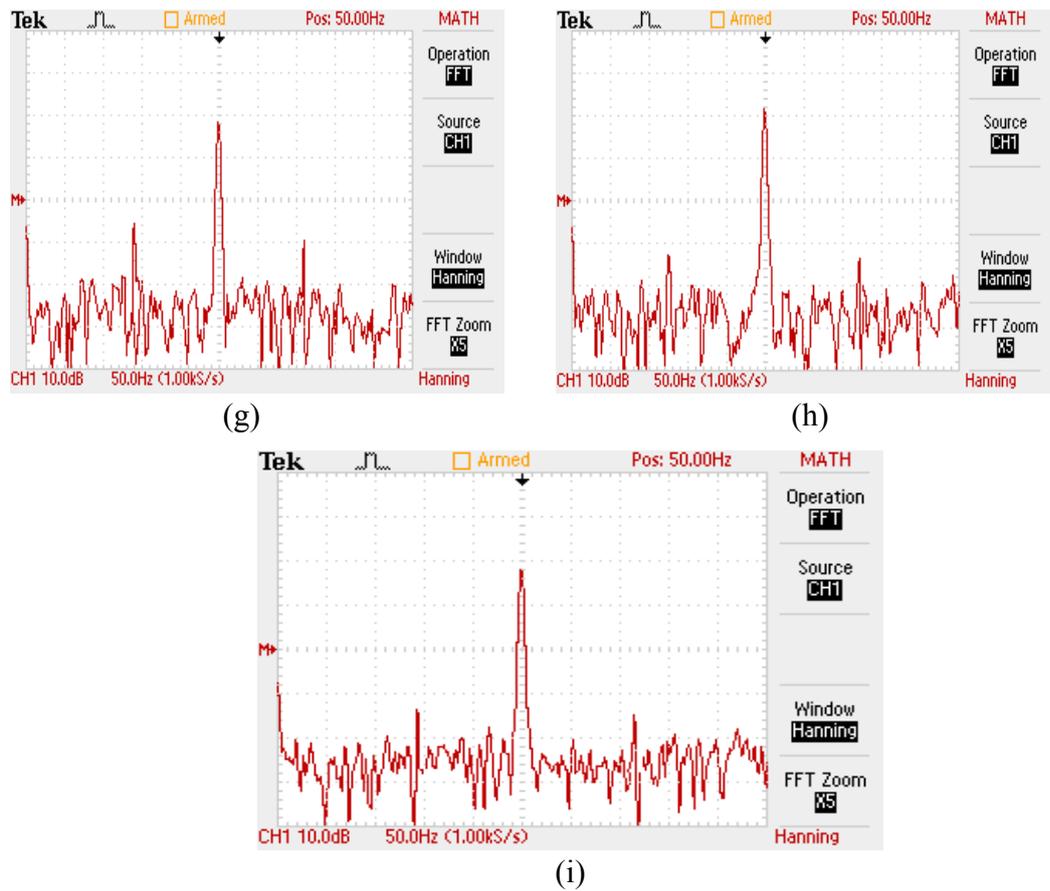


Figure 6.3 (a)-(i): Individual FFT spectrum analysis of each electric Motor (1)-(9) without load

As we can easily observe from Figure 6.3, Motor 1 has the highest faults sideband at certain frequencies (29 Hz & 71 Hz) compared with the other motors, which show clear symptoms of the BRB fault. Also, the influence of sideband-amplitude values decrease according to their distance from Motor 1. The distance between the motors is already known.

### 6.3 Wireless Sensors and Related Network

The development process of a WSN system contains both hardware and software requirements. The test-bed environment in developing the application is set up with commercially available hardware components. A software application is then developed, and tested on a hardware platform. This section describes the hardware system used, followed by the software written and implemented for the motor fault diagnosis applications.

### 6.3.1 Motor Wireless Sensor Hardware

The hardware used was the Arduino development board as a base, with the wireless sensor SD board shield on top, and then an XBee module on the top level, which makes the system look easy to use, as illustrated in Figure 6.4.

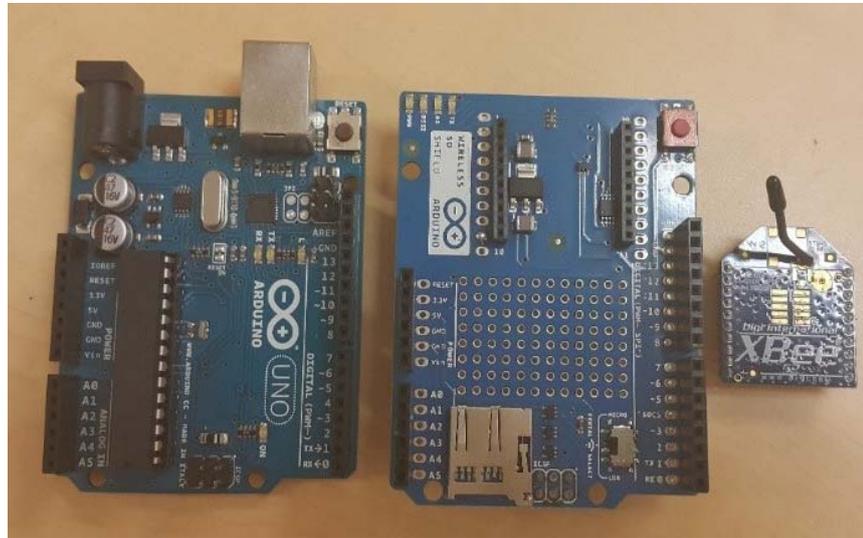


Figure 6.4: Arduino experimental node

Arduino is an open-source platform for electronics prototype development, based on flexibility and cost [137]. The ADC on Arduino can acquire the input values from different analogue sensors, including an electric current sensor. For that, analog to digital conversion was done on Micro Controller, using the C programming language of microcontroller software (based on wire connectivity). An ATMEGA328P low-power 8-bit CMOS microcontroller was chosen for programming based on AVR's better RISC architecture [138]. The ATMEGA328P was chosen because of its free programming environment, easy reference design and it being easy to attach one's own sensors around the board. By executing complex commands in a particular single clock cycle, ATMEGA328P can easily manage throughputs up to 1 MIPS per MHz, allowing the developed system to be designed to improve the processing speed.

A split-core current transducer (ECS1030-L72) was chosen as the electric current sensor because of its non-invasive nature. The sensor can be easily clamped onto the each motor sensing point without interrupting power into the circuit breaker. The VAC output of the sensor could not directly interface with the Arduino board because of the maximum limits of reverse voltage on the MCU

pins [139]. A rectifier was therefore required to ensure that a negative voltage would not be applied to the pins of the MCU. Two rectifier circuits were designed for attaching the sensor with Arduino: a full-wave rectifier using a diode bridge, and a half-wave precision rectifier an OP AMP. The diode bridge was also paired with a smoothing capacitor and discharging resistor to reduce the voltage ripple. Figure 6.5 shows a schematic circuit diagram of the designed circuit that is required between the electric current sensor and Arduino as follows:

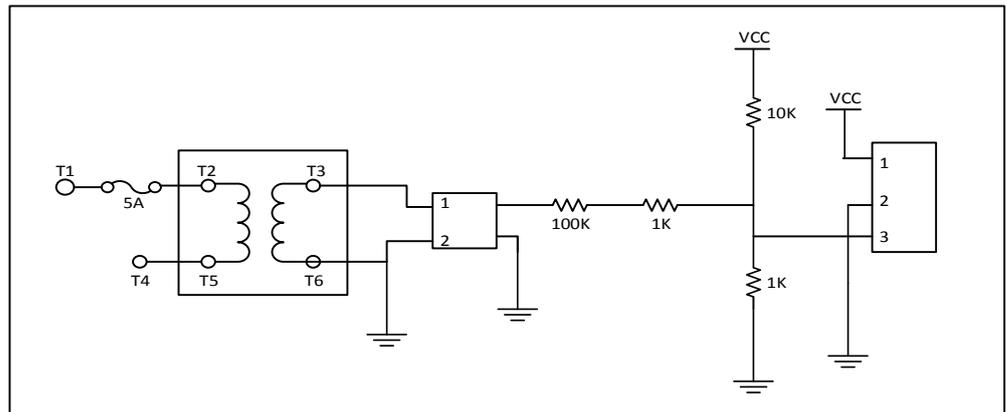


Figure 6.5: Circuit diagram of electric current sensor board

Following Figure 6.6 shows the experimental setup of Arduino with a current sensor to acquire the current signals for Arduino analysis that will be described in following section of this chapter.

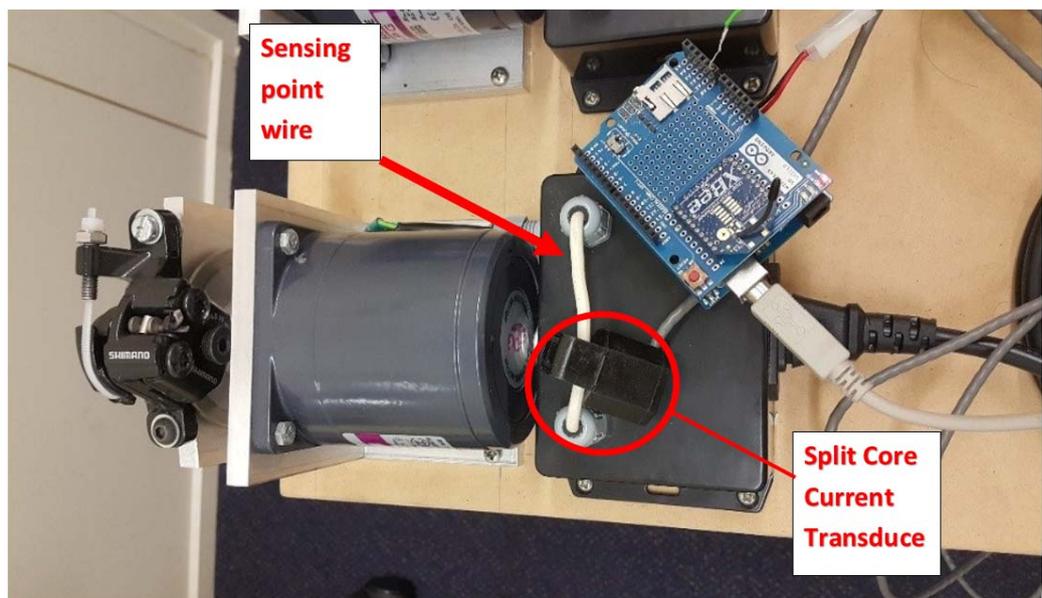


Figure 6.6: Arduino module connectivity using electric current sensor with motor sensing point

To measure rotor speed, the position difference during the time difference must be known. A simple way has been proposed for measuring a high rotor-speed: to count the position difference using the position counter within a defined time period. Thus, the speed can be easily calculated because it is related to the position difference. Since this measurement resolution depends on the number of encoder pulses over a defined period, the resolution is low and fails in the low-speed region.

Therefore, to measure the rotor speed in a low speed range, the time difference is measured between two encoder edges using a developed circuit, as shown in the following Figures 6.7-6.8. This circuit is mounted near the motor rotor to capture the encoder pulses.

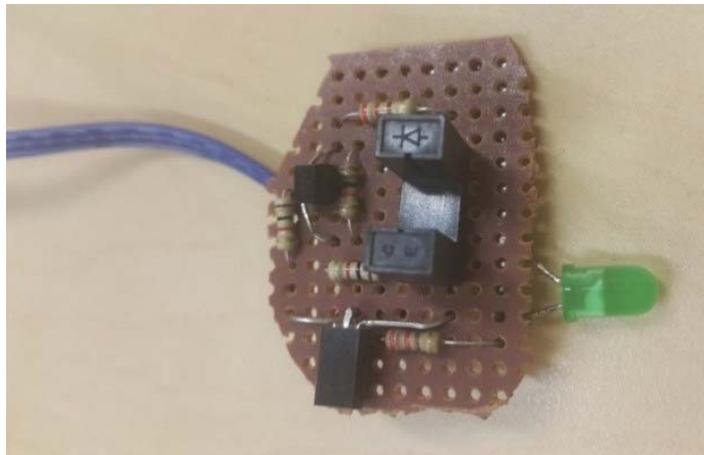


Figure 6.7: Speed sensor measuring circuit

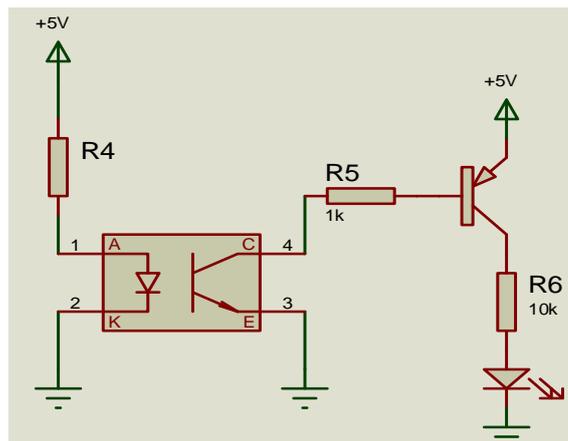


Figure 6.8: Circuit diagram of motor incremental motion indicator

The speed is then measured, inversely related to the measured time difference. However, this technique requires a high-frequency clock for timing, and resolution over a high-speed range is low. Therefore, a technique is used in

software coding that calculates the speed as a division of the number of encoder pulses and the precisely measured time-duration between these pulses. Each value is very close to the referenced manual motor speed as shown in the following Figure 6.9:

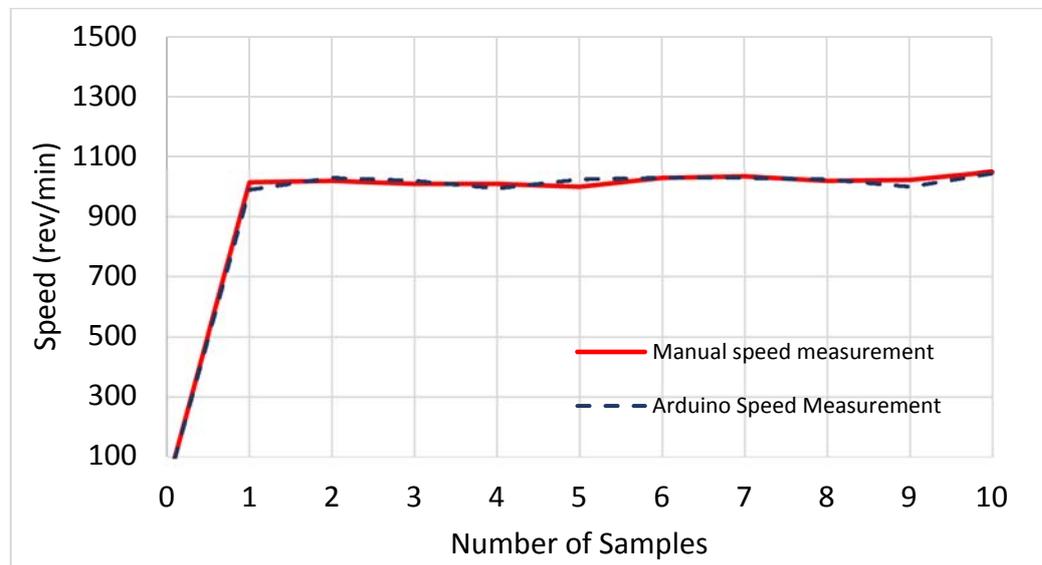


Figure 6.9: Comparison between the reference motor speed and sensor measurement

As discussed in chapter 4, the required ideal speed for Motor 1 can vary between 1200-1500, but due to applying the brake, the speed of motor reduced to 980-1050. As we can observe from Figure 6.9, there is slight variation between both measurements, because the manual tachometer has more intelligence and measures the speed with accuracy. But Arduino sensor shows the accuracy of the calculated data is near the expected speed.

### 6.3.2 System Software Design

Software design of this system has two main parts. Arduino boards need to be programmed in order to measure sensor readings and forward them whenever ordered. Arduino boards are programmed using C language, using the function of the Arduino board in this system for logging of sensor data and processing them according to user request. Current and speed sensors are attached with different sensor reading techniques. Therefore, according to sensor specifications, the required code was added into the board by using Arduino built-in libraries. Matlab program was used to get the data into the computer via coordinator USB connection, and store them into mat file for further analysis.

Apart from data logging, Arduino boards were programmed to send the status of each sensor, that is, number of readings logged, time intervals and units used to log sensor readings. As mentioned earlier, Arduino Uno R3 board has 2K byte of EPROM, which is used for temporary storage of data [136]. In case of loss of data packets from end node to coordinator, the end node always has keep backup data to send again on request.

Before constructing a physical prototype for a test-bed environment, Proteus ISIS Professional was used to simulate the Arduino's connectivity with the electric current sensor, and check whether the design circuit was functioning properly or not. Figure 6.10 shows the layout of the logical prototype in the software.

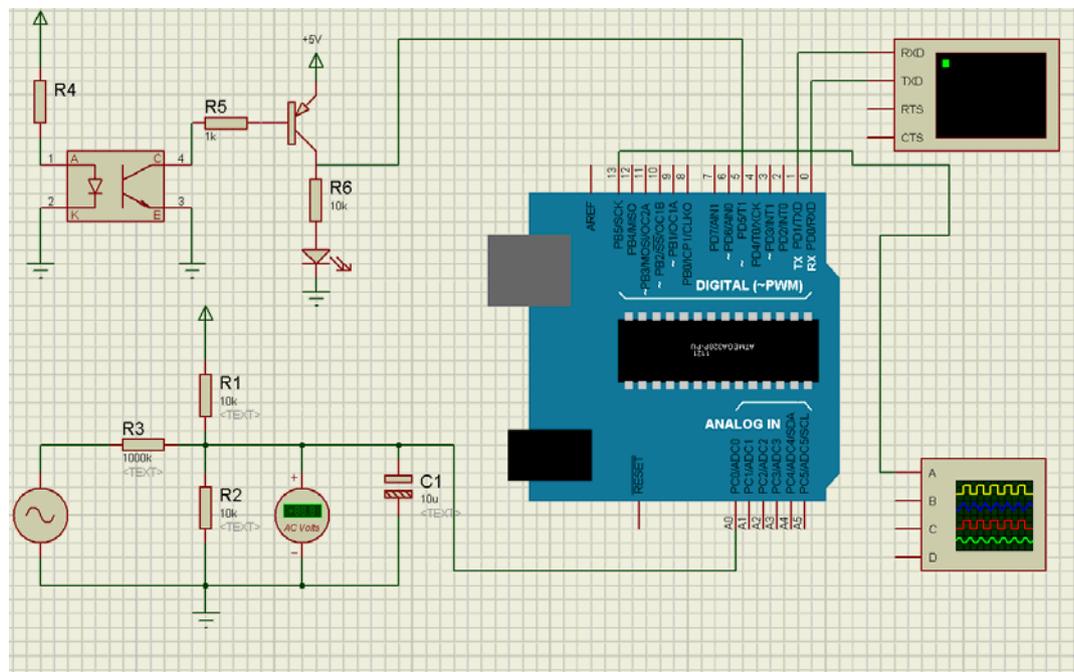


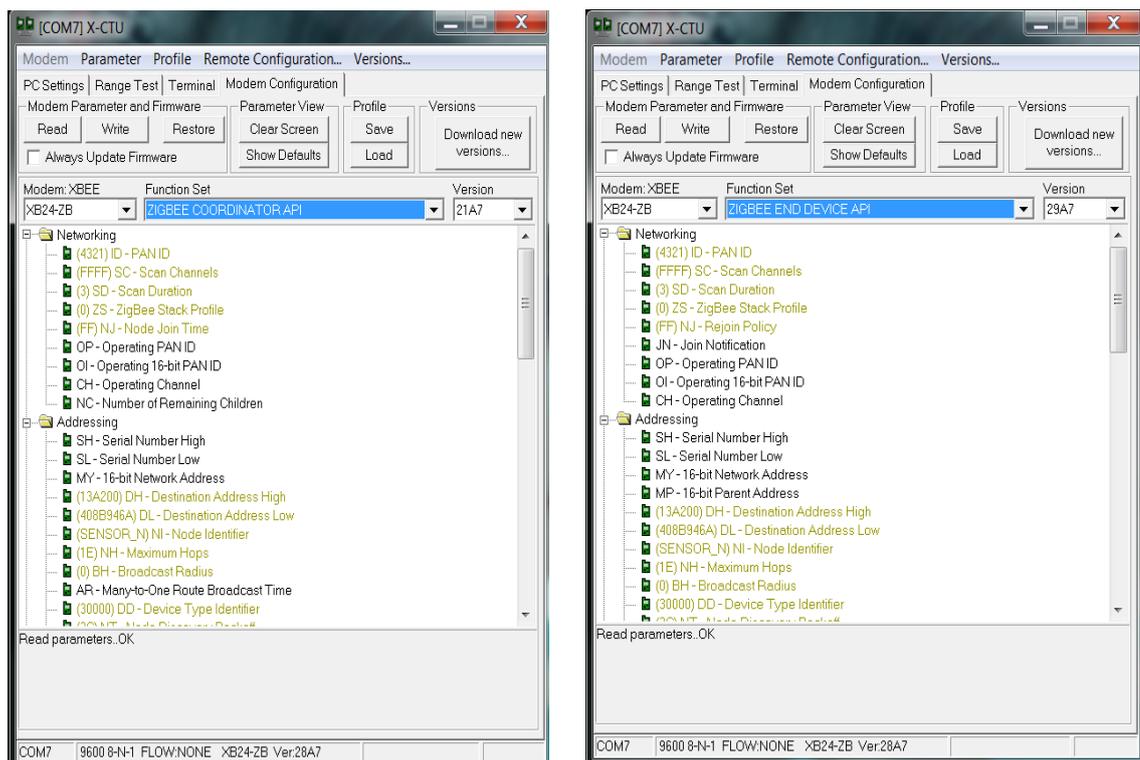
Figure 6.10: Schematic layout of Arduino for testing in Proteus ISIS Professional

## 6.4 XBee PRO ZB (S2B) Module

The XBee RF module provides an interface via a logic asynchronous serial port to the host device. When connecting a serial port, the Arduino unit can be configured and communicate with any voltage-compatible UART. It can easily be linked through a level translator (analog-to-digital) to any serial port device via a USB interface board or RS 232 [135]. The speed- sensor signal data is injected into module UART via the digital pin 6 slot as a serial signal. The current sensor values are entered from pin A0 as an analog signal. Then on analog-to-digital

convertor will convert these into digital form to get ready for analysis. The signal will be in idle state when no signal data is being transmitted to the coordinator. Each type of data depends on the microcontroller and the XBee RF module and needs to be set up with compatible configurations (parity, baud rate, data bits, starting and stop bits) [135].

To configure the coordinator and end device mote, the XBee module was set up to work as both coordinator and end node by using X-CTU software in particular settings at a data rate of 9600 baud, as shown in Figure 6.11:



(a)

(b)

Figure 6.11: (a) XBee coordinator API configuration (b) End node device

#### 6.4.1 Modes of Operation and configuration for XBee Module

The XBee module was configured in three modes: idle, transmit and receive modes. The XBee Module is in idle mode when it is not transmitting or receiving any type of packet data. Under the following conditions in idle mode, the module will switch between different modes of operation:

- Transmission mode: (when sequential data is ready to transfer into the buffer in the form of packet)
- Receiving mode: (structured RF signal is received by coordinator through the XBee antenna)

- Sleeping mode (enabled only on end nodes)
- Command-based mode (valid mode sequence command is propagated to all end nodes)

The node will be in transmit mode when the sequential data is received through sensors, and then the data will be converted into packets. The RF module will leave the idle mode and enter into transmission mode for sending the data. The destination address will determine the receiving node. Before transmitting the packet data, the module will confirm that the route and 16-bit network address available to the receiving node has been properly confirmed and established. The following flow chart (Figure 6.12) shows the process when the node is in transmission mode.

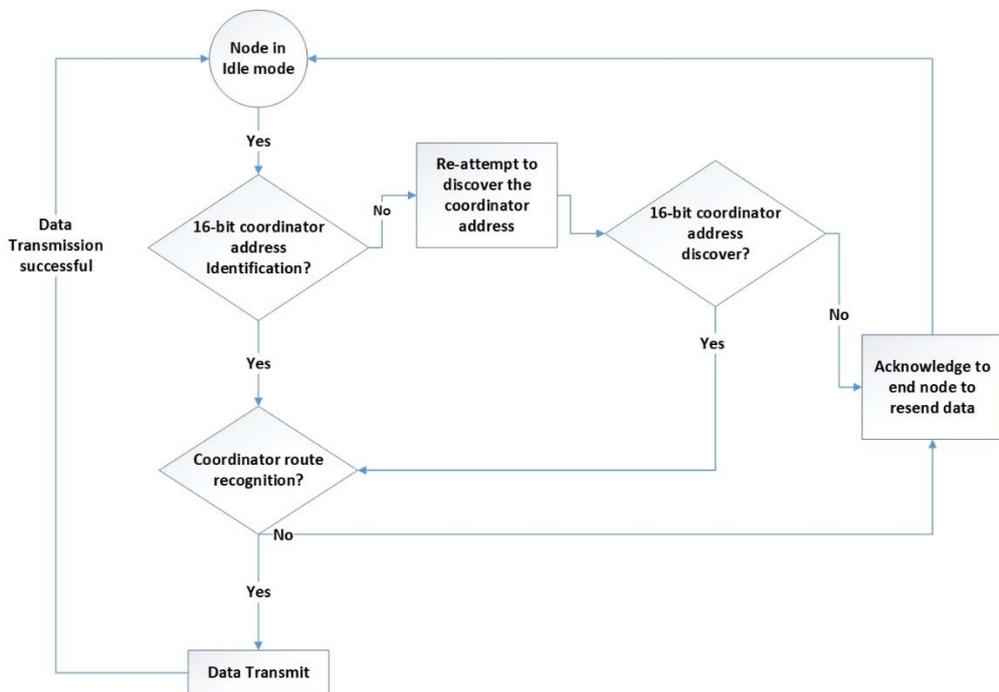


Figure 6.12: End node transmission mode

In Figure 6.12, if the destination address is unknown then the discovery mode of the network address will be performed by the end node. However, if the route to the coordinator is unknown, the route discovery process will carry on until a valid connection has not been established. If the coordinator address is not discovered, acknowledgment will be send to the end node to resend command to discover the route. When the data is transmitted between the nodes, a network-level acknowledgment is sent back through the established route to the sending end-node device. This acknowledgement shows the sender node that the packet

was successfully delivered to the destination node. In case of not receiving an acknowledgement, the sender node will re-send the data over the network.

In receiving mode, if the valid RF data packet is received, the serial data will be transferred to the serial transmit buffer.

During sleep mode, the node will be almost completely turned off and incapable of receiving or sending messages until it wakes back up.

And in a command-based mode, if the node wants to check the availability of the coordinator only, rather than to transfer the data, once the radio is in command mode, it listens for coordinator feedback for a while. If, within the next 10 seconds there is no feedback, the XBee automatically drops out of command mode and goes back into idle mode and waits for the data.

## 6.5 Architecture of Wireless Sensor Network

With reference of Figure 6.1, the diagram of the system shown is in Figure 6.13 and consists of a coordinator and several end nodes. Each motor has on individual end node for the collection of data over the XBee communication link.

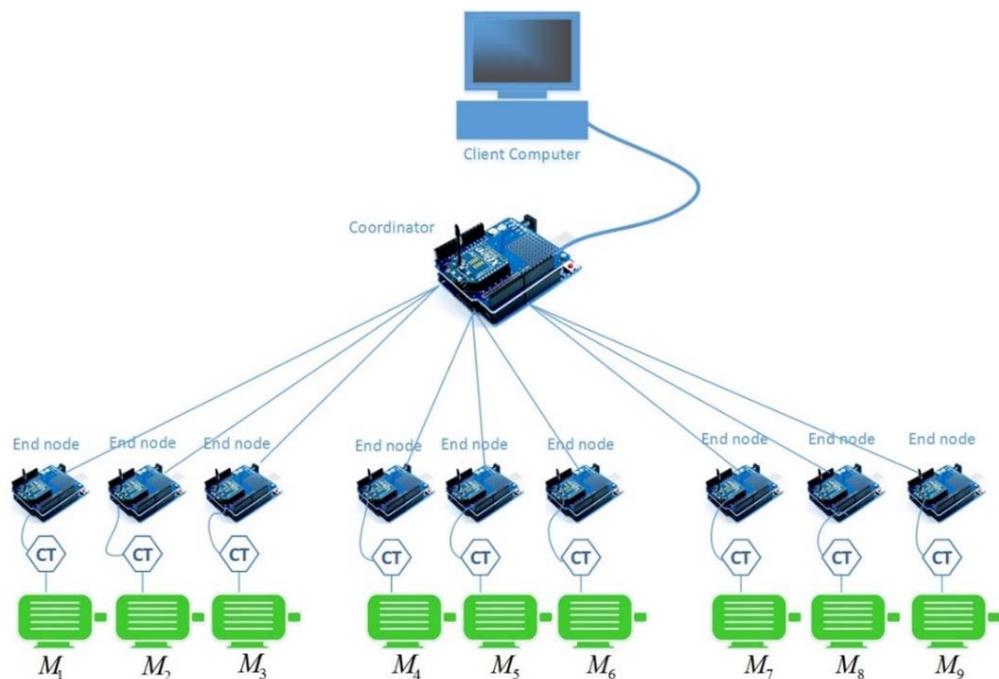


Figure 6.13: System diagram of network

In Figure (6.13), the role of coordinator is central point of the system for collecting each end-node sensor reading. The function of the coordinator is divided into three: applying the ANN technique on sensor data to identify the condition of the motor, displaying the reading on the hyper terminal and the Xbee

interface of the wireless sensor network. A polling method is done on the coordinator side, in which the coordinator sends a request to each end-node sequentially to send sensor readings back to the coordinator. This process is helpful in avoiding the collision of data from multiple end-nodes simultaneously to the coordinator. In case of collision occurs, the coordinator will identify the data as invalid input data, and resend a request to the end-node to transmit the same packet.

The role of the end node device within the network, is to collect the features from the electric current and speed sensors and store them in EPROM, and wait for the coordinator data-reading request (polling) and try to synchronise and give a response to the coordinator with the sensor's values. The Arduino Xbee shield receives the data from the electric current and speed sensors via input pins A<sub>0</sub> & 5 and then serialises the data before passing them to the coordinator. The Xbee uses libraries *<Xbee.h>* for transmission of data between Arduino and Xbee modules and *<FreqCounter.h>* for sideband measurements from the speed sensor, and *<threshold.h>* is used for defining the threshold level to ignore irrelevant sidebands. The flow diagrams (Figure 6.14) present the roles of the coordinator and end-node devices within network as per set of configurations.

In this research, the WSN network system has used byte-oriented data packet transmission by assigning start flag, address, control byte, information and End flag. Flags are used to indicate 'start of packet' and 'end of packet'. In this communication, a user can send a command as a command-frame structure from the coordinator side to the end-node device to start any new task, such as to start taking readings or to send readings stored in its EPROM, or to read electric current and speed sensor values and send them back. Once an end-node receives this command data packet it checks the address for this data packet. PAN ID will match with a pre-defined address and then it will do the ordered operation, or it will simply discard that data packet. The end-node will send similar data packets with information to the coordinator. On the computer side, the coordinator will check data packets for authenticity of sender.

