AN APPROACH TO CONDITION MONITORING AND FAULT DIAGNOSIS OF INDUCTION MACHINES USING KEY PERFORMANCE INDICATORS

by

Forrest Edward Suter

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Electrical Engineering

Charlotte

2017

Approved by:

____
Dr. Robert Cox

____
Dr. Babak Parkhideh

____
Dr. Madhav Manjrekar

ABSTRACT

FORREST EDWARD SUTER. An Approach To Condition Monitoring And Fault Diagnosis Of Induction Machines Using Key Performance Indicators. (Under the direction of DR ROBERT COX)

One of the great difficulties associated with monitoring the condition of an electric motor is that the user must typically possess some degree of expertise to distinguish between a normal operating condition and a potential failure. The typical approach is to monitor various electrical and mechanical parameters such as current and vibration. Key Performance Indicators (KPI) are extracted from these parameters and are monitored for changes. The changes could result from any number of sources, including those related to normal operating conditions. For instance, cyclically varying loads and mechanical unbalances can have the same effect on KPIs extracted from the motor current spectrum as a broken rotor bar or bent shaft. These types of issues, in combination with the vast amount of available information, make it difficult to determine a set of rules for fault detection. When using hard-and-fast rules, for instance, it is possible to omit certain situations out of ignorance or fail to implement rules that deal with the dynamic nature of certain conditions. Without knowing all possible fault conditions and symptoms, the developer finds himself in a difficult position. In this thesis, a new approach is proposed that does not require a rule based approach but instead relies on an unsupervised fault detection algorithm that learns the normal operating condition of a particular motor and then monitors for changes that indicate a potential fault condition has occurred.

DEDICATION

This thesis is dedicated to my wife, Gwen, for her love and support throughout my education.

ACKNOWLEDGEMENTS

Special thanks for my advisor Dr. Robert Cox for his teaching, support and encouragement during the development of this research.

I would also like to thank Dr. Babak Parkhideh and Dr. Madhav Manjrekar for their reviews and discussions during this research.

Finally, I would like gratefully acknowledge AREVA NP for the generous donation of the SpectraQuest Machinery Fault Simulator and accelerometers that were used during this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AC	Alternating Current
AFW	Auxiliary Feedwater
CT	Current Transformer
DAQ	Data Acquisition
DC	Direct Current
DCS	Distributed Control System
DFT	Discrete Fourier Transform
DTFT	Discrete-time Fourier Transform
EPIC	Energy Production and Infrastructure Center
EPRI	Electric Power Research Institute
FFT	Fast Fourier Transform
FLA	Full Load Amps
HI	Health Indicator
HP	Horsepower
HPI	High Pressure Injections
ICP	Integrated Circuit Piezoelectric
IEC	International Electrotechnical Commision
IEEE	Institute of Electrical and Electronic Engineers
IEPE	Internal Electronic Piezoelectric
IFFT	Inverse Fast Fourier Transform
ISO	International Standards Organization

KPI	Key Performance Indicators
LCD	Liquid Crystal Display
MCC	Motor Control Center
MCSA	Motor Current Signature Analysis
MMF	Magneto Motive Force
NEMA	National Electrical Manufacturers Association
NILM	Non-Intrusive Load Monitoring
ORNL	Oak Ridge National Laboratory
PCA	Principal Component Analysis
PCB	Printed Circuit Board
PE	Piezoelectric
PR	Piezoresistive
RCP	Reactor Coolant Pumps
RMS	Root Mean Square
RPM	Revolutions Per Minute
SWGR	Switchgear
TGP	Turned Ground Polished
THD	Total Harmonic Distortion
TIR	Total Indicated Runout
VA	Volt Amperes
VFD	Variable Frequency Drives
WH	Winding Harmonics

1 INTRODUCTION

The majority of industrial applications and power generation facilities utilize induction motors for the transformation of electrical power to the mechanical form. The induction motor is commonly regarded as the workhorse of industry as there are multiple applications which generally include pumps, conveyors, machine tools, presses and packaging equipment. Some of these applications are in hazardous locations, operating under harsh environments, such as the chemical or nuclear power industries. The reliability, efficiency and rate of maintenance are the characteristics of the induction motor that have led to their widespread use in industrial processes. In addition, the production requirements of most industrial processes can be fulfilled by the wide range of power ratings, which is from hundreds of watts to megawatts, of the induction motor. A growing number of induction motors operate using inverter drives. In this case, the motor is not directly connected to the power grid but inverter-fed. The inverter provides voltage of variable amplitude and frequency in order to vary the mechanical speed.

However, the operating conditions for induction motors within the industrial applications make motors vulnerable to many types of faults. A fault condition that is not detected at an incipient stage may become catastrophic with the induction motor and associated equipment suffering severe damage. Left undetected, motor faults cascade into motor failure, which in turn contributes to unscheduled production shutdowns. Such shutdowns are costly, in terms of production time, maintenance cost, wasted raw materials in manufacturing and lost revenue in power generation. Common faults occurring in electrical motor drive systems can be classified as follows:

- Electrical faults: stator winding short circuit, broken rotor bar, broken end-ring, and inverter faults.
- Mechanical faults: rotor eccentricity, bearing faults, shaft misalignment, load faults: unbalance, gearbox fault or general failure in the load part of the drive.

Several alternatives have been used in industry to prevent severe damage to induction motors. Scheduled maintenance is implemented to verify the integrity of the motor, lubrication problems, bearing conditions, stator winding and rotor cage integrity. However, most maintenance is performed when the motor is de-energized, which implies production shutdown. Redundancy is one method to prevent production shutdowns, but not induction motor failure. Employing redundancy requires two sets of equipment, including the induction motors; the first set operates unless there is failure where the second set takes over. This solution is not feasible in many industrial applications due to physical space and high cost limitations.

Monitoring and fault detection improves the reliability and availability of the existing system. Since the various faults degrade relatively slowly, there is a potential for fault detection at an early stage. This avoids sudden and total failures, which have serious consequences. In the context of condition monitoring it is important to differentiate fault detection from fault diagnosis. Fault detection is the decision if the fault is present or not, while fault diagnosis provides more information about the root cause of the failure or localization of the failure. This information can be used to minimize downtime and to schedule adequate maintenance action. Using a signal analysis approach to condition monitoring, no dynamic model of the real process is required with the fault detection being based on measured quantities. The fault detection and diagnosis is only based on processing and analysis of the measured signal.

The studies of induction motor behavior during abnormal conditions due to the presence of the fault and the possibility to detect these abnormal conditions have been a challenge to Electrical Engineers. There are many condition monitoring techniques available including axial flux, temperature, chemical, noise and vibration monitoring; although these require expensive sensors or specialized tools. Vibration monitoring is, however, the most established and commonly used approach for detecting mechanical faults.

The basic quantities associated with electromechanical plants, such as current and voltage are readily measured by tapping into the existing voltage and current transformers that are already installed as part of the protection system. As a result, current monitoring is non-invasive and can be implemented in the distribution systems Switchgear or Motor Control Centers (MCC) remotely from the motors being monitored [1]. A technique called 'Motor Current Signature Analysis (MCSA) is based on stator current monitoring of induction motors; therefore it is not very expensive.

