An Intelligent Embedded System For Condition Monitoring Of Industrial Drives

A Project Report

Submitted by

G. RAYAPPAN - 71205415006

in partial fulfillment for the award of the degree
of

Master of Engineering
in
Power Electronics and Drives

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE – 641 006

ANNA UNIVERSITY: CHENNAI 600 025

JULY 2007

BONAFIDE CERTIFICATE

Certified that this project report entitled **"An intelligent embedded system for condition monitoring of industrial drives"** is the bonafide work of

Mr. G.Rayappan Register No. 71205415006

Who carried out the project work under my supervision.

Signature of the Head of the Department
Prof.K.Regupathy Subramanian

Signature of the Guide
Dr. V. Duraisamy

Certified that the candidate with university Register No:.71205415006 was examined in Project viva- voce examination held on 4.-.07.-.2007

Internal Examiner

K.J. Ramachandran
External Examiner 4/7/07

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY COIMBATORE - 641 006

National Conference on

INTELLIGENT COMPUTING TECHNIQUES FOR REGULATED AND DE-REGULATED POWER SYSTEMS (INCOT 07)

Organised by

School of Electrical Sciences

Department of Electrical & Electronics Engineering

Karunya UNIVERSITY

(Estd under sec. 3 of the UGC Act 1956)

Certificate

This is to certify that

Dr/Mr/~~Mrs~~/~~Ms~~ G. RAYAPPAN

KUMARAGURU COLLEGE OF TECHNOLOGY - COIMBATORE

participated / presented

a paper titled AN INTELLIGENT EMBEDDED SYSTEM FOR ONLINE CONDITION MONITORING OF INDUSTRIAL DRIVES *in the National Conference On 31st March 2007.*

Co-author(s):

Prof. P.V. John	Prof. Thomas Joseph	Dr. P.K. Philip
CONVENOR	DIRECTOR	VICE- CHANCELLOR

ABSTRACT

Electric motors play a very important part in safe and efficient running of any industrial plant. Early detection of abnormalities in the motor would help to avoid costly breakdowns. The bearings play an important role in the reliability and performance of all motor systems. Majority of faults in motors are due to bearing failure. The results of many studies show that bearing problem account for over 40% of all motor failures. Hence, it is essential to check motor condition from time to time.

This project aims at the implementation of motor current signature analysis for detecting the bearing fault in squirrel cage induction motor. This proposed scheme monitors the stator current spectrum to detect the bearing faults and to extract fault signature by using Fast Fourier Transform analyzer. For fault diagnosis, neural network and fuzzy logic techniques are used. The fuzzy logic based fault detection scheme is implemented in real time. The experimental results are presented. Since this scheme detects the faults at their earlier stage, the maintenance can be carried out in an organized manner, which reduces the down time and repairing cost. This approach is validated in a 1 HP 415V 50HZ 960-rpm three phase induction motor.

ஆய்வுச்சுருக்கம்

மின் மோட்டார்கள் தொழிற்சாலையின் பாதுகாப்பு மற்றும் தரமான இயக்கத்திற்கு முக்கிய பங்கு வகிக்கின்றன. மின் மோட்டார்களில் வரும் பழுதுகளை முன் கூட்டியே அறிந்து கொண்டால் மிகப்பெரிய சேதாரத்தை தடுக்க முடியும். மோட்டார்களில் பேரிங்குகள் முக்கிய பங்கு வகிக்கின்றன. பெரும்பான்மையான பழுதுகள் பேரிங் பழுது மூலமே வருகிறது. 40 விழுக்காடு பழுதுகள் பேரிங் பழுது மூலமே வருகிறது. இதனை ஆய்வுகள் மற்றும் ஆராய்ச்சிகள் மூலம் உறுதி செய்யப்பட்டுள்ளது. எனவே மோட்டார்களை மேற்பார்வையிடுவது முக்கியத் தேவையாகும்.

இந்த ஆய்வு பேரிங் பழுதுகளை மேற்பார்வையிட மோட்டார் கரண்ட் சிக்னேச்சர் (MCSA) முறையானது பயன்படுத்தப்பட்டுள்ளது. இந்த முறை மோட்டார் மின் அதிர்வுகளை கண்டறிவதன் மூலம் பேரிங் பழுதுகளை அறிய முடிகிறது. இதற்கு பாஸ்ட் ஸ்போரியர் மாற்றுதல் முறை பயன்படுத்தப்பட்டுள்ளது. மேலும் பழுதுகளை ஆய்வு செய்ய ஃநியூரல் வலைப்பின்னர் மற்றும் ஃபஸ்ஸி முறையானது பயன்படுத்தப்பட்டுள்ளது. மேலும் ஃபஸ்ஸி முறையானது நடைமுறையில் செயல்படுத்தப்பட்டுள்ளது. இதன் மூலம் பழுதுகளை முன் கூட்டியே அறிய முடிகிறது. மேலும் இந்த ஆய்வு ஆய்வகத்தில் ஒரு குதிரை திறன் கொண்ட மோட்டாரில் செயல்படுத்தப்பட்டு ஆய்வு முடிவுகள் தரப்பட்டுள்ளன.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowns all efforts with success.

I would like to express our heartful thanks to our beloved Principal **Dr. Joseph V. Thanikal**, B.E., M.E., Ph.D., PDF, CEPIT, for his support.

I take pleasure in thanking **Prof. K. Regupathy Subramanian**, B.E (Hons), M.Sc., MIEEE, IES, Head of the Department, Electrical and Electronics Engineering, Kumaraguru College of Technology, for his constant encouragement

My sincere thanks to guide **Dr. V. Duraisamy**, M.E., Ph.D., MISTE, AMIE, MIEEE, Professor, Department of Electrical and Electronics Engineering, Kumaraguru College of Technology, who guided me throughout the project and encouraged successfully to complete my work.

I am also grateful to the dynamic support of all our staff and friends, who helped me in many ways during the course of this project.

CONTENTS

Title	Page No.
Bonafide Certificate	ii
Proof publishing a paper in the conference	iii
Abstract in English	iv
Abstract in Tamil	v
Acknowledgement	vi
Contents	vii
List of Tables	x
List of Figures	xi
Abbreviations & symbols	xii

CHAPTER 1 INTRODUCTION

1.1 BEARING FAULTS IN INDUCTION MACHINE	1
1.2 NEED FOR MONITORING SYSTEM	2
1.3 OBJECTIVE	2
1.4 BEARING STRUCTURAL DEFECTS	2
1.5 LITERATURE SURVEY	3
1.6 ORGANIZATION OF THE THESIS	5

CHAPTER 2 BEARING FAULT DETECTION

2.1. PROBLEM DEFINITION	6
2.2 FAULT DETECTION SCHEME	8
2.3 EXPRIMENTAL SETUP FOR DATA ACQUISTION	9
2.4 MOTOR CURRENT SIGNATURE ANALYSIS (MCSA)	10
2.5 EXPRIMENTAL RESULTS	11

CHAPTER 3 NEURAL NETWORK BASED FAULT DIAGNOSIS

3.1 INTRODUCTION TO NEURAL NETWORK	14
3.1.1 NEURON MODEL	15
3.1.2 ACTIVATION FUNCTIONS	16
3.1.3 LEARNING RULES	18

3.2 BACK-PROPAGATION NEURAL NETWORK 18
3.2.1 CHOICE OF PARAMETERS FOR NETWORK TRAINING 21
3.2.2 LEARNING RATE 21
3.2.3 MOMENTUM FACTOR 22
3.3 STRUCTURE OF BP NETWORK FOR FAULT DETECTION 22
3.4 SIMULATION RESULTS 23

CHAPTER 4 FUZZY LOGIC BASED FAULT DIAGNOSIS
4.1 INTRODUCTION 25
4.2 MAMDANI FUZZY LOGIC INFERENCE SYSTEM 26
4.2.1 FUZZIFIER 27
4.2.2FUZZYRULES 28
4.2.3 INFERENCE ENGINE 29
4.2.4 DEFUZZIFIER 30
4.3 FUZZY LOGIC BASED FAULT DETECTION METHOD 30
4.4 SIMULATION RESULTS 33
4.5 COMPARISON OF NEURAL NETWORK AND FUZZY BASED FAULT DIAGNOSIS SYSTEM 33

CHAPTER 5 HARDWARE IMPLEMENTATION
5.1 SCHEMATIC DIAGRAM 36
5.2HARDWARE DESCRIPTION 37
5.3 VOLTAGE REGULATORS 38
5.4 VOLTAGE MEASUREMENT 38
5.5 CURRENT MEASUREMENT 39
5.6 SPEED MEASUREMENT 39
5.7 PIC INTERFACE 40
5.8 LCD DISPLAY 41
5.9 LCD MODULE – PIN DETAILS 42
5.10ALARM CIRCUIT 43
5.11 MICROCONTROLLER 43
5.11.1 NTRODUCTION TO PIC 44
5.11.2 PIC 16F877 44
5.11.3 SPECIAL FEATURES OF PIC MICROCONTROLLER 44

5.11.4 PERIPHERAL FEATURES 45
5.11.5 PIC 16F877 - PIN CONFIGURATION 46
5.12 I/O PORTS 46
5.12.1 PORT A AND TRISA REGISTER 46
5.12.2 PORT B AND TRISB REGISTER 47
5.12.3 PORT C AND THE TRISC REGISTER 48
5.12.4 PORT D AND TRISD REGISTERS 48
5.12.5 PORT E AND TRISE REGISTER 48
5.13 SOFTWARE DESCRIPTION
5.13.1 ALGORITHM FOR PIC PROGRAMMING 48
5.13.2 FUZZY BASED PROGRAMMING 51

CONCLUSION **53**

REFERENCES **54**

APPENDIX A **56**

APPENDIX B **64**

LIST OF TABLES

Table	Title	Page No.
2.1	Rated parameters of the machine under test	9
2.2	Experimental Results of the defective Shaft end bearing	13
2.3	Experimental Results of the defective Fan end bearing	13
3.1	Results for neural network based fault diagnosis	23
4.1	Fuzzy Rules	31
4.2	Results for Fuzzy logic based fault diagnosis	33
4.3	Comparison of Neural Network and Fuzzy Based Fault Diagnosis	34
5.2	Pin details of the LCD Module	42

LIST OF FIGURES

Figure	Title	Page No.
1.1	Faults in Induction motors	1
1.2	Misalignment of the bearing	3
2.1	Dimensions of a ball bearing	7
2.2	Single phase stator current monitoring scheme	8
2.3	Testing equipment & Experimental setup	9
2.4	Bearings drilled with holes	10
2.6	Current spectrum for healthy machine	11
2.8	Current spectrum for faulty machine with shaft end	12
2.10	Current spectrum for faulty machine fan end	12
3.1	Structure of biological neuron	14
3.2	Single – Input Neuron without Bias	15
3.3	Single Input Neuron with Bias	16
3.4	Hard Limit Activation Function	16
3.5	Linear Activation Function	17
3.6	Log-Sigmoid Activation Function	17
3.7	Architecture of BP Neural Network	19
3.8	Structure of BP Network for Fault Detection	23
3.9	Epoch Vs Error Characteristics	24
4.1	Mamdani Fuzzy Logic Inference Systems	26
4.2	Triangular membership functions	28
4.3	Graphical interpretation of fuzzification, inference	29
4.4	Height Defuzzification Method	30
4.5	Input Membership Functions for speed	32
4.6	Input Membership Functions for current	32
4.7	Output Membership Functions for diagnosis	32

4.8	Surface Viewer	33
5.1	Schematic diagram	36
5.2	+5 V Power supply	37
5.4	Voltage measurement circuit	38
5.5	Current measurement circuit	39
5.6	Speed measurement circuit	40
5.8	Alarm circuit	43
5.11	Experimental results fuzzy logic based bearing fault detection	52

ABBREVIATIONS AND SYMBOLS

ABBREVIATIONS

AC	alternating current
CT	current transformer
FFT	fast fourier transform
FIS	fuzzy inference system
HP	horse power
Hz	hertz
I / O	input / output
LCD	liquid crystal display
MCSA	motor stator current signature analysis
PC	personal computer
PIC	peripheral interface controller
rpm	revolutions per minute
RS	recommended standard
V	voltage

SYMBOLS

f_{bng}	characteristic frequency of the bearing
f_e	electrical supply frequency
m	integral multiplier
f	frequency
f_v	characteristic vibration frequency
f_o	characteristic outer race frequency
f_i	characteristic inner race frequency
f_b	characteristic ball frequency
f_c	characteristic cage frequency

N	number of rollers or balls
f_r	rotational frequency of rotor
d	ball diameter
D	pitch diameter
e	sliding factor of rolling element
p	number of pole pairs
α	contact angle of rolling element
b	ball
bng	bearing
c	cage
e	electrical supply
i	inner
o	outer
r	rotor
v	vibration

CHAPTER 1
INTRODUCTION

The simple, robust design and construction of AC induction motor have encouraged their successful in industry for many years. However, these motors are required to operate in highly corrosive and dusty environments. These factors coupled with the natural aging process of any motor make the motor subject to faults. These faults if undetected, contribute to the degradation and eventual failure of the motors. As it is not economical to introduce redundant backup motors, condition monitoring for induction motor is important for safe operation. In order to keep the motor condition, techniques such as fault monitoring, detection, classification and diagnosis have become increasingly essential. Earlier detection of the fault reduces repair cost and motor outage time thereby improving safety.

In general, condition monitoring schemes have concentrated on specific failures modes in one of three phase induction motor components: the stator, the rotor, or bearings. Even though thermal and vibration monitoring have been utilized for decades, most of the recent research has been directed toward electrical monitoring of the motor with emphasis on inspecting the stator current of the motor.

1.1 BEARING FAULTS IN INDUCTION MOTORS

Bearings play an important role in the reliability and performance of all motor systems. In addition, most faults arising in motors are often linked to bearing faults. The result of many studies show that bearing problems account for over 40% of all machine failures. The several faults and its percentage are shown in the Figure 1.1

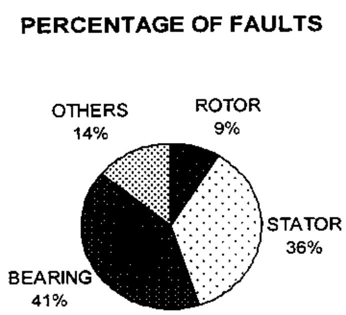


Figure 1.1 Faults in induction motor

1.2 NEED FOR MONITORING SYSTEM

Machine condition monitoring is gaining importance in industry because of the need to increase reliability and to decrease the possibility of production loss due to machine breakdown. By comparing the signals of a machine running in normal and faulty conditions, detection of faults like mass unbalance, rotor rub, shaft misalignment, gear failures, and bearing defects is possible. These signals can also be used to detect the incipient failures of the machine components, through the online monitoring system, reducing the possibility of catastrophic damage and the downtime. Although often the visual inspection of the frequency domain features of the measured signals is adequate to identify the faults, there is a need for a reliable, fast, and automated procedure of diagnostics. Artificial intelligence techniques like Neural Fuzzy techniques can be implemented in the system for automated detection and diagnosis of machine conditions.

1.3 OBJECTIVE

To design and implement an intelligent embedded system for condition monitoring of industrial drives at their earlier stage.

1.4 BEARING STRUCTURAL DEFECTS

Rolling element bearings generally consist of two rings, an inner and an outer race, between which a set of balls or rollers rotate in raceways. Under normal operating conditions of balanced load and good alignment, fatigue failure begins with small fissures, located between the surface of the raceway and the rolling elements, which gradually propagate to the surface generating detectable vibrations and increasing noise levels. Continued stress causes fragments of the material to break loose, producing a localized fatigue phenomenon known as flaking or spalling. Once started, the affected area expands rapidly contaminating the lubricant and causing localized overloading over the entire circumference of the raceway. Eventually, the failure results in rough running of the bearing. While this is the normal mode of failure in rolling element bearings, there are many other conditions which reduce the time to bearing failure. These external sources include contamination, corrosion, improper lubrication, improper installation or brinelling.

Contamination and corrosion frequently accelerate bearing failure because of the harsh environments present in most industrial settings. Dirt and other foreign matter that is commonly present often contaminate the bearing lubrication.