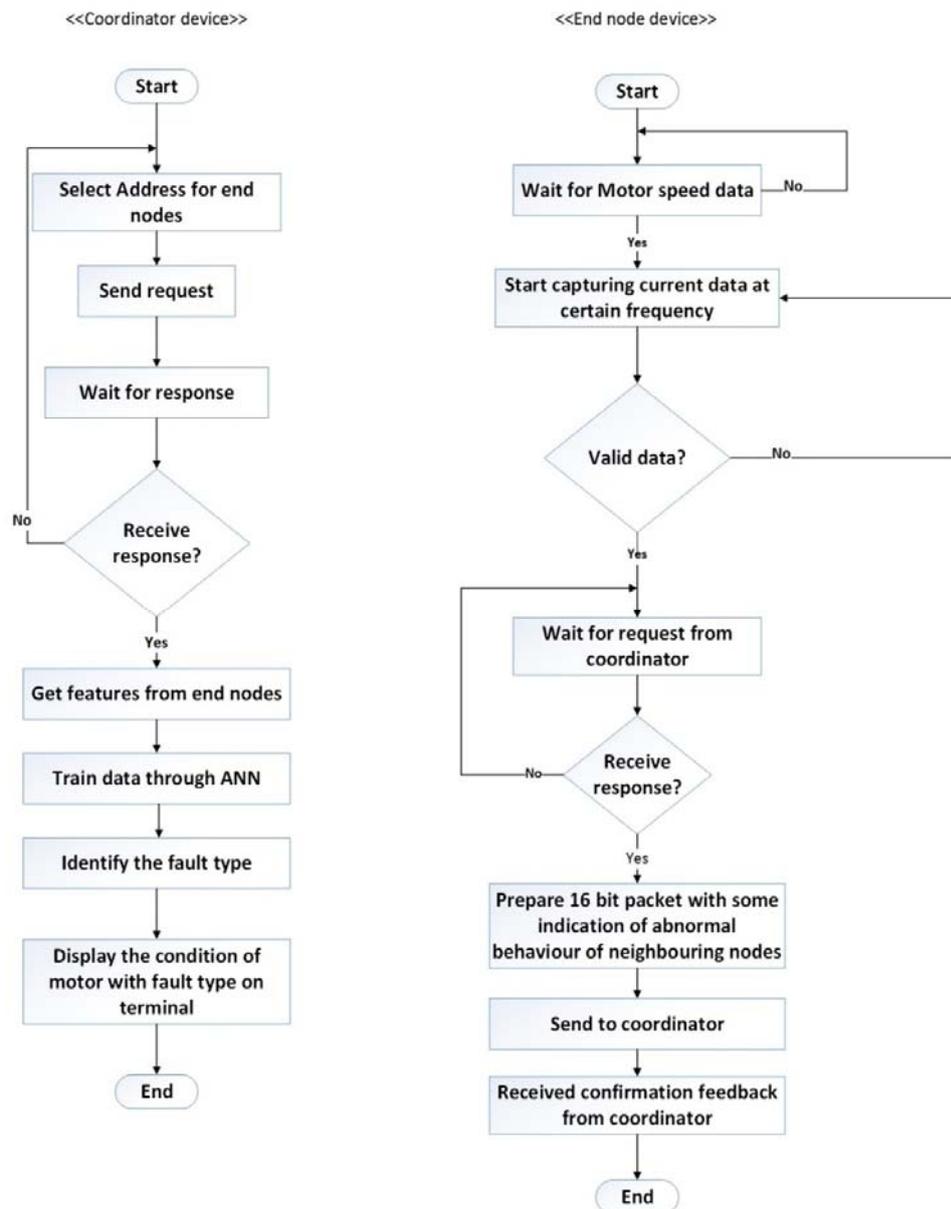


Figure 6.14: Role of coordinator and end-node devices within a network

### 6.5.1 Operation of the Arduino Sensor Node

In this study, the Arduino node has to configure the sample data per second at 50 Hz. The data is sequentially collected for one second in multiple samples and stored in the Arduino buffer. When all the packets are transmitted to the coordinator, the end-node goes into sleep mode until the next instructions are received from the coordinator. This system uses byte-oriented data packet transmission by assigning start flag, address, control byte, information and end flag. Flags are used to indicate ‘start of packet’ and ‘end of packet’. The data is transferred from end-node to coordinator in the form of different digital packets. Each measured data packet uses two bytes of memory at a 9600K Baud data rate.

In Figure 6.15 shows the packet structure and packet data communication flow. Each packet consists of 16 data bytes. In the data pole for each packet, eight bytes are used for collected data, two bytes for the time stamp of the header, three bytes for packet information and two bytes for the mark of Cyclic Redundancy Check (CRC) to check the results.

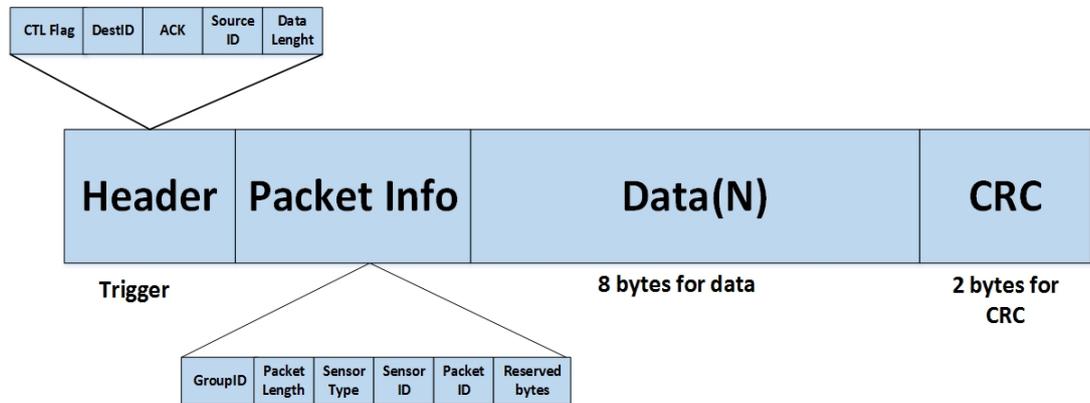


Figure 6.15: Frame Structure of Data Packet

In Figure 6.15, for sending a data packet, the application layer initiates the communication stack with the header and packet information that includes destination ID, source ID, acknowledgment, group ID, packet length, sensor types, sensor ID, and packet ID. Every byte in the packet is coded and transmitted over the same channel. Simultaneously, a CRC code is computed over the entire packet and transmitted. On the receiving end, the coordinator collects the coded bytes and decodes them. If the packet contents are incomplete, it will be considered a damaged packet.

The Figure 6.16 shows each activity of the microcontroller that plays a key role in the Arduino sensor board for data fusion, and transmitting the data to the base station as follows:

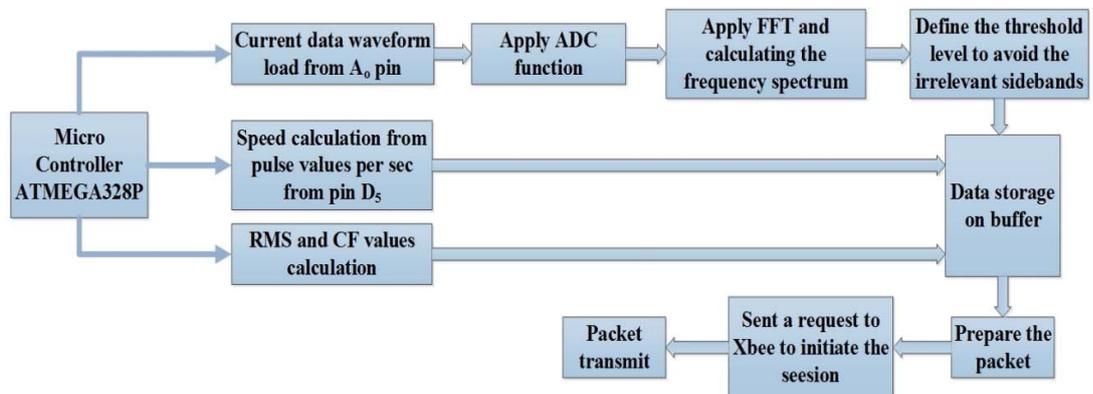


Figure 6.16: Step-by-step activities for data fusion, transformation, packet structuring and transmission in a micro controller

## 6.6 Arduino Fault diagnosis Case Study: Experimental Results for Two Faulty Motor

This case study demonstrates how to propagate a noisy fault signal, and its influence on other neighbouring motors' characteristic according to their distance. Hence, Motor 1 was chosen to be a faulty motor and faults were created artificially in Motor 1 to develop the motor-fault scenario that as follows:

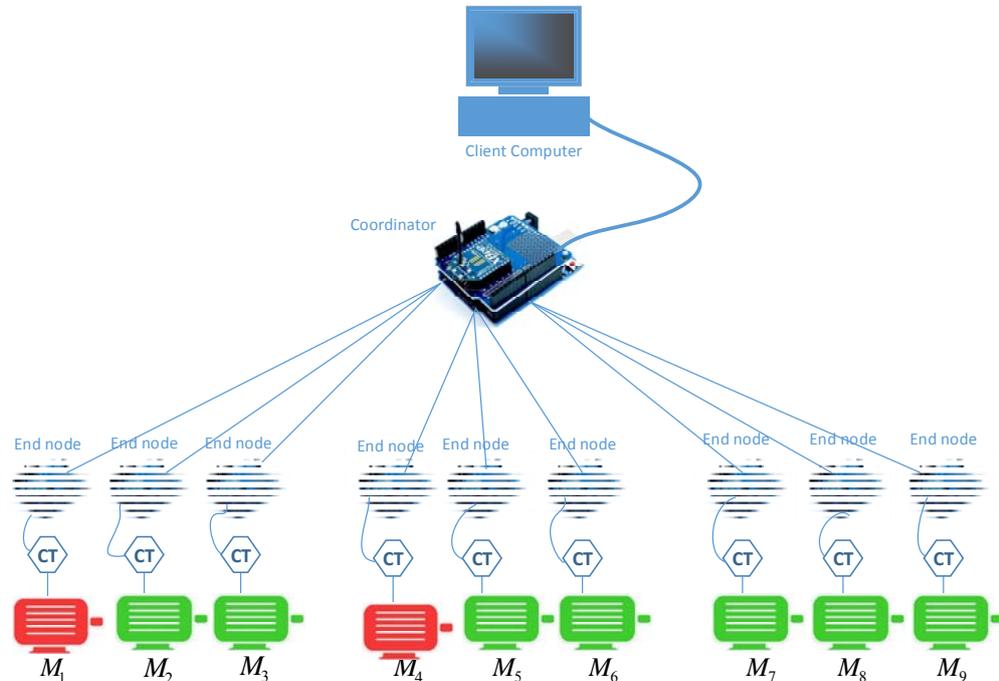


Figure 6.17: Sensing point analysis using Arduino over a network

As discussed, faults inside a motor are related to a set of frequency sideband components. Therefore, faults are represented in a power-line network by a set of frequency points with different amplitude ratios. With each motor, data collection is carried out from CT's connected with Arduino pin ( $A_o$ ). Speed sensors are mounted on motor rotors to measure the speed in RPM at certain times. In Figure 6.18, the fault-symptom sideband components have significant observable impacts on the electric current spectrum of other motors. This impact is due to the propagation of a fault signal through the main power-line. The terminal output (Figure 6.18) shows the fault-diagnosis process between coordinator and end nodes in both healthy and faulty conditions. The  $x$  value show the points for the waveform, and  $fr$  is the frequency-domain points vector that sends to the coordinator for analysis. A total of 20 values were defined for the band-peak vector for ANN analysis and this decides the motor's condition and fault type. After ANN analysis, the coordinator will separate the suspected



Table 6.2 shows the spectrum of all the motors within a network, acquired from Arduino end node modules and store in Matlab for analysis as follows:

Table 6.2: Significant fault frequencies at targeted sensing points

<b>Motors</b>	<b>Speed</b>	<b>Side band electric current peaks</b>	<b>Condition</b>
<b>Motor 1</b>	1012	[0 25 34 50 62 75 0 0 0 0 0 155 0 0 0 0 0 0 0 0];	Broken Rotor Fault Symptoms and noisy signal
<b>Motor 2</b>	1200	[50 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];	Healthy Motor
<b>Motor 3</b>	1220	[50 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];	Healthy Motor
<b>Motor 4</b>	1200	[50 52 0 0 0 0 0 0 0 0 145 0 0 0 0 0 0 0 0];	Unknown fault or some suspected noise peak
<b>Motor 5</b>	1210	[50 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];	Healthy Motor
<b>Motor 6</b>	1280	[50 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];	Healthy Motor
<b>Motor 7</b>	1250	[50 52 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];	Healthy Motor with some suspected noise peak
<b>Motor 8</b>	1280	[50 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];	Healthy Motor
<b>Motor 9</b>	1250	[50 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];	Healthy Motor

As we can observe from Table 6.2, Motor 1 has significant sidebands that clearly show the symptom of a BRB fault with the combination of frequency points 25 & 75 around the fundamental frequency. In Motor 1, a suspected sideband component appears at point 155, but the system would consider this a noise signal that comes from other source. Motors 2, 3, 5, 6, 8 and 9 show their state as healthy with no fault-sideband components. Motors 4 and 7 show some fault components but are showing some suspected noise signal at frequencies 52 Hz & 145 Hz, but not the targeted fault symptoms.

Operation of all motors is related to their transient condition at start-up. This transitory state of a motor appears as a dramatic increase of motor speed from a nominal state to the motor's nominal speed, as shown in Figure 6.19 following. To measure the speed, we take 10 samples at each rotating pulse generation counting the motor at different intervals of time. All the measurements were taken from the Arduino speed sensor individually from each motor. However, due to the limitation of resources and speed sensors, we took the samples sequentially because the speed of the motor does not show high variation in a steady state until abnormal behaviour of the motor interferes with the rotating pulse.

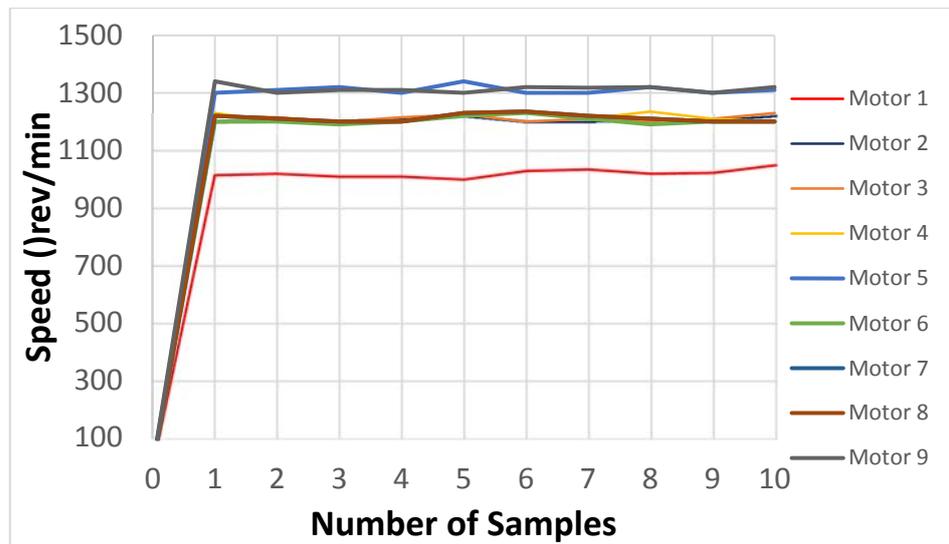


Figure 6.19: Measurement of all motors speed during and after the start-up process

It can easily be observed in Figure 6.19 that Motor 1 comes in at a range 1010 – 1045 rpm, which is bit low speed compared with other motors, due to the brake load. Any variation in the estimated speed of motor may create abnormality in the proper speed. There is a big variation in Motors 5 and 9, because they are running on a high-power rating compared with the neighbouring motors. But we can observe that there is a significant variation between points 5 to 8 due the abnormal behaviour in the same type of Motor 1. The other motors generate a rotating pulse in the normal range, that is, 1200-1250 rpm, which is are the required range of motor speed.

## 6.7 Chapter Summary

This chapter focused on the development of WSN test-bed architecture based on Arduino Xbee modules. The wireless sensor nodes interacted with a physical motor that offered a representation of the modelled environment. The sensor node was successfully tested with the single motor and achieved the required results. Different output results were presented by using the manual current probe and Arduino measurements, which show close accuracy in capturing data in a real-time environment. The Chapter 7 is based on different case scenarios with the combinations of single or multiple faults to prove the concept and accuracy of the developed prototypes hardware.

## RESULTS AND ANALYSIS

### 7.1 Introduction

This chapter focuses on the evaluation and implementation of distributed diagnosis concepts, based on simulation and experimental results. Here, the spectrums of electric current signals related to each motor in a network will be examined, to estimate detectability of injected faults, and the approximate consistency of the diagnosis process. Fault type, motor speed, strength of fault, other features and relative location in the network are the significant variables of the diagnostic process. This chapter discussed different case studies and their simulation results based on different fault indices, propagation of fault signals over power-line network and the use of ANN to employ a smart diagnosis and decision-making approaches in interpreting possible fault indices.

### 7.2 Case Study 4: Faulty Motor within Network with Single Fault Symptoms

In this case study, a targeted fault was artificially created in Motor 1 in a network (see Figure 7.1). Initially, a uniformly running industrial-motor network was chosen to experiment with the same specifications that are discussed in Table 4.3 (Chapter 4) for a model (Figure 4.3).

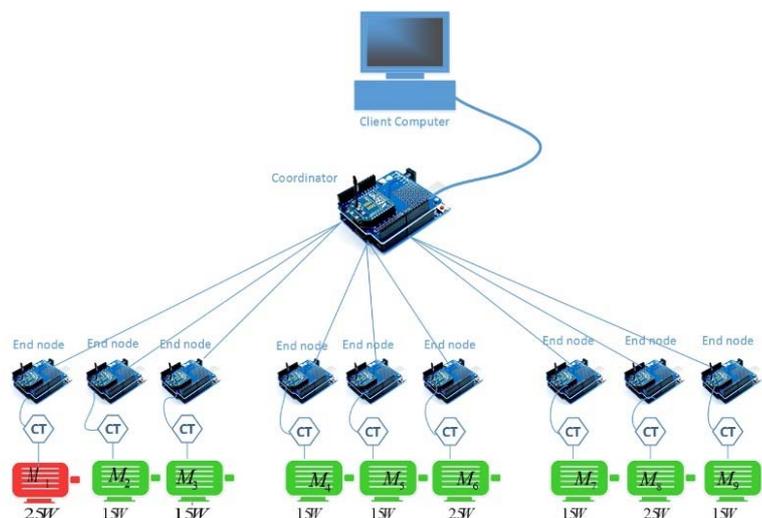
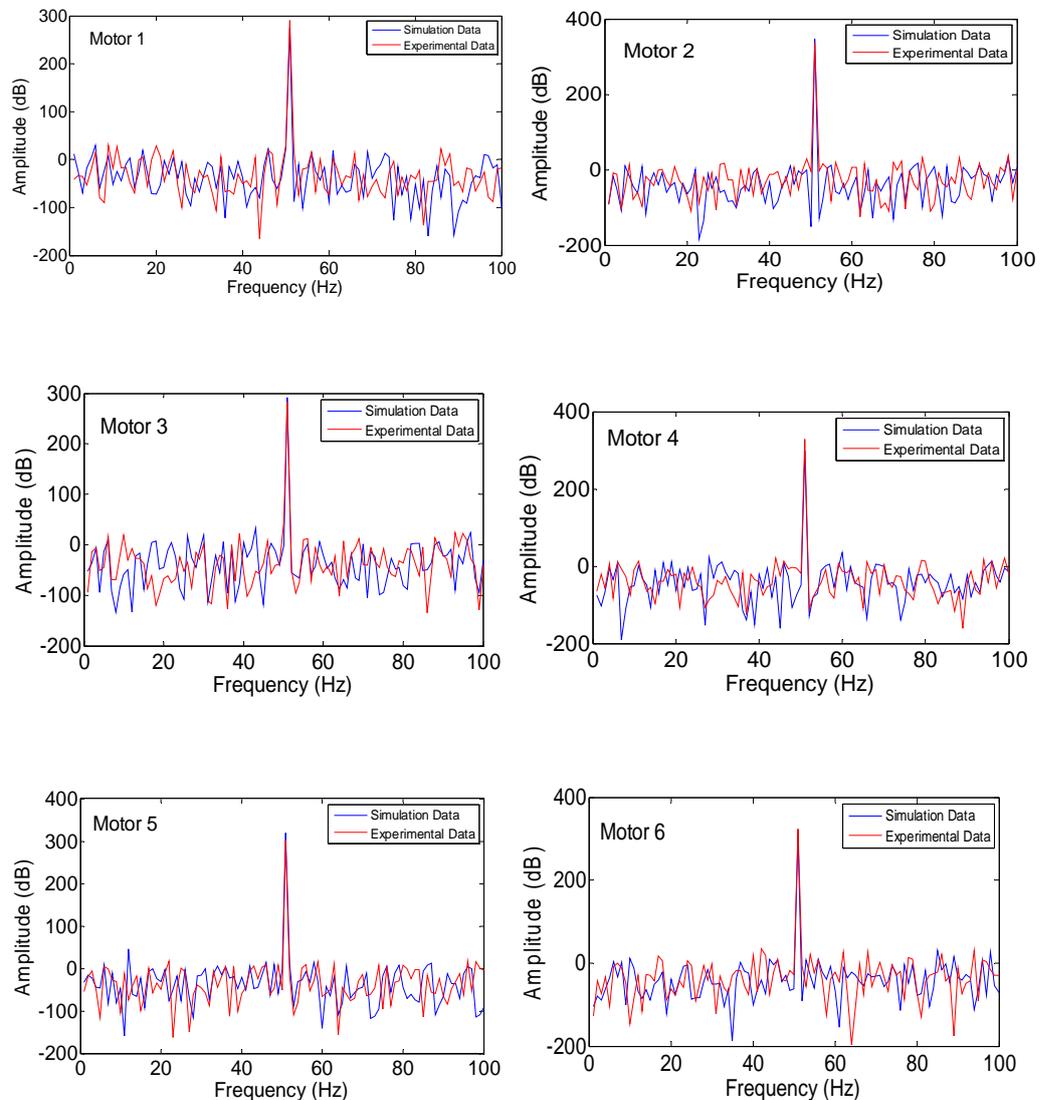


Figure 7.1: Single faulty motor within network

All data was captured from each motor through an Arduino wireless node in normal condition. A BRB fault was then injected into motor to estimate the severity of fault of Motor 1 (25W) and any significant influence on other motors through the main power-line. Same configuration was set on the simulation motor to compare the results with the experimental setting.

Impedance on every motor in the main power-line is configured to  $1\Omega$  (resistance), and overall impedance of every bus to the main power bus was configured to  $0.6\Omega$  (inductive) and  $0.6\Omega$  (resistive). A strong motor (25W) was chosen for manifesting the signal into low-power motors within the same bus, and others. The electric current spectrum of all the motors prior to the manifestation of the fault signal is shown in Figure 7.2 as follows:



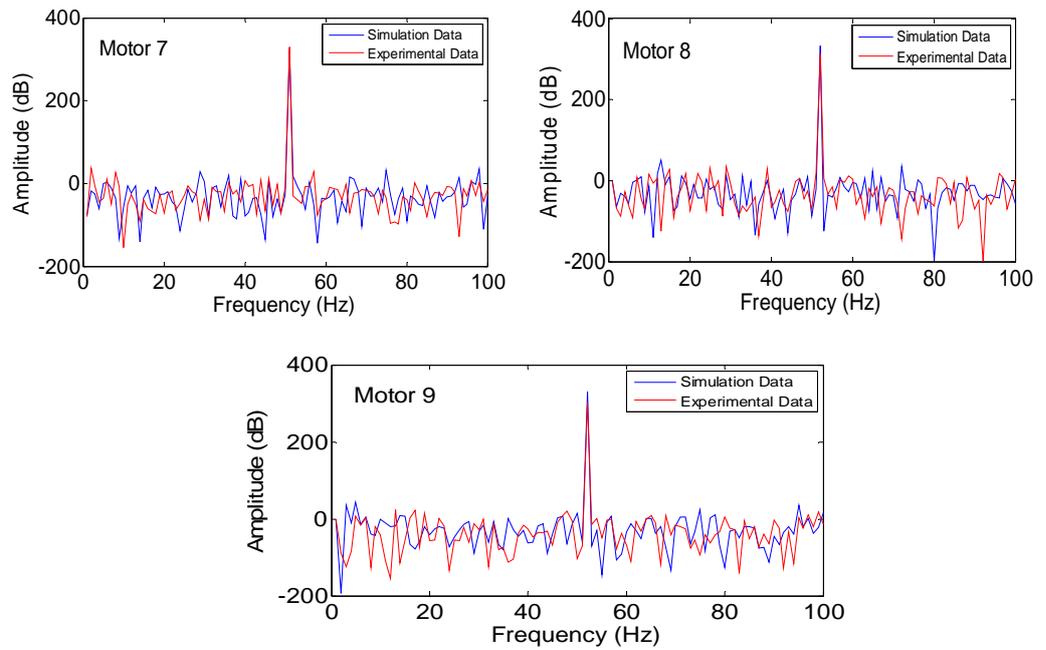
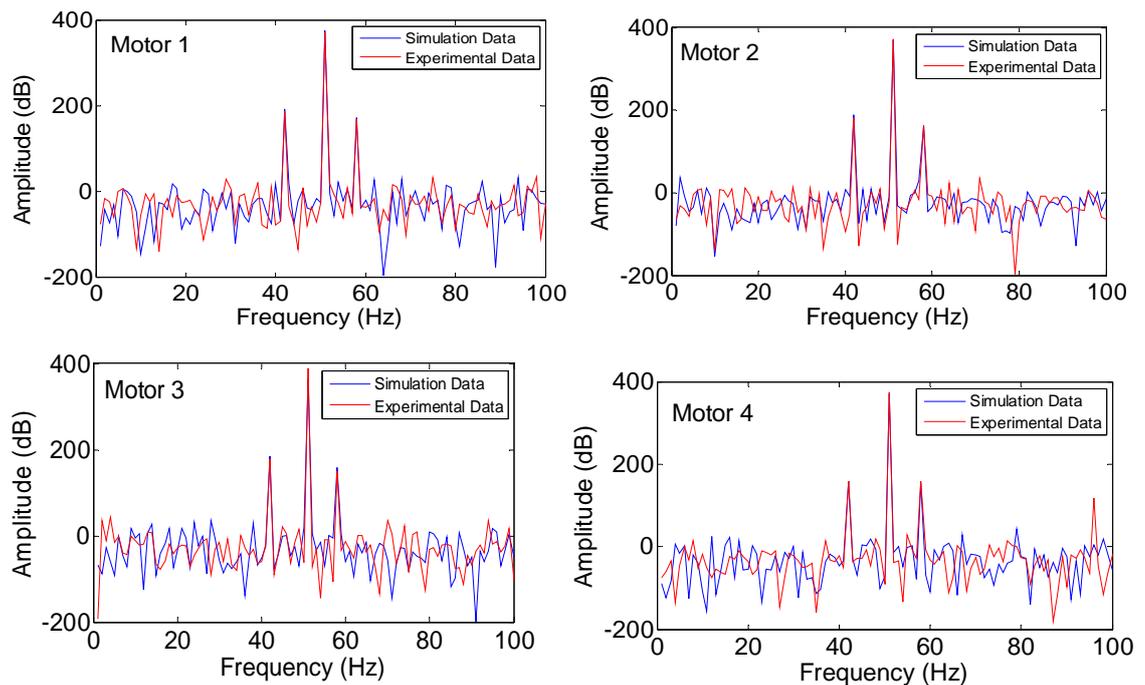


Figure 7.2: Electric current spectrum analysis of all motors at no-load with no fault condition

As shown in Figure 7.2, after applying FFT on the electric current signal of all the motors, at no load, we observe that all the motor signals show a normal condition and no sidebands appeared on any motor's spectrum. Then, at full load, significant sideband frequencies appeared at all the motors' spectrums, with different amplitude values to show the influence of different loads. Figure 7.3 shows the mirroring BRB faulty sidebands around the fundamental frequency among all the motors with varying amplitude values as follows:



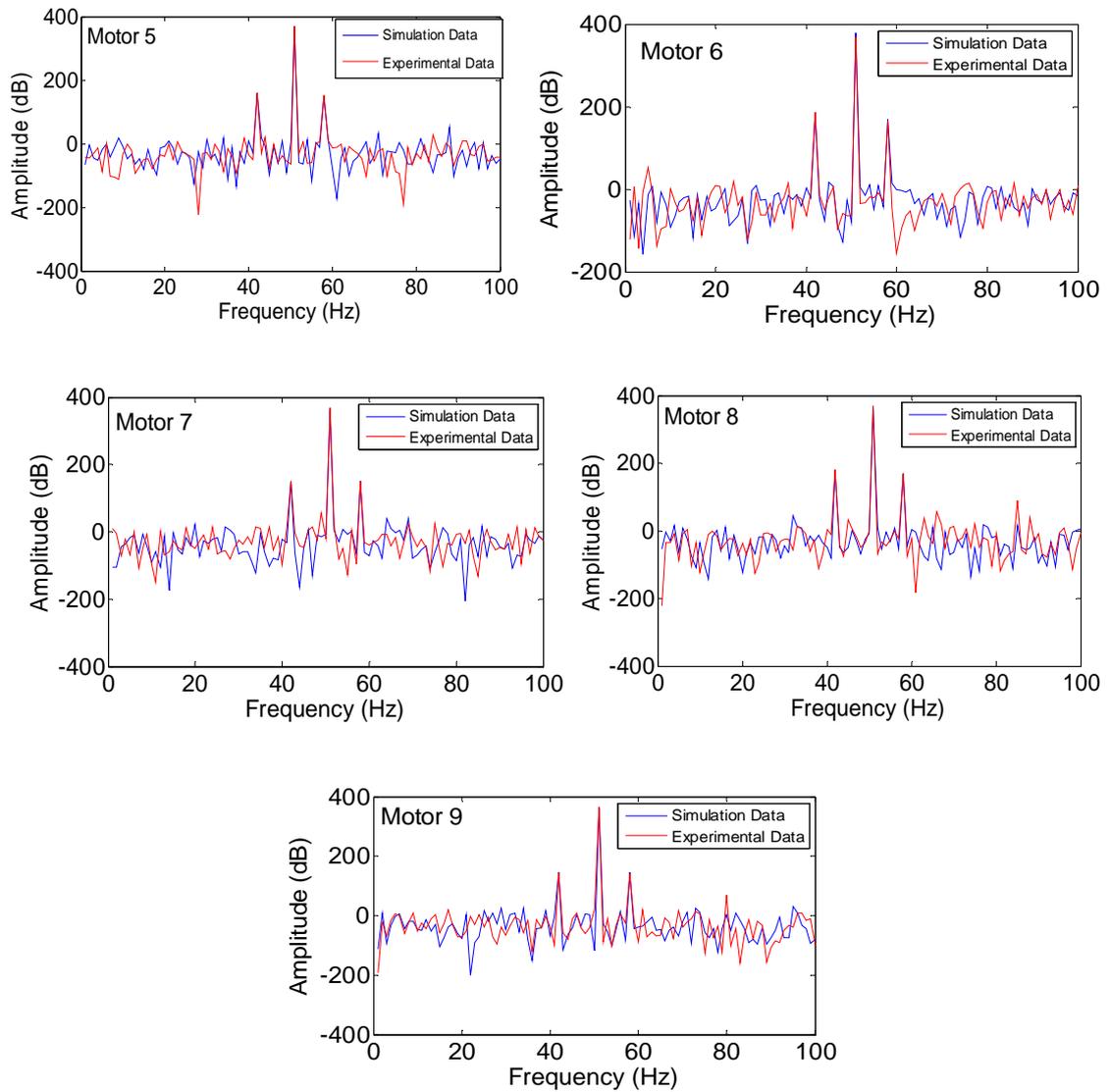


Figure 7.3: Electric current spectrum analysis of all motors at full-load and BRB fault conditions

The Figure 7.4 presents the analysis of electric current spectrum of all the motors, and their amplitude values. It can be observed that a BRB fault-signal mirroring sideband appeared at the same frequency point but with a different amplitude rate, according to the size of motor and its distance from Motor 1.