Both vibration analysis and MCSA are commonly based on Fourier Transform in particular the Fast Fourier Transform (FFT) since it easy to implement and represents the spectra understandably. However, these have not been combined into a method for detecting and diagnosing fault conditions. Some of the methods require the operator or maintenance mechanic to physically go out to the motor and take measurements with handheld meters. Therefore, there is a need for reliable fault detection strategies for electrical and mechanical faults. It is useful to have a means for continuous remote monitoring of induction machines in unmanned/ hazardous locations (such as remote mining sites or petroleum processing plants) and in critical applications where the highest reliability is required.

1.1 Proposed Idea

The approach proposed in this thesis is a two-step process. The first step is to detect faults at an early stage based on comparison of key features of the equipment. The second step is to diagnosis the fault by close examination of the extracted features from recorded data. During Step 1, features of a healthy system are learned during normal operating conditions, and these features are used to create a vector space [2]. During operation, measured features are mapped onto this vector space. If a fault develops, the associated mapping will change. This approach is based on a modification of a common facial-recognition algorithm, and it essentially detects that the feature space no longer appears, as it should when healthy. The proposed approach uses a single quantity known as the Health Indicator to determine that system behavior has changed. This approach is shown to be far more powerful than more simplistic, deterministic rules. Once a change has been observed, the underlying indicators can be captured. The corresponding data sets can be used immediately for fault detection. In Step 2, individual fault related-features are investigated to diagnosis the fault condition. This diagnosis could be performed by human experts and computerized algorithms.

The goal of this research is a set of algorithms that are capable of detecting faults and diagnosing their causes so that equipment can be repaired before significant damage occurs. These algorithms can be operated in real time during equipment operation. At present, automated algorithms are not fully developed and remain a future goal for most organizations. This thesis provides an approach that could be a major stepping-stone for those goals.

2 LITERATURE REVIEW

This chapter presents a literature review of condition monitoring for induction motors. The basic description of the physical wonders related to induction motors and the common faults experienced in induction motors are discussed, as well as the commonly monitored parameters in induction motors.

2.1 Induction Motor

Industrial process and power generation, worldwide, utilize induction motors in the conversion of electrical energy into mechanical energy. The reliability, efficiency and rate of maintenance are the characteristics of the induction motor that have led to their widespread use in industrial applications, including pumps, fans, air compressors, mixers, and conveyor belts, as well as many other industrial applications. Moreover, induction motors may be supplied directly from a constant frequency sinusoidal power supply or by an Alternating Current (AC) variable frequency drive. Due to the large range of types and applications of induction motors, the focus of this discussion will be on those studied in this thesis. In other words, the focus is on the three-phase induction motor, which is a type of asynchronous motor.

Induction motors have uniform air-gaps unlike the Direct Current (DC) motors. The stator is constructed of high-grade steel laminations; refer to Figure 2.1.1 [3]. The three phase windings are placed in slots cut on the inner surface of the stator frame. The rotor windings can either be squirrel cage or wound-rotor type. The squirrel cage rotors are embedded with copper or aluminum rods shorted on both ends by copper or aluminum rings. The wound rotors have a similar form of windings as the stator. The terminals of the windings are connected to slip-rings [4].The squirrel cage motors are more rugged and economical compared to the wound-rotor motors. The windings on the stator and rotor (wound-rotor type) are distributed over several slots to improve the Magneto Motive Force (MMF) waveform and to create a smooth torque output developed by the machine. The MMF is a measure of the strength of the magnetic field, it is proportional to the number of turns in the coil and current that flows through this coil. The measure of MMF in a coil is ampere-turns. A current flowing in a distributed winding results in essentially a sinusoidal space distribution of MMF. If balanced three phase currents flow through these three-phase distributed windings, a constant rotating magnetic field of constant speed and amplitude is produced in the air-gap and this will induce current in the rotor circuit to produce torque [4].

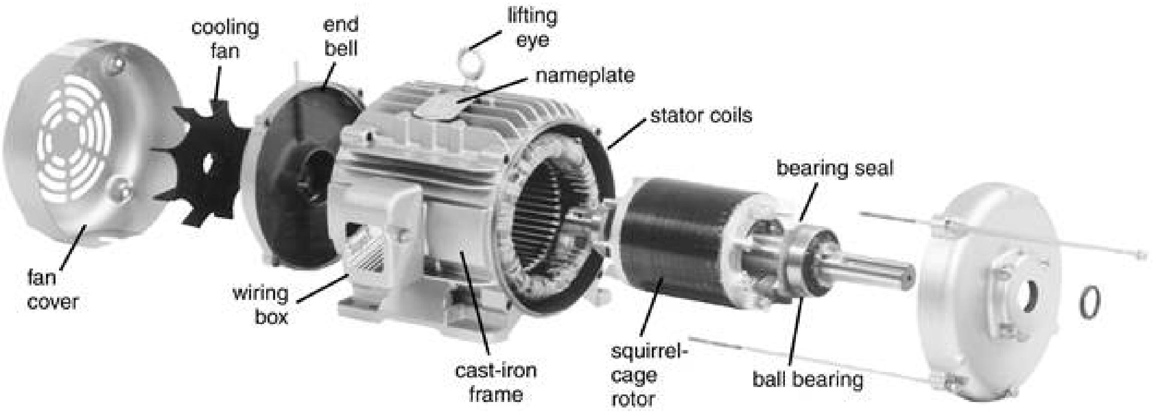


Figure 2.1.1: Exploded View of a Three-Phase Induction Motor (Courtesy of Baldor Electric)

The rotor circuit can be left open or closed circuited depending on the mode of operation. In standstill, the rotor circuit is considered an open circuit, if the three-phase windings are connected to a three-phase supply; a rotating magnetic field is produced in the air gap, which rotates at synchronous speed n_s.

$$n_s = \frac{120}{p} f_1 \tag{2.1}$$

Where frequency f_1 is the power supply frequency and p is the number of pole pairs. Voltages are induced in both the stator and rotor windings at the same frequency f_1 due to the rotating magnetic field. During operation the rotor circuit is closed, the induced voltages in rotor produces rotor currents that produce its own field, which interacts with that of the air-gap magnetic field to produce a torque. The rotor will rotate in the direction of magnetic field in a manner such that the relative speed between the rotor windings and rotating decreases. The rotor, under steady state conditions will rotate at speed n_r, which is less than the synchronous speed n_s of the rotating field in the air-gap [4]. The difference in speed between rotor and synchronous speed is referred to as slip s.

$$s = \frac{n_s - n_r}{n_s} \tag{2.2}$$

The voltage and current in the rotor circuit are induced at frequency f_2:

$$f_2 = s f_1 \tag{2.3}$$

In conventional squirrel cage motors, at full load conditions the slip and current are low but the power factor and the efficiency are high. However, during direct-on line, startup conditions the torque and power factor are low but the current is high. These starting currents are in the order of 500 to 800% of the full load current. If the load requires a large starting torque, the motor will accelerate slowly. This will cause the starting currents to flow for a longer period of time, thereby creating overheating and damage to the insulation. In the Figure 2.1.2, a typical torque speed characteristic curve of an induction motor is shown [3].