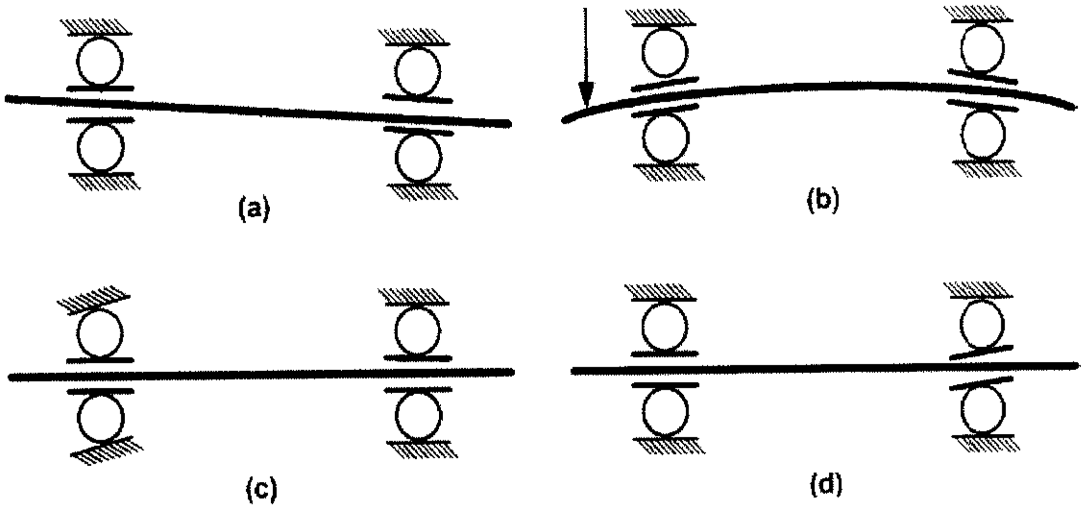


Figure1.2 Misalignment of the bearing

(a) Misalignment (out-of-line), **(b)** Shaft deflection,
(c) Crooked or tilted outer race, **(d)** Crooked or tilted inner race.

Bearing corrosion is produced by the presence of water, acids, deteriorated lubrication and even perspiration from careless handling during installations. Improper lubrication includes both under- and over-lubrication. In either case, the rolling elements are not allowed to rotate on the designed oil film causing increased levels of heating. The excessive heating causes the grease to break down, which reduces its ability to lubricate the bearing elements and accelerates the failure process. Installation problems are often caused by improperly forcing the bearing onto the shaft or in the housing. This produces physical damage in the form of brinelling or false brinelling of the raceways which leads to premature failure. Misalignment of the bearing, which occurs in the four ways depicted in Figure 1.2, is also a common result of defective bearing installation. The most common of these is caused by tilted races. Brinelling is the formation of indentations in the raceways as a result of deformation caused by static overloading.

1.5 LITERATURE SURVEY

Induction Motors are a critical component of many industrial processes and are frequently integrated in commercially available equipment and industrial processes. Motor-driven equipment often provides core capabilities essential to business success and to safety of equipment and personal. There are many published techniques and many commercially available tools to monitor induction motors to insure a high degree of reliability uptime. In spite of these tools, many companies are still faced with unexpected system failures and reduced motor lifetime. Environmental, duty, and installation issues may combine to accelerate motor failure far sooner than the designed motor lifetimes. These studies specifically apply to machines, which are operated in industrial and commercial installations. The results of these studies show that bearing problems account for over 40% of all machines failures. Over the fast several decades, rolling-element (ball and roller) bearings have been utilized in many electric machines while sleeve (fluid-film) bearings are installed in only the largest industrial machines. In the case of induction motors, rolling element bearings are overwhelmingly used to provide rotor support. (Kryter et al 1989).

In general, condition monitoring schemes have concentrated on sensing specific failures modes in one of three phase induction motor components: the stator, the rotor, or the bearings. Even though thermal and vibration monitoring have been utilized for decades, most of the recent research has been directed toward electrical monitoring of the motor with emphasis on inspecting the stator current of the motor. In particular, a large amount of research has been directed toward using the stator current spectrum to sense rotor faults associated with broken rotor bars and mechanical unbalance. (Cardoso et al1993)

All of the presently available techniques require the user to have some degree of expertise in order to distinguish a normal operating condition from a potential mode. This is because the monitored spectral components (either vibration or current) can result from a number of sources, including those related to normal operating conditions (schoen et al 1995).This requirement is even more acute when analyzing the current spectrum of an induction motor since a multitude of harmonics exist due to both the design and construction of the motor and the variation in the load torque which are not related to the health of motor typically have exactly the same effect on the load current. Therefore, systems to eliminate induction motors arbitrary load effects in currents –based monitoring (schoen et al 1997).

Penman et al (1986) suggested the condition monitoring of the dynamic performance of electrical drives received considerable attention in recent years. Many condition monitoring methods have been proposed for different type of rotating machine faults detection and localization. In fact large electro machine systems are often equipped with mechanical sensors, primarily vibration sensors based on proximity probes. Those, however, are delicate and expensive. Moreover, in many situations, vibration-monitoring methods are utilized to detect the presence of an incipient bearing failure. However, in Steele (1982) said that the stator current monitoring can provide the same indications without requiring access to the motor. This thesis demonstrates the feasibility of bearing detection by correlating the characteristic bearing frequencies to the spectral components of the stator current.

1.6 ORGANIZATION OF THE THESIS

This report presents about the various structural defects occurring in a bearing and different techniques which can be implemented to detect those faults at their earlier stage.

Chapter 1 introduces about the fundamental structure of a bearing, modes of failure and the necessity of an online monitoring system. The methodology and the details about the induction motor taken for study and experimentation, the results of the experimental study and the block diagram has been presented in the Chapter2. Neural network based fault diagnosis are described in Chapter 3. Fuzzy logic based fault diagnosis are described in Chapter4 .Hardware implementation and its description are given in the Chapter 5.

CHAPTER 2

BEARING FAULT DETECTION

2.1. PROBLEM DEFINITION

The relationship of the bearing vibration to the stator current spectrum can be determined by remembering that any air-gap eccentricity produces anomalies in the air-gap flux density. In the case of a dynamic eccentricity that varies with rotor position, the oscillation in the air-gap length causes variations in the air-gap flux density. This variation affects the inductance of the machine producing stator current harmonics. Since ball bearings support the rotor, any bearing defect produces a radial motion between the rotor and the stator of the machine. The cause of air-gap eccentricity, these variations generate harmonic stator currents at predictable frequencies, related to the vibration and electrical supply frequencies by

$$f_{bng} = |f_e \pm m \cdot f_v|. \qquad (2.1)$$

Where $m = 1, 2, 3, \ldots, etc$ and fv is one of the characteristic vibration frequencies.

The characteristic frequency of bearing failure (bearing pass frequency) is the inverse number of the time between occurrences of bearing impulses. This frequency can be calculated by the aid of known bearing geometry and rotational speed. The dimensions of a bearing are given in the Figure2.1. An outer race defect causes impulse when ball or roller passes the defected area of race. The theoretical frequency is thus

$$f_o = \frac{N}{2} \cdot f_r \cdot \left(1 - \frac{d}{D} \cdot \cos\alpha\right), \qquad (2.2)$$

where N is the number of balls or rollers, fr is the rotational speed of rotor, d is the diameter of the ball, D is the pitch diameter, α is a contact angle of rolling element. The ball pass frequency of defect on inner race is

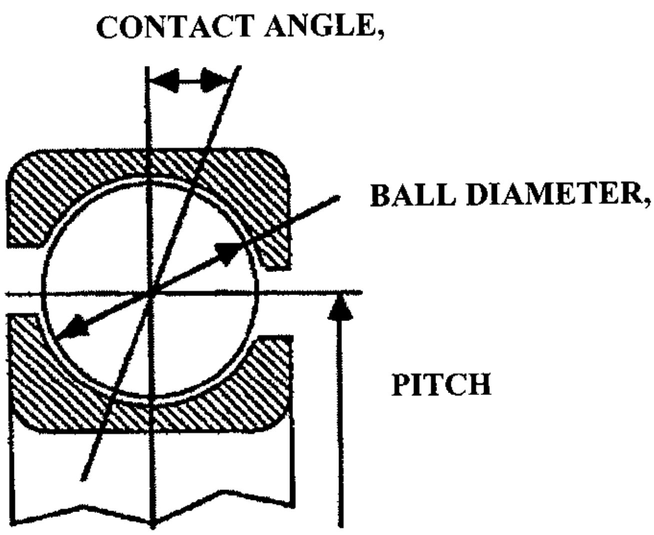


Figure 2.1 Dimensions of a Ball Bearing

$$f_i = \frac{N}{2} \cdot f_r \cdot (1 + \frac{d}{D} \cdot \cos\alpha). \tag{2.3}$$

The ball spin frequency is

$$f_b = \frac{D}{2d} \cdot f_r \cdot (1 - \left(\frac{d}{D}\right)^2 \cdot \cos^2\alpha). \tag{2.4}$$

and the cage fault frequency is

$$f_c = \frac{1}{2} \cdot f_r \cdot (1 + \frac{d}{D} \cdot \cos\alpha). \tag{2.5}$$

The frequencies of equations (3), (4), and (5) are valid for ideal bearing. In practice, the roller elements not only rotate on races but also slide. This can be taken into account by multiplying the theoretical frequencies with a sliding factor 'e' that usually takes value between 0.8 and 1.0. Very often in literature and in practice the above equations are replaced by approximate equations. For example, for outer race defect

$$f_o = 0.4 \cdot N \cdot f_r. \tag{2.6}$$

and for inner race defect

$$f_i = 0.6 \cdot N \cdot f_r. \tag{2.7}$$

The simplified equations are used for two reasons, the geometry of the bearing is often not known and the actual condition monitoring device can calculate easily the frequencies of Equations (6) and (7) for couple of possible numbers of rolling elements.

2.2 FAULT DETECTION SCHEME

The purpose of the monitoring system is to measure the induction motor stator current and to analyze these data determining the vibration frequencies on the bearing. The stator current is sensed in any one of the three phases of the induction motor and its equivalent voltage signal is given to the sound cord of a PC. The analog signal Captured through the sound cord and it converts the sampled signal whose frequency is 11.025 kHz, to the frequency domain using Fast Fourier transform (FFT) algorithm. The current spectrum is generated by t he FFT algorithm with 131072 points and includes only the magnitude information in decibels for each frequency component. The magnitude corresponding to the fault frequencies are extracted and it is given to the fault detection algorithm which is implemented using Fuzzy logic technique. Condition of the bearing will be given as a result of that fuzzy module. Using the FFT analyzer the spectral values obtained and the required side band at ($f^{*}_{bng} = |f_s \pm m f_{i,0}|$) value is measured. The single phase stator current monitoring scheme is shown in the Figure 2.2.

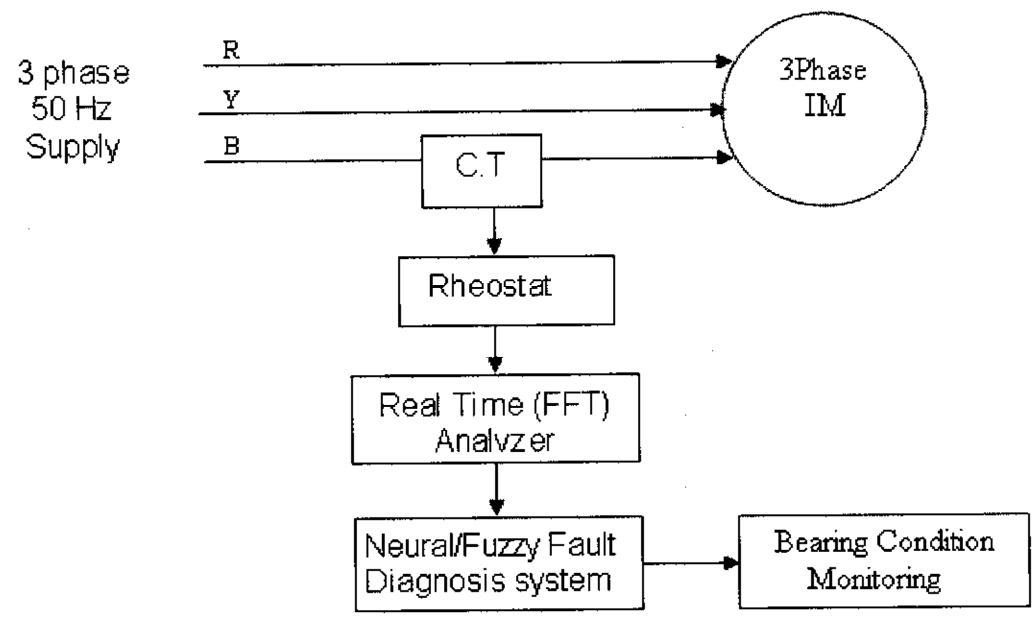


Figure 2.2 Single-Phase Stator Current Monitoring scheme.

2.3 EXPRIMENTAL SETUP FOR DATA ACQUISTION

To illustrate the fault detection scheme a 1 HP, six-pole induction motor is used .The rating of motor is given in Table 2.1. Figure 2.3 shows the experimental setup and for data acquisition.

The bearings of the induction motor are single row, deep groove ball bearings, type 6204Z (Shaft end) and 6203Z (Fan end). Each bearing has 8 balls. Experiments were conducted on 5 bearings: two of these are undamaged (healthy), while three bearings were drilled through the outer race and inner race with holes of diameters 2mm and 3mm as illustrated in Figure 2.4.

Table 2.1 Rated parameters of the machine under test.

Type	Three phase Induction motor
Power	1 HP
Voltage	415 V
Frequency	50 Hz
Current	1.7 A
Speed	960 rpm
Pole pairs	3

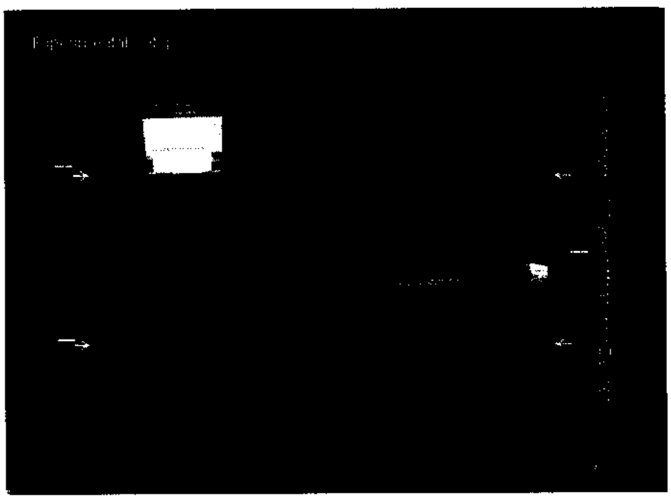


Figure 2.3 Testing equipment & experimental setup

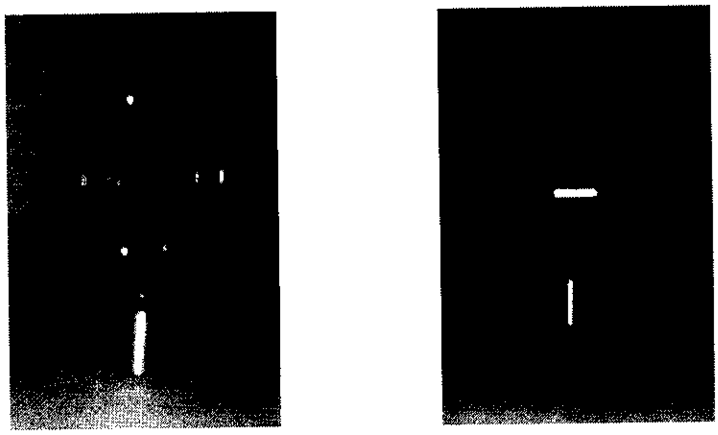


Figure 2.4 Bearings drilled with holes

Experimentation has been conducted by using faulty bearings. Bearing fault is created by drilling holes of various diameter (say 2mm or 3mm) in the race- ways both inner and outer which is similar to bearing faults

Two bearings of 6204Z and one bearing of 6203Z type were damaged and taken for experimental. While these are not realistic bearing failures, the artificial bearing faults produce characteristic fault frequencies and the type of fault is determined by the current spectra.

2.4 MOTOR CURRENT SIGNATURE ANALYSIS (MCSA)

From the bearing data sheet, the outside diameter of a 6204Z bearing is 47mm and inside diameter is 20 mm. Assuming that the inner and the outer races have the same thickness gives the pitch diameter as equal to 34.15mm (D = 34.15mm). The bearing has eight balls ($N = 8$) with approximate diameters of 7.85mm (d = 7.85mm). Assuming a contact angle $\theta = 0^\circ$ and motor operation at a rated shaft speed of 960rpm, the characteristic race frequencies of the shaft-end bearing are calculated using equation 2.2 and 2.3 as fo = 73.93 Hz and fi = 118.07 Hz for the test motor.

The results show that for a bearing which was damaged from the outer raceway and inner races with holes, the characteristic frequencies could be seen in the current spectrum.

The current spectra of the test motor are shown in figures (figure 2.5 to figure 2.7). The frequency components in the current spectra of the motor with defective bearing at shaft end are $|fe + 1 \cdot fo| = 123.93$, $|fe + 2 \cdot fo| = 197.86$, $|fe+3 \cdot fo| = 271.79$

and $|fe+4 \cdot fo|$ = 345.72 Hz frequencies. It is shown that these components are visible only in the plots of the defective bearing.

Current measurements for the damaged bearings were repeated under loaded operation of the induction machine. The current harmonics predicted for rated speed operation can still be found in the current spectrum. This indicates that, regardless of the load level of the machine, the bearing components are still detectable in the current spectrum. It is important to note that the frequency components produced by the bearing defect are relatively small when compared to the rest of the current spectrum. The largest components present in the current spectra occur at multiples of the supply frequency and are caused by saturation, winding distribution and supply voltage.