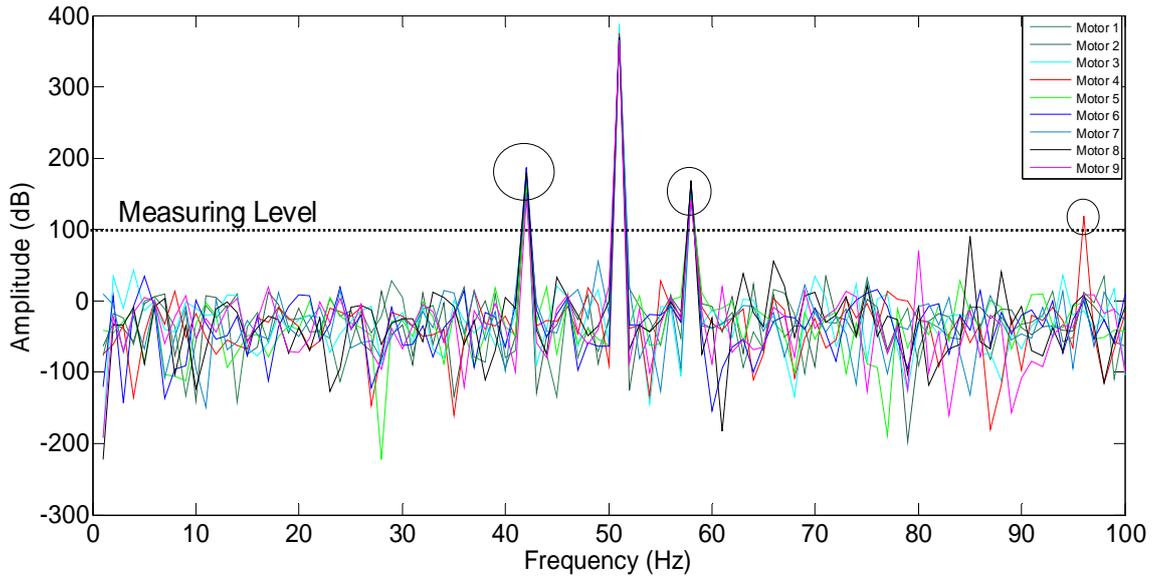


Figure 7.4: Multi-frequency fault propagation by Motor 3 at full-load

A magnified closer view of the abnormal frequencies 42 Hz and 58 Hz shows the shared influence of the faulty motor on other targeted motors and a change in spectrum, mirroring a BRB fault signal in the following Figure 7.5.

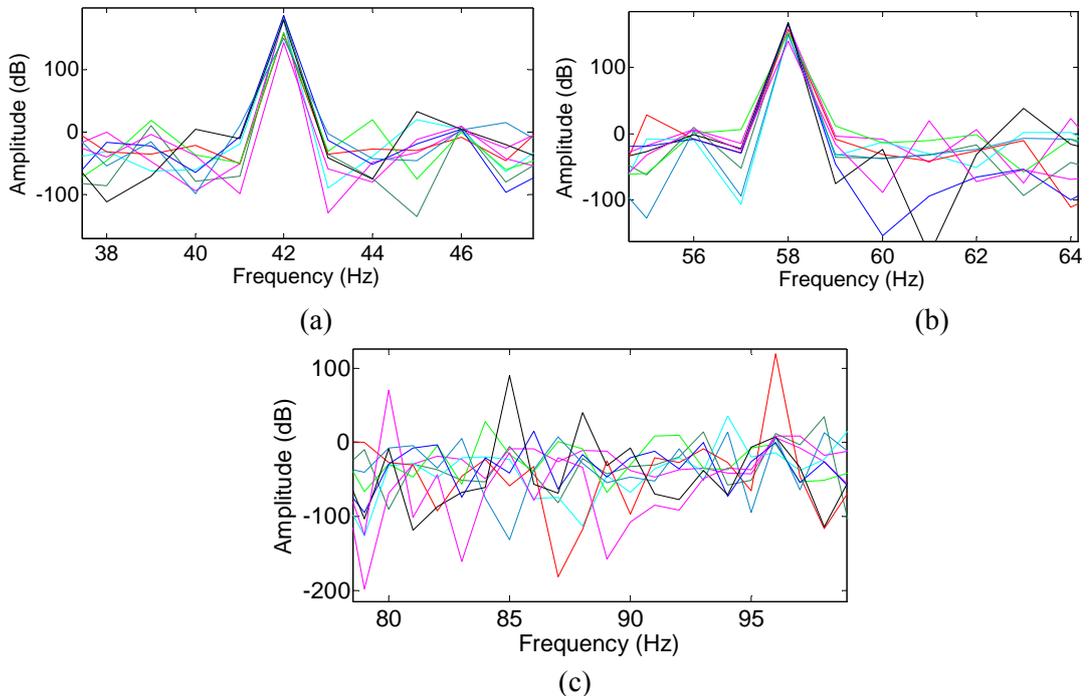


Figure 7.5: Faulty signal at significant frequency points  
 (a) 42 Hz (b) 58 Hz (c) 81 Hz, 85 Hz and 97 Hz

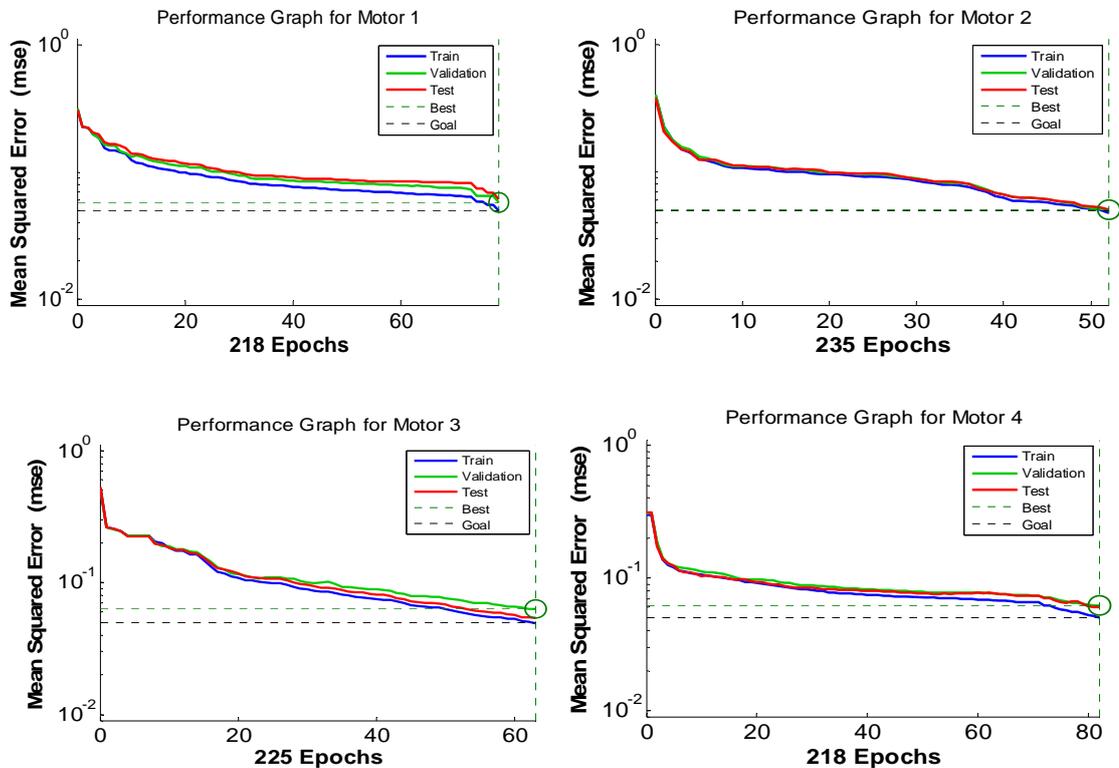
Some abnormal frequencies were also identified during the experiment at 81 Hz, 85 Hz and 97 Hz. After analysis, these frequencies did not show the



Table 7.2: Learning data set for training process for multiple motors in case study 4

Motors	Features					
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
<b>Motor 1</b>	1012	0.22	[42;50;58]	[186.5; 370.6;168.9]	0.250	0.312
<b>Motor 2</b>	1200	0.07	[42;50;58]	[180.2;365;164.1]	0.147	0.200
<b>Motor 3</b>	1220	0.06	[42;50;58]	[179.4;365.2;160.8]	0.142	0.191
<b>Motor 4</b>	1200	0.07	[42;50;58;97]	[159;372;158.5;102.5]	0.147	0.200
<b>Motor 5</b>	1210	0.06	[42;50;58]	[158.6;372.4;152]	0.122	0.210
<b>Motor 6</b>	1280	0.01	[42;50;58]	[151.3;364;150.7]	0.150	0.215
<b>Motor 7</b>	1250	0.03	[42;50;58]	[148.1;365.2;150.2]	0.164	0.210
<b>Motor 8</b>	1280	0.01	[42;50;58;81;88]	[143.4;372;164.1;90;43]	0.150	0.190
<b>Motor 9</b>	1250	0.03	[42;50;58;81]	[142.1;364;140.4;75]	0.164	0.210

As we can deduce from Table 7.2 after comparing the feature ranges, Motor 1 was affected by the BRB fault and Motors 2 and 3 are the most affected motors, due to being on the same sub-bus. However, because of the distance between motors, it does not show a strong value for the BRB fault. In this situation, the system considered these changes an unknown fault in other motors (in which bearing, stators etc. faults are included). The Arduino coordinator can easily analyse from the features the condition of each motor, and identify the fault type through any change in behavioural characteristics, as follows in Figure 7.6.



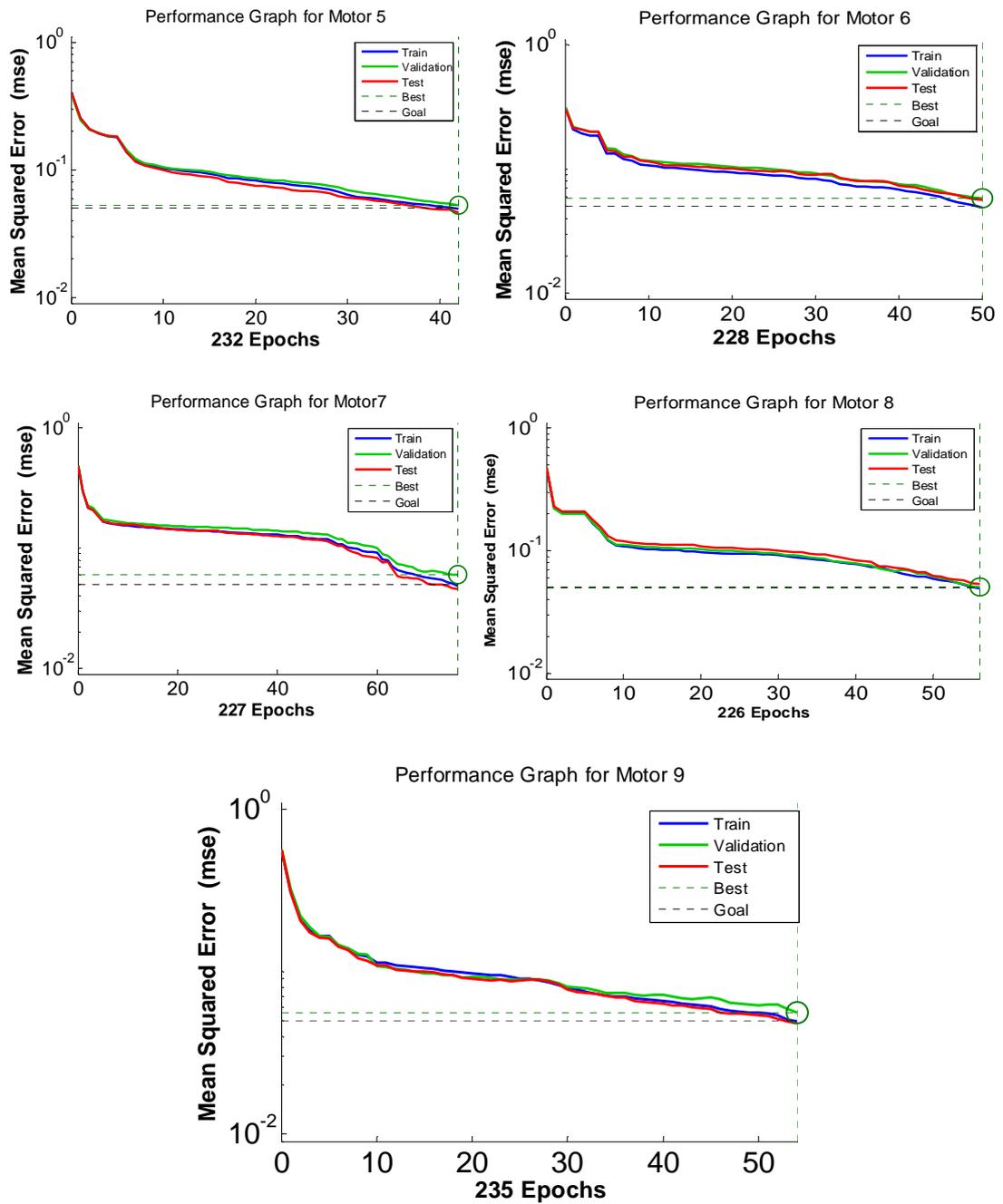


Figure 7.6: Performance graphs of all motors using [6x10x4] neural network architecture

The fewest possible number of epochs were also used during the training period to achieve the required accuracy, which shows the best performance in the targeted architecture, along with the smallest error percentage, that is, between 218-235 epochs. Figure 7.6 shows the training performance graph of the neural network [6x10x4], which attained the best result of the other NN structures in the least time, and achieved the required percentage of results.

After measuring the performance of each motor, the second means of testing the performance of the neural network was to measure the classification confusion matrices for the various types of error that occurred during the training process. Figure 7.7 presents the confusion matrices for the three process phases of training, testing and validation of each motors respectively.

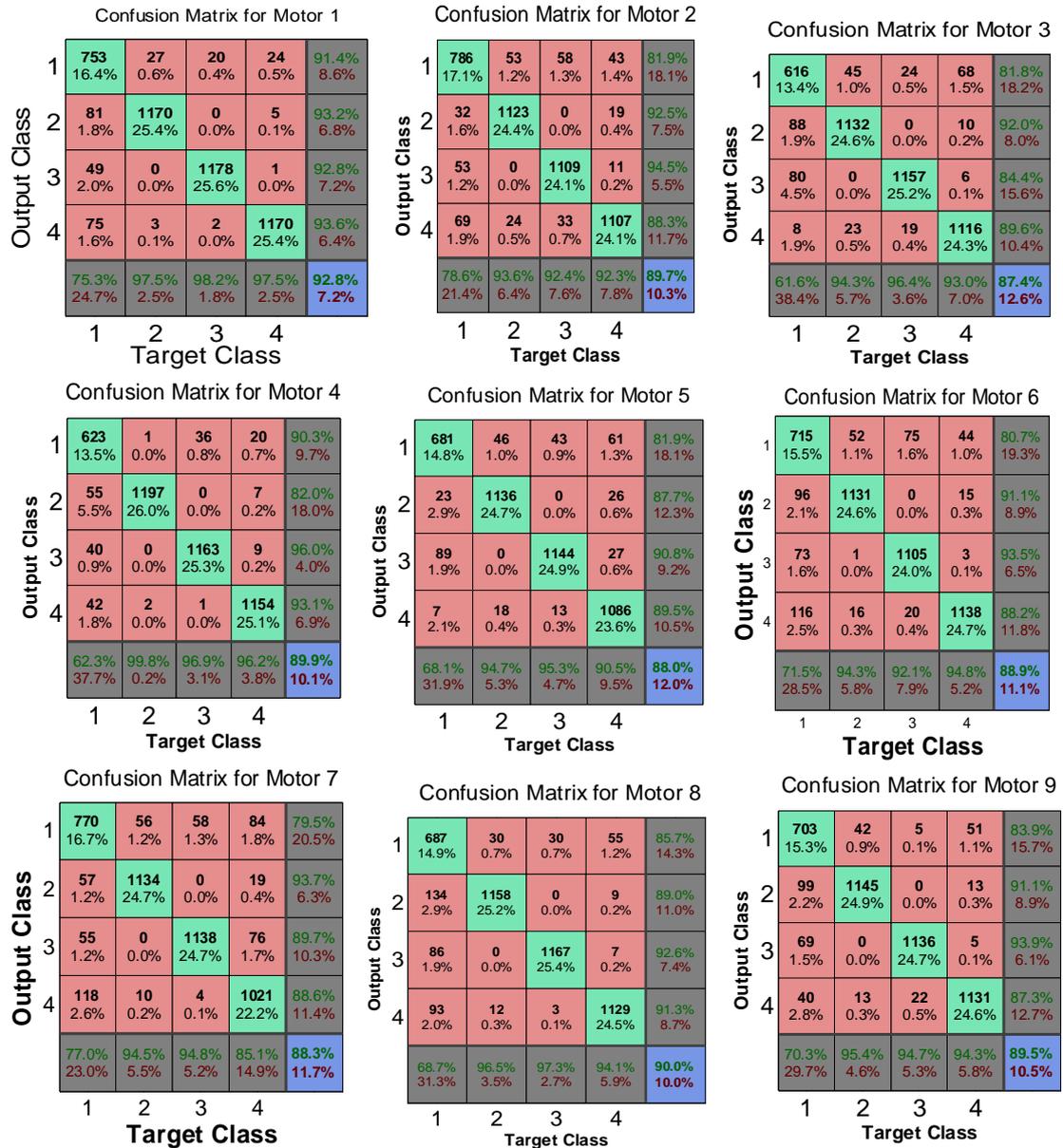


Figure 7.7: Confusion matrices of all motors using targeted and output classes

A total of four targeted and predicted classes (horizontal and vertical) were defined to compare each motor-feature data set. In the case of successful classification of a targeted class trial, the diagonal cells are shown in green. Each diagonal cell indicates the number of cases that have been correctly classified by the network, to identify feature condition, whether healthy or faulty, for each

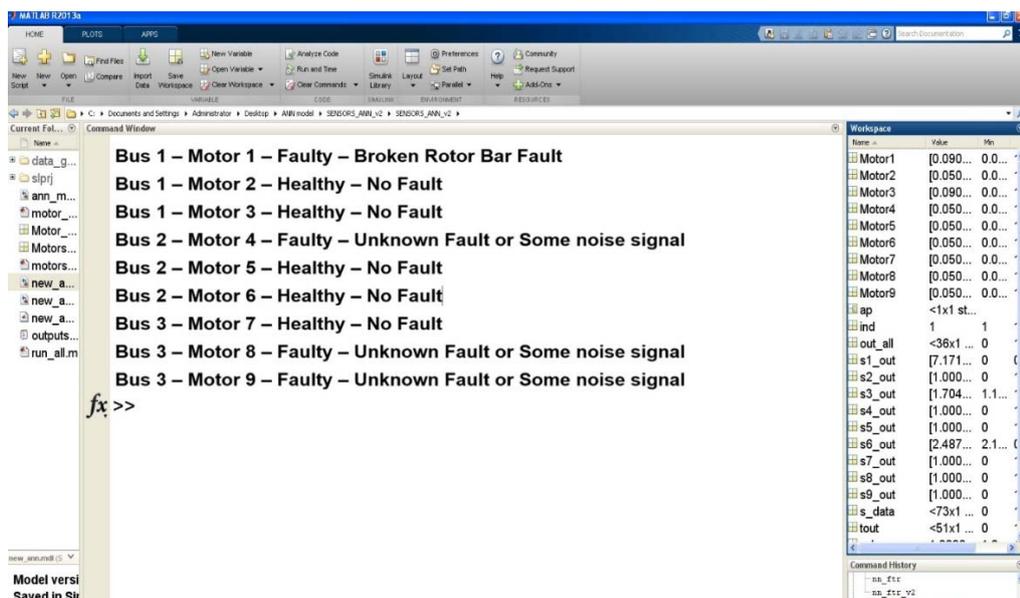
motor. The non-diagonal cells in red specify the number of cases that have been wrongly classified by the ANN, where it did not identify the condition of features. The last cell, in blue, in each matrix indicates the total percentage of cases that were classified correctly in green and vice-versa in red.

Table 7.3: Best Performances for classification of all motors for case study 4

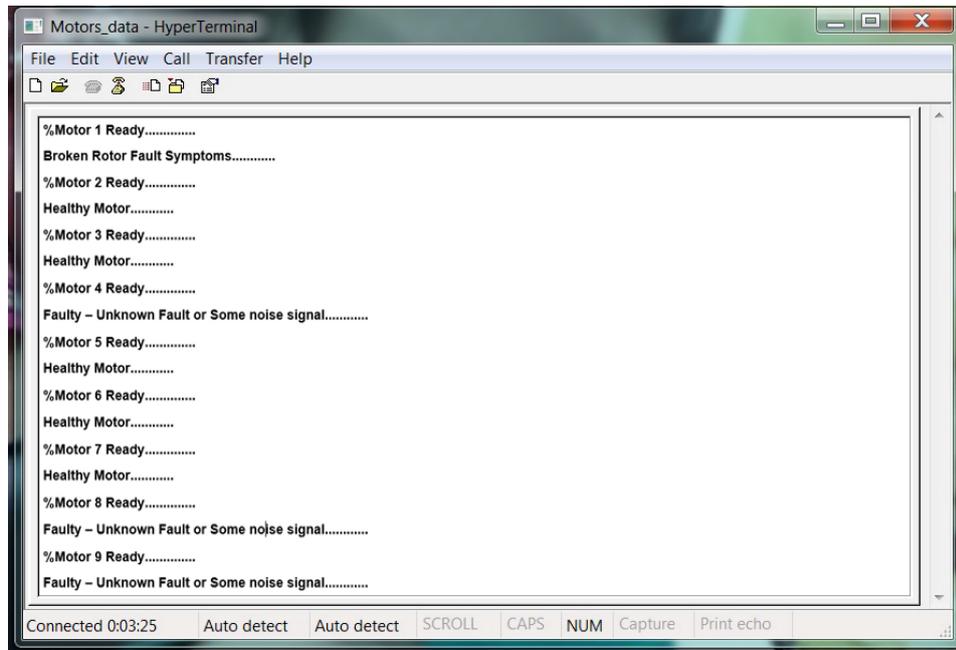
Architecture	Motors	MSE Performance	No. of Epochs	Accuracy (%)	Classification error (%)
<b>[6x10x4]</b>	<b>M1</b>	0.0518	218	92.8	7.2
	<b>M2</b>	0.0483	235	89.7	10.3
	<b>M3</b>	0.0512	225	87.4	12.6
	<b>M4</b>	0.0522	218	89.9	10.1
	<b>M5</b>	0.0495	232	88.0	12.0
	<b>M6</b>	0.0504	228	88.9	11.1
	<b>M7</b>	0.0497	227	88.3	11.7
	<b>M8</b>	0.0502	226	90.0	10.0
	<b>M9</b>	0.0496	235	89.5	10.5

With reference to Figure (5.5) in Chapter 5, it can be seen that through the chosen neural network architecture [6x10x4] in Table 7.3, a satisfactory accuracy was achieved in fault detection in the feature vector, ranging from 87 to 92 percent. This reflects the performance efficiency of the ANN algorithm, in reducing the level of uncertainty in decision-making in single motor-fault, where frequency signal makes the diagnostic process more complex.

Figure 7.8 shows a comparison between a Matlab simulation and an experimental result for validity of the developed prototypes as follows:



(a)



(b)

Figure 7.8: Comparison of motor condition outputs  
 (a) Matlab simulation (b) Test-bed experimental results

### 7.3 Case Study 5: Dissimilar Faulty Motors with Multiple Faults in the Same Bus

To examine the functionality of simulation prototypes, various further conditions of multiple motors within a network were investigated in this case study. For that, two different sizes of motor were chosen from the same bus, to observe the influence of a fault signal on other motors, as shown in Figure 7.9 as follows. The network shown in following figure was selected for the experiment.

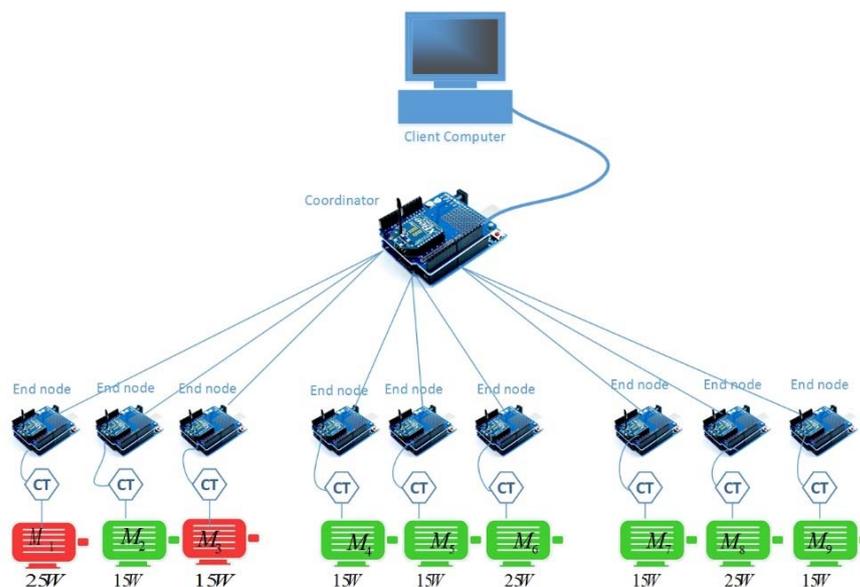
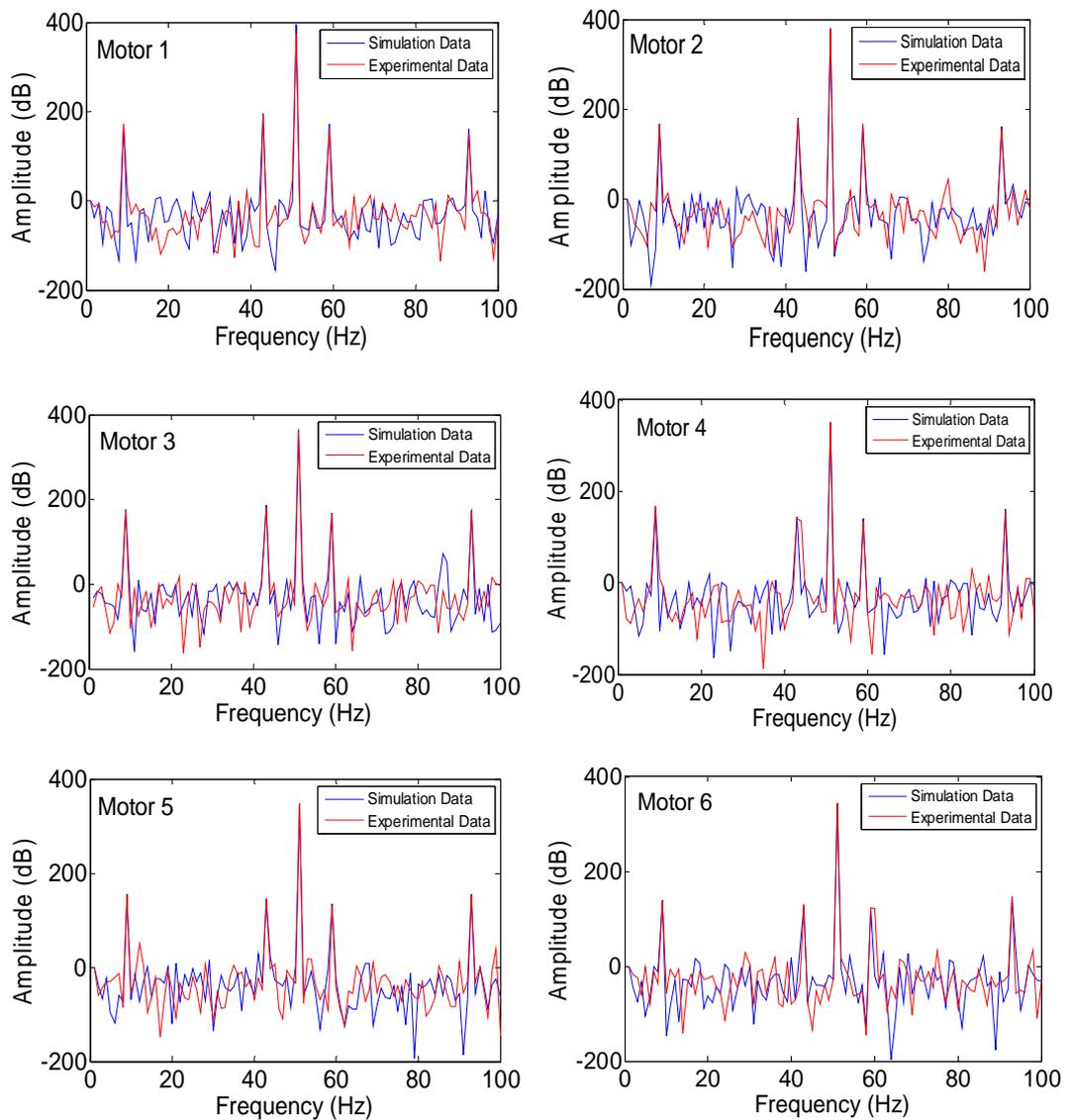


Figure 7.9: Multiple faulty motors within the same bus

Referring to Figure 7.10, BRB and eccentricity faults are observable to analyse the attenuation strength of the fault spectrum and validate its ability to diagnose the fault in each faulty situation. For the case study, BRB fault was integrated in Motor 1 and an air-gap eccentricity fault in Motor 3 at the full-load level, to observe the significant frequency sidebands. In this case study, along with all motor-level sensing points, there were other sensing points also considered, to observe the strength of the signal. Figure 7.10 shows the mirrored sideband around the fundamental frequency among all the motors with varying amplitude values as follows:



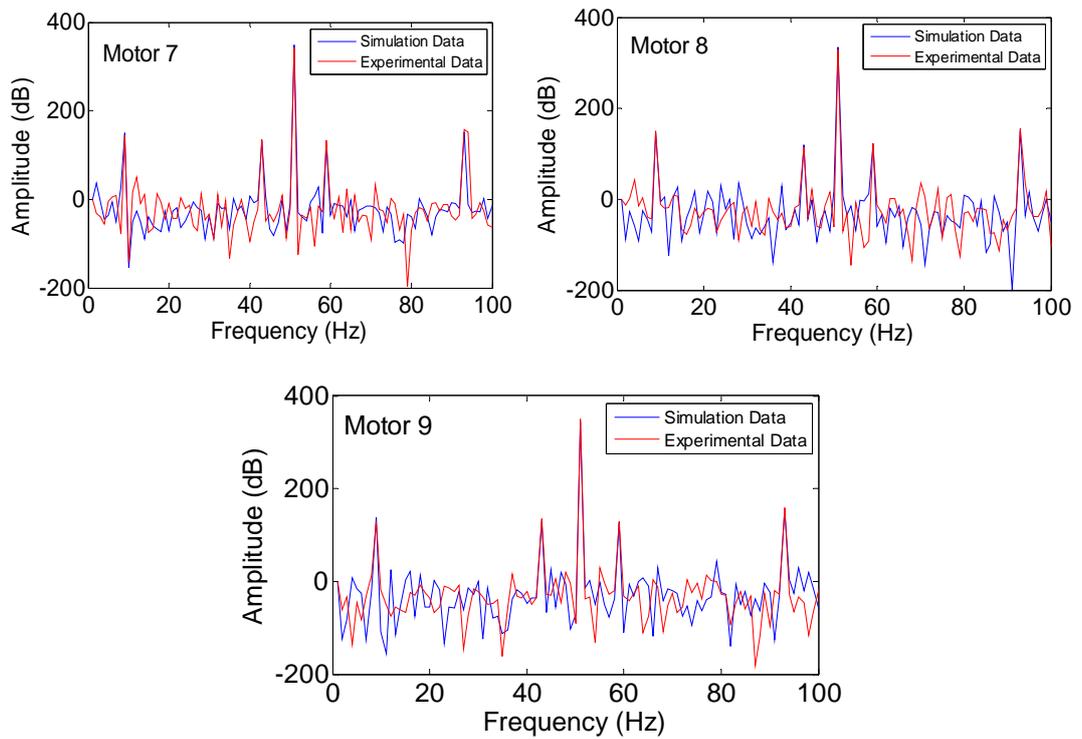


Figure 7.10: Electric current spectrum of all motors at full-load in BRB and ECE fault conditions

Figures 7.11 and 7.12 present the electric current spectrum from all the motors, and their amplitude values. It can be observed from Figure 7.12 that BRB fault-signal mirroring sideband (at frequency points 43 Hz and 57 Hz) appeared at the same frequency point, but had a different amplitude rate, according to the size of motor and distance from Motor 1. The ECE fault was propagated from Motor 3 (at frequency points 9 and 91) and manifested into other motor's electrical current signals.