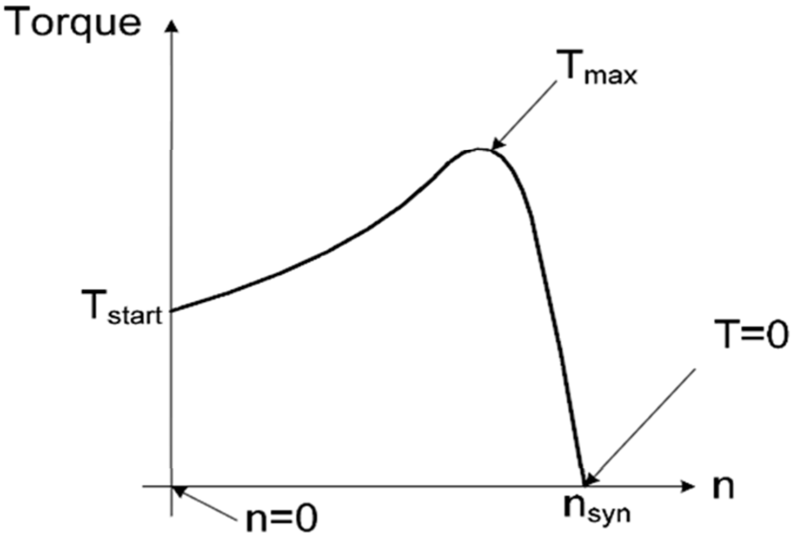


Figure 2.1.2: Typical Induction Motor Torque Speed Curve

2.2 Induction Motor Faults

Induction motors are frequently exposed to non-ideal or detrimental operating environments. These circumstances include; insufficient lubrication, inadequate cooling, frequent start/stop conditions, and overload. During these conditions, the motor is subjected to undesirable stresses, which makes the motor susceptible to faults or failure [5]. According to surveys by the Electric Power Research Institute (EPRI), the majority of the faults in induction motors statistically occur as follows: stator faults (37%), rotor cage (10%) failures (broken rotor bars/end-rings), air-gap irregularities and others (12%) and bearing failures (41%) [6]. These faults account for more than 90% of the induction failures [7], [8], [9], [10]. These common faults can be categorized into two groups:

electrical and mechanical faults. Electrical faults include stator and rotor faults, while mechanical faults include bearing faults and air-gap eccentricities.

The stator faults can be classified as lamination or frame fault (ground or circulation current) and stator winding faults. According to surveys, stator-winding faults account for 30 - 40% of all electrical failures [7], [8], [10]. Stator winding faults can be either turn-to-turn (short circuit within the same phase) or other phase combinations (phase-to-phase, phase to ground and or three phase faults). Most of the phase faults are characterized by high fault currents, which facilitate detecting them, whereas turn-to-turn faults (particularly with lower number of faulted windings) are characterized with lower fault current levels, which make these faults difficult to detect [20]. Many of these faults are caused by a combination of various stresses acting up on the stator, which can be categorized as: thermal, electrical, mechanical and environmental [7], [10], [11].

Thermal stresses are the most recognized causes for insulation degradation and ultimate motor failure. These can be categorized into three types: aging, cycling and overloading. As a rule of thumb, for every 10°C increase in temperature, the insulation life is halved due to thermal aging. Winding failure will occur irrespective of the degree of thermal aging if the other stresses mentioned above increase. The effect of temperature on thermal aging can be reduced to prolong thermal life by reducing the operating temperature or increasing the insulation class of winding material used. Thermal overloading occurs due to variation of the applied voltage, unbalanced phase voltages, obstructed ventilation and higher ambient temperature. As a rule of thumb, for every 3.5% voltage unbalances the winding temperature increases by 25% in phase with the highest current. Cycling gives expansion and contraction in the insulation system resulting in a weakening effect [7], [10].

Mechanical stresses that accelerate the degradation of insulation include coil movement and rotor striking the stator. The current in the stator windings produce a force on the coils that is proportional to the square of the current. This force is maximal under starting cycle or transient overloads, causing the coils to vibrate at twice the line frequency in both the radial and tangential directions. This movement causes damage to insulation. The rotor can strike the stator due to a number of reasons, such as shaft deflection, air-gap eccentricities and broken rotor bars. If the strike occurs during the starting cycle, the force of the rotor can cause the stator laminations to puncture the insulation, whereas during full speed steady operation it can cause premature grounding of the coil in the stator slots [7], [10].

Broken rotor bars account for more than 5% of the electric motor failure of induction motors [8], [9], [12]. There are numerous reasons for rotor bars and end rings breakages, these include magnetic stresses caused by electromagnetic stresses; thermal stresses due to abnormal operating duty including imbalance and overload conditions; inadequate casting, fabrication procedures or overloading; contamination and abrasion due to poor operating conditions; and lack of maintenance. Because of the reasons mentioned above, the rotor bars may be damaged and simultaneously an imbalance condition may occur. Rotor asymmetries result in asymmetrical distribution of rotor currents. An incipient broken rotor bar aggravates exponentially in time, as excessive current flow is concentrated on adjacent bars instead of the broken bar, which propagates electrical stresses to adjacent areas. This progressively deteriorates the rotor part, degrades the motor's overall performance, and shortens its lifetime [8], [13].

Bearing faults account for 40% of induction motor failures [8], [14]. A majority of induction motors use ball or rolling element bearings, which consist of an inner and outer ring with a set of rolling elements placed in raceways rotating inside these rings [14]. Bearings operate in non-ideal conditions and are subjected to excessive loading which usually causes premature failures. Overheating due to temperatures in excess of 204.4°C can anneal the ball and ring material. The result of loss in hardness reduces the bearing capacity causing early failure. In extreme cases, the balls and rings will deform. The increase in temperature can also degrade or destroy the lubricant. Brinelling, which occurs when the loads exceed elastic limit of the ring material. Contamination due to dirty work areas, tools, fixtures, hands and grinding operations. Corrosion resulting from exposure to corrosive fluids or a corrosive atmosphere. Lubrication failures typically caused by restricted lubricant flow and excessive temperatures that degrade the lubricant properties. Misalignment caused by exceeding the alignment tolerances. The misalignment results in rises of temperatures of the bearings. Bearing faults or defects can be classified as outer race, inner race, rolling element and cage defects.