2.5 EXPRIMENTAL RESULTS

The current spectrum of healthy and faulty machine is shown in Figures (Figure 2.5 to Figure 2.7)

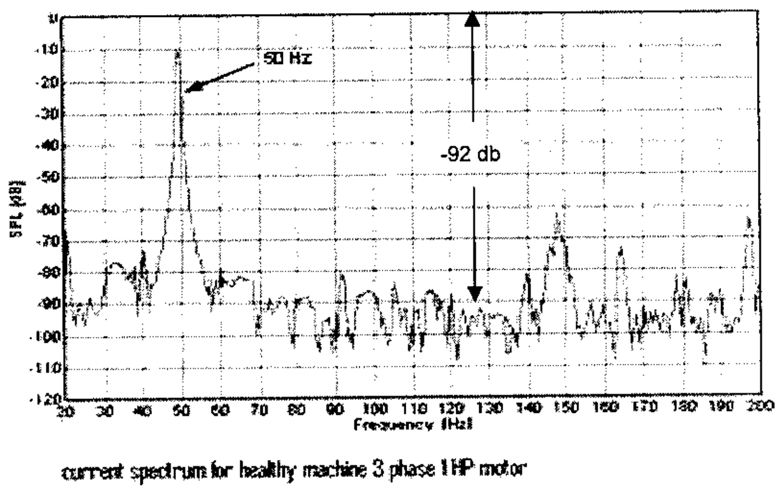


Figure 2.5 Current spectrum for healthy machine

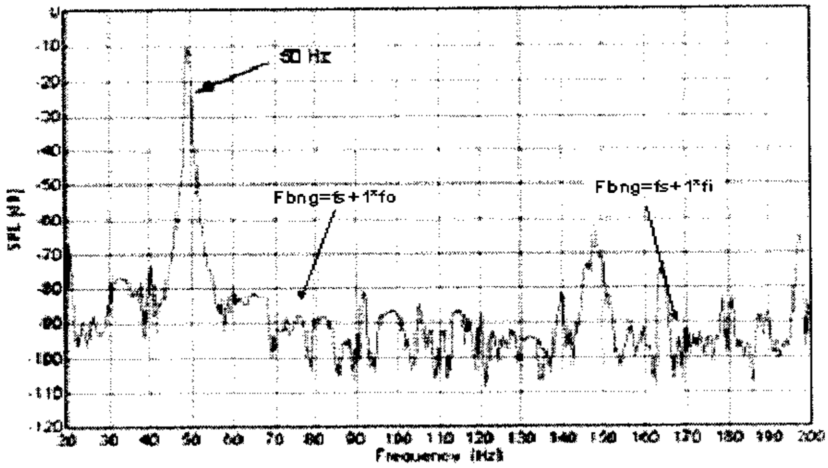


Figure 2.6 Current spectrum for faulty machine with shaft end

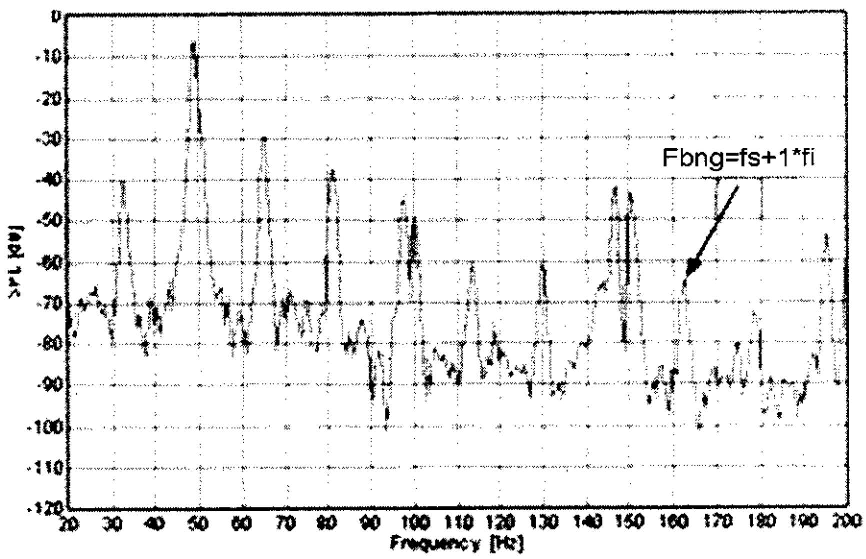


Figure 2.7 Current spectrum for faulty machine fan end

A comparative study has been made with the current spectrum of motor with healthy bearing and with a faulty one in both shaft end and fan end of the test motor, which is shown in Figures (Figure 2.5 to Figure 2.7). Also the data at different characteristic frequencies are shown in the Table 2.2 & 2.3

Table 2.2 Experimental results of the defective shaft end bearing

Characteristic Frequencies	Outer Race $f_o = 73.43$			Inner Race $f_i = 118.067$		
M (harmonic order)	1	2	3	1	2	3
Frequency (Hz) At $f_{(bng)}$	123.93	197.86	271.79	168.067	286.134	404.20
Harmonic amplitude for faulty machine in (dB)	-29.68	-27.75	-35.36	-35.19	-33.46	-36.68
Harmonic amplitude for healthy machine in (dB)	-38.25	-36.48	-48.13	-40.18	-44.20	-50.92

Table 2.3 Experimental results of the defective fan end bearing

Characteristic Frequencies	Outer Race $f_o = 73.43$			Inner Race $f_i = 118.067$		
M (harmonic order)	1	2	3	1	2	3
Frequency (Hz) At $f_{(bng)}$	124.1	198.22	272.33	167.89	285.79	403.69
Harmonic amplitude for faulty machine in (dB)	-35.57	-28.34	-40.87	-38.79	-36.89	-45.85
Harmonic amplitude for healthy machine in (dB)	-38.79	-36.89	-45.85	-41.33	-44.42	-47.11

CHAPTER 3

NEURAL NETWORK BASED FAULT DIAGNOSIS

3.1 INTRODUCTION TO NEURAL NETWORK

An artificial neural network is an information processing system that has certain performance characteristics in common with biological neural networks. Laurene Fausett (2004) explains that the artificial neural networks have been developed as generalization of mathematical models of human cognition or neural biology, based on assumptions that:

- Information processing occurs at many simple elements called neurons.
- Signals are passed between neurons over connection links.
- Each connection link has as associated weight, which, in a typical neural net, multiplies the signal transmitted.
- Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine it output signal.

A biological neuron has three types of components that are of particular interest in understanding an artificial neuron: its dendrites, soma and axon. Dendrites receive signal from other neurons. The signals are electrical impulses that are transmitted across a synaptic gap by means of a chemical process. The soma or cell body sums the incoming signals. When sufficient input is received, the cell fires; that is, it transmits a signal over its axon to other cells. Figure3.1 shows the structure of biological neuron.

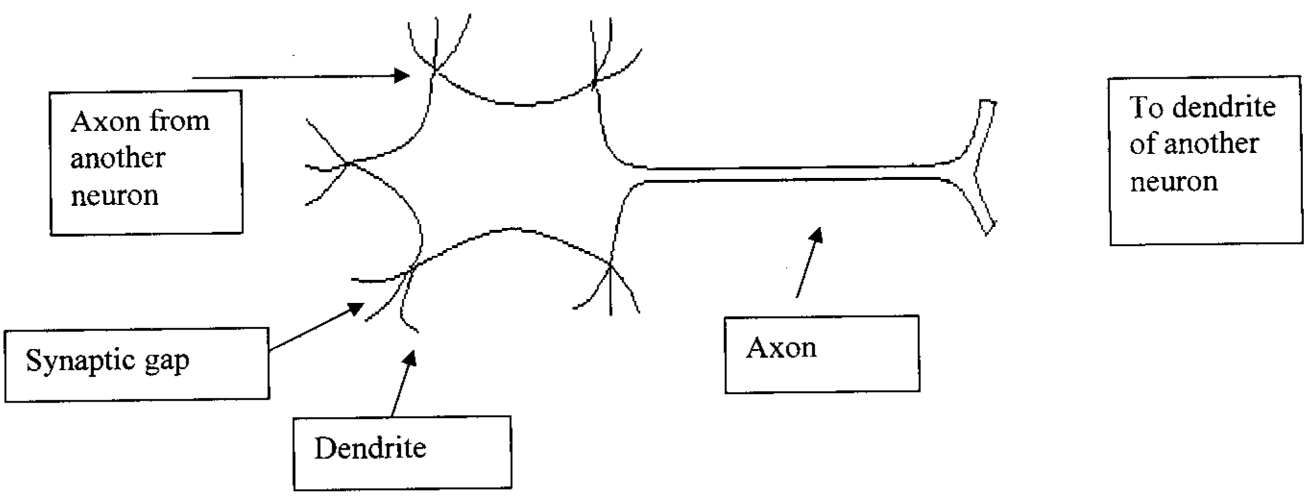


Figure 3.1 Structure of biological neuron

An Artificial Neural Network is characterized by,

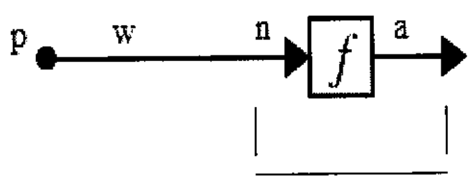
- Its pattern of connections between the neurons (called its architecture)
- Its method of determining the weights on the connections (called its training or learning, algorithm), and
- Its activation function

The network function is determined largely by the connections between elements. Therefore, a neural network can be trained to perform a particular function by adjusting the values of the connections (weight) between the elements commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Figure3.2 shows back-propagation neural network

The network weight is adjusted based on a comparison of the output and the target, until the network output matches the target.

3.1.1 NEURON MODEL

Figure 3.2 shows a neuron with a single scalar input with no bias. The scalar input p, is transmitted through a connection that multiplies its strength by the scalar weight w, to form the product wp, again a scalar. Here the weighted input wp is the only argument of the activation function f, which produces the scalar output a.

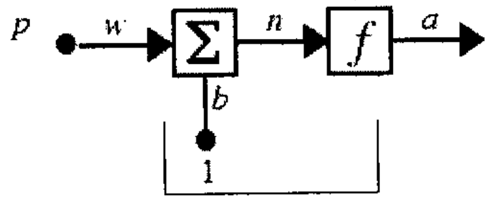


Neuron without bias

$a = f(wp)$

Figure3.2 Single – Input Neuron without Bias

Figure3.3 shows a neuron with a scalar input, with scalar bias. The bias is much like a weight, except that it has a constant input of 1. The activation function net input n, again a scalar, is a sum of the weighted input wp and the bias b., this sum is the argument of the activation function f. f is an activation function, typically a step function or a sigmoid function, that takes the argument n and produces the output a. w and b are both adjustable parameters of the neuron.



Neuron with bias

a = f (wp + b)

Figure3.3 Single Input Neuron with Bias

The central idea of neural networks is that such parameters can be adjusted so that the network exhibits some desired or interesting behavior. Thus, we can train the network to do a particular job by adjusting the weight or bias parameters, or perhaps the network itself will adjust these parameters to achieve some desired end.

3.1.2 ACTIVATION FUNCTIONS

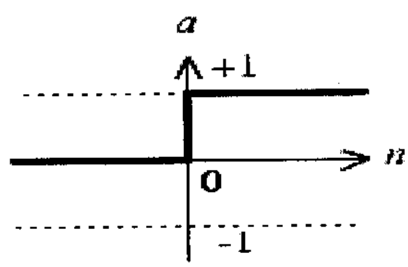
An activation function may be linear or a non-linear function of an. A particular activation function is chosen to satisfy some specification of a problem that the neuron is attempting to solve. There are three most commonly used activation function. They are

(a) Hard limit activation function

(b) Linear activation function

(c) Log-sigmoid activation function

(a) Hard limit activation function:



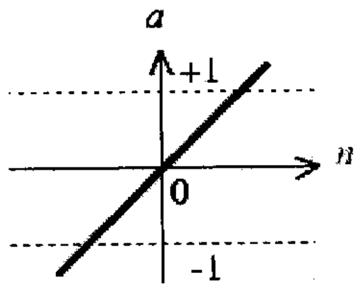
a = hardlim (n)

Figure3.4 Hard Limit Activation Function

Figure 3.4 shows the graphical representation of the hard limit activation function. The hard limit activation function sets the output of the neuron to 0 if the function argument is less than 0, or 1 if its argument is greater than or equal to 0.

b..Linear activation function:

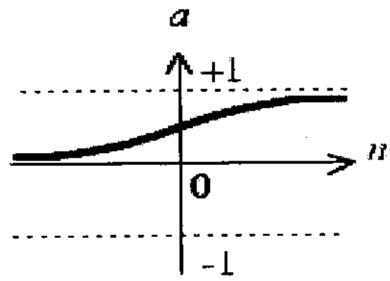
The output of a linear activation function is equal to its input. The output (a) versus input (p) characteristic of a single-input linear neuron is shown in Figure 3.5.



a = purelin (n)

Figure3.5 Linear Activation Function

(b) Log-sigmoid activation function:



a = logsig (n)

Figure3.6 Log-Sigmoid Activation Function

Figure 3.6 shows the log-sigmoid activation function. This activation function takes the input (which may have any value between plus and minus infinity) and squashes the output into the range 0 to 1, according to expression

$$a = 1 / 1 + e^{+n} \qquad (3.1)$$

This activation function is commonly used in multilayer networks that are trained using the back-propagation algorithm, in part because this function is differentiable.

supervised training. The LMS algorithm will adjust the weights and biases to minimize the mean square error, where the error is the difference between the target output and the network output. The perceptron-net is incapable of implementing certain elementary functions. These limitations were overcome with improved (multilayer) perceptron networks. Performance learning is another important class of learning law, in which the network parameters are adjusted to optimize the performance of the network. Back propagation (BP) algorithm can be used to train multilayer networks. As with the LMS learning law, BP is an approximate steepest descent algorithm, in which the performance index is mean square error. The difference between the LMS algorithm and back propagation is only in the way in which the derivatives are calculated. The single-layer perceptron like networks are only able to solve linearly separable classification problems. Multilayer perceptron, trained by BP algorithm were developed to overcome these limitations and is currently the most widely used neural network. In addition, multi-layer networks can be used as universal function approximators. A two-layer network, with sigmoid-type activation functions in the hidden layer, can approximate any practical function, with enough neurons in the hidden layer. The Figure 3.7 shows the Architecture of BP Neural Network.

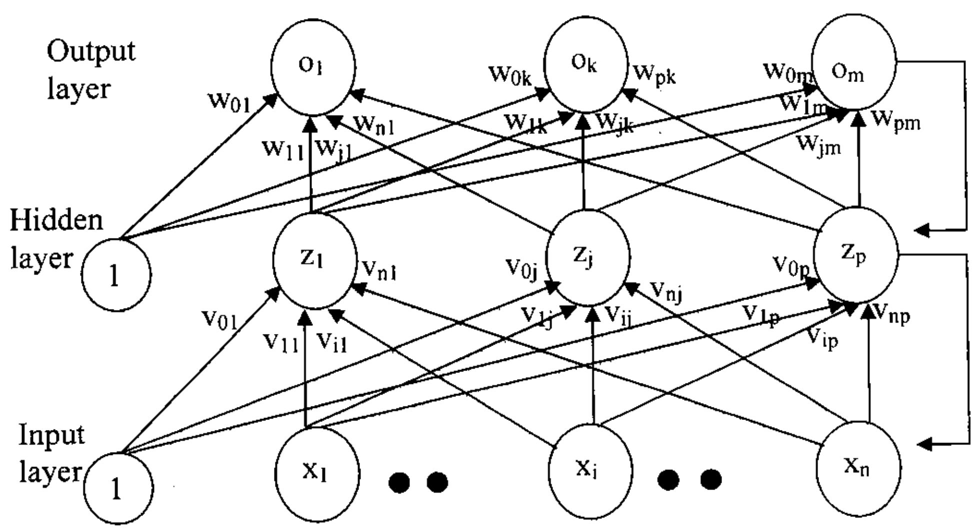


Figure 3.7 Architecture of BP Neural Network

The BP algorithm uses the chain rule in order to compute the derivatives of the squared error with respect to the weights and biases in the hidden layers. It is called

BP because the derivatives are computed first at the last layer of the network, and then propagated backward through the network, using the chain rule, to compute the derivatives in the hidden layers.

The BP training algorithm is an interactive gradient algorithm designed to minimize the mean square error between the actual output of a feed-forward net and the desired output.

Step 0: Initialize weights.

Step 1: While stopping condition is false, do steps 2-9,

Step 2: For each training pair, do steps 3-8,

Feed forward:

Step 3: Each input unit (X_i, i=1...n) receives input signal x_i and broadcasts this signal to all units in the layer above (the hidden units).

Step 4: Each hidden unit (Z_j, j =1...n) sums it weighted input signals,

$$Z\_in_j = v_{oj} + \sum_{i=1}^{n} xivij,$$

applies it activation function to compute its output signals,

$Zj=f(z\_in_j)$,and sends this signal to all units in the layer above .