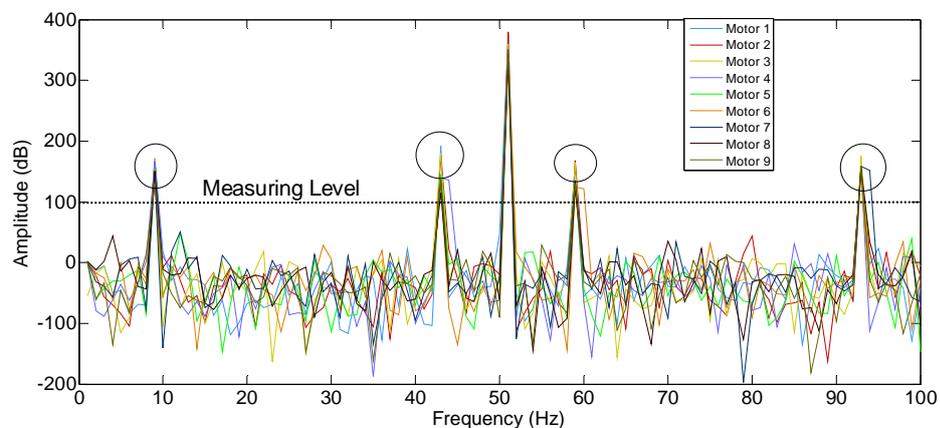


Figure 7.11: Multi-frequency fault propagation by Motor 3 and Motor 1 at full-load, to observe the BRB and eccentricity fault influences on different motors

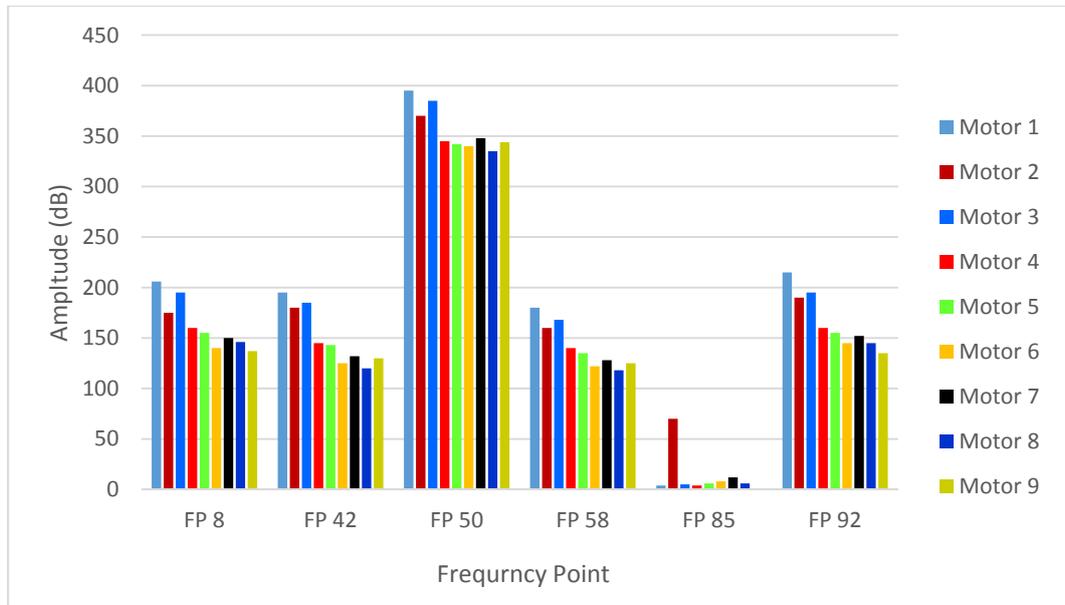


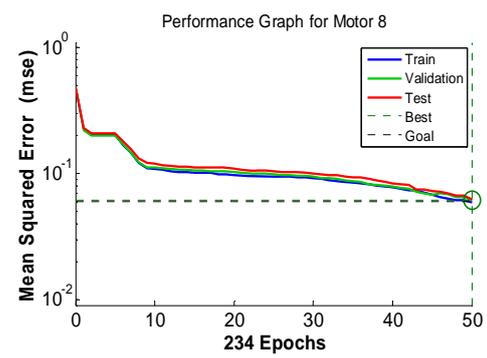
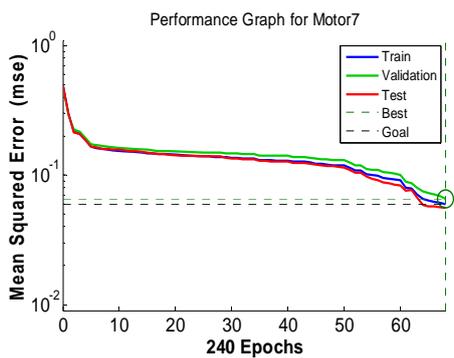
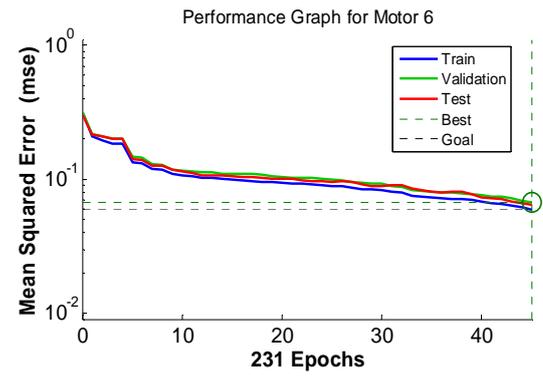
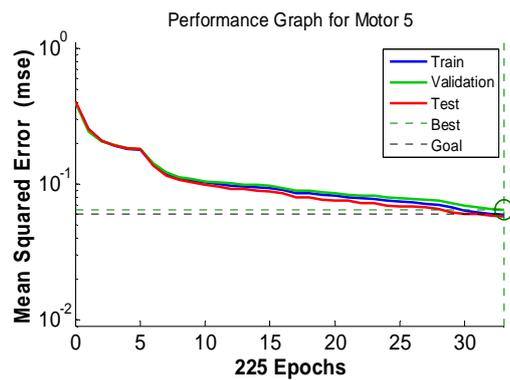
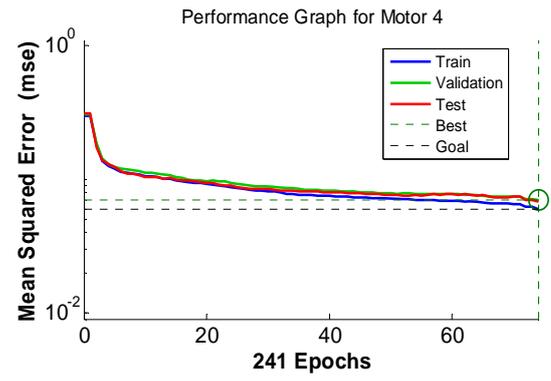
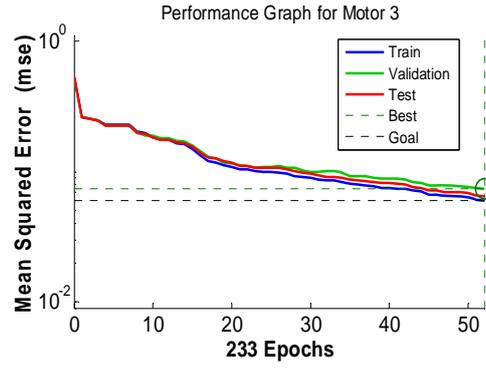
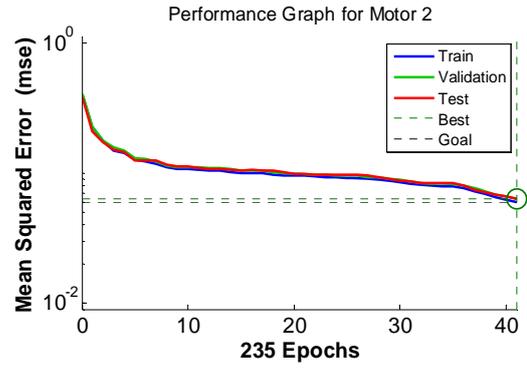
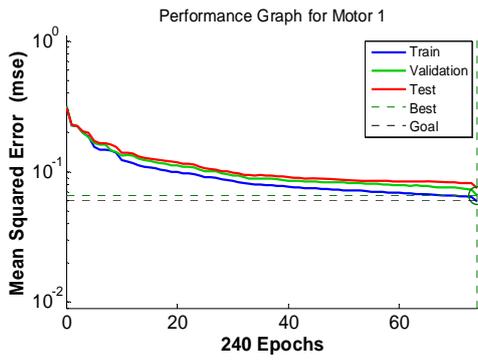
Figure 7.12: Analysis and observation chart of multi-frequency fault influence of Motor 1 and Motor 3

The following Table 7.4 presents the learning data sets of all motors for training to decrease the chance of confusion in decision-making.

Table 7.4: Learning data set for training process for multiple motors for case study 5

Motors	Features					
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
<b>Motor 1</b>	1012	0.21	[8;42;50;58; 92]	[185;195;395;180;2;160]	0.250	0.312
<b>Motor 2</b>	1200	0.07	[8;42;50;58;85;92]	[165;180;370;160;60;162]	0.147	0.200
<b>Motor 3</b>	1130	0.19	[8;42;50;58; 92]	[175;185;385;168;5;175]	0.142	0.191
<b>Motor 4</b>	1200	0.07	[8;42;50;58; 92]	[160;145;345;140;4;160]	0.231	0.290
<b>Motor 5</b>	1210	0.06	[8;42;50;58; 92]	[155;143;342;135;6;155]	0.122	0.210
<b>Motor 6</b>	1280	0.01	[8;42;50;58; 92]	[140;125;340;122;8;145]	0.150	0.215
<b>Motor 7</b>	1250	0.03	[8;42;50;58; 92]	[150;132;348;128;12;152]	0.164	0.210
<b>Motor 8</b>	1280	0.01	[8;42;50;58; 92]	[146;120;335;118;6;145]	0.150	0.190
<b>Motor 9</b>	1250	0.03	[8;42;50;58; 92]	[137;130;344;125;1;135]	0.164	0.210

Figures 7.13 and 7.14 present performance graphs and confusion matrices for the three process phases of training, testing and validation of each motor respectively.



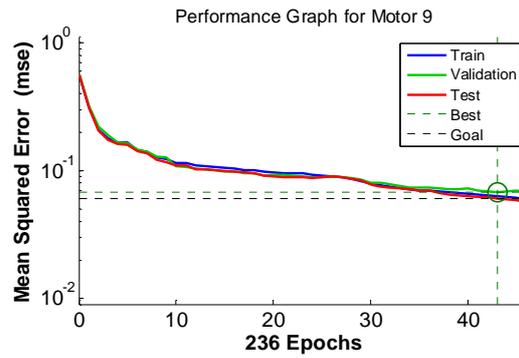


Figure 7.13: Measurement of performance graphs of all motors using neural network architecture

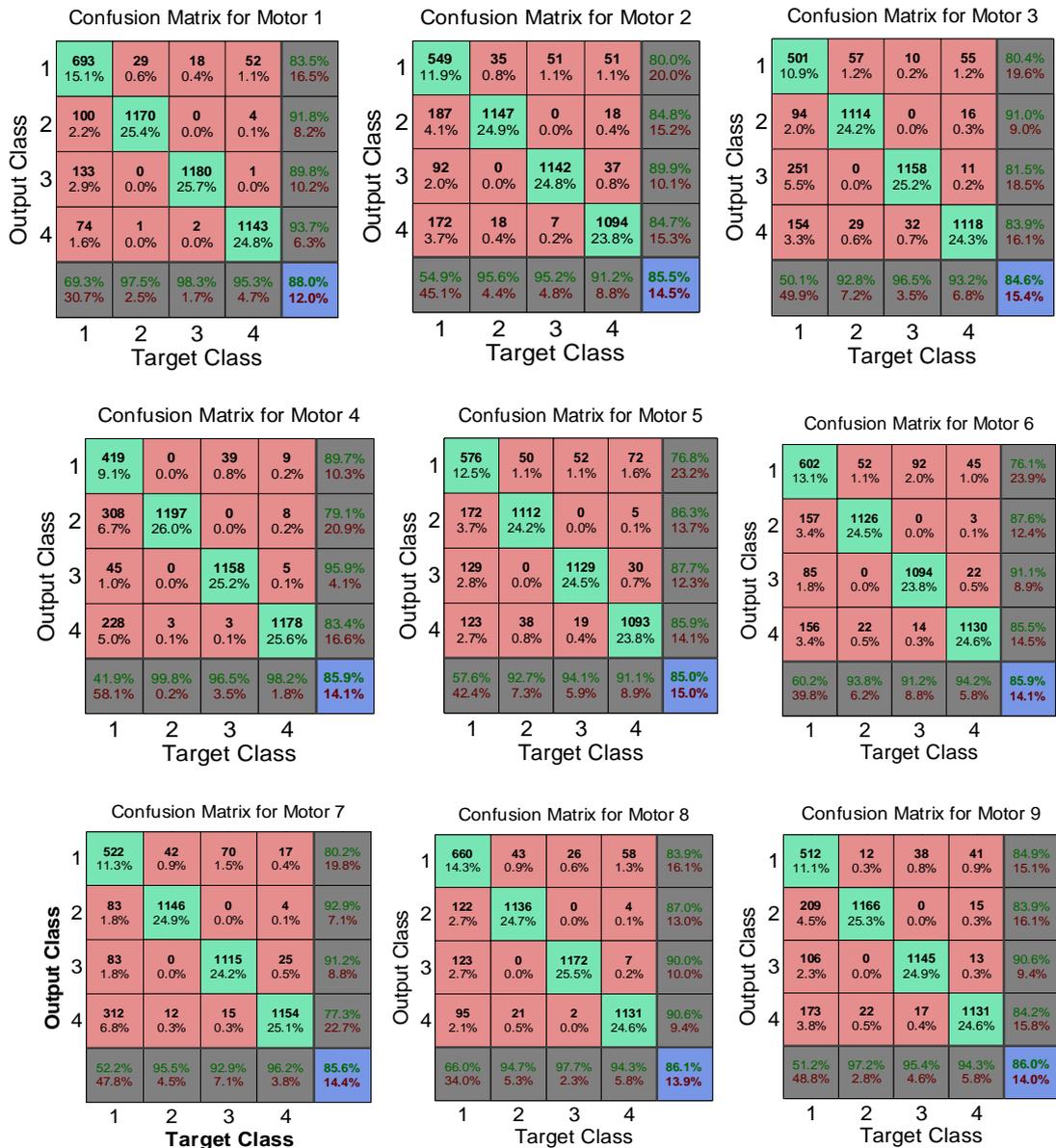


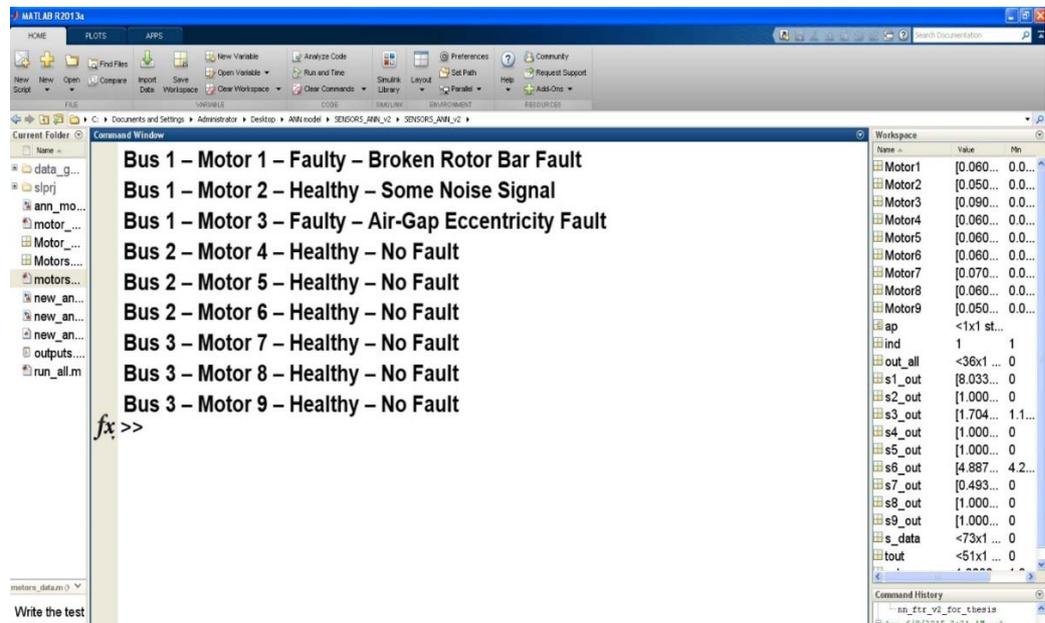
Figure 7.14: Confusion matrices of all motors using targeted and output classes

Table 7.5: Best performances for classification for case study 5

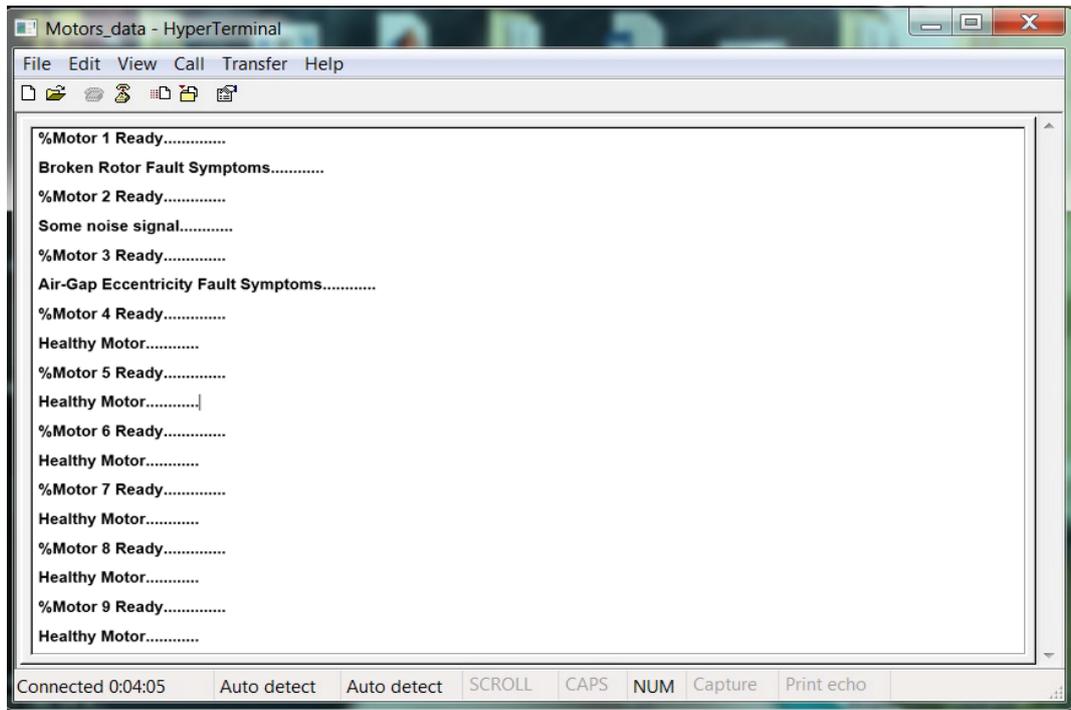
Motors	MSE Performance	No. of Epochs	Accuracy (%)	Classification error (%)
M1	0.0596	240	88.0	12.0
M2	0.0631	235	85.5	14.5
M3	0.0617	233	84.6	15.4
M4	0.0622	241	85.9	14.1
M5	0.0613	225	85.0	15.0
M6	0.0622	231	85.9	14.1
M7	0.0610	240	85.6	14.4
M8	0.0600	234	86.1	13.9
M9	0.0629	236	86.0	14.0

It can be seen that through Table 7.5, a satisfactory accuracy was achieved in fault detection in the feature vector, ranging from 84 to 88 percent. This reflects the performance efficiency of the ANN network, in reducing the level of uncertainty-management in decision-making in multiple-motor fault diagnosis, where multiple fault frequency signals make the detection process more complex.

With the comparison of different features from each motor, Figure 7.15 shows a comparison between the Matlab simulation and the experimental results for validity of the developed prototypes, and diagnosis of the condition of all motors within a network as follows.



(a)



(b)

Figure 7.15: Comparison of motor condition outputs  
 (a) Matlab simulation (b) Experimental results

## 7.4 Case Study 6: Dissimilar Faulty Motors with Multiple Faults in Different Power Buses

In this case study, an incident event consisting of two different faults in same sized of motors was investigated in different buses, in the presence of another fault and noise influences that were generated from other motors in the same bus. For that, two similar sized of motors (Motor 5 and Motor 9) were chosen from the buses 2 and 3, to observe the influence of faulty signals on other motors within the same or different buses, when the same type of fault signal is propagated over the network. To create the more complexity in the network signal, another high-power Motor 1 was chosen to have a high strength faulty signal, to create discrimination between the same type of fault signal, and identification of a suspected faulty source within the motor network, as shown in Figure 7.16. The network shown in the following figure was selected for the experiment. To observe the intensity of the fault signal, three sensing points were considered in this case study, to determine the fault-signal strength at different locations within network.

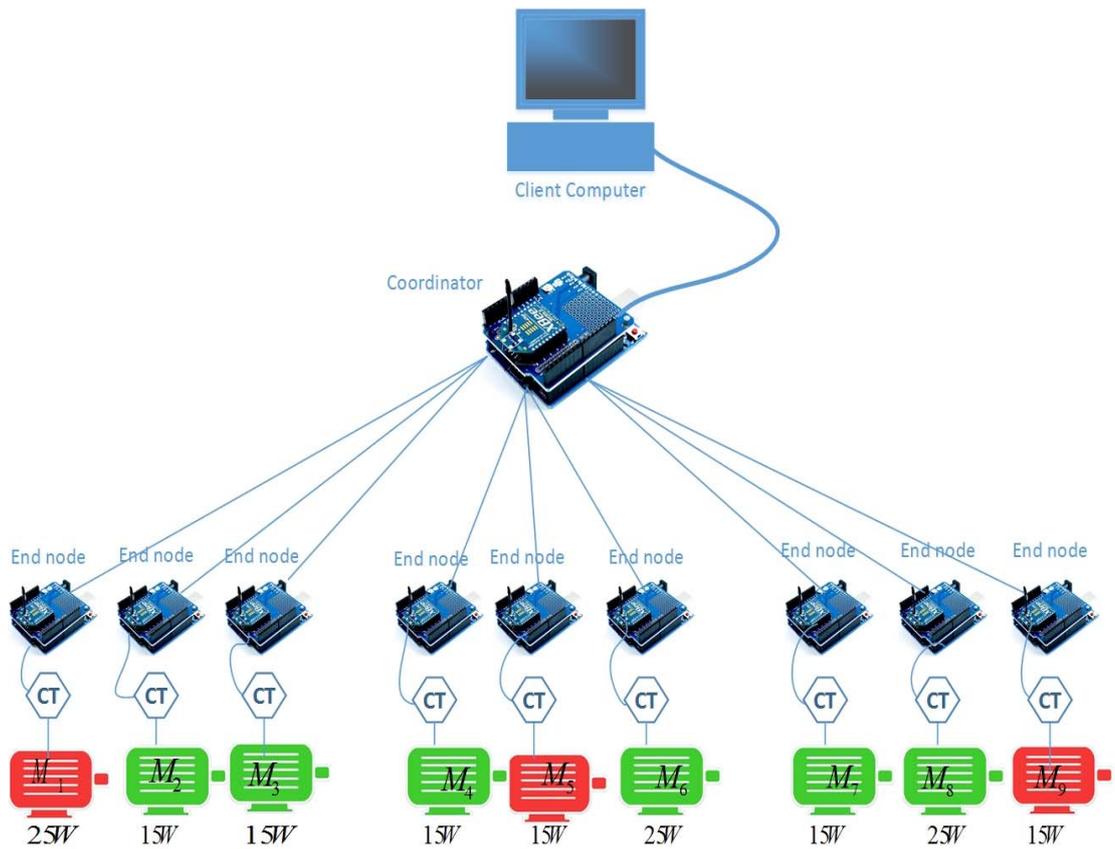
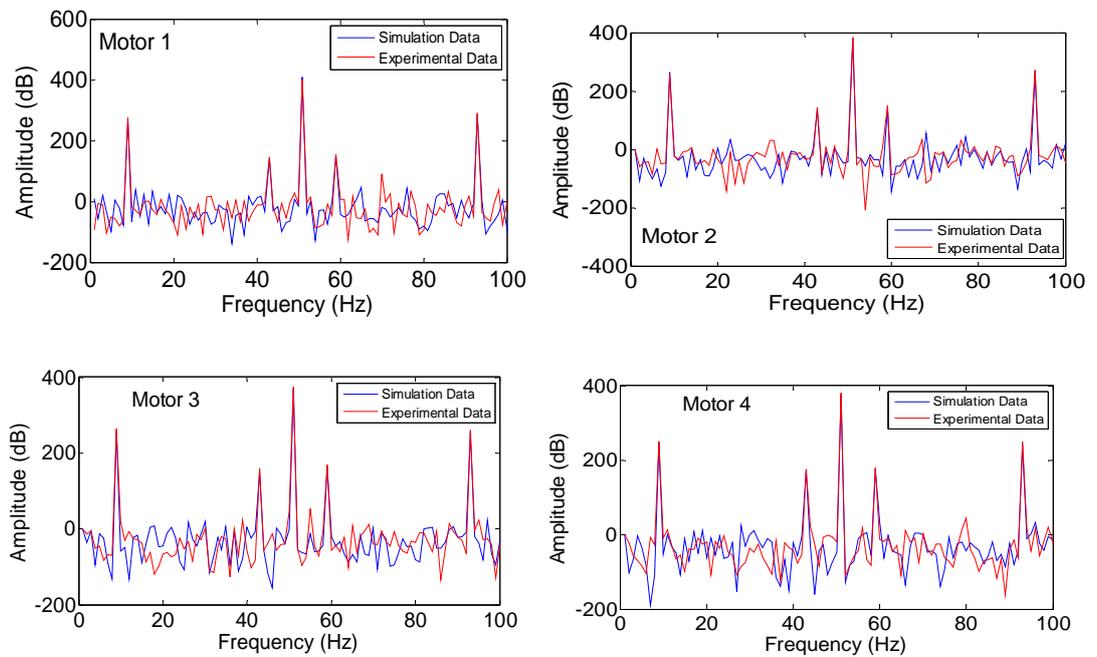


Figure 7.16: Multiple faulty motors in different buses with Faults

Figure (7.17) shows the mirroring sideband around the fundamental frequency among all the motors with varying amplitude values as follows:



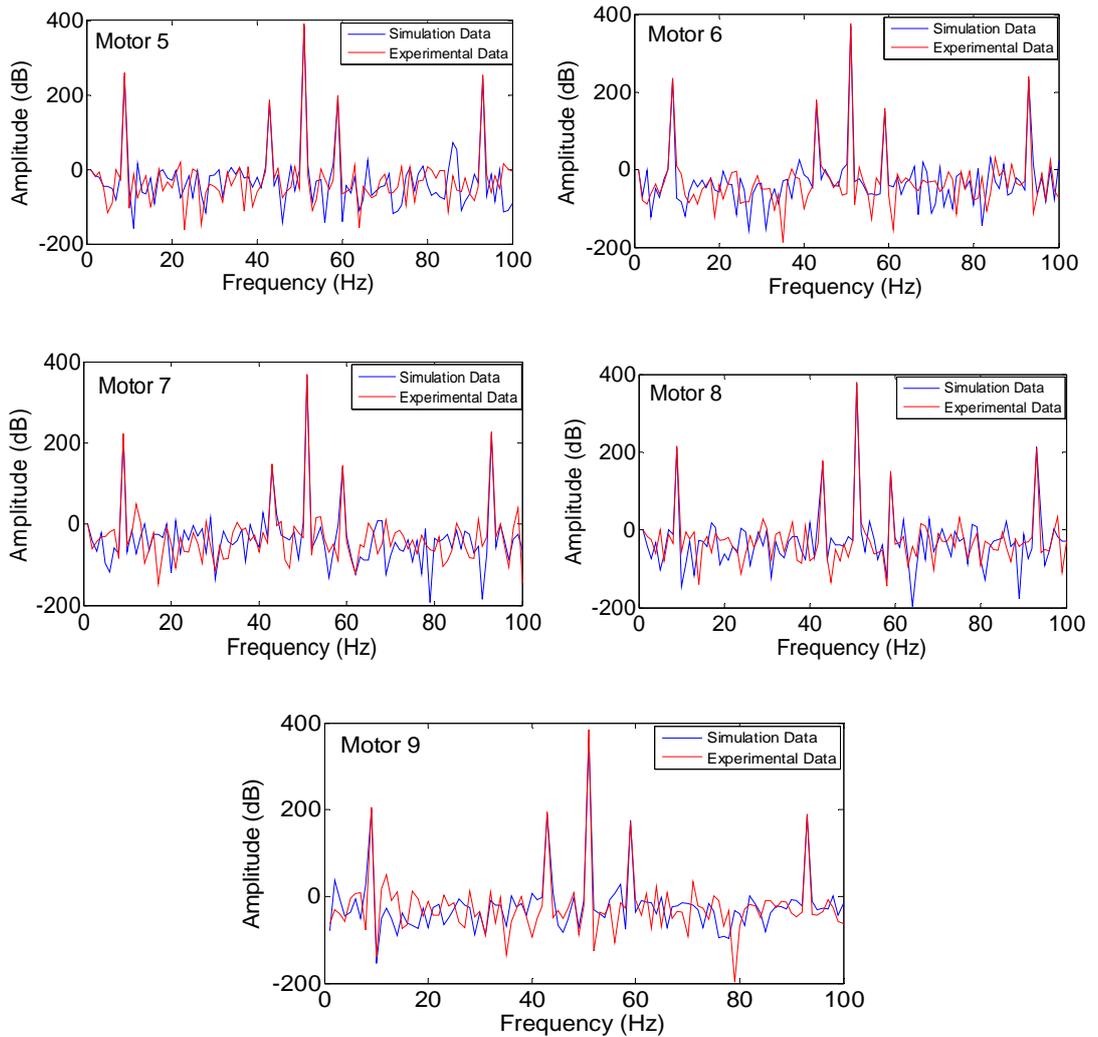


Figure 7.17: Current spectrum of all motors at full-load in BRB and ECE fault conditions in multiple motors

Figures 7.18-7.19 present an analysis of the electric current spectrum from all motors, and their amplitude values. It can be observed from Figure 7.19 that and eccentricity fault signal mirroring sideband appeared at the same frequency point but at a different amplitude rate, according to the size of motor and its distance from Motor 1. And BRB fault was propagated from Motor 5 and Motor 9 with different intensity of fault-signal. It can be observed that Motor 1 is under a low influence from the BRB fault due to its distance from other faulty motors. But the amplitude value is prominent, due to its own eccentricity fault.