Air-gap eccentricities are conditions that occur when there is a non-uniform distance between the rotor and stator in the air-gap. Generally, there are three types of eccentricity: static, dynamic and mixed. In static eccentricity, the rotor geometrical center is identical to the rotational center but displaced with respect to the stator geometrical center. The point of minimal gap length is stationary with respect to stator. This type of eccentricity is caused by misalignment of the rotor axis within the stator. This condition may be attributed to manufacturing tolerances, an oval stator core, incorrect bearing positioning or bearing wear. In dynamic eccentricity, the rotor geometrical center is different from the rotational center. The rotational center is identical to the stator geometrical center. The point of minimal gap is moving with respect to the stator. This type can also be caused by manufacturing tolerances, bearing wear and bent shaft or flexible shaft. Mixed eccentricity is a combination of both static and dynamic eccentricity. The rotor geometrical center, rotor rotational center and stator geometrical center are different. The point of minimal gap is also moving with respect to the stator. The major risk of these eccentricities is mechanical contact of the rotor and stator, which would result in damage for the machine [8], [15].

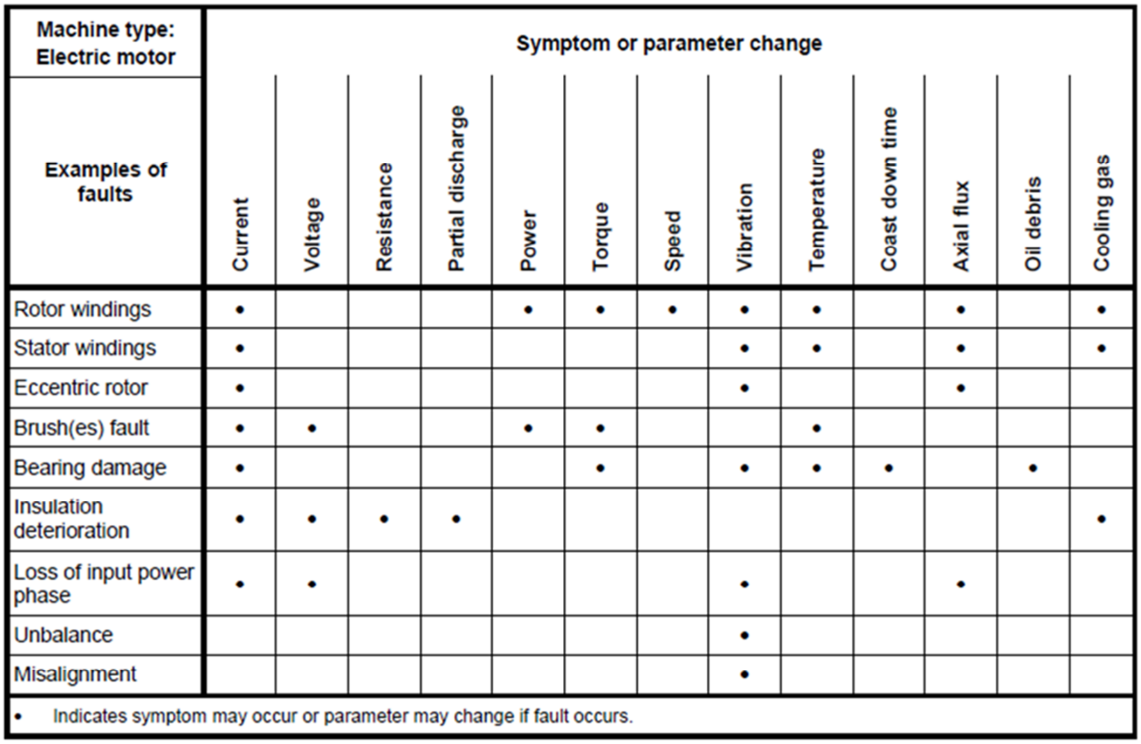
2.3 Condition Monitoring

Condition monitoring is a continuous evaluation process of the health of the facilities equipment through its service life. It is essential to detect faults while they are still developing; this is referred to as incipient fault detection. The incipient fault detection of motor failure provides a safe operating environment. Using a comprehensive condition monitoring scheme for continuous assessment of the induction has become increasingly important. By condition monitoring induction motors, adequate warning to imminent failure can be provided to the operator, which makes it possible to schedule future maintenance and repair work. This can reduce down time and optimize the maintenance schedule.

The condition parameters or symptoms for determining various faults in induction machines recognized by International Standards Organization are displayed in Table

2.3.1 [16]. The most prominent parameters used include current, vibration, temperature and axial flux [16]. These prominent parameters can be categorized as: invasively (Axial flux) and non-invasively (noise, vibration and current) monitored parameters.

Table 2.3.1: Parameters Used to Detect Faults in Induction Motors



Machine type: Electric motor	**Symptom or parameter change**												
Examples of faults	**Current**	**Voltage**	**Resistance**	**Partial discharge**	**Power**	**Torque**	**Speed**	**Vibration**	**Temperature**	**Coast down time**	**Axial flux**	**Oil debris**	**Cooling gas**
Rotor windings	•				•	•	•	•	•		•		•
Stator windings	•							•	•		•		•
Eccentric rotor	•							•			•		
Brush(es) fault	•	•			•	•			•				
Bearing damage	•					•		•	•	•		•	
Insulation deterioration	•	•	•	•									•
Loss of input power phase	•	•						•			•		
Unbalance								•					
Misalignment								•					
• Indicates symptom may occur or parameter may change if fault occurs.													

Axial flux leakage is a result of stator and rotor currents on the machine extremities; in the coil ends and the rotor end rings. Axial flux can always be detected even with symmetrical supply due to the effect of imperfections in the production process. Therefore, the axial flux can be a good indicator of an induction motor's health. However, the measurement is invasive since a search coil is wound around an end-turn in front of the machine. The coil is perpendicular to the machine and shaft. The search coil measures the sum of the flux from stator winding and flux from the rotor winding [17], [18].

Temperature or thermal measurement involves bulk temperature measurement or parameter estimation. Researchers have developed two thermal models to categorize electric machines: the finite element analysis model and the lumped parameter thermal model. The finite element analysis model handles heat conduction problems more accurately and is well suited for steady state or transient problems of large machines where asymmetries are a common feature but are computationally intensive [7], [19]. The lumped parameter thermal model is the most popular for temperature estimation. It is composed of different thermal resistors and capacitors [7]. In a turn-to-turn fault, the temperature rises in the region of the fault; however, this might be too slow to detect the incipient fault before it progresses to a more severe fault [20].

Vibration analysis is the most established technology and the most tangible. Almost all machines vibrate, and the connection between these vibrations and the machine condition is that both are easily measured and interpreted. The major benefit of vibration monitoring is that different mechanical processes within the machine (eg. imbalances, gear-mesh, bearing faults) all produce energies at different frequencies. Separating these different frequencies from one another through spectrum analysis enables a whole new level of detail to be seen [21]. The monitoring and frequency analysis will be discussed later.

The MCSA is an electric machinery technology developed by the Oak Ridge National Laboratory (ORNL). It exploits the intrinsic ability of the electric motors and generators to act as transducers by using non-intrusive current clamps. The motor current provides the means of detecting small variations in time dependent loads and speed variation generated anywhere within the electro mechanical system, which converts them into revealing signatures that can be used to determine equipment degradation and incipient failures [22], [23]. There are numerous applications of using MCSA on equipment health published among the nuclear-generation, industrial and defense industries. In motor applications, the stator current is monitored for detection of different faults in induction motors. The monitoring methods and fault frequency analysis will be discussed later.