Step 5: Each output unit (y_k, k=1...m) sums its weighted input signals,

$$Y\_In_k = W_{ok}=\sum_{j=1}^{P} ZjWjk$$

And applies its activation function to compute its signals,

$$Yk = f(y\_ink).$$

Back propagation of error:

Step 6: Each output unit (y_k,k=1...m) receives a target pattern corresponding to input training pattern, computes its error information term ,

$\delta_k=(t_k-y_k)f'(y\_in_k)$,

Calculates its weight correction term (used to update w_{jk} later),

$$\Delta wjk = \alpha\delta kzj,$$

Calculates its bias correction term (used to update Wok later)

$$\Delta wok = \alpha\delta k$$

Step7: Each hidden unit (Z_j, j =1...p) sums its delta inputs

$$\delta\_inj = \sum_{k=1}^{m} \delta k wjk,$$

multiplies by the derivative of its activation function to calculate its error information term

$$\delta j = \delta\_inj f'(z\_inj),$$

Calculates its weight correction term (used to update V_{ij} later),

$$\Delta vij = \alpha \delta j xi,$$

And calculates its bias correction term (used to update V_{oj} later),

$$\Delta voj = \alpha \delta j.$$

Update weights and biases:

Step8: Each output unit (Y_k, k=1...m) updates its bias and weights (j=0...p):

$$W_{jk}(new)=w_{jk}(old)= \Delta wjk,$$

Each hidden unit (Z_j, j...p) updates its bias and weights (i=0...n);

$$Vij(new)=vij(old) + \Delta vij$$

Step 9: Test stopping condition.

3.2.1 CHOICE OF PARAMETERS FOR NETWORK TRAINING

When the basic BP algorithm is applied to a practical problem the training may take days or weeks of computer time. This has encouraged considerable research on methods to accelerate the convergence of the algorithm. The research on faster algorithms falls roughly into two categories; the first category involves the development of heuristic techniques, which arises out of a study of the distinctive performance of the standard BP algorithm. These heuristic techniques include such ideas varying the learning rate, using momentum and rescaling variables. Another category of research has focused on standard numerical optimization techniques.

3.2.2 LEARNING RATE

The speed of training the BP network is improved by changing the learning rate during training. Increasing the learning rate on flat surfaces and then decreasing the learning rate when slope increases can increase the process of convergence. If the learning rate is too large, it leads to unstable learning. And if it is too small, it leads to incredibly long training times. Hence care has to taken while deciding learning rate. There are many different approaches for varying the learning rate. The learning rate is varied according to the performance of the algorithm. The rules of the variable learning rate BP algorithm are:

1. If the squared error increases by more than some set percentage ξ (typically one to five percent) after weight update, then the weight update is discarded, the learning rate is multiplied by some factor $o < p < 1$, and the momentum coefficient γ (if it is used) is set to zero.

2. If the squared error decreases after a weight update, then the weight update is accepted and the learning rate is multiplied by some factor $\eta > 1$. If γ has been previously set to zero, it is reset to its original value.

3. If the squared error increases by less than ξ then the weight update is accepted but the learning rate is unchanged. If γ has been previously set to zero, it is reset to its original value.

3.2.3 MOMENTUM FACTOR

In BP with momentum, the weight change is in a direction that is a combination of the current gradient and the previous gradient. This is a modification of gradient descent whose advantage arises chiefly when some training data are very different from the majority of the data. By the use of momentum larger training rate can be used, while maintaining the stability of the algorithm. Another feature of momentum is that it tends to accelerate convergence when the trajectory is moving in a consistent direction. The larger the value of γ, the more the momentum the trajectory has. The momentum coefficient is maintained with the range [0, 1].

3.3 STRUCTURE OF BP NETWORK FOR FAULT DETECTION

An artificial neural network is composed of neurons with a deterministic activation function. The neural network is trained by adjusting the numerical value of the weights will contain the non-linearity of the desired mapping, so that difficulties in the mathematical modeling can be avoided. The BP training algorithm is used to adjust the numerical values of the weights and the internal threshold of each neuron. The network is trained by, initially selecting small random weights and internal threshold and then presenting all training data. Weights and thresholds are adjusted after every training example is presented to the network; until the weight converges or the error is reduced to acceptable value. The value of learning rate and momentum factor are respectively 0.0012 and 0.85. Figure 3.8 shows the structure of BP Network for Fault Detection.

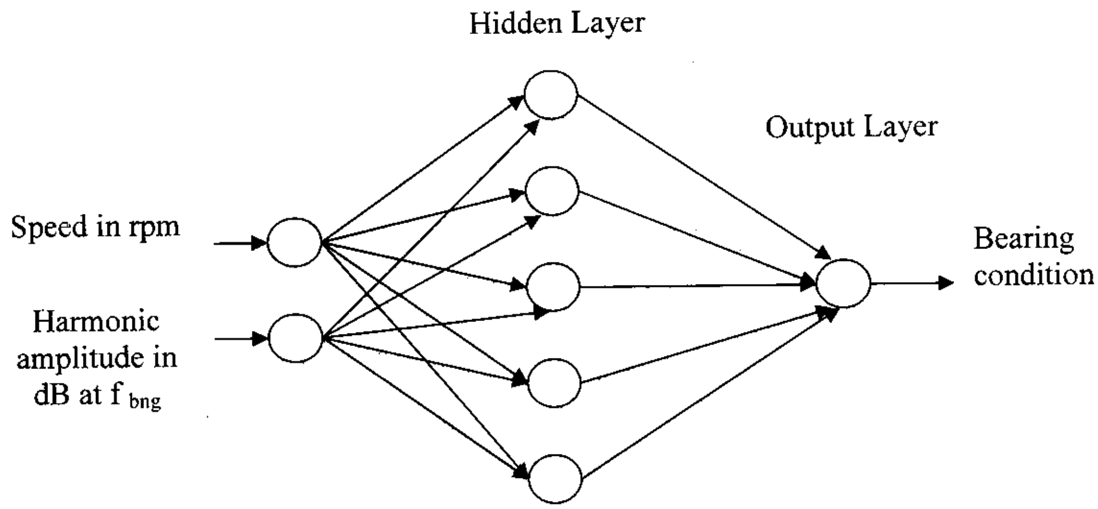


Figure 3.8 Structure of BP Network for Fault Detection

3.4 SIMULATION RESULTS

Table 3.1 shows that input and output of the BP network

Table 3.1 BP network input and output

Speed in rpm	Harmonic Amplitude in (dB)	Target
860	-45	0
880	-65	0.5
920	-75	1
950	-85	1
Average error		0.177

Feed forward neural networks with two layers are used. The network consists of two input neuron, five hidden neurons and one output neuron. BP algorithm is used for training. The activation function in the first layer is log–sigmoid, and the output layer transfer function is tan-sigmoid function is the output layer. The training function used is trainlm. Figure 3.9 shows the performance characteristics of the BP network.

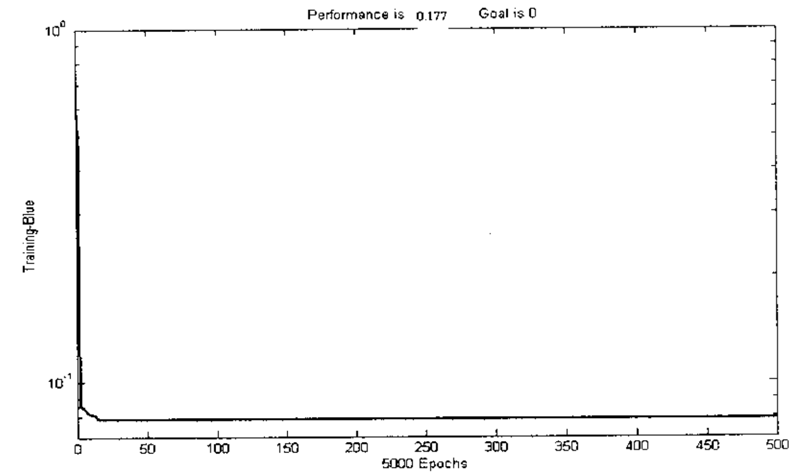


Figure 3.9 Epoch Vs Error Characteristics

CHAPTER 4

FUZZY LOGIC BASED FAULT DIAGNOSIS

4.1 INTRODUCTION

Problems in the real world quite often turn out to be complex owing to an element of uncertainty either in the parameters which define the problem or in the situations in which the problem occurs.

The uncertainty may arise due to partial information about the problem, or due to information which is not fully reliable, or due to inherent imprecision in the language with which the problem is defined, or due to receipt of information from more than one source about the problem which is conflicting. It is in such situations that fuzzy set theory exhibits immense potential for effective solving of the uncertainty in the problem. Fuzziness means 'vagueness'. Fuzzy set theory is an excellent mathematical tool to handle the uncertainty arising due to vagueness.

Fuzzy logic systems are universal function approximators. In general, the goal of the fuzzy logic system is to yield a set of outputs for given inputs in a non-linear system, without using any mathematical model, but by using linguistic rules. It has many advantages. They are

- Fuzzy logic is conceptually easy to understand. The mathematical concepts behind fuzzy reasoning are very simple. What makes fuzzy better is the "Naturalness" of its approach and not its far-reaching complexity.
- Fuzzy logic is flexible. With any given system, it's easy to massage it or layer more functionality on top of it without starting again from scratch.
- Fuzzy logic is tolerant of imprecise data. Everything is imprecise if you look closely enough, but more than that, most things are imprecise even on careful inspection. Fuzzy reasoning builds this understanding into the process rather than tacking it onto the end.
- Fuzzy logic can model nonlinear functions of arbitrary complexity. You can create a fuzzy system to match any set of input-output data. This process is made particularly easy by adaptive techniques like Adaptive Neuro-Fuzzy Inference Systems (ANFIS), which are available in the Fuzzy Logic Toolbox.

- Fuzzy logic can be built on top of the experience of experts. In direct contrast to neural networks, which take training data and generate opaque, impenetrable models, fuzzy logic lets you rely on the experience of people who already understand your system.
- Fuzzy logic can be blended with conventional control techniques. Fuzzy systems don't necessarily replace conventional control methods. In many cases fuzzy systems augment them and simplify their implementation.
- Fuzzy logic is based on natural language. The basis for fuzzy logic is the basis for human communication. This observation underpins many of the other statements about fuzzy logic.

4.2 MAMDANI FUZZY LOGIC INFERENCE SYSTEM

Mamdani-type of fuzzy logic controller contains four main parts, two of which perform transformations. The four parts are

- Fuzzifier (transformation 1)
- Fuzzy rule base
- Inference engine(fuzzy reasoning, decision-making logic)
- Defuzzifier(transformation 2)

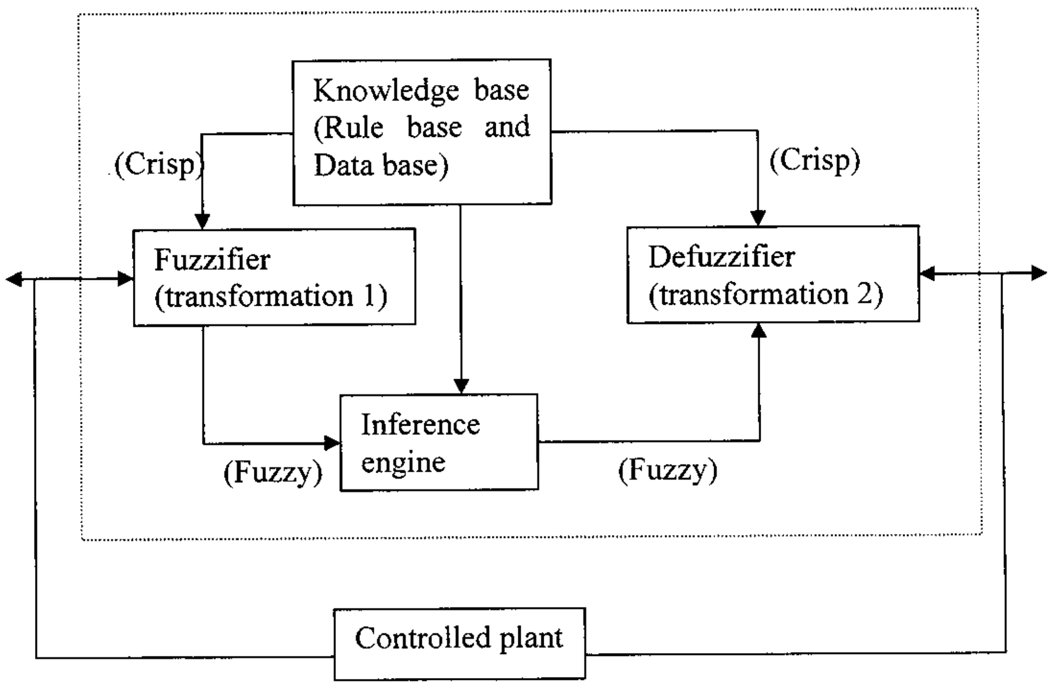


Figure 4.1 Mamdani Fuzzy Logic Inference Systems

4.2.1 FUZZIFIER

The fuzzifier performs measurement of the input variables (input signals, real variables), scale mapping and fuzzification (transformation 1).thus all the monitoring input signals are scaled and fuzzification means that the measured signals (crisp input quantities which have numerical values) are transformed into fuzzy quantities. This transformation is performed by using membership functions. In a conventional fuzzy logic controller, the number of membership functions and the shapes of these are initially determined by the user. A membership function has a value between 0 and 1, and it indicates the degree of belongingness of a quantity to a fuzzy set. If it is absolutely certain that the quantity belongs to the fuzzy set, then its value is 1(it is 100% certain that the quantity belongs to this set), but if it is absolutely certain that it does not belong to this set then its value is 0. Similarly if for example the quantity belongs to the fuzzy set to an extent of 50%, then the membership function is 0.5.

There are many types of different membership functions, piecewise linear or continuous. Some of these are smooth membership functions, e.g. bell-shaped, sigmoid, Gaussian etc. and others are non-smooth, e.g. triangular, trapezoidal etc. the choice of the type of membership function used in a specific problem is not unique. Thus it is reasonable to specify parameterized membership functions, which can be fitted to a practical problem. If the number of elements in the universe X is very large or if a continuum is used for X then it is useful to have a parameterized membership function, where the parameters are adjusted according to the given problem. Parameterized membership functions play an important role in adaptive fuzzy systems, but are also useful for digital implementation. Due to their simple forms and high computational efficiency, simple membership functions, which contain straight line segments, are used extensively in various implementations. Obviously, the triangular membership function is a special case of the trapezoidal one.

$$\mu A'(x;a,b,c) = \begin{cases} 0 & x<a \\ (x-a) / (b-a) & a \le x \le b \\ (c-x) / (c-b) & b \le x \le c \\ 0 & x>c \end{cases} \qquad (4.1)$$

Triangular membership function depends on three parameters a, b, c and can be described as follows by considering four regions.

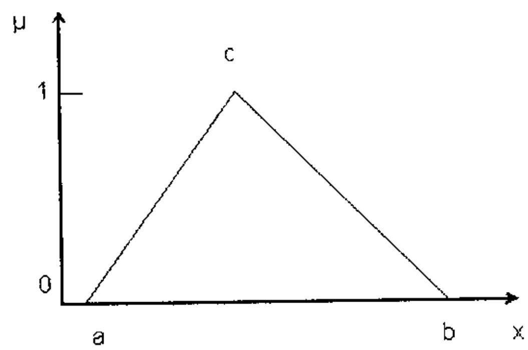


Figure 4.2 Triangular membership functions

A triangular membership function is shown in Figure 4.2 is used for both the input and output variable and the points a, b, c are also denoted. Alternatively, it is possible to give a more compact form

$\mu^{A'}(x; a, b, c) = \max \{\min [(x-a) / (b-a), (c-x) / (c-b)], 0\}$ (4.2)

The detection of bearing fault severity is considered by utilizing Mamdani-style fuzzy inference and using as input variables are speed and current .Low, medium, and high are the three membership functions used for the input variables. Poor, fair, and good are the membership functions used for output variable.

4.2.2 FUZZY RULES

The knowledge base consists of the data base and the linguistic control rule base. The data base provides the information which is used to define the linguistic control rules and the fuzzy data manipulation in the fuzzy logic controller. The rule base specifies the control goal actions by means of a set of linguistic control rules. In other words, the rule base contains rules such as would be provided by an expert. The fuzzy logic controller looks at the input signals and by using the expert rules determines the appropriate output signals (control actions). The rule base contains a set of if-then rules. The main methods of developing a rule base are:

- Using the experience and knowledge of an expert for the application and the control goals;
- Modeling the control action of the operator;
- Modeling the process;
- Using a self-organized fuzzy controller;
- Using artificial neural networks;

When the initial rules are obtained by using expert physical considerations, these can be formed by considering that the three main objectives to be achieved by the fuzzy logic controller are:

- Removal of any significant errors in the process output by suitable adjustment of the control output;
- Ensuring a smooth control action near the reference value (small oscillations in the process output are not transmitted to the control input);
- Preventing the process output exceeding user specified values;

By considering the two dimensional matrix of the input variables, each subspace is associated with a fuzzy output situation.