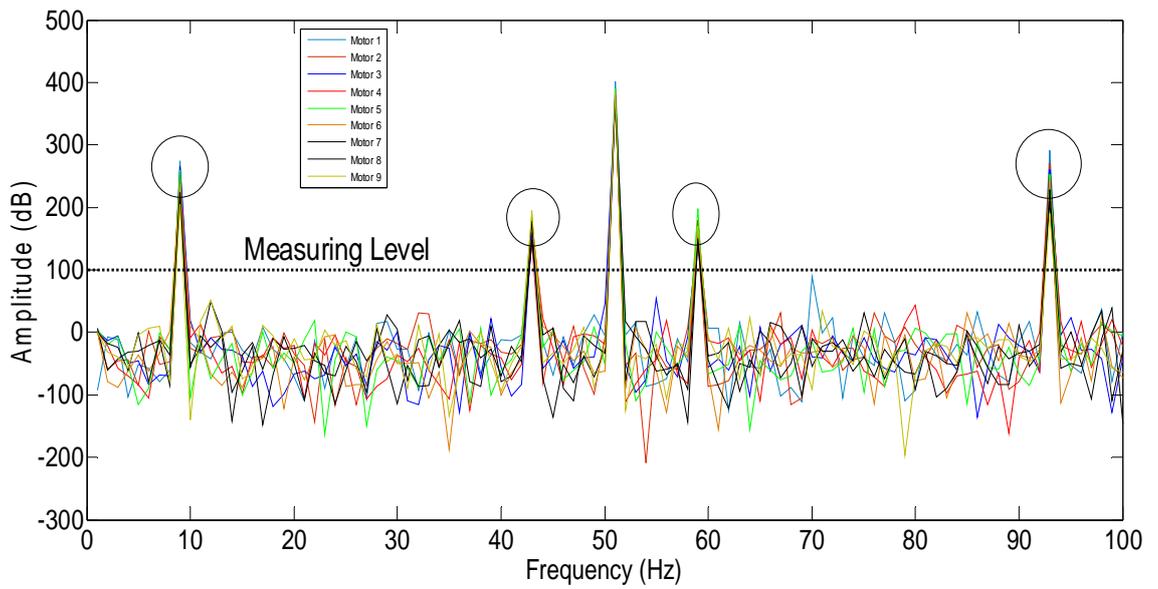


Figure 7.18: Multi-frequency fault propagation by motors 1, 5 and 9 at full-load, to observe the BRB and Eccentricity faults, influence on different motors according to their distance and size

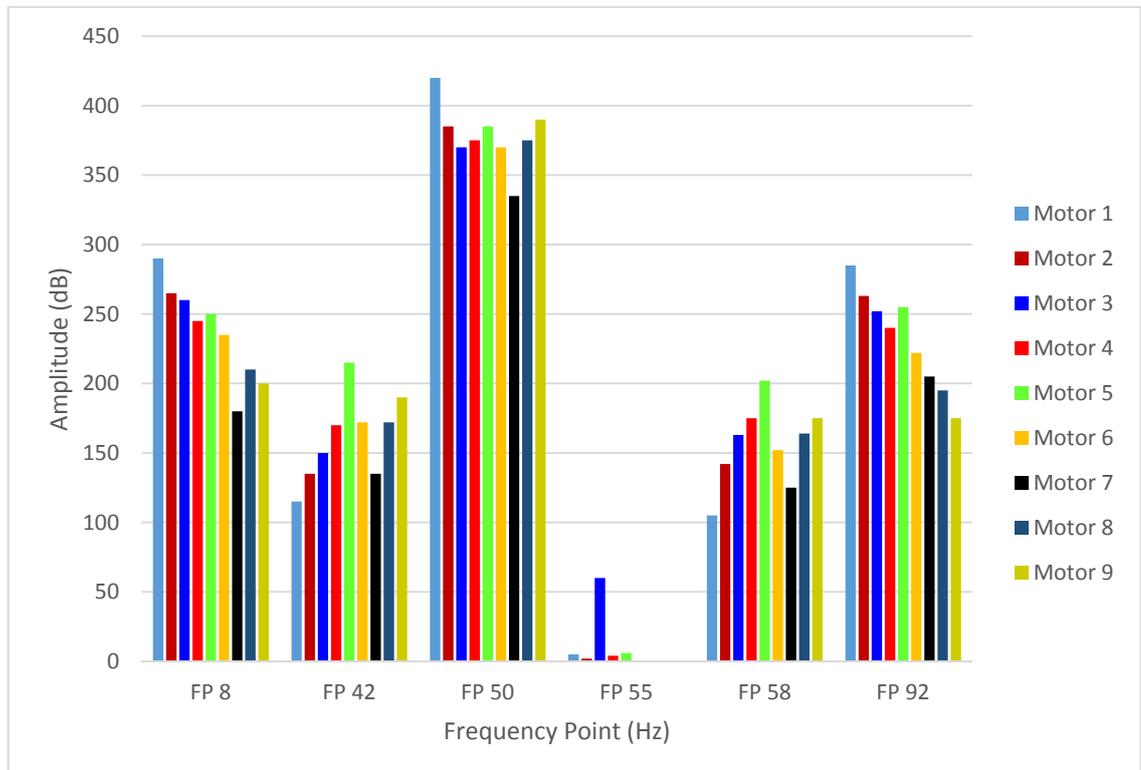
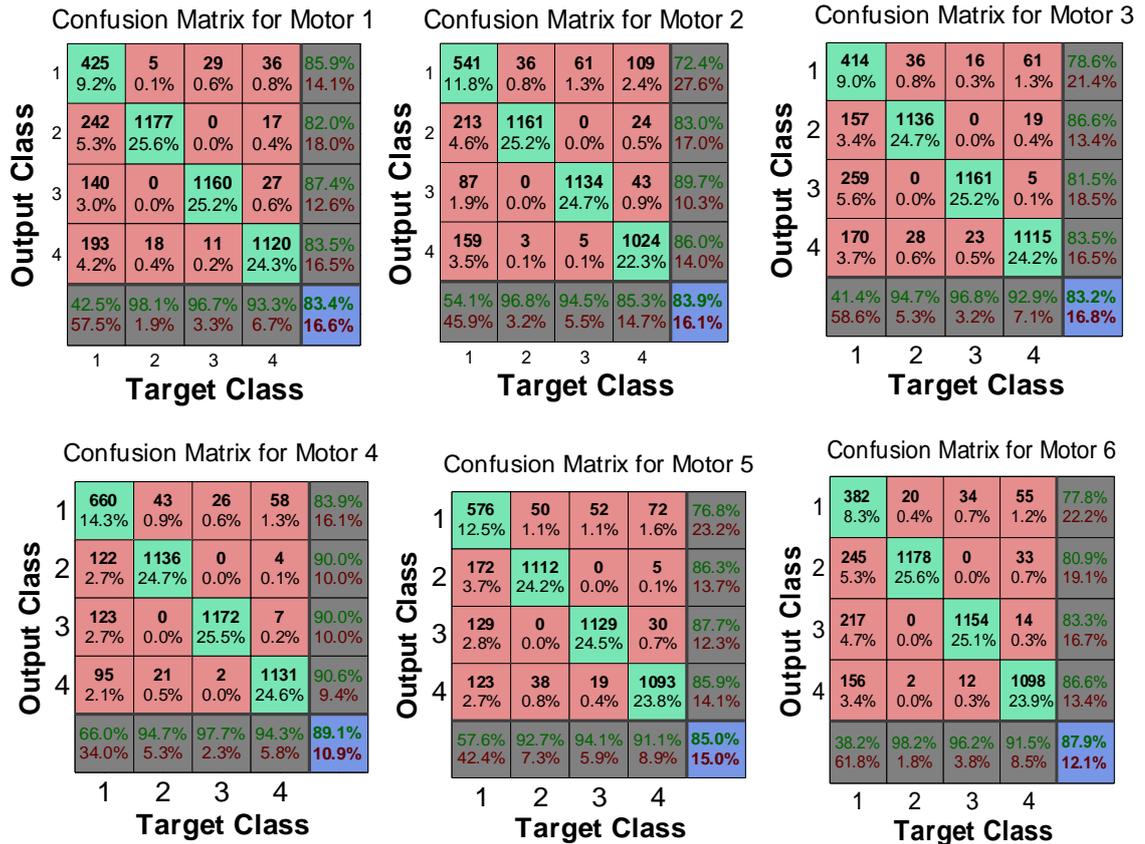


Figure 7.19: Analysis and observation chart of multi-frequency fault influence by Motors 1, 5 and Motor 3

Table 7.6: Learning data set for training process for multiple motors for case study 6

Motors	Features					
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
<b>Motor 1</b>	1012	0.22	[9;43;50;57; 60;91]	[280;140;419;168;2;285]	0.250	0.312
<b>Motor 2</b>	1200	0.07	[9;43;50;57; 91]	[265;135;380;142;263]	0.147	0.200
<b>Motor 3</b>	1220	0.06	[9;43;50;57; 60; 91]	[260;150;370;163;60;252]	0.142	0.191
<b>Motor 4</b>	1200	0.07	[9;43;50;57;60; 91]	[245;170;375;175;3;240]	0.147	0.200
<b>Motor 5</b>	1150	0.19	[9;43;50;57;60; 91]	[250;180;385;190;5;245]	0.210	0.301
<b>Motor 6</b>	1280	0.01	[9;43;50;57; 91]	[235;172;370;152;232]	0.150	0.215
<b>Motor 7</b>	1250	0.03	[9;43;50;57; 91]	[220;145;365;142;225]	0.164	0.210
<b>Motor 8</b>	1280	0.01	[9;43;50;57; 91]	[210;172;375;164;214]	0.150	0.190
<b>Motor 9</b>	1150	0.19	[9;43;50;57; 91]	[200;190;390;175;190]	0.201	0.299

Figure (7.16) presents the confusion matrices for the three process phases of training, testing and validation of each motors, respectively.



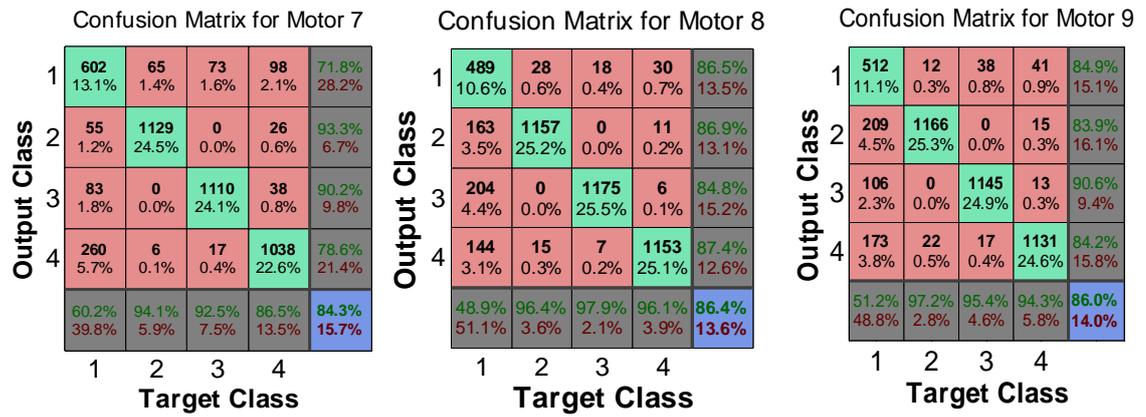


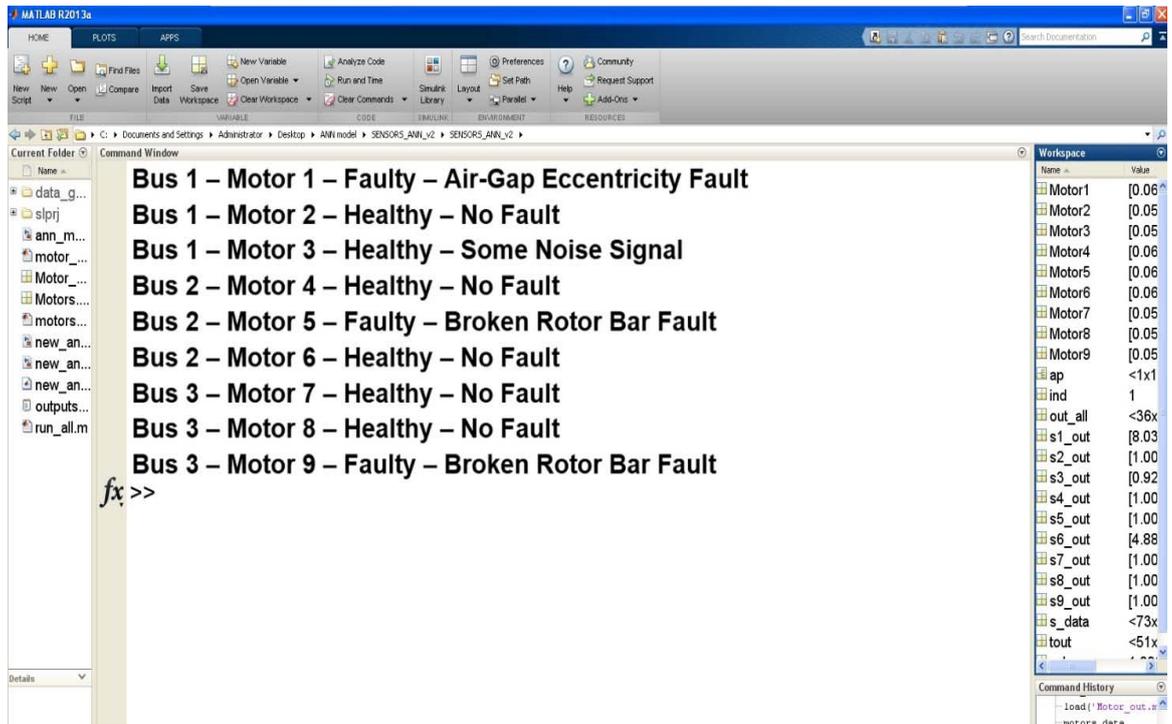
Figure 7.20: Confusion matrices of all motors using targeted and output classes

Table 7.7: Best performances for classification for case study 6

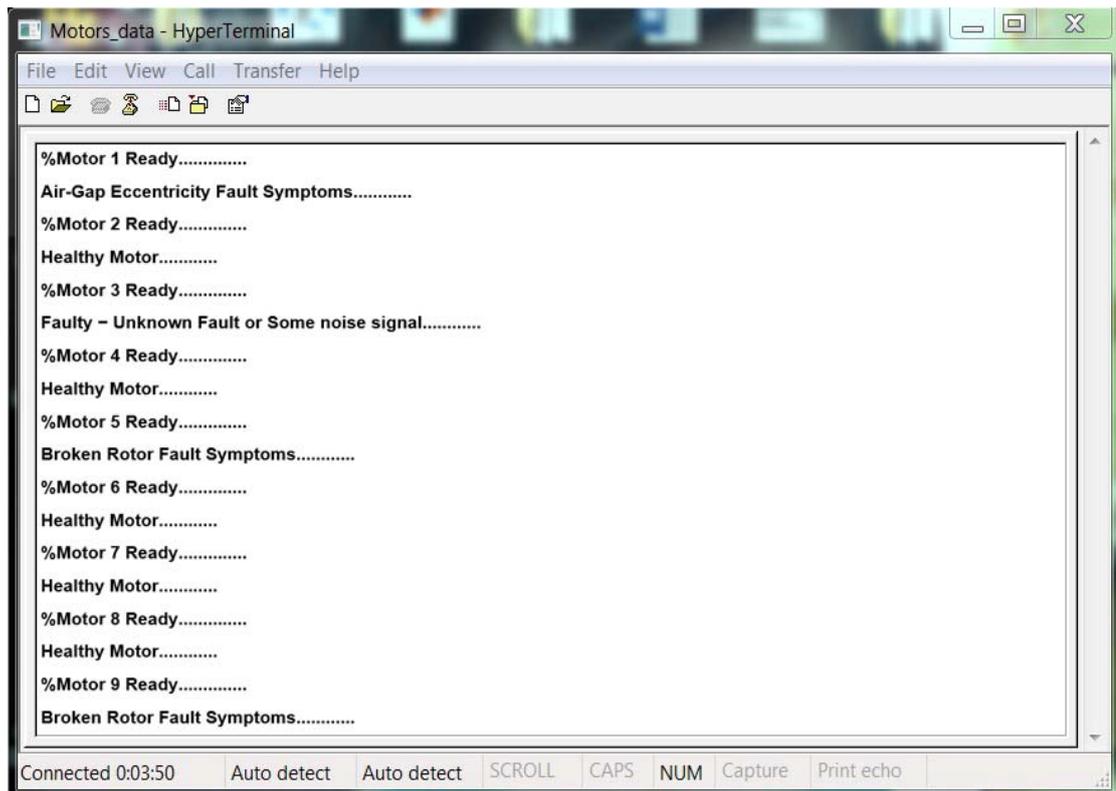
Motors	MSE Performance	No. of Epochs	Accuracy (%)	Classification error (%)
M1	0.0506	245	83.4	16.6
M2	0.0595	237	83.9	16.1
M3	0.0615	228	83.2	16.8
M4	0.0644	235	89.1	10.9
M5	0.0610	222	85.0	15.0
M6	0.0625	235	87.9	12.1
M7	0.0617	242	84.3	15.7
M8	0.0610	238	86.4	13.6
M9	0.0614	239	86.0	14.0

It can be seen, in Table 7.7, that a satisfactory accuracy was achieved in fault detection in the feature vector, ranging from 83 to 89 percent. This reflects the performance efficiency of the ANN network in the presence of multiple faults, when the same types of fault are mirror manifesting into other motor spectrums and creating doubt in identifying the source of fault.

In the comparison of different features from each motor, Figure 7.17 compare the Matlab simulation and the experiment results, for the validity of the developed prototypes and diagnosis of the condition of all motors within the network, as follows:



(a)



(b)

Figure 7.21: Comparison of motor condition outputs  
(a) Matlab simulation (b) Experimental results

## 7.6 Chapter Summary

A number of experimental and simulation case studies were investigated by evaluating the concept of propagation of fault signals from signals and groups of motors. As presented in this chapter, the components of electric current signals in all motors are closely manifested and influenced by neighbouring motors. Fault type, motor speed, strength of fault, other features and location as significant variables in diagnosis process. The results reflected the efficiency of the ANN network, in reducing the level of uncertainty in management of decision-making in multiple-motor fault diagnosis, where multiple-faults frequency signals and similar types of fault manifest symptoms that make the detection procedure more complex.

## CONCLUSIONS AND FUTURE RECOMMENDATIONS

### 8.1 Introduction

In this thesis, motivation and research objectives have been described, including the shortcomings in individual motor-fault diagnosis. A diagnostic approach is proposed, to utilise the propagation and manifestation of fault-frequency components over networks, and reduce fault diagnosis confusion in a distributed-motor environment. To prove the concept, an industrial distributed-motor network model was scaled down to generate simulation data, based on various case studies. To reduce uncertainty for the management of decision-making, an ANN model was developed, to analyse this in both single and multiple motor spectrum, where some motors are under the heavy influence of faulty signals. Finally, the concept was tested in an Arduino based test-bed environment.

In the remainder of this chapter, different contributions from multiple phases of this research are discussed. Finally, recommendations for feasible advancement of this work, possible future research and research shortcomings are presented.

### 8.2 Achievements of Research Tasks and their Future Scope

This research project has successfully demonstrated the importance in-network fault diagnosis as an efficient replacement for individual single-motor diagnoses. The project took advantage of multiple fields of study in industrial fault diagnosis to formulate the concept of fault signal propagation within power-line network, and manifesting faulty signals into healthy signals while they are travelling in a scaled-down distributed power-line system. A systematic method was employed to estimate the influence of a propagated fault signal (electric current signal) into main a power-line network. This showed how it changes the spectrum behaviour of motor features in electric currents within a power-line.

Analysis for estimation of attenuation factors in propagated signals was carried out to evaluate the possible paths of different signals over a network. This analysis has been presented to estimate the sources of faults on multiple path routes in power-line, and anticipated fault representation around the power-line network.

In the final part of this chapter, conclusions relating to this research as identified in the main objectives formulated in Chapter 1 are presented.

## **I. Multi-Motor Simulation Model**

A scale-down multi-motor network simulation model was developed for this study, to observe the behaviour of an overall power-line network containing mechanical faults in a given electrical motor within the network. This research has demonstrated that a Simulink model of multiple motors connected with main power-line can have a significant combined influence on an industrial system's overall operation. A fault- injection model block at the motor level was present to inject BRB and eccentricity faults into any motor. This model was then extended to the mechanism of the propagation of a fault from one motor to others. Two motor faults (BRB and ECE) were effectively modelled by imperfect corresponding representatives of the spectrum fault patterns described in previous literature. The model also considered heterogeneity in motor sizes for simulating the impact of higher powered motor's signals on a network. Propagation of these fault patterns, with the dynamic extension of electrical and mechanical faults, may stimulate interest in the diagnostic process. This simulation model facilitates the environment for testing sensor networks and data fusion approaches to facilitate better intelligence for fault identification and location.

## **II. Knowledge-based Solutions to Interpreting Fault Information**

This research has presented a supervised distributed ANN able to identify multiple fault types (BRB & ECE), as well as the location of fault events within an industrial motor network. Features have been extracted from electric current signals, based on different frequency components and associated amplitude values, with each fault type. A set of significant fault features, such as synchronised speed, rotor slip, amplitude value of each fault frequency component, and RMS and CF values were used to train these features using the BP algorithm. The simulated architecture worked well with

the significant feature sets selected, and accurate fault detection has been achieved.

Classification performance achieved satisfactory results for both healthy and defective conditions, including multiple fault-type identifications. In order to analyse the different characteristics of faulty signals in a power-line network and diagnose faults correctly, different significant aspects were considered carefully, including multiple motor sizes in a distributed network, noise level, measurement of attenuation factors and similarity between different fault symptoms. These factors have a significant influence on electric current signals and may contribute to creating confusion in fault diagnosis. Different case studies have been discussed, where high-power motors had a significant influence on low-power motors and created confusion in the same types of fault symptoms, over whether it was generating it themselves or manifesting from other motors over the network. The relationship of the features with these faults has also been discussed. To improve efficiency, various types of neural network architectures were tested for training, to adjust the weights of the hidden layer until the targeted output was achieved. It was proved through the results that accuracy in identifying motor condition depends on the neural network architecture being satisfactory.

### **III. Wireless Sensor Network Test-Bed Environment**

In order to prove the propagation of faulty signals within a power-line network, and detection of the fault type, a typical example of an experimental WSN motor network was modelled, based on Arduino Xbee modules and multiple-sized motors. This architecture evidently presented a good real-time representation of an industrial multi- motor network modelling environment, where all motors in parallel connected with the main power-line. Different output results clearly show the comparison between the simulation and sensor measurements, which show a close accuracy in capturing data in a real-time environment. Furthermore, development of the wireless node-level feature extraction technique has been demonstrated for data fusion, using MCSA at end node-level and decision-level fusion was implemented at the node coordinator for efficient fault diagnosis, to create more complex detection.

All the measurements have been taken from an Arduino speed sensor individually to each motor. However, due to the limitation of resources, time

and sensors, we took the signature samples sequentially, because motor speed does not show a high variation in steady state, until an abnormal behaviour of the motor interferes with the rotating pulse.

### **8.3 Recommendations and Future Research**

This thesis has presented a wireless node-level solution that deals with consistency and reliability issues in fault diagnosis in industrial multi-motor networks. This section suggests some promising directions for multiple fault diagnosis in a distributed multi-motor environment. Concerning future enhancement of fault diagnosis, further improvements could be explored to improve overall system operation.

- I. This research utilised the FFT technique to analyse the current spectrum to identify the behaviour of motor spectrums. The study could be expanded further by employing similar, or more advanced, signal processing techniques such as STFT, or wavelet transforms. This could be helpful in improving the reliability and accuracy of distributed-motor fault diagnosis
- II. In addition, further simulation work could enhance the understanding of significant frequencies related to induction motor faults, and improve the overall effectiveness of fault detection. BRB and ECE faults have been considered, as an instance of analysis of distributed-fault diagnosis. Some other types of motor faults such as inter-term faults, bearing faults and stator winding-related faults could be an extended work to prove the concept of signal propagation over multi-motor network
- III. The influence of faulty gearbox components in the current spectrum remains to be investigated
- IV. The influences of non-stationary processes on the motor stator current spectrum need to be further explored for distributed fault diagnosis purposes
- V. A typical example of a single-phase small motors test-bed environment has been utilised to formulate the concept of propagation of faulty signals within power-line networks. Other electrical drives, including 3-phase induction motors and transformers in simulation and practical prototypes, could be further investigated to measure the non-linear attenuation level and strength of faulty signals due to overload of saturated signals from transformers in higher frequency components. This improvement would

be very helpful in creating the more complexity in multi- frequency signals within power-line networks for the fault diagnosis process

- VI. A supervised ANN method has been effectively utilised to evaluate knowledge-based numerical features to overcome possible confusion in the identification of fault indices. This technique would not be suitable in those situations where analysis requires long processing time for large feature sets. In this situation, other decision-making techniques including fuzzy logic, neuro-fuzzy, weighted fusion, D-S evidence theory, Bayesian inference, Kalman filter and Genetic algorithm could be the potential methods for future expansion of distributed signature analysis and fault diagnosis
- VII. This research utilised the Arduino development board R3 with an ATMEGA328P low-power 8-bit CMOS microcontroller for data fusion at node level. It has 1K bytes EEPROM, and 32 KB of programmable system flash. Due to the limitations of memory and a low-bit microcontroller, this node would not be suitable for higher data analysis and transformation when overloading the saturated signal from high-power electrical drives such as transformers propagated over the power-line network. Hence, it is very important to have a significant focus on hardware node selection and development, appropriate to working in any industrial environment for distributed fault diagnosis
- VIII. A multi-hop mesh networking protocol may be needed to achieve more sophisticated sensor fusion procedures

In conclusion, distributed signature analysis can be seen as pioneering work in combining multidisciplinary research on industrial motor fault diagnosis. This study unlocks future research potential for a plethora of specific research focuses.

# References

- [1] J. Faiz and M. Ojaghi, "Different indexes for eccentricity faults diagnosis in three phase squirrel-cage induction motors: A review," *Mechatronics*, vol. 19, no. 1, pp. 2-13, 2009.
- [2] W. Deleroi, "Squirrel cage motor with broken bar in the rotor physical phenomena and their experimental assessment," in *International Conference on Emergency Medicine (ICEM 1982)*, Budapest, Hungary, 1982.
- [3] N. Elkasabgy, A. Eastham and G. Dawson, "Detection of broken bars in the cage rotor on an induction machine," *IEEE Transactions on Industry Applications*, vol. 28, no. 1, p. 165 –171, 1992.
- [4] J.H. Jung, J.J. Lee, and B.H. Kwon, "Online Diagnosis of Induction Motors Using MCSA," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 6, p. 1842 – 1852, 2006.
- [5] M. Benbouzid, "A review of induction motors signature analysis as a medium for faults detection," in *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (IECON '98)*, 1998.
- [6] G. Kliman, R. Koegl, J. Stein, R. Endicott and M. Madden, "Non-invasive detection of broken rotor bars in operating induction motors," *IEEE Transactions on Energy Conversion*, vol. 3, no. 4, p. 873 –879, 1988.
- [7] A. Bellini, F. Filippetti, G. Franceschini, C. Tassoni and G. Kliman, "Quantitative evaluation of induction motor broken bars by means of electrical signature analysis," *IEEE Transactions on Industry Applications*, vol. 37, p. 1248 –1255, Sep. 2001.
- [8] B. Gaydon, "Instrument to detect induction motor rotor circuit defects by speed fluctuation measurements," in *International Conference on Advances in Medical Signal and Information Processing*, 1979.

- [9] F. Filippetti, G. Franceschini, C. Tassoni and P. Vas, "AI techniques in induction machines diagnosis including the speed ripple effect," *IEEE Transactions on Industry Applications*, vol. 34, no. 1, pp. 98 - 108, Jan. 1998.
- [10] A. Sadoughi, M. Ebrahimi and E. Razaeei, "A new approach for induction motor broken bar diagnosis by using vibration spectrum," in *International Joint Conference SICE-ICASE*, 2006.
- [11] K. Cho, J. Lang and S. Umans, "Detection of broken rotor bars in induction motors Using state and parameter estimation," *IEEE Transactions on Industry Applications*, vol. 28, no. 3, pp. 702 - 709, 1992.
- [12] M. G. Melero, M. F. Cabanas, F. R. Faya, C. H. Rojas and J. Solares, "Electromagnetic torque harmonics for on-line inter turn short circuits detection in squirrel cage induction motors," in *8th European Conference on Power Electronics and Applications*, Switzerland, 1999.
- [13] D. Dorrell, W. Thomson and S. Roach, "Analysis of airgap flux current, and vibration signals as a function of the combination of static and dynamic airgap eccentricity in 3-phase induction motors," *IEEE Transactions on Industry Applications*, vol. 33, no. 1, p. 24 – 34, 1997.
- [14] S. Fruechtenicht, E. Pittius and H. Seinsch, "Diagnostic system for three-phase asynchronous machines," in *International Conference on Advances in Medical Signal and Information Processing*, 1989.
- [15] M. Benbouzid and H. Nejjari, "A simple fuzzy logic approach for induction motors stator condition monitoring," in *IEEE International Electric Machines and Drives Conference, IEMDC*, 2001.
- [16] B. Ayhan, M. Y. Chow and M. H. Song, "Multiple discriminant analysis and neural-network-based monolith and partition fault detection schemes for broken rotor bar in induction motors," *IEEE Transactions on Industrial Electronics*, vol. 4, no. 53, pp. 1298 - 1308, 2006.
- [17] L. Cristaldi, M. Lazzaroni, A. Monti, F. Ponci and F. Zocchi, "A genetic algorithm for fault identification in electrical drives: a comparison with neuro-fuzzy

- computation," in *21st IEEE Instrumentation and Measurement Technology Conference IMTC 04*, 2004.
- [18] F. Filippetti, G. Franceschini and T. Carla, "Neural networks aided on-line diagnostics of induction motor rotor faults," in *Conference Record - IAS Annual Meeting (IEEE Industry Applications Society)*, 1993.
- [19] M. Haji and H. Toliyat, "Pattern recognition - a technique for induction machines rotor broken bar detection," *IEEE Transactions on Energy Conversion*, vol. 16, no. 4, p. 312 – 317, 2001.
- [20] A. da Silva, R. Povinelli and N. Demerdash, "Induction machine broken bar and stator short-circuit fault diagnostics based on three-phase stator current envelopes," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 3, p. 1310 – 1318, 2008.
- [21] X. Xue, V. Sundararajan and L. Gonzalez-Argueta, "Sensor fusion for machine condition monitoring," in *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, California, USA, 2008.
- [22] J. Siau, A. Graff, W. L. Soong and N. Ertugrul, "Broken Bar Detection in Induction Motors Using Current and Flux Spectral Analysis," in *Australian Universities Power Engineering Conference*, Christchurch, New Zealand, 2003.
- [23] D. H. Hwang, Y. W. Youn, J. H. Sun and Y. H. Kim, "Robust Diagnosis Algorithm for Identifying Broken Rotor Bar Faults in Induction Motors," *Journal of Electrical Engineering & Technology*, vol. 9, no. 1, pp. 37-44, 2014.
- [24] K. M. S. Rama and K. SrikondaHari, "Neural network for the diagnosis of rotor broken faults of induction motors using MCSA," in *7th International Intelligent Systems and Control (ISCO)*, 2013.
- [25] Y. H. Kim, Y. W. Youn, D. H. Hwang, J. H. Sun and D. S. Kang, "High-Resolution Parameter Estimation Method to Identify Broken Rotor Bar Faults in Induction Motors," *IEEE Transactions on Industrial Electronics*, vol. 9, no. 60, pp. 4103-4117, 2013.