The most commonly monitored parameters or signals in condition monitoring; vibration and current will be discussed further in Sections 2.4 and 2.5 respectively. These parameters indicate a fault condition by either an increase or decrease in the overall value or by some change in characteristic value. For such parameters, simple measurements of overall values may not be sufficient to reveal a fault condition. As a result, spectral and phase (in the case of vibration) measurements may be required to reveal changes attributed to faults. The accuracy required for condition monitoring parameters is not as absolute as the one required for performance measurement. The monitoring methods and frequency domain fault characteristics of these types of signals are discussed in the next two sections.

2.4 Methods for Monitoring Vibration

All machines under normal operation produce oscillation motion such as vibration. There are two types of vibration - benign and serious vibration. Benign vibrations are characteristics of regular operating conditions. Examples of benign vibrations are as follows: blade-passing frequency; 120Hz motor hum frequency; pure tone from motors driven by VFD; gear mesh frequencies; and broadband turbulence from pumps or fans. The amplitude will vary from machine to machine and is a measure of the quality of manufacturing and load condition. Increase in normal levels advocates a reason for investigation but not for alarm [24].

Serious vibration cause accelerated wear and premature failure. The serious vibrations can be categorized into two: forced and natural. The corrective actions of these two serious vibrations are very different. Forced vibrations are caused by mechanical (misalignment, unbalance, worn bearing, bent rotor etc.) and electrical (shorted turns, rotor cage faults, etc.) defects. These defects can be corrected by alignment, balancing or changing the faulty component, whereas natural vibrations are caused by a structural effect, where the structure behaves like a mechanical amplifier to sensitive frequencies. Symptoms of natural vibrations are: abnormally high amplitudes of vibrations; strongly directional vibrations; variation of vibration amplitude; and rumbles during coast up and coast down of motor. There are three main tests used to verify resonance: impact or hammer test on the major components to determine their natural frequencies, variable shaker, and coast-up and down test. Thus, corrective measures involve changing the speed, adding damping, changing the natural frequency of the responding part, dynamic absorber and reducing source input [24].

The commonly measured vibration signals are displacement, velocity and acceleration. Displacement measures the distance that the moving component has moved relative to a fixed reference and it indicates stress. It is a predominate factor at low speeds 1200rpm (below 20Hz). Velocity measures the instantaneous rate of change of the displacement and it indicates fatigue. The amplitude is constant throughout all frequencies, and as a result, it is preferred for condition monitoring where the amplitude has the same meaning at all frequencies, which simplifies the interpretation of the severity and establishing the alarm limits. Velocity is generally used between 10Hz (600rpm) and 1kHz (60,000rpm). Acceleration measures the instantaneous rate of change of velocity and it indicates force. It is a predominate factor at high speeds 60,000rpm (1kHz) [25].

In general, there are three categories of vibration sensors based on their electrical outputs: displacement sensors, velocity sensors and accelerometers. Each sensor type serves a specific purpose and is restricted by inherent constraints [25]. Displacement sensors, such as proximity switches and strain gauges are non-contact and measure shaft movement or displacement relative to the probes itself. Velocity sensors, such as tachogenerators, unlike the displacements sensors, have contact with the moving components to measure the motion. Accelerometer sensors are usually classified into two: Piezoelectric (PE) and Piezoresistive (PR) accelerometers.

The PE accelerometers are the most widely used for testing and measurement [26] compared to the PR accelerometers since they provide a wide frequency range and are available in wide range of sensitivity, size and weight. There are two subdivisions of the PE accelerometer: the basic charge mode piezoelectric and the voltage mode Internal Electronic Piezoelectric (IEPE) accelerometers. The basic charge mode PE requires use of special low noise cables (which tend to be expensive compared to the standard commercial coaxial cable and charge amplifiers), thus IEPE type is preferred. These vibration sensors need to be attached to the surface of the machines. In the field of vibration monitoring the PE accelerometer is the most commonly used, which permits the vibration signal to be converted to either a velocity or displacement signal.

2.5 Methods of Current Monitoring

In the stator current of every induction motor there are two types of frequency harmonics that are always present: the Winding Harmonics (WHs) and the Principal Slot Harmonics (PSHs). The important ones for a three-phase induction motor are the first order WH (also called the fundamental component), the fifth order (WH5), and seventh order (WH7). The frequency of the *n* order is WH (WH*n*) is proportional to the fundamental supply frequency [1].

It is possible to obtain information about the health and the integrity of an induction motor through the monitoring and analysis of the electrical signature, as mentioned previously. The instrumentation system usually consists of the following items according [1]: a Current Transformer (CT) to sense the signal; resistive shunt across the output of the CT; a MCSA instrument to produce the current spectrum or signature. The current measurements can be collected by measuring the secondary side of a CT or around the motor cable using a clip on current transformer. Usually only one current signal of one motor phase is analyzed for its current frequency content. The reason for this is that, the rotating flux waves produced by the different faults cut all three stator phase windings; thus, the corresponding currents are induced in each of the three phases. High frequency response range is required for the CTs to cover the frequency range of components that can be induced due to different faults mechanism and to cater for inverter-fed induction motor drives. The frequency domain based methods of the line current are generally known as MCSA [1].

2.6 Summary

In summary, the basic concepts of the induction motor have been presented. The common types of faults associated with induction motors were identified and categorized into electrical and mechanical faults. These common faults have also been described. Furthermore, the most prominent parameters used to detect these faults, namely motor current and vibration have been identified and discussed briefly. CTs are usually used to sense the current signal in current monitoring. Vibration monitoring is more mature and established technique for detecting faults. In vibration monitoring, the velocity is the preferred signal used to quantify the level of vibration since the amplitude is constant throughout all frequencies. The accelerometers are usually used to measure vibration. The acceleration can be integrated to obtain the velocity signal. The IEPE accelerometers are widely used to measure the accelerations.

The next chapter discusses the development of an algorithm to extract key features from the vibration and current signals for condition monitoring of induction machines.

3 KEY PERFORMANCE INDICATORS IN INDUCTION MACHINES

A method for the processing of raw sensor signals from induction machines into Key Performance Indicators (KPIs) for condition monitoring is presented in this chapter. This thesis proposes a condition monitoring approach that is intended to be more automated with less dependence on the standard rules based approach. The first step is to extract KPIs or features from various raw input signals. Specifically, the input signals are the line voltages, phase currents, motor speed, and vibration signals from accelerometers. Note that the majority of signals can be taken from existing distribution system equipment. As in the case of a power generation facility, the line voltages and phase currents can be taken directly from the CTs and Potential Transformers (PTs) that are installed in MCCs and Switchgear (SWGR) for relaying purposes. The data can then be collected and processed within the facilities existing Distributed Control System (DCS). The processing includes steps such as Root-Mean-Squared (RMS) calculations and FFTs. The processed data can then be input into an unsupervised fault detection algorithm that will alert operators of potential fault conditions. At that point KPIs can then be analyzed to determine the specific fault condition.