4.2.3 INFERENCE ENGINE

It is the kernel of a fuzzy logic controller and has the capability both of simulating human decision–making based on fuzzy concepts and of inferring fuzzy control actions by using fuzzy implication and fuzzy logic rules of inference as shown in Figure 5.3. In other words, once all the monitored input variables are transformed into their respective linguistic variables, the inference engine evaluates the set of if-then rules and thus result is obtained which is again a linguistic value for the linguistic variable. This linguistic result has to be then transformed into a crisp output value of the fuzzy logic control.

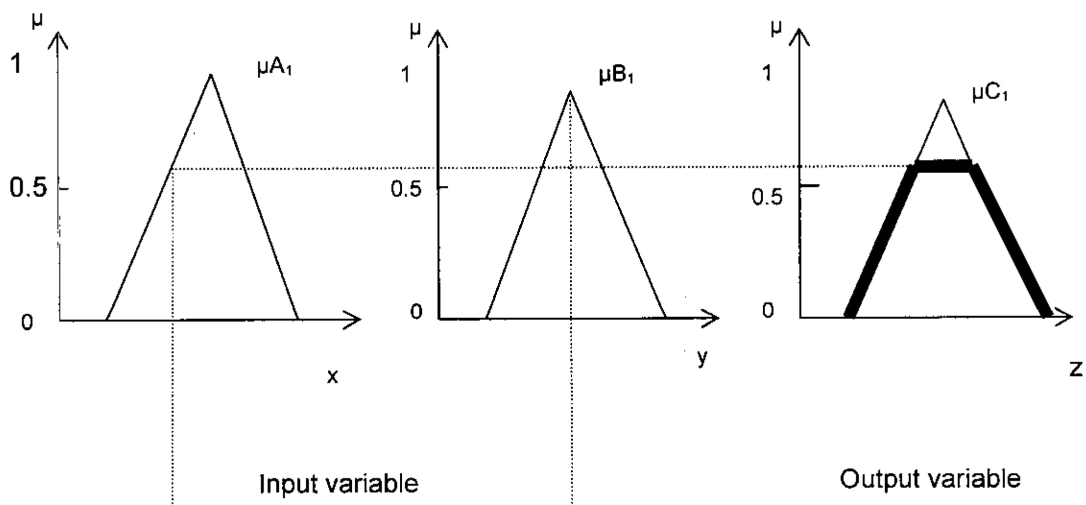


Figure 4.3 Graphical interpretation of fuzzification, inference

4.2.4 DEFUZZIFIER

The second transformation is performed by the defuzzifier which performs scale mapping as well as defuzzification. The defuzzifier yields a non-fuzzy, crisp control action from the inferred fuzzy control action by using the consequent membership functions of the rules. There are many defuzzification techniques. They are centre of gravity method, height method, mean of maxima method, first of maxima method, sum of maxima. In this project height method defuzzification technique is used as shown in Figure 5.4. In this method, the individual output membership functions for each rule are used (e.g. if for the fuzzy AND the min operator is used, then these are clipped membership functions) and first, the peak values (height), p_k, of the (clipped) consequent membership functions of all rules that have fired are multiplied by the ordinates of these membership functions(c_k). In a second step, these products are added and then divided by the sum of the peak values of the (clipped) consequent membership functions. It follows that the output value is

$$Z^{*H} = \frac{\sum p_k c_k}{\sum p_k} \qquad (4.3)$$

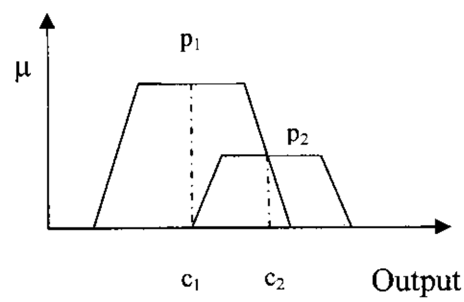


Figure 4.4 Height Defuzzification Method

4.3 FUZZY LOGIC BASED FAULT DETECTION METHOD

The fuzzy inference system used is Mamdani. Triangular membership function is used for both the input speed and current and for the output diagnosis. Three membership functions for the input variables as well as for output variable are selected.

Table 4.1 Fuzzy Rules

CURRENT	SPEED	DIAGONOSIS
LOW	LOW	BAD
LOW	MEDIUM	FAIR
LOW	HIGH	GOOD
MEDIUM	LOW	FAIR
MEDIUM	MEDIUM	FAIR
MEDIUM	HIGH	GOOD
HIGH	HIGH	BAD
HIGH	MEDIUM	FAIR
HIGH	LOW	BAD

Table 4.2 gives the nine fuzzy rules used for bearing fault detection. The triangular membership function is used for fuzzification, the ranges setting for each function is follows, For speed, "low" speed between 860 rpm to 885 rpm, "medium" between 880 rpm to 920 rpm, "high" between 918 rpm to 960 rpm .For current spectral value, "low" amplitude between -95 db to -73 db, "medium" between -75 db to -58 db, "high" between -62 db to -40 db. For output; "poor" between 0 to 0.001, "fair" between 0 to1, "good" between .99 to 1. The membership functions used for simulation are shown in Figures (Figure4.5– Figure 4.7). The Figure 4.8 shows the surface viewer of the FFD.

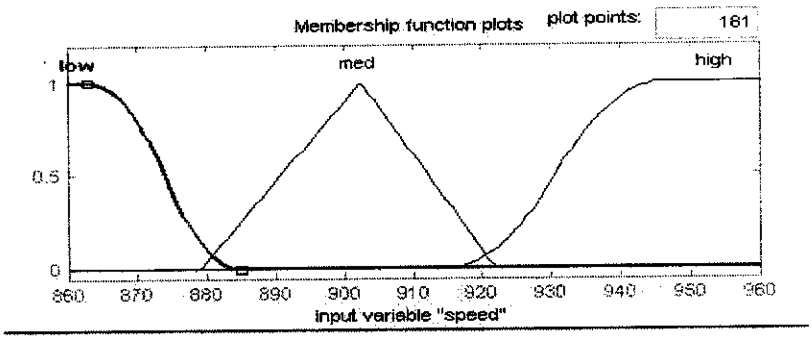


Figure 4.5 Input Membership Functions for speed

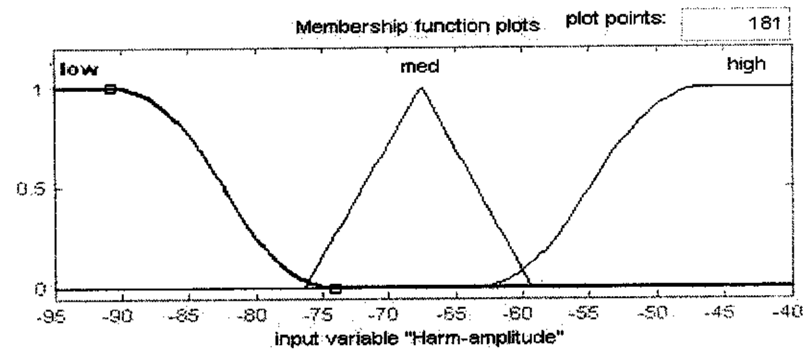


Figure 4.6 Input Membership Functions for current

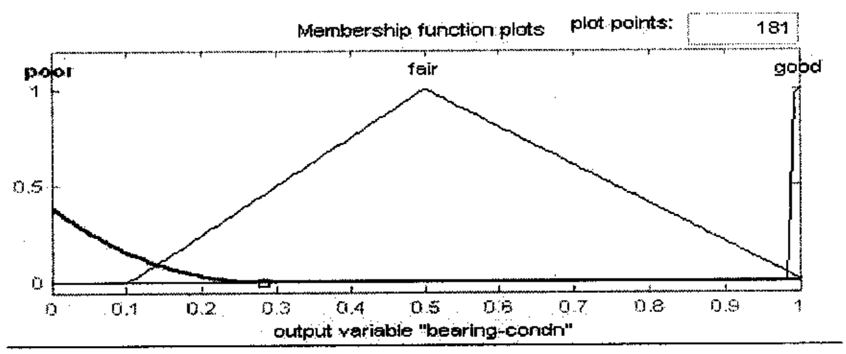


Figure 4.7 Output Membership Functions for diagnosis

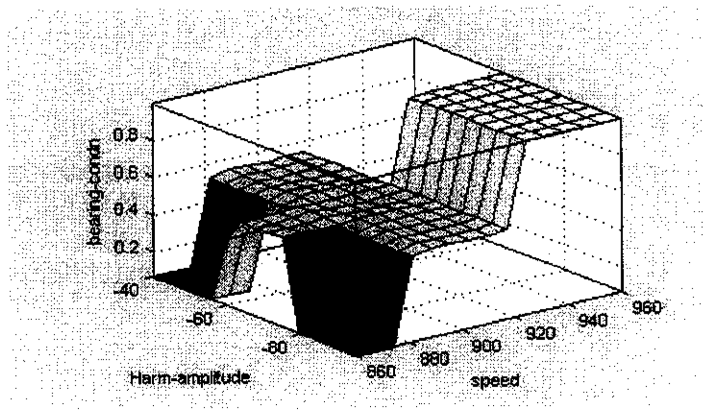


Figure 4.8 Surface Viewer

4.4 SIMULATION RESULTS

The difference between the target and the actual is calculated for different inputs. The results obtained are shown in Table 4.2

Table 4.2 Simulation results

Speed	Amplitude in db	Target	Obtained value	% of error	Bearing condition
860	-45	0	0.0669	0	Poor
880	-65	0.5	0.5	0	Fair
950	-85	1	0.984	0.4	Good
Average error				0.1333	

4.5 COMPARISON OF NEURAL NETWORK AND FUZZY BASED FAULT DIAGNOSIS SYSTEM

Neural network approach is a black box approach, where the expert knowledge is hidden in the black box system in the form of weights and biases of the neural network. However, in fuzzy logic based system the actions of a human expert are clearly present in the rule base. Comparison of both the neural network and fuzzy based fault diagnoses for both the online and offline is given in Table 4.2

Table 4.3 Comparison of Neural Network and Fuzzy Based Fault Diagnosis

% Error	Neural Network Diagnosis	Fuzzy Logic Diagnosis
	0.177	0.133

From the above table, it is inferred that the fuzzy fault diagnosis gives reduced error compared with neural network based diagnosis.

CHAPTER 5
HARDWARE IMPLEMENTATION

For hardware implementation MCSA method is used. Figure 5.1 shows the schematic diagram of hardware implementation of MCSA method. This method monitors the frequency components of stator current spectrum for fault detection. The fuzzy logic is used for fault diagnosis. The detailed information about the fuzzy logic inference system and ranges of input & output parameters of the system are discussed in the chapter 4.

5.1 SCHEMATIC DIAGRAM

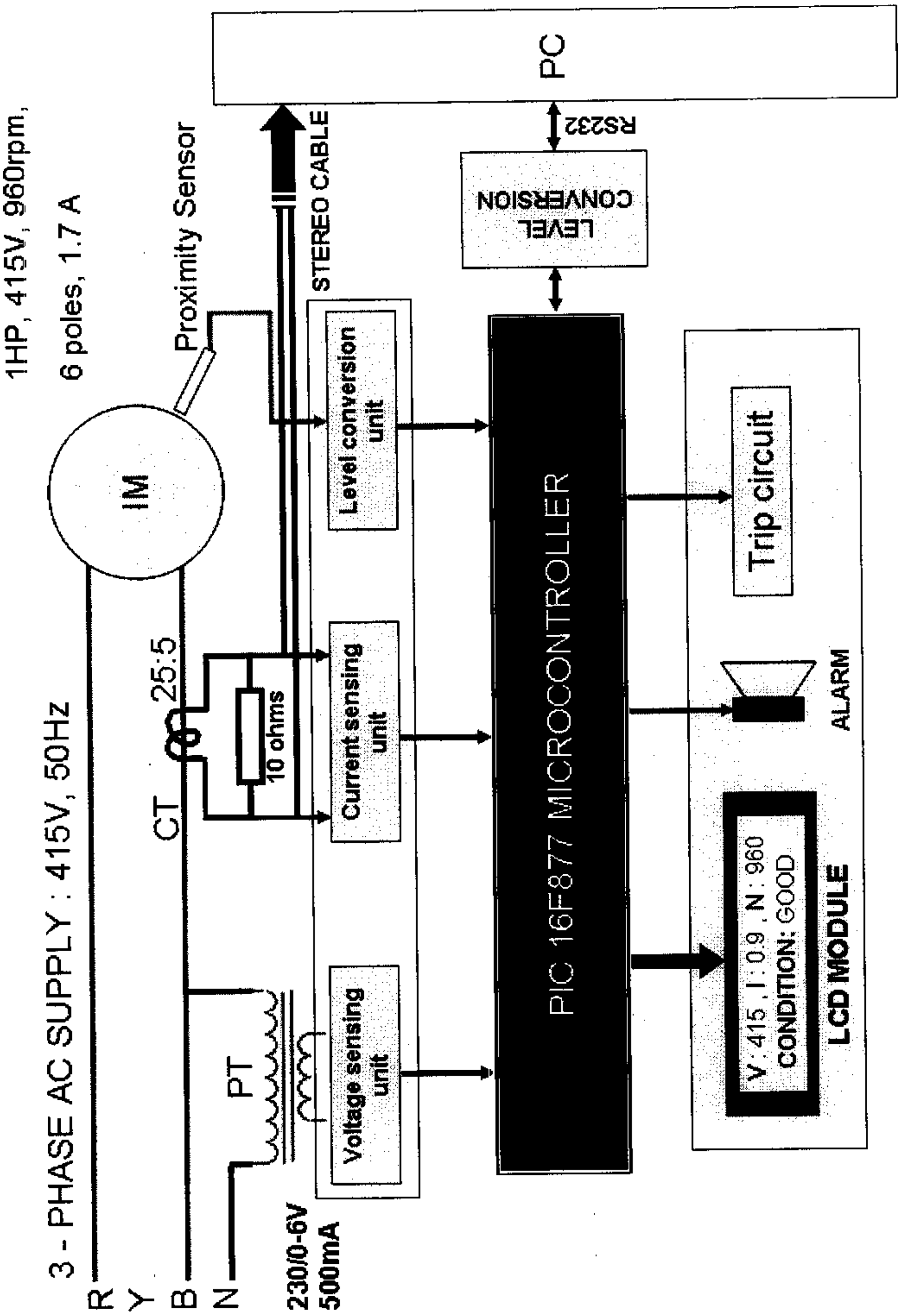


Figure 5.1 schematic Diagram

5.2 HARDWARE DESCRIPTION

In this hardware, it have several measurement modules such as voltage, current and speed measurement. In addition to this, PIC motherboard is the core part. These units require a DC supply ranging from +5 V to +12 V. Thus a regulated power supply circuit is designed for +5 V and +12 V.

The commonly available source of 230 V, 50 Hz AC is utilized and it is stepped down to the required maximum voltage, say 12 V AC. Then it is rectified, filtered and regulated to the required output voltage. +5 V power supply is shown in the figure 5.2. +12 V power supply is shown in the figure 5.3.