- [26] A. J. F. Gomez and T. J. Sobczyk, "Motor current signature analysis apply for external mechanical fault and cage asymmetry in induction motors," in *9th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED)*, 2013.
- [27] K. N. Gyftakis , D. K. Athanasopoulos and J. C. Kappatou, "Broken bar fault diagnosis in single and double cage induction motors fed by asymmetrical voltage supply," in *9th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED)*, 2013.
- [28] S. Bindu and V. V. Thomas, "Diagnoses of internal faults of three phase squirrel cage induction motor - A review," in *International Conference on Advances in Energy Conversion Technologies (ICAECT)*, 2014.
- [29] L. Eren and M. J. Devaney, "Effect of current resampling in Motor Current Signature Analysis," in *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2013.
- [30] M. Y. Kaikaa and M. Hadjami, "Effects of the simultaneous presence of static eccentricity and broken rotor bars on the stator current of induction machine," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2452-2463, 2014.
- [31] F. Gomez, A. Dziechciarz and T. .J. Sobczyk, "Mathematical modelling of eccentricities in induction machines by the monoharmonic model," in *9th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED)*, 2013.
- [32] C. Bruzzese, "Validation of sequence circuits useful for split-phase current signature analysis (SPCSA) and diagnosis of eccentric-rotor traction cage motors," in *3rd International Conference on Electric Power and Energy Conversion Systems (EPECS)*, 2013.
- [33] I. Ahmed, M. Ahamed, K. Imran, M. S. Khan and S. J. Akhtar, "Detection of eccentricities faults in machine using Frequency Spectrum technique," *International journal of computer and Electrical Engineering*, vol. 3, no. 1, pp. 1793-1863, 2011.

- [34] W. T. Thomson, D. Rankin and D. G. Dorrell, "On-line current monitoring to diagnose air gap eccentricity-an industrial case history of a large high-voltage three-phase induction motor," in *IEEE International Electric Machines and Drives Conference Record*, 1997.
- [35] S. Nandi and H. A. Toliyat, "Condition monitoring and fault diagnosis of electrical machines: A review," in *IEEE Thirty-Fourth IAS Annual Meeting Record of the Industry Applications Conference*, 1999.
- [36] M. Djeddi, P. Granjon and B. Leprettre, "Bearing Fault Diagnosis in Induction Machine Based on Current Analysis Using High-Resolution Technique," in *IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives, (SDEMPED)*, 2007.
- [37] M. Eftekhari, M. Moallem, M. Sadri and Shojaei, "Review of induction motor testing and monitoring methods for inter-turn stator winding faults," in *21st Iranian Conference Electrical Engineering (ICEE)*, 2013.
- [38] S. Bindu and V. V. Thomas, "Diagnoses of internal faults of three phase squirrel cage induction motor - A review," in *2014 International Conference on Advances in Energy Conversion Technologies (ICAECT)*, 2014.
- [39] I. Ahmed and M. Ahmed, "Comparison of stator current, axial leakage flux and instantaneous power to detect broken rotor bar faults in induction machines," in *Australasian Universities Power Engineering Conference (AUPEC '08)*, 2008.
- [40] L. Hou and N. W. Bergmann,, "Novel Industrial Wireless Sensor Networks for Machine Condition Monitoring and Fault Diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 10, pp. 2787-2798, Oct. 2012.
- [41] S. Fruechtenicht, E. Pittius and H. Seinsch, "Diagnostic system for three-phase asynchronous machines," in *International Conference on Advances in Medical Signal and Information Processing*, 1989.
- [42] M. Benbouzid and H. Nejjari, "A simple fuzzy logic approach for induction motors stator condition monitoring," in *IEEE International Electric Machines and Drives Conference, IEMDC*, 2001.

- [43] B. Ayhan, M. Y. Chow and M. H. Song, "Multiple discriminant analysis and neural-network-based monolith and partition fault-detection schemes for broken rotor bar in induction motors," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 4, p. 1298 –1308, June 2006.
- [44] L. Cristaldi, M. Lazzaroni, A.Monti, F. Ponci and F. Zocchi, "A genetic algorithm for fault identification in electrical drives: a comparison with neuro-fuzzy computation," in *21st IEEE Instrumentation and Measurement Technology Conference (IMTC 04)*, May 2004.
- [45] F. Filippetti, G. Franceschini and T. Carla, "Neural networks aided online diagnostics of induction motor rotor faults," in *Conference Record - IEEE Industry Applications Society (IAS) Annual Meeting*, 1993.
- [46] E. Shahriar, M. R. Ahsan, T. Chong and UiPil, "Fault diagnosis of induction motors utilizing local binary pattern-based texture analysis," *EURASIP Journal on Image and Video Processing*, vol. 1, pp. 1-11, 2013.
- [47] A. Gheitasi, A. A. Anbukey and T. T. Lie, "Distributed Intelligence for Diagnosis of Industrial Motors Internal events," in *20th International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, 2012.
- [48] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess and B. Wu, *Signal Processing and Database Management Systems*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2007.
- [49] J. H. Jung, J. J. Lee and B. H. Kwon, "Online Diagnosis of Induction Motors Using MCSA," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 6, pp. 1842 - 1852, Dec. 2006.
- [50] D. Dorrell, W. Thomson and S. Roach, "Analysis of airgap flux, current, and vibration signals as a function of the combination of static and dynamic airgap eccentricity in 3-phase induction motors," *IEEE Transactions on Industry Applications*, vol. 33, no. 1, pp. 24 - 34, Jan 1997.
- [51] S. Nandi and H. A. Toliyat, "Condition monitoring and fault diagnosis of electrical machines - A review," in *Thirty-Fourth IAS Annual Meeting on Industry Applications Conference*, 1999.

- [52] N. Tandon and A. Choudhury, "A review of vibration and acoustic measurement for the detection of defects in rolling element bearings," *Tribology International*, vol. 32, no. 8, pp. 469-480, 1999.
- [53] Y. H. Kim, Y. W. Youn, D. H. Hwang, J. H. Sun and D. S. Kang, "High-Resolution Parameter Estimation Method to Identify Broken Rotor Bar Faults in Induction Motors," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4103-4117, 2013.
- [54] M. A. Kadir and T. Hiyama, "Fault classification based artificial method of induction motors bearing," *International Journal of innovative computing, information and control*, vol. 7, no. 9, pp. 5477-5494, 2011.
- [55] P. Sanchez, M. Riera-Guasp, J. A. D. Antonino, J. F. Roger , J. C. Perez and R. P. Puche, "Instantaneous Frequency of the Left Sideband Harmonic During the Start-Up Transient: A New Method for Diagnosis of Broken Bars," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 11, pp. 4557-4570, Nov. 2009.
- [56] J. A. D. Antonino, M. G. Riera, J. R. Folch and P. Molina , "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. 9, pp. 990-996, July-Aug. 2006.
- [57] M. S. Pineda, P. Jover, M. Riera, A. Arkkio and J. R. Folch, "DWT analysis of numerical and experimental data for the diagnosis of dynamic eccentricities in induction motors," *Mechanical Systems and Signal Processing*, vol. 21, pp. 2575-2589, Feb. 2007.
- [58] J. A. Daviu, P. J. Rodriguez, M. R. Guasp, M. P. Sánchez and A. Arkkio, "Detection of Combined Faults in Induction Machines with Stator Parallel Branches through the DWT of the startup current," *Mechanical Systems and Signal Processing, Elsevier*, vol. 23, no. 7, pp. 2336-2351, Oct. 2009.
- [59] R. Guasp, J. A. D. Antonino, R. Folch and P. M. P. Molina , "The Use of the Wavelet Approximation Signal as a Tool for the Diagnosis of Rotor Bar Failures," *IEEE Transactions on Industry Applications*, vol. 44, no. 3, pp. 716-726, May-june 2008.

- [60] R. Guasp, J. A. D. Antonino, M. S. Pineda, R. P. Puche and J. C. Perez, "A General Approach for the Transient Detection of Slip-Dependent Fault Components Based on the Discrete Wavelet Transform," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4167-4180, Dec. 2008.
- [61] H. Yang, J. Mathew and L. Ma, "Vibration feature extraction techniques for fault diagnosis of rotating machinery: A literature survey," in *Proceedings of the 10th Asia-Pacific Vibration Conference*, Gold Coast, Australia, 2003.
- [62] W. J. Wang and P. D. McFadden, "Application of wavelets to gearbox vibration signals for fault detection," *Journal of Sound and Vibration*, vol. 192, pp. 927-939, 1996.
- [63] E. M. Liggins, L. D. Hall and J. Llinas, *Handbook of multisensor data fusion: Theory and practice (Second Edition)*, CRC press, 2008.
- [64] G. Vachtsevanos, F. Lewis, M. Roemer and B. Wu, *Intelligent fault diagnosis and prognosis for engineering systems*, Hoboken, USA: John Wiley & Sons, Inc., 2007.
- [65] L. Hou and W. N. Bergmann, "Induction motor condition monitoring using industrial wireless sensor networks," in *2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Dec. 2010.
- [66] N. Mehala and R. Dahiya, "A comparative study of FFT, STFT and WAVELET techniques for induction machine fault diagnostic analysis," in *International Conference on Computational Intelligence, Man Machine Systems and Cybernetics*, Cairo, Egypt, 2008.
- [67] J. D. Antonino, G. Riera, J. L. Pons, P. Jongbin, L. B. Sang, Y. Jiyeon and C. Kral, "Detection of Broken Outer-Cage Bars for Double-Cage Induction Motors Under the Startup Transient," *IEEE Transactions on Industry Applications*, vol. 48, no. 5, pp. 1539-1548, Sept-Oct. 2012.
- [68] N. Tandon and A. Choudhury, "A review of vibration and acoustic measurement for the detection of defects in rolling element bearings," *Tribology International*, vol. 32, no. 8, pp. 469-480, 1999.

- [69] M. Iorgulescu and R. Beloiu, "Study of DC motor diagnosis based on the vibration spectrum and current analysis," in *2012 International Conference on Applied and Theoretical Electricity (ICATE)*, Oct. 2012.
- [70] Y. Gritli, D. Tommaso, R. Miceli, F. Filippetti and C. Rossi, "Vibration signature analysis for rotor broken bar diagnosis in double cage induction motor drives," in *2013 Fourth International Conference on Power Engineering, Energy and Electrical Drives (POWERENG)*, May 2013.
- [71] Y. Gritli, D. Tommaso, R. Miceli and C. Rossi, "Diagnosis of mechanical unbalance for double cage induction motor load in time-varying conditions based on motor vibration signature analysis," in *2013 International on Renewable Energy Research and Applications (ICRERA)*, Oct. 2013.
- [72] M. Tsytkin, "Induction motor condition monitoring: Vibration analysis technique - a twice line frequency component as a diagnostic tool," in *IEEE International Electric Machines & Drives Conference (IEMDC)*, May 2013.
- [73] P.V. Raj, K. Natarajan and G. T. Girikumar, "Induction motor fault detection and diagnosis by vibration analysis using MEMS accelerometer," in *2013 International Conference on Emerging Trends in Communication, Control, Signal Processing & Computing Applications (C2SPCA)*, Oct. 2013.
- [74] L. Hong and S. J. Dhupia, "A time-domain fault detection method based on an electrical machine stator current measurement for planetary gear-sets," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2013.
- [75] K. R. Patel, S. Agrawal and N. C. Joshi, "Induction motor bearing fault identification using vibration measurement," in *2012 Students Conference on Engineering and Systems (SCES)*, March 2012.
- [76] L. Jiang, X. Fu, J. Cui and Z. Li, "Fault detection of rolling element bearing based on principal component analysis," in *24th Chinese Control and Decision Conference (CCDC)*, May 2012.

- [77] M. Benbouzid, "A review of induction motors signature analysis as a medium for faults detection," in *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society*, 1998.
- [78] S. Krishna, R. S. Merugu and K. SrikondaHari, "Neural network for the diagnosis of rotor broken faults of induction motors using MCSA," in *7th International Conference on Intelligent Systems and Control (ISCO)*, Jan. 2013.
- [79] S. Krishna, R. S. Merugu and S. K. Ravi, "Fault diagnosis of induction motor using Motor Current Signature Analysis," in *2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT)*, March 2013.
- [80] S. Das, P. Purkait and S. Chakravorti, "Separating induction Motor Current Signature for stator winding faults from that due to supply voltage unbalances," in *1st International Conference on Power and Energy in NERIST (ICPEN)*, Dec. 2012.
- [81] L. Hong and J. S. Dhupia, "A time-domain fault detection method based on an electrical machine stator current measurement for planetary gear-sets," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2013.
- [82] A. J. Fernandez Gomez and T. J. Sobczyk, "Motor current signature analysis apply for external mechanical fault and cage asymmetry in induction motors," in *9th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED)*, Aug. 2013.
- [83] A. Gheitasi, A. A. Anbukey and T. T. Lie, "Distributed Intelligence for Diagnosis of Industrial Motors Internal events," in *20th International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, 2012.
- [84] P. M. Frank and X. Ding, "Survey of robust residual generation and evaluation methods in observer-based fault detection systems," *Journal of Process Control*, vol. 7, no. 6, pp. 403-424, 1997.
- [85] M. H. Hafizi and A. Izadian, "Model-based fault diagnosis of a DC-DC boost converters using hidden Markov model," in *IEEE International Conference on Electro/Information Technology (EIT)*, May 2013.

- [86] C. Yu, W. Wang, C. Fujun and Y. Fuwen, "Model-based fault diagnosis methods: A survey," in *27th Chinese Control Conference*, July 2008.
- [87] H. Su and K. T. Chong, "Induction machine condition monitoring using neural network modelling," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 241- 249, Feb. 2007.
- [88] S. S. Refaat, H. Abu-Rub, M. S. Saad, A. Zahab and E. M. Iqbal, "ANN-based for detection, diagnosis the bearing fault for three phase induction motors using current signal," in *IEEE International Conference on Industrial Technology (ICIT)*, Feb. 2013.
- [89] T. Boukra, A. Lebaroud and G. Clerc, "Neural-Network approaches for classification of induction machine faults using optimal time-frequency representations," in *10th International Conference on Environment and Electrical Engineering (EEEIC)*, May 2011.
- [90] Y. Nyanteh, C. Edrington, S. Srivastava and D. Cartes, "Application of Artificial Intelligence to Real-Time Fault Detection in Permanent-Magnet Synchronous Machines," *IEEE Transactions on Industry Applications*, vol. 49, no. 3, pp. 1205-1214, 2013.
- [91] T. Eldin, H. R. Emara, A. Zahab and S. S. Refaat, "Monitoring and Diagnosis of External Faults in Three Phase Induction Motors Using Artificial Neural Network," *IEEE Power Engineering Society General Meeting*, pp. 24-28, June 2007.
- [92] H. Arabaci and O. Bilgin, "Diagnosis of broken rotor bar faults by using frequency spectrum of stator current envelope," in *20th International Conference on Electrical Machines (ICEM)*, Sep. 2012.
- [93] A. Drira and N. Derbel, "Classification of rotor fault in induction machine using Artificial Neural Networks," in *8th International Multi-Conference on Systems, Signals and Devices (SSD)*, March 2011.
- [94] S. Hamdani, O. Touhami, R. Ibtouen and M. Fadel, "Neural network technique for induction motor rotor faults classification-dynamic eccentricity and broken bar

- faults," in *IEEE International Symposium on Diagnostics for Electric Machines Power Electronics & Drives (SDEMPED)*, 2011.
- [95] M. A. Kadir, S. Sharifah, A. Wahab, M. Helmy and M. I. Ghazali, "Using artificial neural network to monitor and predict induction motor failure," in *International engineering convention*, Jeddah, Saudi Arabia, 2007.
- [96] Y. M. Yang, G. Zhe-xue and X. Yong-cheng, "Fault Diagnosis of Complex Systems Based on Multi-sensor and Multi-domain Knowledge Information Fusion," in *IEEE International Conference on Networking, Sensing and Control, (ICNSC 08)*, April 2008.
- [97] N. Wang, Y. Liang and H. Zhu, "Research on Fault Diagnosis Based on D-S Evidential Reasoning," in *2011 International Conference on Intelligent Computation Technology and Automation (ICICTA)*, March 2011.
- [98] J. Seshadrinath, B. Singh and B. K. Panigrahi, "Incipient Interturn Fault Diagnosis in Induction Machines Using an Analytic Wavelet-Based Optimized Bayesian Inference," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 990-1001, May 2014 .
- [99] S. Kumar, J. Prakash and S. S. Kumar, "Detection of Broken Rotor Bars in Induction Motor Using Derivative Free Kalman Filters," in *International Conference on Process Automation, Control and Computing (PACC)*, July 2011.
- [100] V. P. Mini, S. Setty and S. Ushakumari, "Fault detection and diagnosis of an induction motor using fuzzy logic," in *IEEE Region 8 International Conference on Computational Technologies in Electrical and Electronics Engineering (SIBIRCON)*, July 2010.
- [101] M. Chun-yu, Z. Guang-Wen and X. Yu-kun, "Research of Rotating Machinery Fault Diagnosis Based on Fuzzy Neural Network and Information Fusion," in *2014 International Symposium on Computer, Consumer and Control (IS3C)*, June 2014.
- [102] J. Seshadrinath, B. Singh and B. K. Panigrahi, "Incipient Interturn Fault Diagnosis in Induction Machines Using an Analytic FFT-Based Optimized Bayesian Inference," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 990-1001, Oct. 2014.

- [103] X. Zhang, P. Zhang and C. Liu, "Fault diagnosis of rotating machinery based on evidence theory of evidence entropy," in *9th International Conference on Electronic Measurement & Instruments, (ICEMI '09)*, Aug. 2009.
- [104] A. Gheitasi, A. Al-Anbuky and T. T. Lie, "Impact of propagation of fault signals on industrial diagnosis using current signature analysis," in *Universities Power Engineering Conference (AUPEC)*, Sept. 2011.
- [105] L. Changning, Q. Miao and M. Pecht, "The research on multipath propagation of vibration signal in mechanical fault system," in *Prognostics and System Health Management Conference (PHM-Shenzhen)*, May 2011.
- [106] R. F. Mustapa, M. S. Serwan, N. Hamzah and Z. Zakaria, "Hypothesis testing for fault analysis and the propagation of faulted voltage through transformer connections," in *2011 IEEE Student Conference on Research and Development (SCORED)*, Dec. 2011.
- [107] A. Gheitasi and A. A. Anbuky, "Industrial Fault Signals Propagation and Current Signature Analysis," *Journal of Energy and Power Engineering*, vol. 7, pp. 361-369, 2013.
- [108] S. Korkua, H. Jain, L. Wei-Jen and K. Chiman, "Wireless health monitoring system for vibration detection of induction motors," in *IEEE Industrial and Commercial Power Systems Technical Conference (I&CPS)*, May 2010.
- [109] D. Flammini, E. Marioli, Sisinni and A. Taroni, "Design and implementation of a wireless fieldbus for plastic machineries," *IEEE Transaction on Industry Electronics*, vol. 56, no. 3, p. 747–755, Mar. 2009.
- [110] P. Tiwari, Ballal and F. L. Lewis, "Energy-efficient wireless sensor network design and implementation for condition-based maintenance," *ACM Trans. Sensor Network*, vol. 3, no. 1, pp. 23-28, Mar. 2007.
- [111] F. Salvadori, M. de Campos, P. S. Sausen, R. F. de Camargo, C. Gehrke, C. Rech, M. A. Spohn and A. C. Oliveira, "Monitoring in industrial systems using wireless sensor network with dynamic power management," *IEEE Trans. Instrumental Measurements*, vol. 58, no. 9, p. 3104–3111, Sep. 2009.

- [112] L. Bin and V. C. Gungor, "Online and remote motor energy monitoring and fault diagnostics using wireless sensor networks," *IEEE Trans on Industrial Electron*, vol. 56, no. 11, pp. 4651-4659, Nov. 2009.
- [113] S. Korkua, H. Jain, W. J. Lee and C. Kwan, "Wireless health monitoring system for vibration detection of induction motors," in *Proc. IEEEI&CPS*, 2010.
- [114] L. Hou and N. Bergmann, "Novel industrial wireless sensor networks for machine condition monitoring and fault diagnosis," *IEEE Trans Instrumental. Measuements*, vol. 61, no. 10, p. 2787–2798, Oct. 2012.
- [115] L. Hou and N. Bergmann, "Induction motor fault diagnosis using industrial wireless sensor networks and Dempster-Shafer classifier fusion," in *37th Annual Conference of IEEE Industrial Electron Socirty*, Nov. 2011.
- [116] S. Altaf, A. Al-Anbuky and H. Gholam-Hosseini, "Fault signal propagation in a network of distributed motors," in *IEEE 8th International on Power Engineering and Optimization Conference (PEOCO)*, March 2014 .
- [117] T. Kataoka, H. Uchida, S. Nishhta, T. Kai and T. Fuaabashi, "A method for aggregation of a group of induction motor loads," in *POWERCON*, Perth, Australia, 2000.
- [118] J. Mora-Flòrez, J. Meléndez and G. Carrillo-Caicedo, "Comparison of impedance based fault location methods for power distribution systems," *Electric Power Systems Research*, vol. 78, no. 4, pp. 657-666, April 2008.
- [119] Z. Wang, C. S. Chang and Y. Zhang, "A feature based frequency domain analysis algorithm for fault detection of induction motors," in *6th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, June 2011.
- [120] Z. Jun, D. L. Lubkeman and A. A. Girgis, "Automated fault location and diagnosis on electric power distribution feeders," *IEEE Transactions on Power Delivery*, vol. 12, no. 2, pp. 801-809, Apr 1997.
- [121] M. E. H. Benbouzid, "A Review of Induction Motor Signature Analysis as a Medium for Faults Detection," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, pp. 984,993, October 2000.

- [122] K. S. Low, W. N. N. Win and J. Meng Er, "Wireless Sensor Networks for Industrial Environments," in *International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, Nov. 2005.
- [123] S. Grubic, J. M. Aller, L. Bin and T. G. Habetler, "A Survey on Testing and Monitoring Methods for Stator Insulation Systems of Low-Voltage Induction Machines Focusing on Turn Insulation Problems," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4127-4136, Dec. 2008.
- [124] J. P. Amaro, F.J.T.E. Ferreira, R. Cortesao, N. Vinagre and R. P. Bras, "Low cost wireless sensor network for in-field operation monitoring of induction motors," in *IEEE International Conference on Industrial Technology (ICIT)*, March 2010.
- [125] K.K. Tan, S.N. Huang, Y. Zhang and T.H. Lee, "Distributed fault detection in industrial system based on sensor wireless network," *Computer Standards & Interfaces*, vol. 31, no. 3, pp. 573-578, March 2009.
- [126] F. Nozari, M. D. Kankam and W. W. Price, "Aggregation of Induction Motors for Transient Stability Load Modelling," *IEEE Transactions on Power Systems*, vol. 2, no. 4, pp. 1096-1103, Nov. 1987.
- [127] K. W. Louie and P. Wilson, "Aggregation of Induction Motors Based on their Specifications," in *Canadian Conference on Electrical and Computer Engineering, (CCECE '06)*, May 2006.
- [128] T. Aroui, Y. Koubaa and A. Toumi, "Modelling and Diagnostics of Inductions Machines Under Rotor Failures," *ICGST-ACSE Journal*, vol. 7, no. 2, pp. 9-18, Nov. 2007.
- [129] J. C. Principe, N. R. Euliano and W. C. Lefebvre, *Neural and adaptive systems: fundamentals through simulations*, New York: Wiley, 2000.
- [130] C.M. Bishop, *Pattern Recognition and Machine Learning*, New York: Springer, 2006.
- [131] A.C. McCormick and A. K. Nandi, "Classification of the rotating machine condition using artificial neural networks," *Proceedings of the Institution of Mechanical Engineers - Part C*, vol. 211, pp. 439-450, 1997.

- [132] G. Dreyfus, *Neural networks: methodology and applications*, Springer Verlag, 2005.
- [133] S. S. Refaat, H. Abu-Rub, M. S. Saad and A. Zahab, "ANN-based for detection, diagnosis the bearing fault for three phase induction motors using current signal," in *IEEE International Conference on Industrial Technology (ICIT)*, Feb. 2013.
- [134] S. Altaf, A. Al-Anbuky and H. GholamHosseini, "Fault diagnosis in a distributed motor network using Artificial Neural Network," in *22nd International Symposium on Power Electronics, Electrical Drives Automation and Motion (SPEEDAM 2014)*, Ischia, Italy, Jun 2014.
- [135] Digi International, "XBee / XBee-PRO RF Modules 802.15.4 Product Manual, 1st ed. Minnetonka, MN 55343: pp. 1-105," Digi International, 2015. [Online]. Available: [http://ftp1.digi.com/support/documentation/90000982\\_S.pdf](http://ftp1.digi.com/support/documentation/90000982_S.pdf)
- [136] A. Sapena-Bano, J. Perez-Cruz, M. Pineda-Sanchez, R. Puche-Panadero, J. Roger-Folch, M. Riera-Guasp and J. Martinez-Roman, "Condition monitoring of electrical machines using low computing power devices," in *International Conference on Electrical Machines (ICEM)*, Sept. 2014.
- [137] V. Pratap, P. Neelam and K. Chandrakant, "Real Time DC Motor Speed Control using PID Controller in LabVIEW," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 3, no. 9, pp. 221-229, Sep. 2014.
- [138] B. Surjith, "PC Controlled Wireless Multipledrive System," *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, no. 10, pp. 398-401, Oct. 2014.
- [139] L. Hou and N. Bergmann, "Novel Industrial Wireless Sensor Networks for Machine Condition Monitoring and Fault Diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 10, pp. 2787-2798, 2012.

# Appendix 1

## Motor Model development code

### “Parameters.m”

%%This program describe the motor features that are associated with the Motor Simulink mode. All the parameters can be easy adjust accruing to the motor specifications.

```
clc, close all, close all;
```

```
%%%%%%%%%% MOTOR PARAMETERS %%%%%%%%%%%
```

```
V=110*sqrt(2); %Line voltage value [V]
Rs=2.02 ; %Stator resistance (Ohm)
Rr=4.12 ; %Rotor resistance (Ohm)
Lsl=8.54e-3 ; %Stator leakage inductance (H)
Lrl=3.13e-3 ; %Rotor leakage inductance (H)
Lm=180e-3 ; %Magnetizing inductance (H)
ffreq=50 ; % fundamental frequency (Hz)imdqrn0
wo=2*pi*ffreq ; %fundamental frequency calculation (rad/s)
p=4 ; %number of poles
Jol=0.0147 ; % value of Moment of inertia (kg.m^2)
Bm=1e-4; %Frictional coefficient
TL1=10; %partial load torque [N.m]
TL2=2; %full load torque [N.m]
nsync=120*freq0/p;%Synchronous speed [rpm]
```

```
gam=2*pi/3;
Ls=Lsl+Lm; Lr=Lrl+Lm;
RM=diag([Rr Rr Rs Rs]); % Resistance matrix
% Inductance matrices
Lss=[Ls 0 0 Ls];
Lrr=[Lr 0 0 Lr];
%% L = [Lss Lsr
%% Lsr' Lrr]
%% dLdtheta = [zeros(2) dLsr
%% dLsr' zeros(2)]
```

## “faults.m”

%%%%This program has been used for the creation of faults within motors. Different formulations have been used to associate the faults with the motor and adjust the severity of the fault influence.