Section 3.1 will discuss the basic functions being processed for KPIs. The following Section 3.2 and 3.3 will discuss the calculations that are performed for specific mechanical and electrical features respectively.

3.1 KPI Algorithm Development

MATLAB software was used to develop a function *KPIpro2*. The *KPIpro2* function implements an algorithm for processing raw input signals, recorded from an operating motors, into KPIs. The function will process thirteen input signals in real time. However, for the purposes of development, data was recorded and all analysis performed off-line. The input signals include the three phase currents; the three line-to-line voltages; rotational speed; and the axial, horizontal and vertical and acceleration on two bearing blocks. A total of 186 KPIs are extracted from these raw signals, refer to Appendix C for a description of the KPI values. The processing includes steps such as RMS calculations and FFTs. Since these steps are repeated multiple times on similar signals, the KPI algorithm calls other functions to perform these steps. Thirteen additional functions are called on by *KPIpro2* to extract various KPIs.

The *KPIpro2* takes in the raw signals as an $m \times 13$ matrix, where m is the number of samples taken during testing. In addition, the sampling frequency is an input to the function. This was necessary early in development as different Data Acquisition (DAQ) systems were used that had dissimilar sampling rate capabilities. Note that raw signals provided to the feature-extraction algorithm must be sampled at a high rate. The discrete signal $x[n]$ is a sampled version of the continuous time signal $x[t]$ when acquired through a DAQ. The continuous time signal has to be sampled at a rate greater than twice the maximum frequency contained in the signal in order to avoid aliasing distortions, a factor called the Nyquist Rate [27], [28]. Higher frequency sampling yields a higher number of data points and higher signal resolution. However, the computational requirements to analyze this signal will increase with higher signal resolution. Thus, the sampling frequency selection should be based on the required resolution of the signal and the desired computational accuracy/performance. Therefore, a sampling frequency of 6000 samples per second is needed to avoid aliasing distortion for analyzing signal features up to 3000Hz. The processed features are output at a much lower rate, as multiple data samples are typically required for processing. In this implementation, the features are calculated at 5Hz.

The first steps of the algorithm involved the setup of the number of samples to process per output-sampling rate, vector for time indication and scaling the raw data. A vector *vectorref* is established for use throughout the KPI processing for the number of samples to process per 0.2 seconds. This vector will be used by each additional function called by *KPIpro2*. The $m \times 187$ matrix *KPI* is then pre-allocated for the m equal to the length of *vectorref* vector with all zeros. This step overwriting with zeros was necessary to ensure that data from previous analysis was erased from matrix *KPI* prior to new data being added. The time vector was added as *KPI* (m,187) for plotting KPIs vs time instead of leaving the plots as KPIs vs samples. The final step before calculating KPIs is scaling the raw signals. The voltage scaling is based on the voltage transducer used during testing; refer to Figure 4.1.7 for details of transducers circuit:

$$V_{LINE} = \frac{V_M}{R_M}\frac{1}{K}R_{LINE} \qquad (3.1)$$

Where; $V_M$is the raw signal; R_M is the measuring resistor (110Ω); R_{LINE} is the primary resistor (100kΩ); K is a multiplying factor of 2.5. The current scaling is based on the current transducer used during testing; refer to Figure 4.1.7 for details of transducers circuit:

$$I_{PHASE} = \frac{V_M}{R_M} K \tag{3.2}$$

Where; $V_M$is the raw signal; R_M is the measuring resistor (110Ω); K is a multiplying factor of 1000. The acceleration scaling is based on the sensitivity of the accelerometer used, 100mV per G:

$$Accel = V_M \frac{g}{0.1v} \frac{9.80665\, m/s^2}{g} \tag{3.3}$$

Where; $V_M$is the raw signal converted to meters per second squared. Once the raw signals are scaled and the KPI matrix is allocated, the processing for KPIs is started.

The RMS of the signals is one of the common features calculated for KPIs. The RMS is calculated in both the time domain and the frequency domain. The RMS of the phase currents and line voltages are calculated within the time domain using the following formula:

$$x_{RMS}(t) = \sqrt{\frac{1}{N}\sum_n x^2(t)} \text{ [27]} \tag{3.4}$$

The *KPIrms2* function was developed to perform the equation (3.4) on the voltage and current signals with the results assigned to KPIs (m, 1-6). It should also be noted that RMS could be calculated by:

$$x_{RMS}(t) = \sqrt{\tfrac{1}{T}\int_{t-T}^{t} x(t)^2 dt} \text{ [29], [30]} \tag{3.5}$$

Where T is the period of the input signal and x is the input signal. During development of *KPIpro2* both methods (Equations (3.4) & (3.5)) were used to calculate the RMS of known signals with the results compared. Comparison of known signals showed less than a 0.4% difference in the results of both calculations between each other and oscilloscope measurements. The deciding factor in method to proceed with was processing time. MATLAB function *profile on* was used to determine the processing time of each method. The method using equation (3.4) would take a third of the processing time as the method using equation (3.5). Five minutes of data took 15 seconds for RMS (3.4) while RMS (3.5) took 45 seconds. An additional note, if statements extend processing time; therefore, these statements were reduced or eliminated from functions as much as possible.

The RMS of the acceleration signals are calculated within the frequency domain. From Parseval's relation:

$$\sum_{n=0}^{N-1} \left| x[n] \right|^2 = \frac{1}{N} \sum_{f=0}^{N-1} \left| X[f] \right|^2 \quad [27] \tag{3.6}$$

The RMS in the frequency domain was derived to be:

$$\sqrt{\frac{1}{N} \sum_n x^2(t)} = \sqrt{\frac{1}{N^2} \sum_n \left| X(f) \right|^2} = \sqrt{\sum_n \left| \frac{X(f)}{N} \right|^2} \quad [27] \tag{3.7}$$

This derivation is used as part of the *bands* function to find the RMS of acceleration over bands of frequencies. Refer to Section 3.2 for further discussion of the bands function.