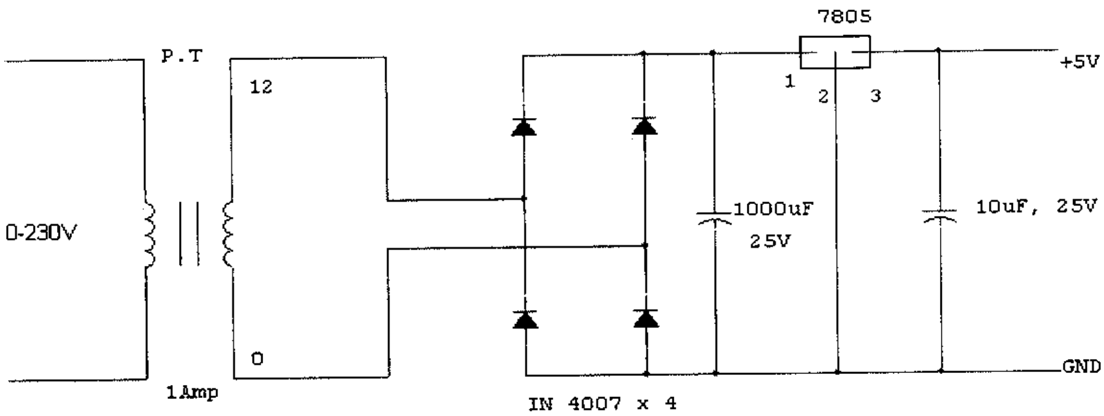


Figure 5.2 +5 V Power supply

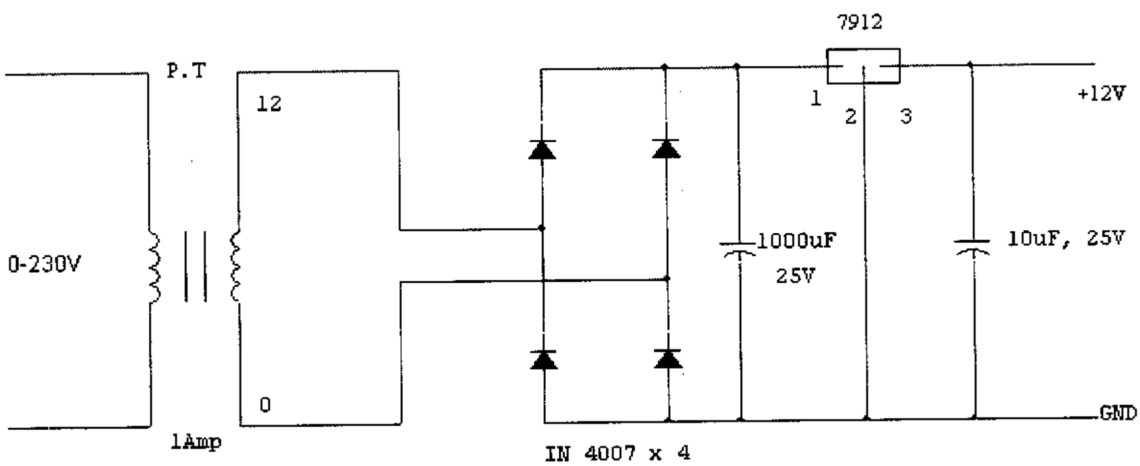


Figure 5.3 +12 V Power supply

AC 230V, 50 Hz supply is given to the primary side of the step down transformer of 230V/ 0-12V type to perform step down operation. The current rating of the transformer is 1A. Now this can be used for rectification purpose. Rectification is achieved using a full bridge rectifier circuit which comprises of four 1N 4007 solid state diodes. Two diodes will conduct during the positive cycle and the other two will conduct during the negative half cycle. The output obtained is not a pure DC and therefore filtration has to be done. Filter circuits usually consist of a capacitor, which smoothens the pulsating DC. It is helpful in reduction of the ripples from pulsating (1000µF/ 25V) and it maintains stability at the load side (10µF/ 25V).

5.3 VOLTAGE REGULATORS

Voltage regulators play an important role in any power supply unit. The primary purpose of a regulator is to aid the rectifier and filter circuit in providing a constant DC voltage to the device. Power supplies without regulators have an inherent problem of changing DC voltage values due to variations in the load or due to fluctuations in the AC line voltage. IC 7805 and IC 7812 are used to provide +5V and +12V regulated DC supply respectively.

5.4 VOLTAGE MEASUREMENT

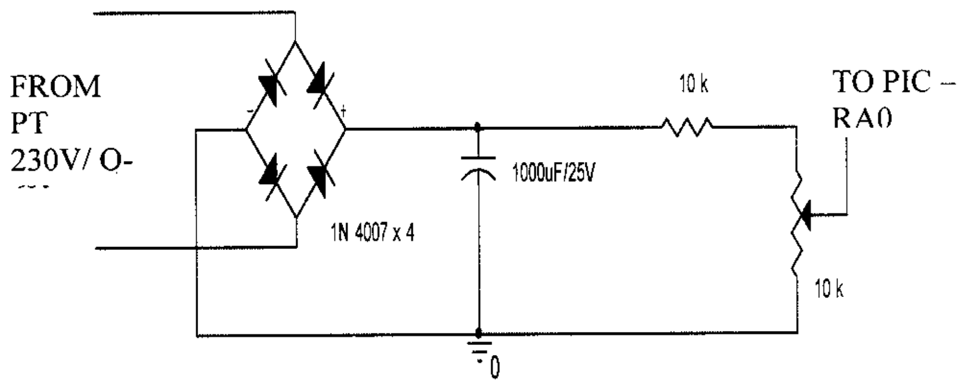


Figure 5.4 Voltage measurement circuit

The voltage measurement circuit of this project is shown in the Figure 5.4. In case of voltage measurement, among three phases of the supply only one of the phases is monitored using a 230V/ 0-6V PT. The AC output of the PT is rectified, filtered and converted into DC. The converted DC voltage is dropped across a variable resistor and given into the PIC microcontroller. This unit is calibrated to

415V (Line to line voltage) and displayed in the LCD. Voltage measured is not used as an input parameter to the system.

5.5 CURRENT MEASUREMENT

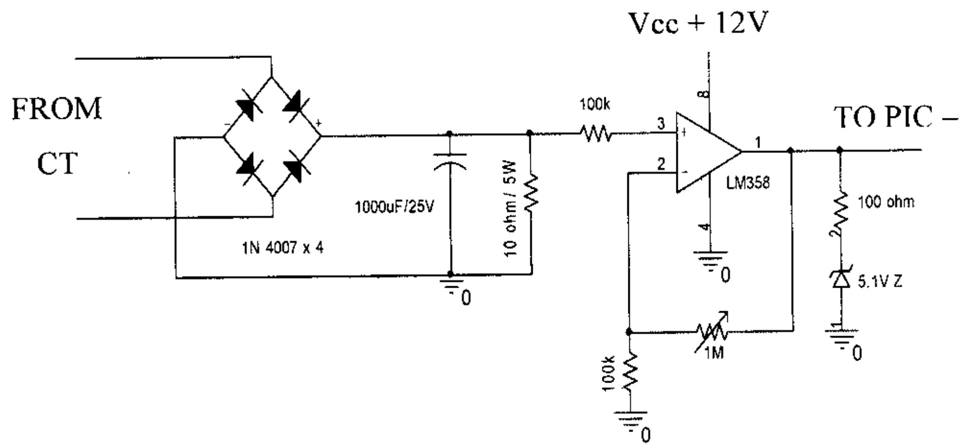


Figure 5.5 Current measurement circuits

The current measurement circuit of this project is shown in the Figure 5.5. A Current transformer (5:1) is connected in series with one of the phases which is connected to the motor. A shunt resistor of value 10 ohm/ 5W is connected across the rectified and filtered signal. Thus the current signal is converted into its equivalent voltage signal. The voltage signal of very small current value is amplified using a negative feedback amplifier and given to the PIC microcontroller. The PIC is calibrated to the original value of current by adjusting the value of the feedback resistor.

5.6 SPEED MEASUREMENT

The speed sensing unit of this project is shown in the Figure 5.5. An inductive type proximity sensor is placed near the metal strip, which is fixed on the rotor whose speed is to be measured. When the rotor rotates, the strip disturbs the magnetic field produced by the sensor in each revolution. The output will be in pulses and the time difference between the pulses are calculated, from this speed can be calculated. The in-built timer control is used for this purpose.

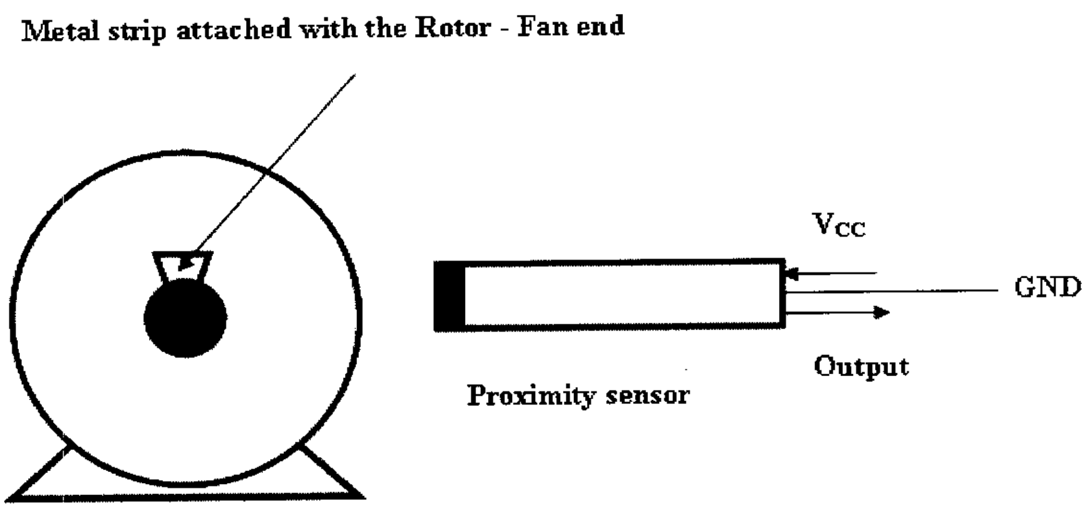
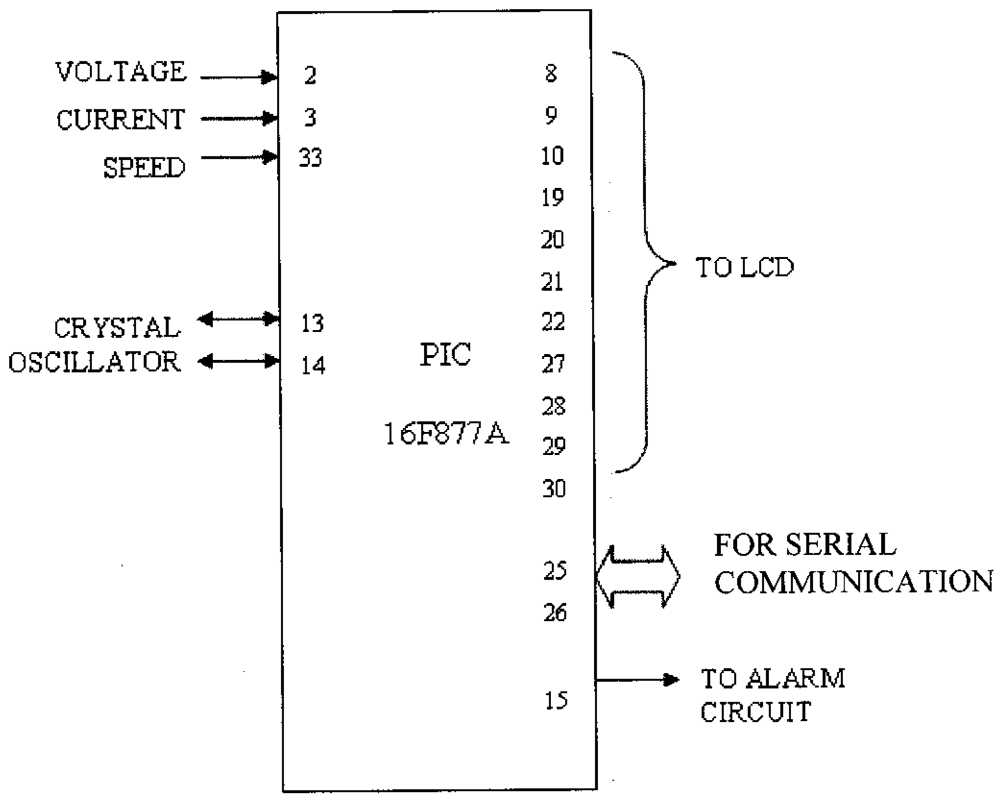


Figure 5.6 Speed measurement circuit

5.7 PIC INTERFACE



The microcontroller interface diagram is shown in the figure the motor parameters are interfaced to the PIC via ports. The port assignment tabular column is shown in Table 5.1

Table 5.1 Port assignments

PORT	ASSIGNMENT
RA0	VOLTAGE SENSE
RA1	CURRENT SENSE
RB0	SPEED SENSE
RC0	ALARM CIRCUIT
RD0 – RD7	LCD DATA INTERFACE
RE0 – RE2	LCD CONTROL PINS
RC6, RC7	RS 232 INTERFACE

The sensed signals are digitalized using the in-built A/ D converter in the PIC microcontroller. The digital value is then converted into its equivalent numeric value and it is displayed in the LCD. These data are given to the Fuzzy module as well as interfaced with the PC through RS232. The PIC also controls the relay - alarm circuit.

5.8 LCD DISPLAY

Liquid crystal displays (LCDs) have materials which combine the properties of both liquids and crystals. Rather than having a melting point, they have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an ordered form similar to a crystal.An LCD consists of two glass panels, with the liquid crystal material sandwiched in between them. The inner surface of the glass plates are coated with transparent electrodes which define the character, symbols or patterns to be displayed polymeric layers are present in between the electrodes and the liquid crystal, which makes the liquid crystal molecules to maintain a defined orientation angle. One each polarizer are pasted outside the two glass panels. This polarizer would rotate the light rays passing through them to a definite angle, in a particular direction When the LCD is in the off state, light rays are rotated by the two polarizer and the liquid crystal, such that the light rays come out of the LCD without any orientation, and hence the LCD appears transparent. When sufficient voltage is applied to the electrodes, the liquid crystal molecules would be aligned in a specific direction. The light rays passing through the LCD would be rotated by the polarizer, which would result in activating / highlighting the desired characters.The LCDs are lightweight with only a few millimeters thickness. Since the LCD's consume less power, they are compatible with low power electronic circuits and can be powered for long durations. The LCD doesn't generate light and so light is needed to read the display. By using backlighting,. reading is possible in the dark. The LCD's have long life and a wide operating temperature range. Changing the display size or the layout size is relatively simple which makes the LCDs more users friendly. The recent advances in technology have resulted in better legibility, more information displaying capability and a wider temperature range. These have resulted in the LCDs being extensively used in telecommunications and entertainment electronics.

The power supply should be of +5V, with maximum allowable transients of 10mv. To achieve a better / suitable contrast for the display, the voltage (VL) at pin 3 should be adjusted properly. A module should not be inserted or removed from a live circuit. The ground terminal of the power supply must be isolated properly so that no voltage is induced in it. The module should be isolated from the other circuits, so that stray voltages are not induced, which could cause a flickering display.

5.9 LCD MODULE – PIN DETAILS

The microcontroller is interfaced with the LCD module to display the parameters measured. The data from the controller is send to the 16 pin LCD module through 8 bit data bus. The pin details about the LCD module used, is given in the Table 5.2.

Table 5.2 Pin details of the LCD module

PIN NUMBER	DETAILS
1, 16	Ground
2	Vcc
3	Contrast control
4	RS
5	RW
6	EN

5.10 ALARM CIRCUIT

Pin RC0 of the microcontroller is assigned to this alarm circuit. It consists of a relay activated buzzer. On logic HIGH signal from PIC, will activate the buzzer. The alarm circuit used in this project is shown in the Figure 5.8

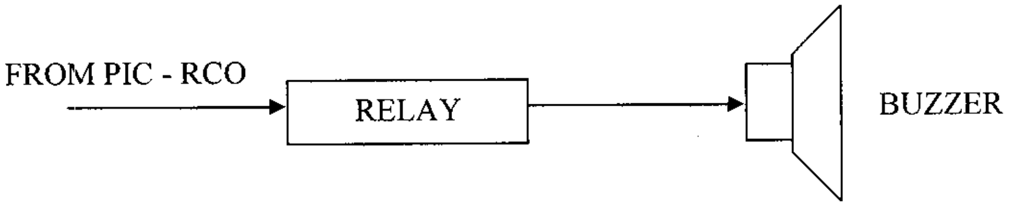


Figure 5.8 Alarm circuit

5.11 MICROCONTROLLER

Microcontroller is a general purpose device, which integrates a number of the components of a microprocessor system on to single chip. It has inbuilt CPU, memory and peripherals to make it as a mini computer. A microcontroller combines on to the same microchip:

- The CPU core
- Memory(both ROM and RAM)
- Some parallel digital i/o

Microcontrollers will combine other devices such as:

- A timer module to allow the microcontroller to perform tasks for certain time periods.
- A serial I/O port to allow data to flow between the controller and other devices such as a PIC or another microcontroller.
- An ADC to allow the microcontroller to accept analog input data for processing.

Microcontrollers are:

- Smaller in size
- Consumes less power
- Inexpensive

Micro controller is a stand alone unit, which can perform functions on its own without any requirement for additional hardware like i/o ports and external memory.

The heart of the microcontroller is the CPU core. In the past, this has traditionally been based on an 8-bit microprocessor unit. For example Motorola uses a basic 6800 microprocessor core in their 6805/6808 microcontroller devices.

In the recent years, microcontrollers have been developed around specifically designed CPU cores, for example the microchip PIC range of microcontrollers.

5.11.1 INTRODUCTION TO PIC:

The microcontroller that has been used for this project is from PIC series. PIC microcontroller is the first RISC based microcontroller fabricated in CMOS (complimentary metal oxide semiconductor) that uses separate bus for instruction and data allowing simultaneous access of program and data memory.

The main advantage of CMOS and RISC combination is low power consumption resulting in a very small chip size with a small pin count. The main advantage of CMOS is that it has immunity to noise than other fabrication techniques.

5.11.2 PIC 16F877:

Various microcontrollers offer different kinds of memories. EEPROM, EPROM, FLASH etc. are some of the memories of which FLASH is the most recently developed. Technology that is used in PIC16F877 is flash technology, so that data is retained even when the power is switched off. Easy Programming and Erasing are other features of PIC 16F877.