Note: This program will not execute properly without the Simulink Model.

```
clear all;
close all;
clc;
ts=1      %time when is rotor bar breakon
a=5       %break bar - parametr
f=50      %Frequency Hz
P=1       %pole number
Rs=7.58   %Ohm
Rb=71.5e-5 %Ohm
Rbb=5     %Ohm
Lb=0.1e-6 %H
Rr=1.5    %H
Re=1.5e-6 %Ohm
Ls=0,4612 %H
L1s=Ls    %H
Nr=16     %Number rotor bars
Le=1e-7   %H end ring segment
r=70e-3   %m
Ns=160    %
g=0.75e-3 %m
l=65e-3   %m lenght of rotor stack
J=0.027   %kg.m2
Us=230*sqrt(2) %V
TL1=5
TL2=20
we=2*pi*f
wk=2*pi*f %rad/s
Ks=sqrt(2/3)
Kr=sqrt(2/Nr)
Ksk=2/3
Krk=2/3*sqrt(Nr/3)
Ur=zeros(Nr,1)
mo=4*pi*1e-7
ar=(2*pi/Nr)
delta=ar/2
Lms=mo*l*r*Ns*Ns*pi/(4*P*P*g)
Lm=Lms*4*sin(P*ar/2)/(pi*Ns)
```

```

Rr=2*(Re+Rb*(1-cos(ar)))
Lr=(mo*1*r*ar/g+2*Le+2*Lb*(1-cos(ar)))

Inv_Beta=zeros(Nr,1)
Beta=zeros(1,Nr)
Pomocny=zeros(Nr,1)
for i=1:Nr
    Inv_Beta(i,1)=exp(-j*ar/2*(i-1))

    Beta(1,i)=exp(j*ar/2*(i-1))
    Pomocny(i,1)=i
end
Re_Beta=real(Beta)
Beta_k_1_1=exp(j*ar/2*(1))

% Resistance Matrix - only with fault
% k-th bar with fault
k1=2 %k1-th bar with fault
Const_dRr1=Krk*Kr*Beta(1,k1-1)*(1-Beta(1,2)) %*(Rbb-Rb)
Beta_k1_1=exp(j*ar/2*(k1-1))

k2=5 %k2-th bar with fault

Const_dRr2=Krk*Kr*Beta(1,k2-1)*(1-Beta(1,2)) %*(Rbb-Rb)
Beta_k2_1=exp(j*ar/2*(k2-1))
Beta_k2_2=exp(j*ar/2*(k2-2))
Const_Tem=Lm/2*1/Kr/Ks*sqrt(Nr/3)
%.....
% TRansformation
%.....
Rk=zeros(4,4)
Rk(1,1)=Rs
Rk(2,2)=Rs
Rk(3,3)=2*(Re+Rb*(1-cos(ar)))*sqrt(3/Nr)
Rk(4,4)=Rk(3,3)

Lk=zeros(4,4)
Lk(1,1)=L1s+3/2*Lms
Lk(2,2)=Lk(1,1)
Lk(1,3)=Ks/Kr*3/2*sqrt(3/Nr)*Lm
Lk(2,4)=Lk(1,3)
Lk(3,1)=Nr/2*sqrt(Nr/3)*Kr/Ks*Lm
Lk(4,2)=Lk(3,1)
Lk(3,3)=Lr

```

$$Lk(4,4)=Lk(3,3)$$

$$Inv\_Lk=inv(Lk)$$

$$Lwk\_12=-(L1s+3/2*Lms)$$

$$Lwk\_21=-Lwk\_12$$

$$Lwk\_14=-3/2*Ks/Kr*\sqrt{3/Nr}*Lm$$

$$Lwk\_23=-Lwk\_14$$

$$Lwk\_32=-Nr/2*\sqrt{Nr/3}*Kr/Ks*Lm$$

$$Lwk\_41=-Lwk\_32$$

$$Lwk\_34=-Lr$$

$$Lwk\_43=-Lwk\_34$$

# Appendix 2

## Neural Network Source code

### “Data\_generation.m”

Note: This program will not execute properly without the Simulink Model.

```
clear all;
close all;

%Feature1 : Attenuation
min1H=0.04;
max1H=0.06;
min1F1=0;
max1F1=0.039;
min1F2=0.061;
max1F2=0.1;

%Feature2 :
min2H=1450;
max2H=1500;
min2F1=1350;
max2F1=1449;
min2F2=1501;
max2F2=1600;

%Feature3 : Peak Value (Assume value of main peak to be 60)
min3H=0;
max3H=22;
min3F1=23;
max3F1=60;

%Feature4 : Frequency Points
%Further divided into three inputs
min4aF1=30;
max4aF1=49;
min4aF2=1;
max4aF2=10;

min4cF1=51;
max4cF1=70;
min4cF2=90;
max4cF2=99;
```

**%Feature5 : RMS**

```
min5H=0.2;  
max5H=0.3;  
min5F1=0.1;  
max5F1=0.19;  
min5F2=0.31;  
max5F2=0.4;
```

**%Feature6 : Crest Factor**

```
min6H=0.20;  
max6H=0.25;  
min6F1=0.1;  
max6F1=0.19;  
min6F2=0.26;  
max6F2=0.3;
```

```
nSamples = 200;
```

```
x4bx = 50.* ones(1,nSamples);
```

**%Inputs for Healty : Output = [1 0 0 0]**

**%Case I - All 4 features are healty**

```
x3x = min3H + (max3F1-min3H). *rand(1,nSamples);  
x4ax1 = min4aF1 + (max4aF1-min4aF1). *rand(1,nSamples/2);  
x4ax2 = min4aF2 + (max4aF2-min4aF2). *rand(1,nSamples/2);  
x4cx1 = min4cF1 + (max4cF1-min4cF1). *rand(1,nSamples/2);  
x4cx2 = min4cF2 + (max4cF2-min4cF2). *rand(1,nSamples/2);  
x1H = min1H + (max1H-min1H). *rand(1,nSamples);  
x2H = min2H + (max2H-min2H). *rand(1,nSamples);  
x5H = min5H + (max5H-min5H). *rand(1,nSamples);  
x6H = min6H + (max6H-min6H). *rand(1,nSamples);
```

```
xH1 = [x1H ; x2H ; x3x ; x4ax1,x4ax2 ; x4bx; x4cx1,x4cx2 ; x5H ; x6H];
```

**%Case II - Only Feature1 is faulty**

```
x3x = min3H + (max3F1-min3H). *rand(1,nSamples);  
x4ax1 = min4aF1 + (max4aF1-min4aF1). *rand(1,nSamples/2);  
x4ax2 = min4aF2 + (max4aF2-min4aF2). *rand(1,nSamples/2);  
x4cx1 = min4cF1 + (max4cF1-min4cF1). *rand(1,nSamples/2);  
x4cx2 = min4cF2 + (max4cF2-min4cF2). *rand(1,nSamples/2);  
x1F1 = min1F1 + (max1F1-min1F1). *rand(1,nSamples/2);  
x1F2 = min1F2 + (max1F2-min1F2). *rand(1,nSamples/2);  
x2H = min2H + (max2H-min2H). *rand(1,nSamples);  
x5H = min5H + (max5H-min5H). *rand(1,nSamples);
```

x6H = min6H + (max6H-min6H). \*rand(1,nSamples);

xH2 = [x1F1,x1F2 ; x2H ; x3x ; x4ax1,x4ax2 ; x4bx; x4cx1,x4cx2 ; x5H ; x6H];

**%Case III - Only Feature2 is faulty**

x3x = min3H + (max3F1-min3H). \*rand(1,nSamples);

x4ax1 = min4aF1 + (max4aF1-min4aF1). \*rand(1,nSamples/2);

x4ax2 = min4aF2 + (max4aF2-min4aF2). \*rand(1,nSamples/2);

x4cx1 = min4cF1 + (max4cF1-min4cF1). \*rand(1,nSamples/2);

x4cx2 = min4cF2 + (max4cF2-min4cF2). \*rand(1,nSamples/2);

x1H = min1H + (max1H-min1H). \*rand(1,nSamples);

x2F1 = min2F1 + (max2F1-min2F1). \*rand(1,nSamples/2);

x2F2 = min2F2 + (max2F2-min2F2). \*rand(1,nSamples/2);

x5H = min5H + (max5H-min5H). \*rand(1,nSamples);

x6H = min6H + (max6H-min6H). \*rand(1,nSamples);

xH3 = [x1H ; x2F1,x2F2 ; x3x ; x4ax1,x4ax2 ; x4bx; x4cx1,x4cx2 ; x5H ; x6H];

**%Case IV - Only Feature5 is faulty**

x3x = min3H + (max3F1-min3H). \*rand(1,nSamples);

x4ax1 = min4aF1 + (max4aF1-min4aF1). \*rand(1,nSamples/2);

x4ax2 = min4aF2 + (max4aF2-min4aF2). \*rand(1,nSamples/2);

x4cx1 = min4cF1 + (max4cF1-min4cF1). \*rand(1,nSamples/2);

x4cx2 = min4cF2 + (max4cF2-min4cF2). \*rand(1,nSamples/2);

x1H = min1H + (max1H-min1H). \*rand(1,nSamples);

x2H = min2H + (max2H-min2H). \*rand(1,nSamples);

x5F1 = min5F1 + (max5F1-min5F1). \*rand(1,nSamples/2);

x5F2 = min5F2 + (max5F2-min5F2). \*rand(1,nSamples/2);

x6H = min6H + (max6H-min6H). \*rand(1,nSamples);

xH4 = [x1H ; x2H ; x3x ; x4ax1,x4ax2 ; x4bx; x4cx1,x4cx2 ; x5F1,x5F2 ; x6H];

**%Case V - Only Feature6 is faulty**

x3x = min3H + (max3F1-min3H). \*rand(1,nSamples);

x4ax1 = min4aF1 + (max4aF1-min4aF1). \*rand(1,nSamples/2);

x4ax2 = min4aF2 + (max4aF2-min4aF2). \*rand(1,nSamples/2);

x4cx1 = min4cF1 + (max4cF1-min4cF1). \*rand(1,nSamples/2);

x4cx2 = min4cF2 + (max4cF2-min4cF2). \*rand(1,nSamples/2);

x1H = min1H + (max1H-min1H). \*rand(1,nSamples);

x2H = min2H + (max2H-min2H). \*rand(1,nSamples);

x5H = min5H + (max5H-min5H). \*rand(1,nSamples);

x6F1 = min6F1 + (max6F1-min6F1). \*rand(1,nSamples/2);

x6F2 = min6F2 + (max6F2-min6F2). \*rand(1,nSamples/2);

```
xH5 = [x1H ; x2H ; x3x ; x4ax1,x4ax2 ; x4bx; x4cx1,x4cx2 ; x5H ; x6F1,x6F2];
```

```
% Now concatenate all the inputs
```

```
xH = [xH1,xH2,xH3,xH4,xH5];
```

```
% Now Generate the corresponding target output vector
```

```
y1 = ones(1,1000);
```

```
y2 = zeros(3,1000);
```

```
yH = [y1 ; y2];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%
```

```
% Inputs for Fault 1 - Output = [0 1 0 0]
```

```
% Case I - Feature1 and 2
```

```
x3x = min3F1 + (max3F1-min3F1).*rand(1,nSamples);
```

```
x4ax = min4aF1 + (max4aF1-min4aF1).*rand(1,nSamples);
```

```
x4cx = min4cF1 + (max4cF1-min4cF1).*rand(1,nSamples);
```

```
x1F1 = min1F1 + (max1F1-min1F1).*rand(1,nSamples/2);
```

```
x1F2 = min1F2 + (max1F2-min1F2).*rand(1,nSamples/2);
```

```
x2F1 = min2F1 + (max2F1-min2F1).*rand(1,nSamples/2);
```

```
x2F2 = min2F2 + (max2F2-min2F2).*rand(1,nSamples/2);
```

```
x5H = min5H + (max5H-min5H).*rand(1,nSamples);
```

```
x6H = min6H + (max6H-min6H).*rand(1,nSamples);
```

```
xF1 = [x1F1,x1F2 ; x2F1,x2F2 ; x3x ; x4ax ; x4bx; x4cx ; x5H ; x6H];
```

```
% Case II - Feature1 and 5
```

```
x3x = min3F1 + (max3F1-min3F1).*rand(1,nSamples);
```

```
x4ax = min4aF1 + (max4aF1-min4aF1).*rand(1,nSamples);
```

```
x4cx = min4cF1 + (max4cF1-min4cF1).*rand(1,nSamples);
```

```
x1F1 = min1F1 + (max1F1-min1F1).*rand(1,nSamples/2);
```

```
x1F2 = min1F2 + (max1F2-min1F2).*rand(1,nSamples/2);
```

```
x2H = min2F1 + (max2F2-min2F1).*rand(1,nSamples);
```

```
x5F1 = min5F1 + (max5F1-min5F1).*rand(1,nSamples/2);
```

```
x5F2 = min5F2 + (max5F2-min5F2).*rand(1,nSamples/2);
```

```
x6H = min6H + (max6H-min6H).*rand(1,nSamples);
```

```
xF2 = [x1F1,x1F2 ; x2H ; x3x ; x4ax ; x4bx; x4cx ; x5F1,x5F2 ; x6H];
```

```
% Case III - Feature1 and 6
```

```
x3x = min3F1 + (max3F1-min3F1).*rand(1,nSamples);
```

```
x4ax = min4aF1 + (max4aF1-min4aF1).*rand(1,nSamples);
```

```
x4cx = min4cF1 + (max4cF1-min4cF1).*rand(1,nSamples);
```

```
x1F1 = min1F1 + (max1F1-min1F1).*rand(1,nSamples/2);
```

```

x1F2 = min1F2 + (max1F2-min1F2).*rand(1,nSamples/2);
x2H = min2F1 + (max2F2-min2F1).*rand(1,nSamples);
x5H = min5H + (max5H-min5H).*rand(1,nSamples);
x6F1 = min6F1 + (max6F1-min6F1).*rand(1,nSamples/2);
x6F2 = min6F2 + (max6F2-min6F2).*rand(1,nSamples/2);

xF3 = [x1F1,x1F2 ; x2H ; x3x ; x4ax ; x4bx; x4cx ; x5H ; x6F1,x6F2];
% Case IV - Feature 2 and 5
x3x = min3F1 + (max3F1-min3F1).*rand(1,nSamples);
x4ax = min4aF1 + (max4aF1-min4aF1).*rand(1,nSamples);
x4cx = min4cF1 + (max4cF1-min4cF1).*rand(1,nSamples);
x1H = min1F1 + (max1F2-min1F1).*rand(1,nSamples);
x2F1 = min2F1 + (max2F1-min2F1).*rand(1,nSamples/2);
x2F2 = min2F2 + (max2F2-min2F2).*rand(1,nSamples/2);
x5F1 = min5F1 + (max5F1-min5F1).*rand(1,nSamples/2);
x5F2 = min5F2 + (max5F2-min5F2).*rand(1,nSamples/2);
x6H = min6H + (max6H-min6H).*rand(1,nSamples);

xF4 = [x1H ; x2F1,x2F2 ; x3x ; x4ax ; x4bx; x4cx ; x5F1,x5F2 ; x6H];
% Case V - Feature 2 and 6
x3x = min3F1 + (max3F1-min3F1).*rand(1,nSamples);
x4ax = min4aF1 + (max4aF1-min4aF1).*rand(1,nSamples);
x4cx = min4cF1 + (max4cF1-min4cF1).*rand(1,nSamples);
x1H = min1F1 + (max1F2-min1F1).*rand(1,nSamples);
x2F1 = min2F1 + (max2F1-min2F1).*rand(1,nSamples/2);
x2F2 = min2F2 + (max2F2-min2F2).*rand(1,nSamples/2);
x5H = min5H + (max5H-min5H).*rand(1,nSamples);
x6F1 = min6F1 + (max6F1-min6F1).*rand(1,nSamples/2);
x6F2 = min6F2 + (max6F2-min6F2).*rand(1,nSamples/2);

xF5 = [x1H ; x2F1,x2F2 ; x3x ; x4ax ; x4bx; x4cx ; x5H ; x6F1,x6F2];
% Case VI - Feature 5 and 6
x3x = min3F1 + (max3F1-min3F1).*rand(1,nSamples);
x4ax = min4aF1 + (max4aF1-min4aF1).*rand(1,nSamples);
x4cx = min4cF1 + (max4cF1-min4cF1).*rand(1,nSamples);
x1H = min1F1 + (max1F2-min1F1).*rand(1,nSamples);
x2H = min2F1 + (max2F2-min2F1).*rand(1,nSamples);
x5F1 = min5F1 + (max5F1-min5F1).*rand(1,nSamples/2);
x5F2 = min5F2 + (max5F2-min5F2).*rand(1,nSamples/2);
x6F1 = min6F1 + (max6F1-min6F1).*rand(1,nSamples/2);
x6F2 = min6F2 + (max6F2-min6F2).*rand(1,nSamples/2);

xF6 = [x1H ; x2H ; x3x ; x4ax ; x4bx; x4cx ; x5F1,x5F2 ; x6F1,x6F2];

```

```

% Now concatenate all the inputs for FAULT 1
xF = [xF1,xF2,xF3,xF4,xF5,xF6];

% Now generate the corresponding target output
y1 = zeros(1,1200);
y2 = ones(1,1200);
y3 = zeros(2,1200);

yF = [y1;y2;y3];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Inputs for Fault 2 - Output = [0 0 1 0]
% Case I - Feature1 and 2
x3x = min3F1 + (max3F1-min3F1).*rand(1,nSamples);
x4ax = min4aF2 + (max4aF2-min4aF2).*rand(1,nSamples);
x4cx = min4cF2 + (max4cF2-min4cF2).*rand(1,nSamples);
x1F1 = min1F1 + (max1F1-min1F1).*rand(1,nSamples/2);
x1F2 = min1F2 + (max1F2-min1F2).*rand(1,nSamples/2);
x2F1 = min2F1 + (max2F1-min2F1).*rand(1,nSamples/2);
x2F2 = min2F2 + (max2F2-min2F2).*rand(1,nSamples/2);
x5H = min5H + (max5H-min5H).*rand(1,nSamples);
x6H = min6H + (max6H-min6H).*rand(1,nSamples);

xF1 = [x1F1,x1F2 ; x2F1,x2F2 ; x3x ; x4ax ; x4bx; x4cx ; x5H ; x6H];
% Case II - Feature1 and 5
x3x = min3F1 + (max3F1-min3F1).*rand(1,nSamples);
x4ax = min4aF2 + (max4aF2-min4aF2).*rand(1,nSamples);
x4cx = min4cF2 + (max4cF2-min4cF2).*rand(1,nSamples);
x1F1 = min1F1 + (max1F1-min1F1).*rand(1,nSamples/2);
x1F2 = min1F2 + (max1F2-min1F2).*rand(1,nSamples/2);
x2H = min2F1 + (max2F2-min2F1).*rand(1,nSamples);
x5F1 = min5F1 + (max5F1-min5F1).*rand(1,nSamples/2);
x5F2 = min5F2 + (max5F2-min5F2).*rand(1,nSamples/2);
x6H = min6H + (max6H-min6H).*rand(1,nSamples);

xF2 = [x1F1,x1F2 ; x2H ; x3x ; x4ax ; x4bx; x4cx ; x5F1,x5F2 ; x6H];
% Case III - Feature1 and 6
x3x = min3F1 + (max3F1-min3F1).*rand(1,nSamples);
x4ax = min4aF2 + (max4aF2-min4aF2).*rand(1,nSamples);
x4cx = min4cF2 + (max4cF2-min4cF2).*rand(1,nSamples);
x1F1 = min1F1 + (max1F1-min1F1).*rand(1,nSamples/2);
x1F2 = min1F2 + (max1F2-min1F2).*rand(1,nSamples/2);
x2H = min2F1 + (max2F2-min2F1).*rand(1,nSamples);
x5H = min5H + (max5H-min5H).*rand(1,nSamples);

```

```

x6F1 = min6F1 + (max6F1-min6F1).*rand(1,nSamples/2);
x6F2 = min6F2 + (max6F2-min6F2).*rand(1,nSamples/2);

xF3 = [x1F1,x1F2 ; x2H ; x3x ; x4ax ; x4bx; x4cx ; x5H ; x6F1,x6F2];
% Case IV - Feature 2 and 5
x3x = min3F1 + (max3F1-min3F1).*rand(1,nSamples);
x4ax = min4aF2 + (max4aF2-min4aF2).*rand(1,nSamples);
x4cx = min4cF2 + (max4cF2-min4cF2).*rand(1,nSamples);
x1H = min1F1 + (max1F2-min1F1).*rand(1,nSamples);
x2F1 = min2F1 + (max2F1-min2F1).*rand(1,nSamples/2);
x2F2 = min2F2 + (max2F2-min2F2).*rand(1,nSamples/2);
x5F1 = min5F1 + (max5F1-min5F1).*rand(1,nSamples/2);
x5F2 = min5F2 + (max5F2-min5F2).*rand(1,nSamples/2);
x6H = min6H + (max6H-min6H).*rand(1,nSamples);

xF4 = [x1H ; x2F1,x2F2 ; x3x ; x4ax ; x4bx; x4cx ; x5F1,x5F2 ; x6H];
% Case V - Feature 2 and 6
x3x = min3F1 + (max3F1-min3F1).*rand(1,nSamples);
x4ax = min4aF2 + (max4aF2-min4aF2).*rand(1,nSamples);
x4cx = min4cF2 + (max4cF2-min4cF2).*rand(1,nSamples);
x1H = min1F1 + (max1F2-min1F1).*rand(1,nSamples);
x2F1 = min2F1 + (max2F1-min2F1).*rand(1,nSamples/2);
x2F2 = min2F2 + (max2F2-min2F2).*rand(1,nSamples/2);
x5H = min5H + (max5H-min5H).*rand(1,nSamples);
x6F1 = min6F1 + (max6F1-min6F1).*rand(1,nSamples/2);
x6F2 = min6F2 + (max6F2-min6F2).*rand(1,nSamples/2);

xF5 = [x1H ; x2F1,x2F2 ; x3x ; x4ax ; x4bx; x4cx ; x5H ; x6F1,x6F2];
% Case VI - Feature 5 and 6
x3x = min3F1 + (max3F1-min3F1).*rand(1,nSamples);
x4ax = min4aF2 + (max4aF2-min4aF2).*rand(1,nSamples);
x4cx = min4cF2 + (max4cF2-min4cF2).*rand(1,nSamples);
x1H = min1F1 + (max1F2-min1F1).*rand(1,nSamples);
x2H = min2F1 + (max2F2-min2F1).*rand(1,nSamples);
x5F1 = min5F1 + (max5F1-min5F1).*rand(1,nSamples/2);
x5F2 = min5F2 + (max5F2-min5F2).*rand(1,nSamples/2);
x6F1 = min6F1 + (max6F1-min6F1).*rand(1,nSamples/2);
x6F2 = min6F2 + (max6F2-min6F2).*rand(1,nSamples/2);

xF6 = [x1H ; x2H ; x3x ; x4ax ; x4bx; x4cx ; x5F1,x5F2 ; x6F1,x6F2];

% Now concatenate all the inputs for FAULT 2
xFF = [xF1,xF2,xF3,xF4,xF5,xF6];

```

% Now generate the corresponding target output

```
y1 = zeros(2,1200);  
y2 = ones(1,1200);  
y3 = zeros(1,1200);
```

```
yFF = [y1;y2;y3];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%Inputs for Unknown Fault : Output = [0 0 0 1]

```
x3x = min3H + (max3H-min3H).*rand(1,nSamples);
```

%Case I - Feature 1 and 2

```
x4ax1 = min4aF1 + (max4aF1-min4aF1).*rand(1,nSamples/2);  
x4ax2 = min4aF2 + (max4aF2-min4aF2).*rand(1,nSamples/2);  
x4cx1 = min4cF1 + (max4cF1-min4cF1).*rand(1,nSamples/2);  
x4cx2 = min4cF2 + (max4cF2-min4cF2).*rand(1,nSamples/2);  
x1F1 = min1F1 + (max1F1-min1F1).*rand(1,nSamples/2);  
x1F2 = min1F2 + (max1F2-min1F2).*rand(1,nSamples/2);  
x2F1 = min2F1 + (max2F1-min2F1).*rand(1,nSamples/2);  
x2F2 = min2F1 + (max2F2-min2F2).*rand(1,nSamples/2);  
x5x = min5F1 + (max5F2-min5F1).*rand(1,nSamples);  
x6x = min6F1 + (max6F2-min6F1).*rand(1,nSamples);
```

```
xU1 = [x1F1,x1F2 ; x2F1,x2F2 ; x3x; x4ax1,x4ax2 ; x4bx; x4cx1,x4cx2 ; x5x ; x6x];
```

%Case II - Feature 1 and 5

```
x4ax1 = min4aF1 + (max4aF1-min4aF1).*rand(1,nSamples/2);  
x4ax2 = min4aF2 + (max4aF2-min4aF2).*rand(1,nSamples/2);  
x4cx1 = min4cF1 + (max4cF1-min4cF1).*rand(1,nSamples/2);  
x4cx2 = min4cF2 + (max4cF2-min4cF2).*rand(1,nSamples/2);  
x1F1 = min1F1 + (max1F1-min1F1).*rand(1,nSamples/2);  
x1F2 = min1F2 + (max1F2-min1F2).*rand(1,nSamples/2);  
x2x = min2F1 + (max2F2-min2F1).*rand(1,nSamples);  
x5F1 = min5F1 + (max5F1-min5F1).*rand(1,nSamples/2);  
x5F2 = min5F2 + (max5F2-min5F2).*rand(1,nSamples/2);  
x6x = min6F1 + (max6F2-min6F1).*rand(1,nSamples);
```

```
xU2 = [x1F1,x1F2 ; x2x ; x3x; x4ax1,x4ax2 ; x4bx; x4cx1,x4cx2 ; x5F1,x5F2 ; x6x];
```

%Case III - Feature 1 and 6

```
x4ax1 = min4aF1 + (max4aF1-min4aF1).*rand(1,nSamples/2);  
x4ax2 = min4aF2 + (max4aF2-min4aF2).*rand(1,nSamples/2);  
x4cx1 = min4cF1 + (max4cF1-min4cF1).*rand(1,nSamples/2);  
x4cx2 = min4cF2 + (max4cF2-min4cF2).*rand(1,nSamples/2);
```

```

x1F1 = min1F1 + (max1F1-min1F1).*rand(1,nSamples/2);
x1F2 = min1F2 + (max1F2-min1F2).*rand(1,nSamples/2);
x2x = min2F1 + (max2F2-min2F1).*rand(1,nSamples);
x5x = min5F1 + (max5F2-min5F1).*rand(1,nSamples);
x6F1 = min6F1 + (max6F1-min6F1).*rand(1,nSamples/2);
x6F2 = min6F2 + (max6F2-min6F2).*rand(1,nSamples/2);

xU3 = [x1F1,x1F2 ; x2x ; x3x; x4ax1,x4ax2 ; x4bx; x4cx1,x4cx2 ; x5x ; x6F1,x6F2];

```

#### %Case IV - Feature 2 and 5

```

x4ax1 = min4aF1 + (max4aF1-min4aF1).*rand(1,nSamples/2);
x4ax2 = min4aF2 + (max4aF2-min4aF2).*rand(1,nSamples/2);
x4cx1 = min4cF1 + (max4cF1-min4cF1).*rand(1,nSamples/2);
x4cx2 = min4cF2 + (max4cF2-min4cF2).*rand(1,nSamples/2);
x1x = min1F1 + (max1F2-min1F1).*rand(1,nSamples);
x2F1 = min2F1 + (max2F1-min2F1).*rand(1,nSamples/2);
x2F2 = min2F2 + (max2F2-min2F2).*rand(1,nSamples/2);
x5F1 = min5F1 + (max5F1-min5F1).*rand(1,nSamples/2);
x5F2 = min5F2 + (max5F2-min5F2).*rand(1,nSamples/2);
x6x = min6F1 + (max6F2-min6F1).*rand(1,nSamples);

xU4 = [x1x ; x2F1,x2F2 ; x3x; x4ax1,x4ax2 ; x4bx; x4cx1,x4cx2 ; x5F1,x5F2 ; x6x];

```

#### %Case V - Feature 2 and 6

```

x4ax1 = min4aF1 + (max4aF1-min4aF1).*rand(1,nSamples/2);
x4ax2 = min4aF2 + (max4aF2-min4aF2).*rand(1,nSamples/2);
x4cx1 = min4cF1 + (max4cF1-min4cF1).*rand(1,nSamples/2);
x4cx2 = min4cF2 + (max4cF2-min4cF2).*rand(1,nSamples/2);
x1x = min1F1 + (max1F2-min1F1).*rand(1,nSamples);
x2F1 = min2F1 + (max2F1-min2F1).*rand(1,nSamples/2);
x2F2 = min2F2 + (max2F2-min2F2).*rand(1,nSamples/2);
x5x = min5F1 + (max5F2-min5F1).*rand(1,nSamples);
x6F1 = min6F1 + (max6F1-min6F1).*rand(1,nSamples/2);
x6F2 = min6F2 + (max6F2-min6F2).*rand(1,nSamples/2);

xU5 = [x1x ; x2F1,x2F2 ; x3x; x4ax1,x4ax2 ; x4bx; x4cx1,x4cx2 ; x5x ; x6F1,x6F2];

```

#### %Case VI - Feature 5 and 6

```

x4ax1 = min4aF1 + (max4aF1-min4aF1).*rand(1,nSamples/2);
x4ax2 = min4aF2 + (max4aF2-min4aF2).*rand(1,nSamples/2);
x4cx1 = min4cF1 + (max4cF1-min4cF1).*rand(1,nSamples/2);
x4cx2 = min4cF2 + (max4cF2-min4cF2).*rand(1,nSamples/2);
x1x = min1F1 + (max1F2-min1F1).*rand(1,nSamples);
x2x = min2F1 + (max2F2-min2F1).*rand(1,nSamples);

```

```

x5F1 = min5F1 + (max5F1-min5F1).*rand(1,nSamples/2);
x2F2 = min5F2 + (max5F2-min5F2).*rand(1,nSamples/2);
x6F1 = min6F1 + (max6F1-min6F1).*rand(1,nSamples/2);
x6F2 = min6F2 + (max6F2-min6F2).*rand(1,nSamples/2);

xU6 = [x1x ; x2x ; x3x; x4ax1,x4ax2 ; x4bx; x4cx1,x4cx2 ; x5F1,x5F2 ; x6F1,x6F2];

% Now concatenate all the inputs
xU = [xU1,xU2,xU3,xU4,xU5,xU6];

% Now generate the corresponding target output
y1 = zeros(3,1200);
y2 = ones(1,1200);

yU = [y1;y2];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Now stack everything together
In = [xH,xF,xFF,xU];
Out = [yH,yF,yFF,yU];

% Now save the training data in data.mat
save('data.mat','In','Out');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

## “Motor\_condition.m”

```

% This will take the Motors output and tell about the corresponding conditions of
the Motors

Clc;

ap = load('Motor_out.mat');
out_all = ap.sen_out(2:37,1);

s1_out = out_all(1:4);
s2_out = out_all(5:8);
s3_out = out_all(9:12);
s4_out = out_all(13:16);
s5_out = out_all(17:20);
s6_out = out_all(21:24);
s7_out = out_all(25:28);

```

```

s8_out = out_all(29:32);
s9_out = out_all(33:36);

% Check the condition of Motor 1 - BUS 1
[val,ind] = max(s1_out);
if (ind == 1)
    display(' Bus 1 - Motor 1 - Healthy - No Fault')
else if (ind == 2)
    display(' Bus 1 - Motor 1 - Faulty - Broken Rotor Bar Fault ')
else if (ind == 3)
    display(' Bus 1 - Motor 1 - Faulty - Air-Gap Eccentricity Fault')
else if (ind == 4)
    display('Bus 1 - Motor 1 - Faulty - Unknown Fault)
else if (ind == 5)
    display('Bus 1 - Motor 1 - Healthy - Some Noise signal)
else
    display('N')
end
end
end
end

% Check the condition of Motor 2 - BUS 1
[val,ind] = max(s2_out);
if (ind == 1)
    display(' Bus 1 - Motor 2 - Healthy - No Fault')
else if (ind == 2)
    display(' Bus 1 - Motor 2 - Faulty - Broken Rotor Bar Fault ')
else if (ind == 3)
    display(' Bus 1 - Motor 2 - Faulty - Air-Gap Eccentricity Fault')
else if (ind == 4)
    display('Bus 1 - Motor 2 - Faulty - Unknown Fault)
else if (ind == 5)
    display('Bus 1 - Motor 2 - Healthy - Some Noise signal)
end
end
end
end

% Check the condition of Motor 3 - BUS 1
[val,ind] = max(s3_out);
if (ind == 1)
    display(' Bus 1 - Motor 3 - Healthy - No Fault')

```

```

else if (ind == 2)
    display(' Bus 1 – Motor 3 – Faulty – Broken Rotor Bar Fault ')
else if (ind == 3)
    display(' Bus 1 – Motor 3 – Faulty – Air-Gap Eccentricity Fault')
else if (ind == 4)
    display('Bus 1 – Motor 3 – Faulty – Unknown Fault)
else if (ind == 5)
    display('Bus 1 – Motor 3 – Healthy – Some Noise signal)
    end
end
end
end
end

```

```

% Check the condition of Motor 4 - BUS 2

```

```

[val,ind] = max(s4_out);
if (ind == 1)
    display(' Bus 2 – Motor 4 – Healthy – No Fault')
else if (ind == 2)
    display(' Bus 2 – Motor 4 – Faulty – Broken Rotor Bar Fault ')
else if (ind == 3)
    display(' Bus 2 – Motor 4 – Faulty – Air-Gap Eccentricity Fault')
else if (ind == 4)
    display('Bus 2 – Motor 4 – Faulty – Unknown Fault)
else if (ind == 5)
    display('Bus 2 – Motor 4 – Healthy – Some Noise signal)
    end
end
end
end
end