Additional fault features extracted are located within the vibration and current spectrum. As discussed in Chapter 2, an increase in magnitude of certain harmonic frequencies will indicate a mechanical and/or electrical fault. The Fourier Transform allows periodic and non-periodic continuous time signals to be described in terms of frequency content given by sinusoidal components [8]. The Fourier Transform has the discrete counterpart, the Discrete-Time Fourier Transform (DTFT). The DTFT is of a discrete time signal $x[n]$ is a function of a continuum of frequencies, and unlike the continuous case, the DTFT is always periodic with a period of 2π [28]. A transform of a discrete-time signal that is a function of a finite number of frequencies is called a Discrete Fourier Transform (DFT). The DFT can be viewed as “discretization in frequency” of the DTFT. The various harmonic frequencies were monitored in the vibration and current spectrum by performing discreet Fourier Transforms on the scaled signals [30]:

$$G_q = \sum_{k=0}^{N_0-1} g_k e^{-jq\Omega_0 k} \tag{3.8}$$

With the inverse formula:

$$g_k = \frac{1}{N_0} \sum_{q=0}^{N_0-1} G_q e^{jk\Omega_0 q} \tag{3.9}$$

Where:

$$\begin{aligned} g_k &= T_s g(kT_s) \qquad & G_{\ q} &= G(qf_0) \\ 2\pi f_0 &= \tfrac{2\pi}{T_0} \qquad & 2\pi f_s &= \tfrac{2\pi}{T_s} \\ N_0 &= \frac{T_0}{T_s} = \frac{f_s}{f_0} \qquad & \Omega_0 &= 2\pi f_0 T_s = \frac{2\pi}{N_0} \end{aligned} \tag{3.10}$$

Equation (3.8) was implemented using the FFT algorithm within MATLAB while Equation (3.9) was implemented using the Inverse Fast Fourier Transform (IFFT). The functions *harmagloc3* and *vibharmmech2* were developed to perform the FFT on the current and vibration signals respectively. After the signals are transformed to the frequency domain, a *find* function is used to call in the specific vibration frequencies associated with fault conditions. The *harmagloc3* and *vibharmmech2* functions will then return the max amplitude at ±1Hz of the specific frequency. The search is done around the specified frequency to reduce chance of error. During development phase of the codes it was observed that the peak may not occur exactly at the frequency but within ±0.5Hz. The specific frequencies looked at are discussed in Sections 3.2 and 3.3.

When performing the FFT, windowing was used on the signal to reduce spectral leakage. Spectral leakage is an undesired phenomenon when the power of the original signal data leaks out over the entire frequency spectrum instead of being concentrated at the points of interest. Spectral leakage produces amplitude bias (the difference between the true value and the expected value) and may conceal the presence of weak signals and prevent their detection. The quality of a signal can be improved by multiplying the signal data with a suitable window function. The simplest window is called the rectangular window; however, other window functions have been developed to achieve better sidelobe levels [30]. Decreasing the sidelobe levels reduces the amplitude bias, although this can be achieved only by broadening the window's mainlobe frequency response thus leading to a reduction in spectral resolution. A Chebyshev window is suitable for close and distant interferences. A rapidly decaying sidelobe window such as the Hann is suitable for distant strong interference as show. A window with low adjacent sidelobes such as the Hamming is suitable for very close interference and increasing sidelobes. In the field of fault detection of induction motors via motor current and vibration signals, the most popular is the Hann also known as the Hanning window [31], [32].

3.2 Mechanical Feature Calculations

The mechanical features that are used for KPIs are extracted from the vibration signals. The features include the velocity RMS, acceleration RMS over bands of frequencies, and the magnitude of specific harmonic frequencies.

The KPI for velocity was calculated by taking integrating the acceleration. A direct integration of the scaled integration was not possible. The low frequency or DC content of the signal causes an effect, which throws off the conversion process. The integration cannot account for this DC content [33]. This error builds during the integration giving a growing or decreasing error effect as shown by Figure 3.2.1. Therefore, the DC portion, from the signal conditioner, needs to be removed before integration. The function *velrms* initial steps are to perform a FFT, as described in Section 3.1, on the acceleration signals. The DC value is then set to zero to remove the DC Component. An IFFT is performed to convert the acceleration back to the time domain. The *velrms* then computes the velocity, v, by doing the cumulative trapezoidal numerical integration of the acceleration, a, and is defined as:

$$v = \frac{1}{F_s}\left(\sum_{i=1}^{N} a\,(i) - \frac{a(1) + a(N)}{2}\right) \tag{3.11}$$

Where F_s is the sampling frequency. Figure 3.2.2 shows the velocity (mm/s) verses time (seconds) after the integration. The velocity waveform observed is as expected based on research [21], [34], [35]. The magnitude of the velocity, 2mm/s, is less than 80% of the NEMA standard vibration limits of 3.8mm/s [35].

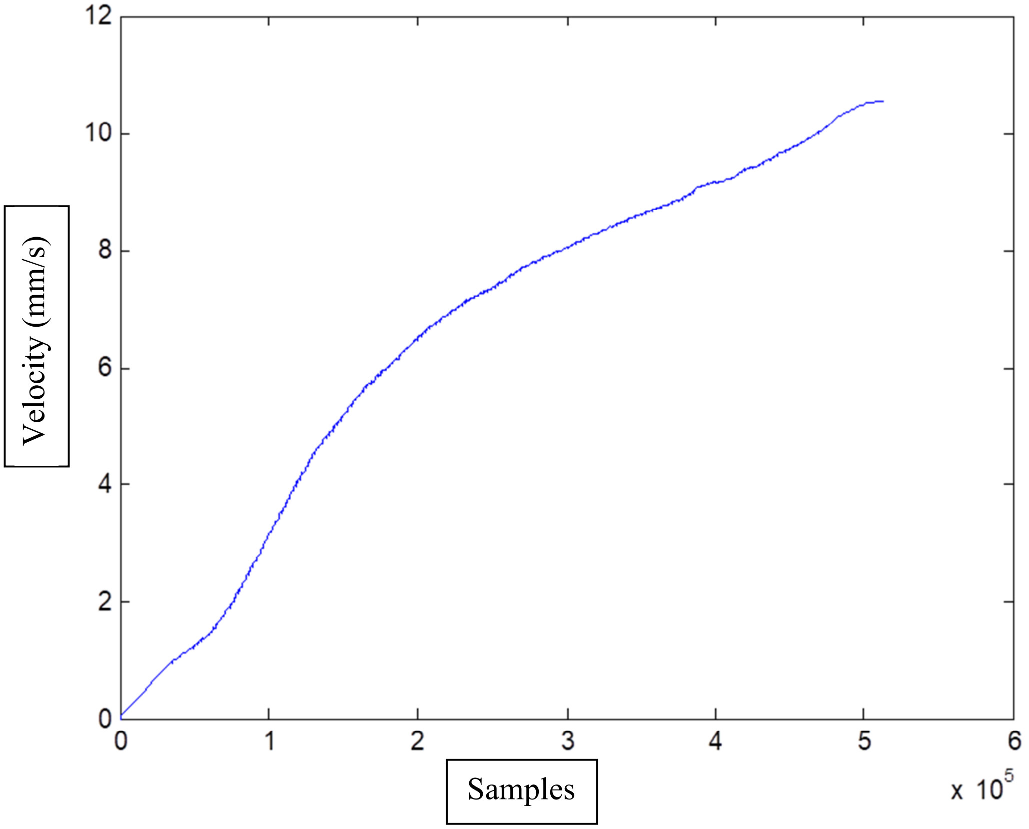


Figure 3.2.1: Integration of Acceleration with DC Component

The final step is to find the RMS of the velocity. The *velrms* function calls the *KPIrms2* function to calculate the RMS of the velocity as described in Section 3.1. The resultant velocity RMS is output to the KPI matrix. The complete function was validated by measuring known signals with the setup shown in Figure 3.2.3. A function generator was used to input various signals to a shaker table. The accelerometers, refer to Section 4.1, were connected to an oscilloscope to observe the recorded vibration. The signal conditioner gain was adjusted for each accelerometer to within a 1% error tolerance with the known input and measurements with handheld vibration meter (Extech 407860) [36] for calibration of the accelerometers and signal conditioner. The vibration signals were then input into the data acquisition system for processing. Comparison of the calculated velocity RMS to the handheld meter showed results with less than 1% error when the vibration peak to peak input was greater than approximately 25mV. When signal was less than 25mV, the calculated and measured velocities were within 5%.