5.11.3SPECIAL FEATURES OF PIC MICROCONTROLLER:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory,
  Up to 368 x 8 bytes of Data Memory (RAM)
  Up to 256 x 8 bytes of EEPROM data memory
- Pin out compatible to the PIC16C73/74/76/77
- Interrupt capability (up to 14 internal/external
- Eight level deep hardware stack

- Direct, indirect, and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC Oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS EPROM/EEPROM technology
- Fully static design
- In-Circuit Serial Programming (ICSP) via two pins
- Only single 5V source needed for programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.5V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
  - ✓ < 2mA typical @ 5V, 4 MHz
  - ✓ 20mA typical @ 3V, 32 kHz
  - ✓ < 1mA typical standby current

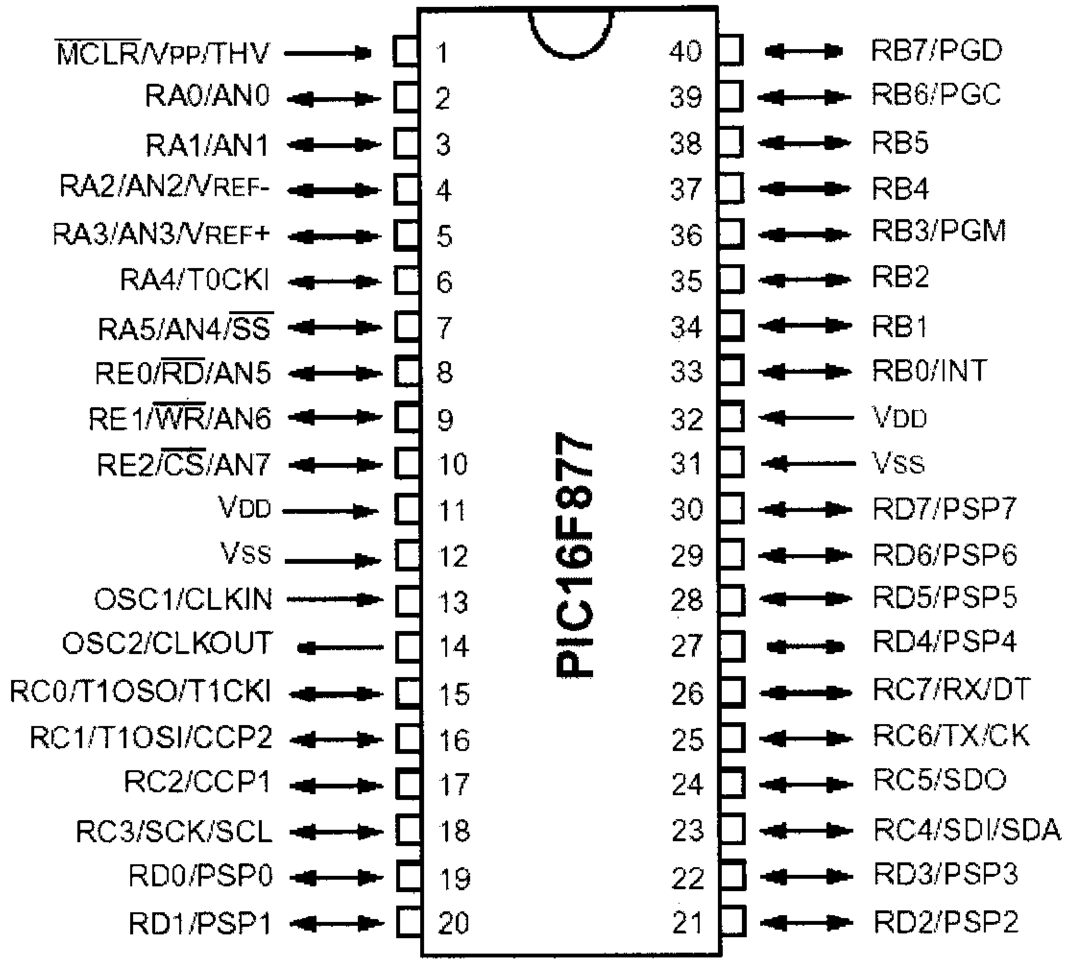
5.11.4 PERIPHERAL FEATURES:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modu
  - ✓ Capture is 16-bit, max resolution is 12.5 ns,
  - ✓ Compare is 16-bit, max resolution is 200 ns,
  - ✓ PWM max. resolution is 10-bi
- 10-bit multi-channel Analog-to-Digital converter
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9- Bit addresses detection.

➢ Brown-out detection circuitry for Brown-out Reset (BOR)

5.11.5 PIC 16F877 - PIN CONFIGURATION

The pin out diagram for 16F877 is shown in the Figure 5.9.



5.12 I/O PORTS

512.1 PORTA AND TRISA REGISTER:

PORTA is a 6-bit wide bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (=1) will make the corresponding PORTA pin an input, i.e., put the corresponding output driver in a Hi-impedance mode. Clearing a TRISA bit (=0) will make the corresponding PORTA pin an output, i.e., put the contents of the output latch on the selected pin. Reading the PORTA register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore a write to a port implies that the port pins are read; this value is modified, and then written to the port data latch. PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

5.12.2 PORTB AND TRISB REGISTER:

PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. Setting a TRISB bit (=1) will make the corresponding PORTB pin an input, i.e., put the corresponding output driver in a hi-impedance mode. Clearing a TRISB bit (=0) will make the corresponding PORTB pin an output, i.e., put the contents of the output latch on the selected pin. Three pins of PORTB are multiplexed with the Low Voltage Programming function; RB3/PGM, RB6/PGC and RB7/PGD. The alternate functions of these pins are described in the Special Features Section. Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups.

Four of PORT B's pins, RB7:RB4, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e. any RB7:RB4 pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RB Port Change Interrupt with flag bit RBIF (INTCON<0>). This interrupt can wake the device from SLEEP.

5.12.3 PORTC AND THE TRISC REGISTER:

PORTC is an 8-bit wide bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (=1) will make the corresponding PORTC pin an input, i.e., put the corresponding output driver in a hi-impedance mode. Clearing a TRISC bit (=0) will make the corresponding PORTC pin an output, i.e., put the contents of the output latch on the selected pin. PORTC is multiplexed with several peripheral functions. PORTC pins have Schmitt Trigger input buffers.

5.12.4 PORTD AND TRISD REGISTERS:

This section is not applicable to the 28-pin devices. PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output. PORTD can be configured as an 8-bit wide microprocessor Port (parallel slave port) by setting control bit PSPMODE (TRISE<4>). In this mode, the input buffers are TTL.

5.12.5 PORTE AND TRISE REGISTER:

PORTE has three pins RE0/RD/AN5, RE1/WR/AN6 and RE2/CS/AN7, which are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers.

The PORTE pins become control inputs for the microprocessor port when bit PSPMODE (TRISE<4>) is set. In this mode, the user must make sure that the TRISE<2:0> bits are set (pins are configured as digital inputs). Ensure ADCON1 is configured for digital I/O. In this mode the input buffers are TTL.

PORTE pins are multiplexed with analog inputs. When selected as an analog input, these pins will read as '0's. TRISE controls the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

5.13 SOFTWARE DESCRIPTION

In the software part, two different modules are present. One is for the implementation of the embedded system in the microcontroller (PIC 16F877A),which based on simulation results.

5.13.1 ALGORITHM FOR PIC PROGRAMMING

The PIC 16F877A microcontroller used in this project is programmed using Hi-tech C. The program is converted into machine language using MPLAB software. The algorithm and flowchart (Figure5.9) for this module are as follows:

Step1. Start the program.

Step2. Assign the ports in the microcontroller as input/ output depending upon the requirements.

Step3. Initialize the Registers as per the requirements.

Step4. Enable the global and peripheral interrupt.

Step5. The current measured in one of the phase is multiplied by a suitable value to obtain the original value.

Step6. Similar procedure is repeated for voltage also.

Step7. Pulses are counted and the speed is calculated.

Step8. The values of voltage, current and speed are given to the LCD display.

Step9. The measured values, current and speed are given to the rule based module for diagnostics.

Step10.Details about the fault is given to the LCD display.

Step11.The values of voltage, current, speed and the diagnostic results are given to the PC.

Step12.If the severity of the fault is more; an alarm will be activated as a warning signal.

Step13.Else step5 to step11 are repeated continuously.

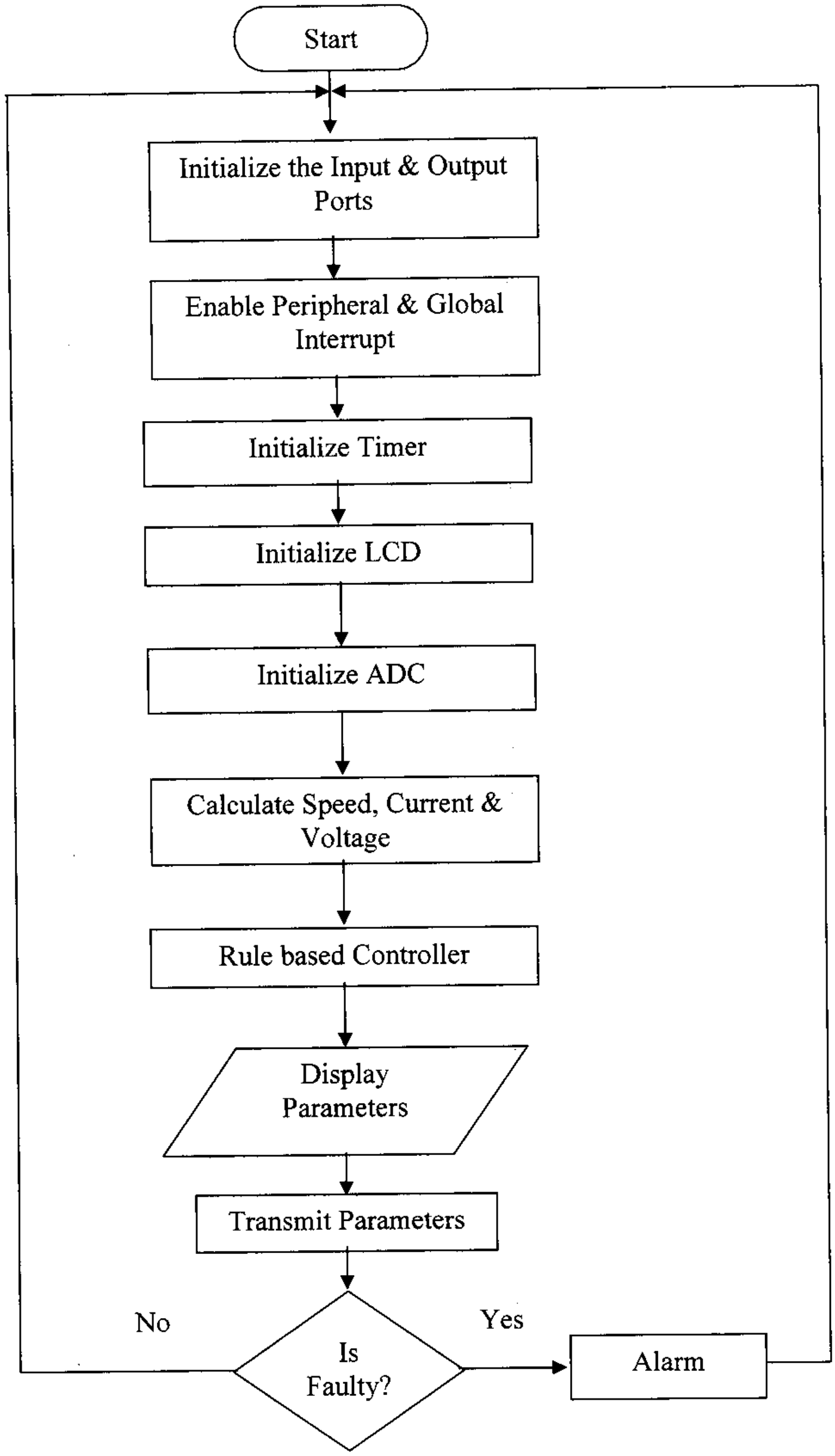


Figure 5.9 Flowchart

5.13.2 FUZZY BASED PROGRAMMING

This is a software developed using MATLAB7 and its features, Graphical User Interface. This is programmed to capture the current signal through sound card and required data is extracted from it. The algorithm and flowchart (Figure 5.10) for this module are as follows:

Step1. Start the program

Step2. Get the analog input through sound card

Step3. Signal processing is done on the signal to make it digitized.

Step4. Fast Fourier Transform is performed on the processed signal

Step5. Required data is acquired and given to the Fuzzy module

Step6. Diagnostic result is displayed on the screen.

Step7. The process is repeated from Step2 to Step6 for every 10 or 20 seconds as per the requirement.

This software is based on stator current analysis. It is featured in such a way that the user can be able to feed the data about the test motor details and the bearing dimensions. From these input data, the developed software will calculate the characteristic frequencies of the bearing failure. Fast Fourier transform is performed and the magnitude components are extracted for the corresponding characteristic frequency.

The current spectrum is displayed for visual inspection at a particular range. The data extracted undergoes fuzzy diagnosis and the result of the fuzzy system is displayed in figure 5.11. It gives the condition of the bearing as the diagnostic result.

This is an online process which keeps repeating the procedure after every 10 or 20 seconds as per our requirements.

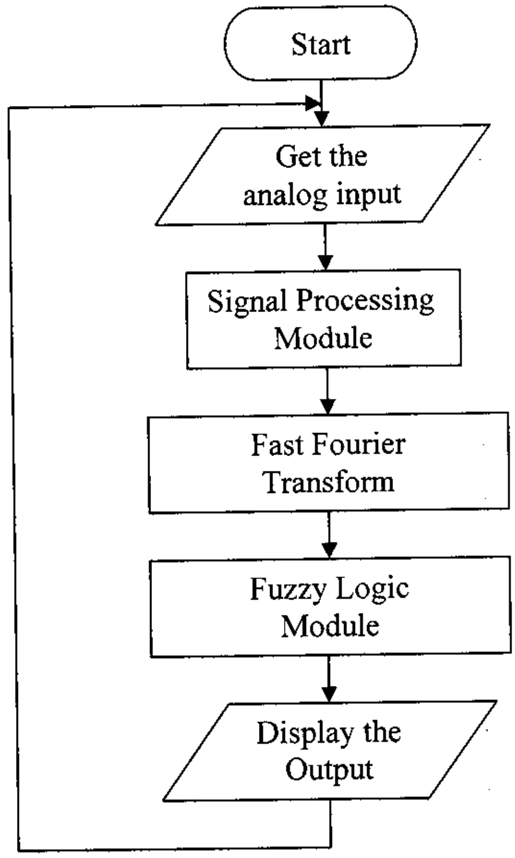


Figure 5.10 Flowchart

5.13.3 EXPERIMENTAL RESULTS

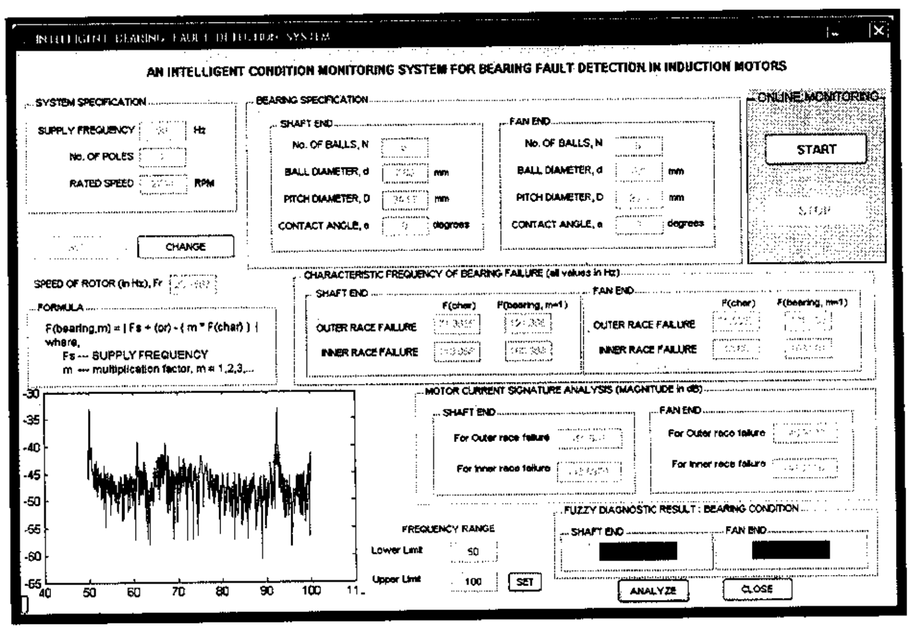


Figure 5.11 experimental results fuzzy logic based bearing fault detection

CONCLUSION

The bearing fault detection methods for three phase induction motor have been investigated using neural network and fuzzy logic. The technique proposed is based on monitoring the current spectrum and speed .The current spectrum value and speed are taken as the inputs for neural network and fuzzy logic fault detector .The performance of neural and fuzzy fault detection is compared in terms of percentage of error. From the simulation results, it is inferred that the fuzzy logic based fault diagnosis gives the reduced percentage error than the neural network based fault diagnosis. Hence fuzzy logic based fault diagnosis is the effective method for fault detection. The fuzzy logic based fault detection scheme has been implemented in real time. The experimental results were verified with 1HP 415 Volts 50Hz 960-rpm three phase induction motor and are presented in this report.