```

```

% Check the condition of Motor 5 - BUS 2

```

```

[val,ind] = max(s5_out);
if (ind == 1)
    display(' Bus 2 – Motor 5 – Healthy – No Fault')
else if (ind == 2)
    display(' Bus 2 – Motor 5 – Faulty – Broken Rotor Bar Fault ')
else if (ind == 3)
    display(' Bus 2 – Motor 5 – Faulty – Air-Gap Eccentricity Fault')
else if (ind == 4)
    display('Bus 2 – Motor 5 – Faulty – Unknown Fault)
else if (ind == 5)
    display('Bus 2 – Motor 5 – Healthy – Some Noise signal)
    end
end

```

```

        end
    end
end

% Check the condition of Motor 6 - BUS 2
[val,ind] = max(s6_out);
if (ind == 1)
    display(' Bus 2 – Motor 6 – Healthy – No Fault')
else if (ind == 2)
    display(' Bus 2 – Motor 6 – Faulty – Broken Rotor Bar Fault ')
else if (ind == 3)
    display(' Bus 2 – Motor 6 – Faulty – Air-Gap Eccentricity Fault')
else if (ind == 4)
    display('Bus 2 – Motor 6 – Faulty – Unknown Fault')
else if (ind == 5)
    display('Bus 2 – Motor 6 – Healthy – Some Noise signal')

    end
end
end
end

```

```

% Check the condition of Motor 7 - BUS 3
[val,ind] = max(s7_out);
if (ind == 1)
    display(' Bus 3 – Motor 7 – Healthy – No Fault')
else if (ind == 2)
    display(' Bus 3 – Motor 7 – Faulty – Broken Rotor Bar Fault ')
else if (ind == 3)
    display(' Bus 3 – Motor 7 – Faulty – Air-Gap Eccentricity Fault')
else if (ind == 4)
    display('Bus 3 – Motor 7 – Faulty – Unknown Fault')
else if (ind == 5)
    display('Bus 3 – Motor 7 – Healthy – Some Noise signal')

    end
end
end
end

```

```

% Check the condition of Motor 8 - BUS 3
[val,ind] = max(s8_out);
if (ind == 1)
    display(' Bus 3 – Motor 8 – Healthy – No Fault')

```

```

else if (ind == 2)
    display(' Bus 3 – Motor 8 – Faulty – Broken Rotor Bar Fault ')
else if (ind == 3)
    display(' Bus 3 – Motor 8 – Faulty – Air-Gap Eccentricity Fault')
else if (ind == 4)
    display('Bus 3 – Motor 8 – Faulty – Unknown Fault)
else if (ind == 5)
    display('Bus 3 – Motor 8 – Healthy – Some Noise signal)
    end
end
end
end

% Check the condition of Motor 9 - BUS 3
[val,ind] = max(s9_out);
if (ind == 1)
    display(' Bus 3 – Motor 9 – Healthy – No Fault')
else if (ind == 2)
    display(' Bus 3 – Motor 9 – Faulty – Broken Rotor Bar Fault ')
else if (ind == 3)
    display(' Bus 3 – Motor 9 – Faulty – Air-Gap Eccentricity Fault')
else if (ind == 4)
    display('Bus 3 – Motor 9 – Faulty – Unknown Fault)
else if (ind == 5)
    display('Bus 3 – Motor 9 – Healthy – Some Noise signal)
    end
end
end
end

```

### **“graphs.m”**

% This program demonstrate the process of loading training data and create a pattern recognition network and Setup division of data for training, validation, testing and draw the confusion matrices and performance results to validate the authenticity of data.

```

clc;
clear all;
close all;

```

```

% Load the Training data
d = load('data.mat');

```

```

inputs = d.In;
targets = d.Out;

% Create a Pattern Recognition Network
hiddenLayerSize = 100;
net = patternnet(hiddenLayerSize);

% Choose Input and Output Pre/Post-Processing Functions

net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing

net.divideFcn = 'dividerand'; % Divide data randomly
net.divideMode = 'sample'; % Divide up every sample
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 10/100;
net.divideParam.testRatio = 10/100;

net.trainFcn = 'trainscg'; % Scaled conjugate gradient

% Choose a Performance Function

net.performFcn = 'mse'; % Mean squared error

% Choose Plot Functions

net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
'plotregression','plotfit'};

net.trainParam.goal = 0.0001;
net.trainParam.max_fail = [d];

% Now form the models of other motors similarly

net1 = net;
net2 = net;
net3 = net;
net4 = net;
net5 = net;
net6 = net;

```

```

net7 = net;
net8 = net;
net9 = net;

% Train all the Networks
RandStream.setGlobalStream(RandStream('mt19937ar','seed',1));
[net,tr] = train(net,inputs,targets);
RandStream.setGlobalStream(RandStream('mt19937ar','seed',2));
[net1,tr1] = train(net1,inputs,targets);
RandStream.setGlobalStream(RandStream('mt19937ar','seed',3));
[net2,tr2] = train(net2,inputs,targets);
RandStream.setGlobalStream(RandStream('mt19937ar','seed',4));
[net3,tr3] = train(net3,inputs,targets);
RandStream.setGlobalStream(RandStream('mt19937ar','seed',9));
[net4,tr4] = train(net4,inputs,targets);
RandStream.setGlobalStream(RandStream('mt19937ar','seed',10));
[net5,tr5] = train(net5,inputs,targets);
RandStream.setGlobalStream(RandStream('mt19937ar','seed',11));
[net6,tr6] = train(net6,inputs,targets);
RandStream.setGlobalStream(RandStream('mt19937ar','seed',12));
[net7,tr7] = train(net7,inputs,targets);
RandStream.setGlobalStream(RandStream('mt19937ar','seed',13));
[net8,tr8] = train(net8,inputs,targets);
RandStream.setGlobalStream(RandStream('mt19937ar','seed',14));
[net9,tr9] = train(net9,inputs,targets);

% Test all the Networks
outputs = net(inputs);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)
outputs1 = net1(inputs);
errors1 = gsubtract(targets,outputs1);
performance1 = perform(net1,targets,outputs1)
outputs2 = net2(inputs);
errors2 = gsubtract(targets,outputs2);
performance2 = perform(net2,targets,outputs2)
outputs3 = net3(inputs);
errors3 = gsubtract(targets,outputs3);
performance3 = perform(net3,targets,outputs3)
outputs4 = net4(inputs);
errors4 = gsubtract(targets,outputs4);
performance4 = perform(net4,targets,outputs4)
outputs5 = net5(inputs);

```

```

errors5 = gsubtract(targets,outputs5);
performance5 = perform(net5,targets,outputs5)
outputs6 = net6(inputs);
errors6 = gsubtract(targets,outputs6);
performance6 = perform(net6,targets,outputs6)
outputs7 = net7(inputs);
errors7 = gsubtract(targets,outputs7);
performance7 = perform(net7,targets,outputs7)
outputs8 = net8(inputs);
errors8 = gsubtract(targets,outputs8);
performance8 = perform(net8,targets,outputs8)

```

```

% Recalculate Training, Validation and Test Performance

```

```

trainTargets = targets .* tr.trainMask{1};
valTargets = targets .* tr.valMask{1};
testTargets = targets .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,outputs);
valPerformance = perform(net,valTargets,outputs);
testPerformance = perform(net,testTargets,outputs);

```

```

% View the Network

```

```

% view(net)

```

```

% Plots

```

```

% Uncomment these lines to enable various plots.

```

```

%figure, plotperform(tr)

```

```

%figure, plottrainstate(tr)

```

```

%figure, plotconfusion(targets,outputs)

```

```

%figure, plotroc(targets,outputs)

```

```

%figure, ploterrhist(errors)

```

```

set(findobj(gca,'type','text'),'fontsize',35)

```

```

% Draw the various plots for all motors

```

```

figure,plotperform(tr),title('Performance Graph for Motor 1');

```

```

figure,plotconfusion(targets,outputs),title('Confusion Matrix for Motor 1');

```

```

%figure,ploterrhist(errors),title('Error Histogram for Sensor 1');

```

```

figure,plotperform(tr1),title('Performance Graph for Motor 2');

```

```

figure,plotconfusion(targets,outputs1),title('Confusion Matrix for Motor 2');

```

```

%figure,ploterrhist(errors1),title('Error Histogram for Sensor 2');

```

```

figure,plotperform(tr2),title('Performance Graph for Motor 3');

```

```

figure,plotconfusion(targets,outputs2),title('Confusion Matrix for Motor 3');

```

```

%figure,ploterrhist(errors2),title('Error Histogram for Sensor 3');

```

```

figure,plotperform(tr3),title('Performance Graph for Motor 4');

```

```

figure,plotconfusion(targets,outputs3),title('Confusion Matrix for Motor 4');

```

```

%figure,ploterrhist(errors3),title('Error Histogram for Sensor 4');

```

```
figure,plotperform(tr4),title('Performance Graph for Motor 5');
figure,plotconfusion(targets,outputs4),title('Confusion Matrix for Motor 5');
%figure,ploterrhist(errors4),title('Error Histogram for Sensor 5');
figure,plotperform(tr5),title('Performance Graph for Motor 6');
figure,plotconfusion(targets,outputs5),title('Confusion Matrix for Motor 6');
%figure,ploterrhist(errors5),title('Error Histogram for Sensor 6');
figure,plotperform(tr6),title('Performance Graph for Motor7');
figure,plotconfusion(targets,outputs6),title('Confusion Matrix for Motor 7');
%figure,ploterrhist(errors6),title('Error Histogram for Sensor 7');
figure,plotperform(tr7),title('Performance Graph for Motor 8');
figure,plotconfusion(targets,outputs7),title('Confusion Matrix for Motor 8');
%figure,ploterrhist(errors7),title('Error Histogram for Sensor 8');
figure,plotperform(tr8),title('Performance Graph for Motor 9');
figure,plotconfusion(targets,outputs8),title('Confusion Matrix for Motor 9');
%figure,ploterrhist(errors8),title('Error Histogram for Sensor 9');
```

# Appendix 3

## Arduino R3 Specifications

The following table shows the technical specifications of an Arduino R3 module [135]:

1	Microcontroller	ATmega328P
2	Input Voltage (recommended)	5-12V
3	Input Voltage (Max. Limit)	5-20V
4	I/O pins	14 ( digital input pins) and 5 (analog signal input pins)
5	DC electric current per I/O pin	40 mA
6	DC electric current for the 3.3V pin	50 mA
7	Speed Clock	16 MHz
8	EEPROM	1K bytes
9	Programmable system flash	32 KB

### “Arduino End node code”

% This program demonstrated the acquiring the electric current and calculate the RPM by counting the position difference using the position counter within a defined time period. Other feature such as detection of significant sidebands, amplitude values, RMS and Crest Factor for the analysis.

```
#include <avr/pgmspace.h>
#include <FreqCounter.h>
#include <Average.h>
#include < threshold.h >

unsigned long freq;

#define F_ADJ 1.94
#define N_MAX 128
#define PI2_OVER_N 0.0491
#define PI2 6.2832

#define FFT_SIZE 256
#define LOG_2_FFT 8

#define N_WAVE 256 /* full length of Sinewave[] */

int f_r[FFT_SIZE];
int f_i[FFT_SIZE];
int NBR[10];
```

//Sample perfect sine wave to reduce the FFT computation at end-node level

```
const int16_t Sinewave[] PROGMEM= {
/* 8-bits */
  +0, +6, +13, +19, +25, +31, +37, +44, +50, +5
6, +62, +68, +74, +80, +86, +92,
  +98, +103, +109, +115, +120, +126, +131, +136, +142
, +147, +152, +157, +162, +167, +171, +176,
  +180, +185, +189, +193, +197, +201, +205, +208, +21
2, +215, +219, +222, +225, +228, +231, +233,
  +236, +238, +240, +242, +244, +246, +247, +249, +25
0, +251, +252, +253, +254, +254, +255, +255,
  +255, +255, +255, +254, +254, +253, +252, +251, +25
0, +249, +247, +246, +244, +242, +240, +238,
  +236, +233, +231, +228, +225, +222, +219, +215, +21
2, +208, +205, +201, +197, +193, +189, +185,
  +180, +176, +171, +167, +162, +157, +152, +147, +14
2, +136, +131, +126, +120, +115, +109, +103,
  +98, +92, +86, +80, +74, +68, +62, +56, +50, +
44, +37, +31, +25, +19, +13, +6,
  +0, -6, -13, -19, -25, -31, -38, -44, -50, -
56, -62, -68, -74, -80, -86, -92,
  -98, -104, -109, -115, -121, -126, -132, -137, -142, -
147, -152, -157, -162, -167, -172, -177,
  -181, -185, -190, -194, -198, -202, -206, -209, -213, -
216, -220, -223, -226, -229, -231, -234,
  -237, -239, -241, -243, -245, -247, -248, -250, -251, -
252, -253, -254, -255, -255, -256, -256,
  -256, -256, -256, -255, -255, -254, -253, -252, -251, -
250, -248, -247, -245, -243, -241, -239,
  -237, -234, -231, -229, -226, -223, -220, -216, -213, -
209, -206, -202, -198, -194, -190, -185,
  -181, -177, -172, -167, -162, -157, -152, -147, -142, -
137, -132, -126, -121, -115, -109, -104,
  -98, -92, -86, -80, -74, -68, -62, -56, -50, -
44, -38, -31, -25, -19, -13, -6
};
```

```
void sample_ADC(int dest[],int n, char channel)
{
  for(int i=0;i<n;i++)
  {
    dest[i]=analogRead(channel)-490;
    delay(2);
  }
}
```

```
void setup() {

  int k=0;
  Serial.begin(9600);
```

```

clear_vector(f_r,NWAVE);
clear_vector(f_i,NWAVE);
delay(3000);
Serial.println("%Ready...1");

get_rpm();
get_rpm();
get_rpm();
get_rpm();
while(frq==0)

get_rpm();

sample_ADC(f_r,NWAVE,0);

Serial.print("x=");
send_vector_to_serial(f_r,250);
int peak=0,rms=0;
long sum=0;
for(k=0;k<NWAVE;k++)
{
  if(peak<abs(f_r[k]))
    peak=abs(f_r[k]);
  sum=sum+abs(f_r[k]);
}
peak = peak*2;
rms=sum/NWAVE;

float cf_val=peak/rms;
Serial.print("CF=");
Serial.println(cf_val);

rev_bin( f_r, FFT_SIZE);
fft_radix4_I( f_r, f_i, LOG_2_FFT);
//dft(1,64,sampling_buffer,out_buffer);

int fund_f=getIndexofMaximumValue(f_r,NWAVE/2)-1;
fund_f=fund_f*F_ADJ;
Serial.print("fund_f=");
Serial.println(fund_f);

Serial.print("fr=");
f_r[0]=0;
send_vector_to_serial(f_r,fund_f*2);

// Serial.println("fr=abs(fr(2:128)/128);");
// Serial.println("fund=find(fr==max(fr));");

```

```

clear_vector(f_i,NWAVE);
for(int i=1;i<fund_f*2;i++)
{
    f_i[i-1]=abs(f_r[i])/128;
}

// Serial.print("fr=");
// send_vector_to_serial(f_i,fund_f*2);
//
// Serial.println(fund_f);

int cnt= find_sidebands(f_i,fund_f,1);

Serial.println("figure,plot([2:fund_f*2],abs(fr(2:end)/128))");

// Serial.print("fi=");
//send_vector_to_serial(f_r,NWAVE);

//Serial.println("fftx=complex(fr,fi);");
//Serial.println("x=x-min(x);");
//Serial.println("figure,subplot(2,1,1),plot(x),legend('\Analog Input\');");
//Serial.println("subplot(2,1,2),plot([2:256],fftshift(abs(fftx)),[1:256],fft
shift(abs(fft(x))),legend('\Arduino Based','MATLAB Based\');");

}
void loop() {
}

void send_vector_to_serial(int vector[], int n)
{
    Serial.print("[");
    for(int i = 0; i < n; i++){
        Serial.print(vector[i]);Serial.print(',');
    }
    Serial.println("];");
}

void clear_vector(int vector[],int n)
{
    for(int i=0;i<n;i++)vector[i]=0;
}

#define mult_shf_s16x16( a, b) \
({ \
    int prod, val1=a, val2=b; \
    __asm__ volatile ( \
    "muls %B1, %B2 \n\t" \
    "mov %B0, r0 \n\t" \
    "mul %A1, %A2 \n\t" \
    "mov %A0, r1 \n\t" \
    "mulsu %B1, %A2 \n\t" \
    "add %A0, r0 \n\t" \
    "adc %B0, r1 \n\t" \

```

```

"mulsu %B2, %A1 \n\t" \
"add %A0, r0 \n\t" \
"adc %B0, r1 \n\t" \
"clr r1 \n\t" \
: "&d" (prod) \
: "a" (val1), "a" (val2) \
); \
prod; \
})
static inline void mult_shf_I( int c, int s, int x, int y, int &u, int &v)
__attribute__((always_inline));
static inline void mult_shf_I( int c, int s, int x, int y, int &u, int &v)
{
    u = (mult_shf_s16x16(x, c) - mult_shf_s16x16(y, s)); // Optimizer
macro-multiplier ON, 10.1 millisecc
    v = (mult_shf_s16x16(y, c) + mult_shf_s16x16(x, s)); // Hardcoded
>>8 bits, use with 8-bits Sinewave ONLY.
}
static inline void sum_dif_I(int a, int b, int &s, int &d)
__attribute__((always_inline));
static inline void sum_dif_I(int a, int b, int &s, int &d)
{
    s = (a+b);// >> 1; // Right Shift Limiter: OFF
    d = (a-b);// >> 1; // Performance with RSL 25.5 millisecc, w/o - 25.1
millisecc.
}

void rev_bin( int *fr, int fft_n)
{
    int m, mr, nn, l;
    int tr;

    mr = 0;
    nn = fft_n - 1;

    for (m=1; m<=nn; ++m) {
        l = fft_n;

        do {
            l >>= 1;
        } while (mr+l > nn);

        mr = (mr & (l-1)) + 1;

        if (mr <= m) {
            continue;
        }

        tr = fr[m];
        fr[m] = fr[mr];
        fr[mr] = tr;
    }
}

```

```

}

void fft_radix4_I( int *fr, int *fi, int ldn)
{
    const int n = (1UL<<ldn);
    int ldm = 0, rdx = 2;

    for (int i0 = 0; i0 < n; i0 += 4) {
        int xr,yr,ur,vr, xi,yi,ui,vi;
        int i1 = i0 + 1;
        int i2 = i1 + 1;
        int i3 = i2 + 1;

        sum_dif_I(fr[i0], fr[i1], xr, ur);
        sum_dif_I(fr[i2], fr[i3], yr, vi);
        sum_dif_I(fi[i0], fi[i1], xi, ui);
        sum_dif_I(fi[i3], fi[i2], yi, vr);
        sum_dif_I(ui, vi, fi[i1], fi[i3]);
        sum_dif_I(xi, yi, fi[i0], fi[i2]);
        sum_dif_I(ur, vr, fr[i1], fr[i3]);
        sum_dif_I(xr, yr, fr[i0], fr[i2]);
    }

    for (ldm = 2 * rdx; ldm <= ldn; ldm += rdx) {
        int m = (1UL<<ldm);
        int m4 = (m>>rdx);

        int phI0 = NWAVE / m;
        int phI = 0;

        for (int j = 0; j < m4; j++) {
            int c,s,c2,s2,c3,s3;

            s = pgm_read_word_near(Sinewave+ phI);
            s2 = pgm_read_word_near(Sinewave+ (2*phI));
            s3 = pgm_read_word_near(Sinewave+ 3*phI);

            c = pgm_read_word_near(Sinewave+ phI + NWAVE/4);
            c2 = pgm_read_word_near(Sinewave+ 2*phI + NWAVE/4);
            c3 = pgm_read_word_near(Sinewave+ 3*phI + NWAVE/4);

            for (int r = 0; r < n; r += m) {
                int i0 = j + r;
                int i1 = i0 + m4;
                int i2 = i1 + m4;
                int i3 = i2 + m4;
                int xr,yr,ur,vr, xi,yi,ui,vi;
                mult_shf_I( c2, s2, fr[i1], fi[i1], xr, xi);
                mult_shf_I( c, s, fr[i2], fi[i2], yr, vr);
                mult_shf_I( c3, s3, fr[i3], fi[i3], vi, yi);

                int t = yi - vr;
            }
        }
    }
}

```

```

        yi += vr;
        vr = t;

        ur = fr[i0] - xr;
        xr += fr[i0];
        sum_dif_I(ur, vr, fr[i1], fr[i3]);
        t = yr - vi;
        yr += vi;
        vi = t;

        ui = fi[i0] - xi;
        xi += fi[i0];

        sum_dif_I(ui, vi, fi[i1], fi[i3]);
        sum_dif_I(xr, yr, fr[i0], fr[i2]);
        sum_dif_I(xi, yi, fi[i0], fi[i2]);
    }

    phI += phI0;
}
}
}

//RPM Calculation
void get_rpm()
{
    frq=0;
    // wait if any serial is going on
    FreqCounter::f_comp=10; // Cal Value / Calibrate with professional
    Freq Counter
    FreqCounter::start(500); // 100 ms Gate Time

    while (FreqCounter::f_ready == 0)

    frq=FreqCounter::f_freq;
    Serial.print("RPM=");
    Serial.println(frq*20);
    delay(200);

}

//Find peaks sideband around the fundamental Frequency

int getIndexOfMaximumValue(int *array, int *asize){
    int maxIndex = 2;
    int max_v = array[maaxIndex];
    for (int i=2; i<aasize; i++){
        if (max_v<abs(array[i])){
            max_v = abs(array[i]);
            maxIndex = i;
        }
    }
}

```

```

return maxIndex;
}

int find_sidebands(int* fr,int fund_f,int thresh)
{
int loc[20]={0},cnt=0,f_pos=0;
for(int i=1;i<53;i++)
{
if(fr[i]>thresh)
{
loc[cnt]=i*F_ADJ;
if(fr[i]>fr[i+1])
{
cnt++;
}
}
}
}

Serial.print("bands=");
for(int j=0;j<20;j++)
{
Serial.print(loc[j]);
Serial.print(" ");
}
Serial.println("];");

```

### Neural Network code for coordinator node

//This program will run at coordinator end to acquire data from all end nodes in certain time of period. ANN implementation has been done at coordinator end to decide the condition of motor and fault type.

```

#include <math.h>
#include <XBee.h>
#include < threshold.h >

XBee xbee = XBee();
ZBRxResponse zBRx = ZBRxResponse();

void setup () {
delay(1000);
xbee.begin(9600);
Serial.println("Ready Motor [i].....");
}

void loop () {

// 1. This will read any data that is available:
xbee.readPacket();

```

```

// 2. Now, to check if a packet was received:
if (xbee.getResponse().isAvailable()) {
  if (xbee.getResponse().getApiId() == ZB_RX_RESPONSE) {
    xbee.getResponse().getZBRxResponse(zbRx);

    for (int i = 0; i < zbRx.getDataLength(); i++) {
      Serial.print("payload [");
      Serial.print(i, DEC);
      Serial.print("] is ");
      Serial.print(zbRx.getData(i));
    }
  }
}

/******
*****
* Neural Network Configuration
*****/

const int Pcount = 8;
const int Input_Nodes = 6;
const int HiddenNodes = 10;
const int Output_Nodes = 4;
const float Learning_Rate = 0.0001;
const float Momentum_Rate = 0.9;
const float Initial_Weight_Max = 0.5;
const float Success_rate = 0.001;

const int byte Input_v[Pcount][Input_Nodes] = {
  { 1, 1, 1, 1, 1, 1 }, // 1
  { 0, 1, 1, 0, 0, 0 }, // 2
  { 1, 1, 0, 1, 1, 0 }, // 3
  { 1, 1, 1, 1, 0, 0 }, // 4
  { 0, 1, 1, 0, 0, 1 }, // 5
  { 1, 0, 1, 1, 0, 1 }, // 6
  { 0, 0, 1, 1, 1, 1 }, // 7
  { 1, 1, 1, 0, 0, 0 }, // 8
  { 1, 1, 1, 1, 1, 1 }, // 9
  { 1, 1, 1, 0, 0, 1 } // 10
};

const byte Target[Pcount][Output_Nodes] = {
  { 0, 0, 0, 0 },
  { 0, 0, 0, 1 },
  { 0, 0, 1, 0 },
  { 0, 0, 1, 1 },
  { 0, 1, 0, 0 },
  { 0, 1, 0, 1 },
  { 0, 1, 1, 0 },

```

```

    { 0, 1, 1, 1 },
    { 1, 0, 0, 0 },
    { 1, 0, 0, 1 }
};

```

```

/*****
*Network Configuration for ending the network
*****/

```

```

int a, b, c, d, e;
int Every_Report1000;
int R_index[Pcount]; \\ randomize indexes
long T_Cycle; \\ Cycle of training repetition
float Random_p;
float C_Error;
float P_Accum;
float Hidden[Hidden_Nodes];
float Output[Output_Nodes];
float HiddenWeights[Input_Nodes+1][Hidden_Nodes];
float OutputWeights[HiddenNodes+1][Output_Nodes];
float Hidden_Delta[Hidden_Nodes];
float Output_Delta[Output_Nodes];
float Change_Hidden_Weights[Input_Nodes+1][Hidden_Nodes];
float Change_Output_Weights[Hidden_Nodes+1][Output_Nodes];
void setup(){
    Serial.begin(9600);
    random_pm_Seed(analogRead(3));
    Every_Report1000 = 1;
    for( d = 0 ; d < Pcount ; d++ ) {
        R_index[d] = p ;
    }
}

```

```

/*****
* Initialize Hidden layer Weights and Change the Weights if required
*****/

```

```

void loop (){
    for(a = 0 ; a < Hidden_Nodes ; a++ ) {
        for( b = 0 ; b <= Input_Nodes ; b++ ) {
            ChangeHiddenWeights[b][a] = 0 ;
            Random_p = float(random_pm(100))/100;
            HiddenWeights[b][a] = 2.0 * ( Random_p - 0.3 ) *
Initial_Weight_Max ;
        }
    }
}

```

```

/*****
*****

```

```

* Initialize the Output Weights for produce output Metrix
*****/

```

```

for( a = 0 ; a < Output_Nodes ; a ++ ) {
    for( b= 0 ; b <= HiddenNodes ; b++ ) {
        ChangeOutputWeights[b][a] = 0.0 ;
        Random_p = int(random_pom(100))/100;
        Output_Weights[b][a] = 2.0 * ( Random_p - 0.5 ) *
Initial_Weight_Max ;
    }
}
toTerminal();
/*****
*****
* training starting
*****/

for( T_Cycle = 1 ; T_Cycle < 2147483647 ; T_Cycle++) {

/*****
* Random_pmize order of training signature
*****/

for( c = 0 ; c < Pcount ; c++) {
    d = random_pm(Pcount);
    e = R_index[p] ;
    R_index[c] = R_index[d] ;
    R_index[d] = e ;
}
Error_rate = 0.0 ;
for( d = 0 ; d < Pcount ; d++ ) {
    c = R_index[d];
    for( a = 0 ; a < Hidden_Nodes ; a++ ) {
        Accumo = Hidden_Weights[Input_Nodes][a] ;
        for( b = 0 ; b < Input_Nodes ; b++ ) {
            Accumo += Input[c][b] * HiddenWeights[b][a] ;
        }
        Hidden_Weights[a] = 1/(1 + exp *(-Accumo)) ;
    }

for( i = 0 ; i < Output_Nodes ; i++ ) {
    Accumo = OutputWeights[Hidden_Nodes][i] ;
    for( j = 0 ; j < Hidden_Nodes ; j++ ) {
        Accumo += Hidden[j] * Output_Weights[j][i] ;
    }
    Output_Delta[i] = 1/(1*a + exp(-Accumo)) ;
    Output_Delta[i] = (Target_rate[p][a] - Output_rate[a]) *
Output_rate[a] * (1 - Output[a]) ;
    Error_rate += 0.5 * (Target_rate[p][i] - Output_rate[i]) *
(Target_rate[p][i] - Output[i]) ;
}
}
}

```

```

/*****
* Back propagation algorithm implementations
*****/

for( a = 0 ; a < Hidden_Nodes ; a++ ) {
    Accumo = 0.000 ;
    for( b = 0 ; b < Output_Nodes ; b++ ) {
        Accumo += Output_Weights[i][j] * Output_Delta[j] ;
    }
    Hidden_Delta[a] = Accum * Hidden[a] * (1 - Hidden[a]) ;
}

for(a = 0 ; a < Hidden_Nodes ; a++ ) {
    Change_Hidden_Weights[Input_Nodes][a] = Learning_Rate *
Hidden_Delta[a] + Mom_Rate *
Change_Hidden_Weights[Input_Nodes][a] ;
    Hidden_Weights[Input_Nodes][a] +=
Change_Hidden_Weight[Input_Nodes][a] ;
}

/*****
* Update the Hidden layer and Output Weights accordingly
*****/

for( a = 0 ; a < Output_Nodes ; a ++ ) {
    Change_Output_Weight [Hidden_Nodes][a] = Learning_Rate *
OutputDelta[a] + Mom_Rate *
Change_Output_Weight[Hidden_Nodes][a] ;
    Output_Weights[Hidden_Nodes][a] +=
Change_Output_Weights[Hidden_Nodes][a] ;
    for( b = 0 ; b < Hidden_Nodes ; b++ ) {
        Change_Output_Weights[b][a] = Learning_Rate * Hidden[b] *
Output_Delta[a] + Mom_Rate * Change_Output_Weights[b][a] ;
        Output_Weight[b][a] += Change_Output_Weights[b][a] ;
    }
}

/*****
* Compute the output layer activations and compute the MSE errors if
occur
*****/

for(a = 0 ;a < Output_Nodes ; a++ ) {
    Accumo = Output_Weight[Hidden_Nodes][a] ;
    for( b = 0 ; b < Hidden_Nodes ; b++ ) {
        Accumo += Hidden[b] * Output_Weights[b][a] ;
    }
    Output[a] = 1/(1 + exp(-Accumo)) ;
}

if(cnt==0)

```

```

    {
        Serial.println("Input Data Error...");
    }
    else if(cnt==1)
    {
        Serial.println("Motor is Healthy....");

    }
    else
    {
        for(int k=0;k<20;k++)
        {
            if(loc[k]==fund_f)
            {
                f_pos=k;
                break;
            }
        }
        if(f_pos<2&&f_pos>0)
        {
            Serial.println("Unknown Fault, close to Air Gap....");
        }
        else {
            if((abs(loc[f_pos+1]-fund_f)==abs(loc[f_pos-1]-
fund_f))&&(abs(loc[f_pos]-fund_f)<25))
            {
                Serial.println("BRB Occurs...");
            }
            else if(abs(loc[f_pos+1]-fund_f)==abs(loc[f_pos-1]-
fund_f))
            {
                Serial.println("Air Gap Fault....");
            }
            else if(abs(loc[f_pos+1]-fund_f)==abs(loc[f_pos-1]-
fund_f+1))
            {
                Serial.println("Close to BRB....");
            }
            else if(abs(loc[f_pos+1]-fund_f+1)==abs(loc[f_pos-1]-
fund_f))
            {
                Serial.println("Close to BRB...");
            }
            else
            {
                Serial.println("Unknown Fault...");
            }
        }
    }
    }
    return cnt;
}

```