REFERENCES

[1] R.C.Kryter et al., "Condition Monitoring Of Machinery Using Motor Current Signature Analysis," Sound Vib.,pp 14-21,Sept.1989.

[2] A.J.M. Cardoso et al., "Computer-Aided Detection Of Air Gap Eccentricity In Operating Three-Phase Induction Motors By Park's Vector Approach," IEEE Trans.Ind.Applicat., Vol.29,pp.897-901, Sept./Oct. 1993.

[3] R.R schoen et al., "Effects Of Time-Varying Loads On Rotor Fault Detection In Induction Machine," IEEE Trans.ind, applicat. Vol, 31, pp 900-906, July/Aug .1995.

[4] R.r. Schoen et al., "Evaluation And Implementation of a System To Eliminate Arbitary Load Effects In Current –Current Based Monitoring Of Induction Machines, "IEEE Trans.Ind .,Vol.33, pp.1571-1577, Nov./Dec. 1997.

[5] J Penman et al.,"Condition Monitoring Of Electrical Drives," Proc .Inst.Elect. Eng., pt.b, Vol. 133, pp 142-148, May 1986.

[6] S. Chen et al., "A New Approach To Motor Condition Monitoring In Induction Motor Drives,"IEEE Trans.Ind.Applicat., Vol.30, pp. 905-911 July/Aug. 1994.

[7] M.E Steele, R .A Ashen, and L.G.knight,"An Electrical Method For Condition Monitoring Of Motors," in int.conf. Ele.Mach –Design and applicat. July 1982, no 213.pp, 231-235.

[8]"Methods Of Motor Current Signature Analysis," Elec.Mach. Power sys, vol, 20, no 5, pp 463-474, Sept, 1992.

[9] R.R. Schoen et al, "Motor Bearing Damage Detection Using Stator Current Monitoring,"IEEE Trans.ind.applicat., vol 31, pp.1274-1279 Nov/Dec 1995.

[10] Riddle J 1955 Ball bearing maintenance (Norman , OK: Univ . Of Oklohama)

[11] Eschmann P,Hasbargen L,weigand K 1958 Ball and roller bearings : Their theory, design, and app-lication (London:K G heyden)

[12] R.L.schiltz,"Forceing Frequency Identification Of Rolling Element Bearings," sound vib, sound Vib. Pp.16-19, May 1990.

[13]Fiorenzo fillippetti, Giovanni Franceschini , and peter vas, "Recent Development Of Induction Motor Drives Fault Diagnosis Using AI Techniques ", IEEE Trans. Industrial Electronics.,vol.47,No.,pp.994-1003.

[14] Fillippetti. F, Franceschini.G and Tassoni.C,' Neural Networks Aided On Line Diagnosis Of Induction Motors Faults',IEEE Trans. Idl .Applicat., Vol 31, No 4 , pp 316-323,1995

[15] Rajasekaran .S and Vijayalaksmi Pai.G.A, ' Neural networks, Fuzzy Logic and Genetic Algoritham Synthesis and Applicatioins', Preentice Hall of India ,1995

APPENDIX A: MATLAB PROGRAMMING CODE

```
function setf_Callback(hObject, eventdata, handles)
        a=str2double(get(handles.freq,'String'));
        b=str2double(get(handles.poles,'String'));
        c=str2double(get(handles.speed,'String'));
        d=str2double(get(handles.N_SE,'String'));
        e=str2double(get(handles.d_SE,'String'));
        f=str2double(get(handles.Dp_SE,'String'));
        g=str2double(get(handles.a_SE,'String'));
        h=str2double(get(handles.N_FE,'String'));
        i=str2double(get(handles.d_FE,'String'));
        j=str2double(get(handles.Dp_FE,'String'));
        k=str2double(get(handles.a_FE,'String'));
        if( isnan(a) )
          warndlg('FREQUENCY FIELD IS EMPTY','! !  Warning  ! !');
        else
          if( isnan(b) )
            warndlg('No. OF POLES FIELD IS EMPTY','! !  Warning  ! !');
          else
            if( isnan(c) )
              warndlg('SPEED FIELD IS EMPTY','! !  Warning  ! !');
            else
              if( isnan(d) )
                warndlg('FIELD IS EMPTY','! !  Warning  ! !');
              else
                if( isnan(e) )
                  warndlg('FIELD IS EMPTY','! !  Warning  ! !');
                else
                  if( isnan(f) )
                    warndlg('FIELD IS EMPTY','! !  Warning  ! !');
                  else
                    if( isnan(g) )
```

```
    warndlg('FIELD IS EMPTY','! !  Warning  ! !');
else
    if( isnan(h) )
        warndlg('FIELD IS EMPTY','! !  Warning  ! !');
    else
        if( isnan(i) )
            warndlg('FIELD IS EMPTY','! !  Warning  ! !');
        else
            if( isnan(j) )
                warndlg('FIELD IS EMPTY','!! Warning !!');
            else
                if( isnan(k) )
                    warndlg('FIELD IS EMPTY','!!Warning!!');
                else
                    set(handles.freq,'Enable','off');
                    set(handles.poles,'Enable','off');
                    set(handles.speed,'Enable','off');
                    set(handles.N_SE,'Enable','off');
                    set(handles.d_SE,'Enable','off');
                    set(handles.Dp_SE,'Enable','off');
                    set(handles.a_SE,'Enable','off');
                    set(handles.N_FE,'Enable','off');
                    set(handles.d_FE,'Enable','off');
                    set(handles.Dp_FE,'Enable','off');
                    set(handles.a_FE,'Enable','off');
                    Frotor = (c)/60;
                    set(handles.Fr,'String',Frotor);

                    Fouter = (d/2)*(Frotor)*(1-((e/f)*(cosd(g))));
                    set(handles.Fo_SE,'String',Fouter);
                    Finner = (d/2)*(Frotor)*(1+((e/f)*(cosd(g))));
                    set(handles.Fi_SE,'String',Finner);
                    Fbouter = a + Fouter;
                    set(handles.Fbo_SE,'String',Fbouter);
```

```
                    Fbinner = a + Finner;
                    set(handles.Fbi_SE,'String',Fbinner);

                    F2outer = (h/2)*(Frotor)*(1-((i/j)*(cosd(k))));
                    set(handles.Fo_FE,'String',F2outer);
                    F2inner = (h/2)*(Frotor)*(1+((i/j)*(cosd(k))));
                    set(handles.Fi_FE,'String',F2inner);
                    F2bouter = a + F2outer;
                    set(handles.Fbo_FE,'String',F2bouter);
                    F2binner = a + F2inner;
                    set(handles.Fbi_FE,'String',F2binner);
                    set(handles.analyze,'Enable','on');
                    set(handles.startf,'Enable','on');
                    set(handles.stopf,'Enable','off');
                    set(handles.setf,'Enable','off');
                    set(handles.changef,'Enable','on');
                   end
                  end
                 end
                end
               end
              end
             end
            end
           end
          end
         end
    clear;

function analyze_Callback(hObject, eventdata, handles)
        global ULIMIT;
        global LLIMIT;
        global FS;
        global SIZE;
```

```
FS = 11025;
SIZE = 131072;
set(handles.ll,'Enable','on');
set(handles.ul,'Enable','on');
set(handles.fset,'Enable','on');
y = wavrecord(5*FS,FS,'double');
Y = fft(y,SIZE);
Pyy = abs(Y)/SIZE;
Plog = 10*log10(Pyy);
f = FS*(1:SIZE)/SIZE;

f1 = str2double(get(handles.Fbo_SE,'String'));
f2 = str2double(get(handles.Fbi_SE,'String'));
f3 = str2double(get(handles.Fbo_FE,'String'));
f4 = str2double(get(handles.Fbi_FE,'String'));

% MAGNITUDE - OUTER - SHAFT END
x=(f-f1)/f1;
for i=1:SIZE
   if( x(i) <0)
      x(i)=x(i)*(-1);
   end
end
[c,l]=min(x);
abc = Plog(l);
set(handles.Mo_SE,'String',abc);

% MAGNITUDE - INNER - SHAFT END
x=(f-f2)/f2;
for i=1:SIZE
   if( x(i) <0)
      x(i)=x(i)*(-1);
   end
end
```

```
out4 = evalfis( mife,fismat);
%SHAFT END DIAGNOSIS
if( out1 > -1 && out1 < 0.1 )
   out1=0;
else
   if( out1 > 0.4 && out1 < 0.6 )
      out1=0.5;
   else
      if( out1 > 0.9 && out1 < 1.1 )
         out1=1;
      end
   end
end
if( out2 > -1 && out2 < 0.1 )
   out2=0;
else
   if( out2 > 0.4 && out2 < 0.6 )
      out2=0.5;
   else
      if( out2 > 0.9 && out2 < 1.1 )
         out2=1;
      end
   end
end

if(out1==0 || out2==0)
   set(handles.secon,'String','Poor');
   set(handles.secon,'BackgroundColor','Red');
else
   if(out1==1 && out2==1)
      set(handles.secon,'String','Good');
      set(handles.secon,'BackgroundColor','Green');
   else
      set(handles.secon,'String','Fair');
```

```
    set(handles.secon,'BackgroundColor','Yellow');
  end
end

%FAN END DIAGNOSIS
if( out3 > -1 && out3 < 0.1 )
  out3=0;
else
  if( out3 > 0.4 && out3 < 0.6 )
    out3=0.5;
  else
    if( out3 > 0.9 && out3 < 1.1 )
      out3=1;
    end
  end
end
if( out4 > -1 && out4 < 0.1 )
  out4=0;
else
  if( out4 > 0.4 && out4 < 0.6 )
    out4=0.5;
  else
    if( out4 > 0.9 && out4 < 1.1 )
      out4=1;
    end
  end
end
if(out3==0 || out4==0)
  set(handles.fecon,'String','Poor');
  set(handles.fecon,'BackgroundColor','Red');
else
  if(out3==1 && out4==1)
    set(handles.fecon,'String','Good');
    set(handles.fecon,'BackgroundColor','Green');
```

```
        else
            set(handles.fecon,'String','Fair');
            set(handles.fecon,'BackgroundColor','Yellow');
        end
    end
    %END OF DIAGNOSIS
```

function closef_Callback(hObject, eventdata, handles)

```
    close;
```

function startf_Callback(hObject, eventdata, handles)

```
    set(handles.analyze,'Enable','off');
    set(handles.changef,'Enable','off');
    set(handles.stopf,'Enable','on');
    set(handles.startf,'Enable','off');
    pause(2);
    global FLAG
    while FLAG==1
        eval('INTELLIGENT_BEARING_FAULT_DETECTION_SYSTEM(''
        analyze_Callback'',gcbo,[],guidata(gcbo))');
        pause(20);
    end
```

function stopf_Callback(hObject, eventdata, handles)

```
    set(handles.analyze,'Enable','on');
    set(handles.changef,'Enable','on');
    set(handles.stopf,'Enable','off');
    set(handles.startf,'Enable','on');
    global FLAG
    FLAG=0;
```

APPENDIX B: PIC PROGRAMMING CODE

```
#include<pic.h>
#include<lcd.h>
unsigned int volt1,cur1;
unsigned char
V1HUN,V1TEN,V1ONE,I1HUN,I1TEN,I1ONE,N1TENTHO,N1THO,N1HUN,N1T
EN,N1ONE;
unsigned int I1,I2,V1,F1,N1,N2,t,rpm;
unsigned int t1,t2,set,fsch,fact;
signed float f;
void main()
{
        TRISA=0XFF;
        TRISD=0X00;
        TRISE=0X00;
        PORTD=0;
        PORTE=0;

        TRISC=0X80;
        PORTC=0;

        SPBRG=0X19;
        BRGH=1;
        TXSTA=0X24;
        RCSTA=0X80;

        TRISB=0x03;
        PORTB=0;
        OPTION=0x88;
        GIE=PEIE=INTE=T0IE=1;

        lcd_init();
```

```
cursor_loc(0X80);
display_string("BEARING COND'T");
cursor_loc(0XC4);
display_string("CHECKER");

delay2();
delay2();
delay2();

lcd_init();

while(1)
{
        ADCON0=0X81;
        delay();
        ADGO=1;
        while(ADGO);        //status check
        volt1=ADRESH*256+ADRESL;
        delay();
        delay();

        ADCON0=0X89;
        delay();
        ADGO=1;
        while(ADGO);        //status check
        cur1=ADRESH*256+ADRESL;
        delay();
        delay();

        V1=volt1/2;
        I1=cur1/1;
        I1=cur1/1;
        I2=I1;
```

```
V1HUN=V1/100;
V1=V1%100;
V1TEN=V1/10;
V1=V1%10;
V1ONE=V1;
I1HUN=I1/100;
I1=I1%100;
I1TEN=I1/10;
I1=I1%10;
I1ONE=I1;
f=1000000/t;
rpm=f*60;
N1=rpm;
N2=N1;
N1TENTHO=N1/10000;
N1=N1%10000;
N1THO=N1/1000;
N1=N1%1000;
N1HUN=N1/100;
N1=N1%100;
N1TEN=N1/10;
N1=N1%10;
N1ONE=N1;

clear_lcd();
cursor_loc(0X80);
display_string("V=");
display_data(V1HUN);
display_data(V1TEN);
display_data(V1ONE);
cursor_loc(0x87);
display_string("I=");
display_data(I1HUN);
display_string(".");
```

```
display_data(I1TEN);
display_data(I1ONE);
cursor_loc(0xC0);
display_string("N=");
display_data(N1THO);
display_data(N1HUN);
display_data(N1TEN);
display_data(N1ONE);

TXREG='V';
while(!TRMT);
delay();
TXREG='=';
while(!TRMT);
delay();
TXREG=0x30+V1HUN;
while(!TRMT);
delay();
TXREG=0x30+V1TEN;
while(!TRMT);
delay();
TXREG=0x30+V1ONE;
while(!TRMT);
delay();
TXREG=';';
while(!TRMT);
delay();
TXREG='I';
while(!TRMT);
delay();
TXREG='=';
while(!TRMT);
delay();
TXREG=0x30+I1HUN;
```

```
delay();
TXREG='B';
while(!TRMT);
delay();
TXREG='=';
while(!TRMT);
delay();
cursor_loc(0xC7);
display_string("B=");

t=0;

if(I2>=70 && I2<90)
{
        if(N2>=850 && N2<890)
        {
                display_string("POOR");

                TXREG='P';
                while(!TRMT);
                delay();
                TXREG='O';
                while(!TRMT);
                delay();
                TXREG='O';
                while(!TRMT);
                delay();

                TXREG='R';
                while(!TRMT);
                delay();

                RC0=1;
        }
```

```
                delay();
                TXREG='O';
                while(!TRMT);
                delay();
                TXREG='O';
                while(!TRMT);
                delay();
                TXREG='D';
                while(!TRMT);
                delay();
                RC0=0;
        }
}

if(I2>=130 && I2<170)
{
        if(N2>=850 && N2<890)
        {
                display_string("GOOD");

                TXREG='G';
                while(!TRMT);
                delay();
                TXREG='O';
                while(!TRMT);
                delay();
                TXREG='O';
                while(!TRMT);
                delay();
                TXREG='D';
                while(!TRMT);
                delay();

                RC0=0;
```

```
}

if(N2>=890 && N2<930)
{
        display_string("FAIR");
        TXREG='F';
        while(!TRMT);
        delay();
        TXREG='A';
        while(!TRMT);
        delay();
        TXREG='I';
        while(!TRMT);
        delay();
        TXREG='R';
        while(!TRMT);
        delay();
        RC0=0;
}

if(N2>=930 && N2<1000)
{
        display_string("POOR");

        TXREG='P';
        while(!TRMT);
        delay();

        TXREG='O';
        while(!TRMT);
        delay();
        TXREG='O';
        while(!TRMT);
        delay();
```

```
                    TXREG='R';
                    while(!TRMT);
                    delay();

                    RC0=1;
                }
            }
            TXREG=' ';
            while(!TRMT);
            delay();
            delay2();

        }
}
delay()
{
        unsigned int i=0;
        for(i=0;i<=400;i++);
}
delay2()
{
        unsigned int j=0;
        for(j=0;j<=40000;j++);
}
void interrupt isr()
{
/**********SPEED START***********/
        if(INTF==1)
        {
            INTF=0;
            t1=t2;
            t2=TMR0;
            if(set==0)
                t=t2-t1;
```

```
            else
            {
                t=((set*256)-t1)+t2;
                set=0;
            }
        }
        if(set==1000)
            set=0;
        if(T0IF==1)
        {
            T0IF=0;
            set++;
        }
/***********SPEED END**********/
}
```