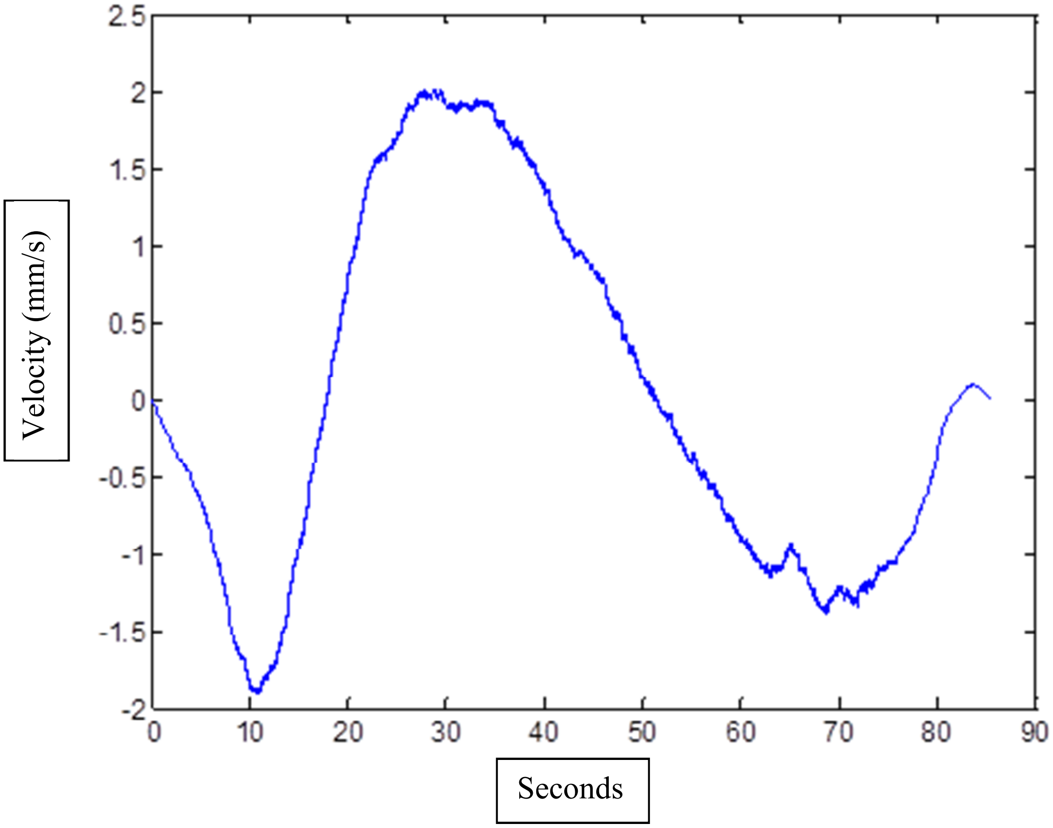


Figure 3.2.2: Integration of Acceleration Without DC Component

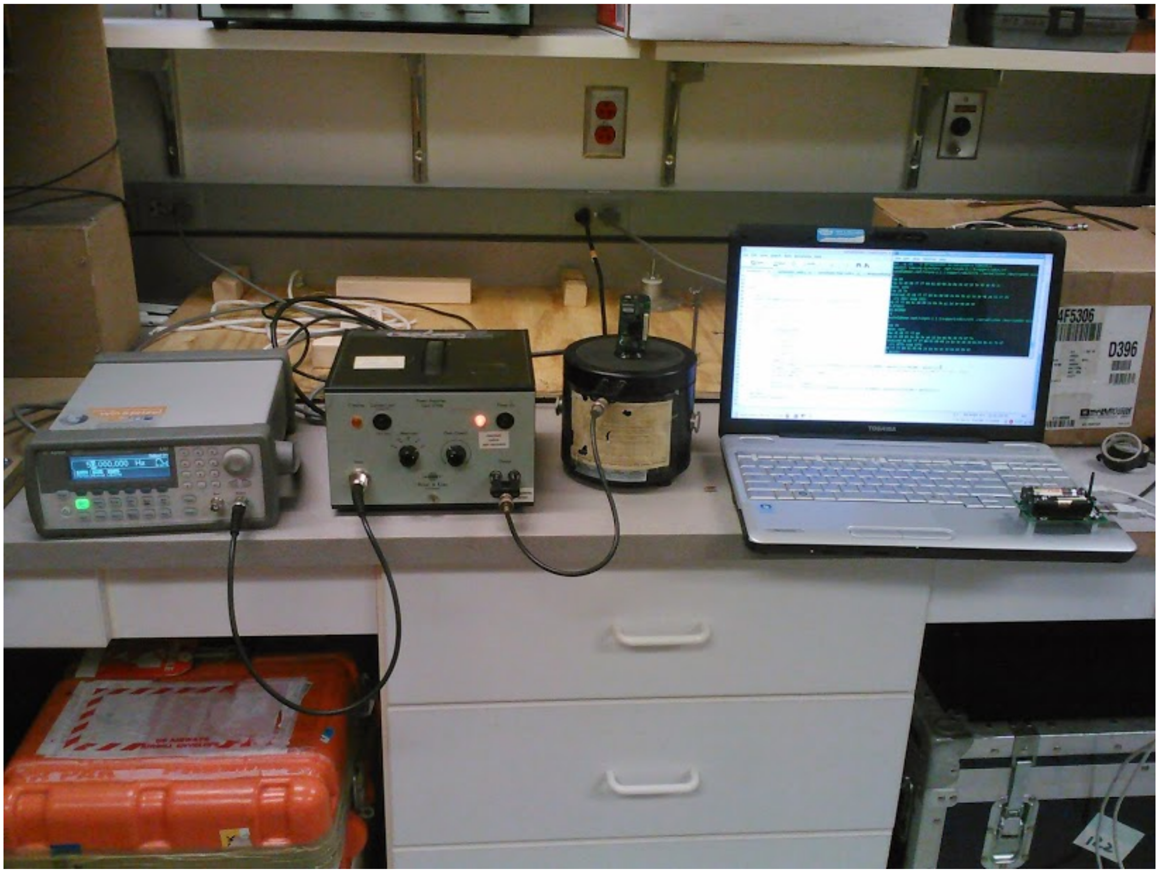


Figure 3.2.3:Shaker and Accelerometer Calibration Setup

KPIs were also calculated for the vibration RMS over bands of frequencies by the *bands* function. Banding is a method of dividing the frequency range into frequency bands and applying a vibration limit to each band. Banding recognizes that the vibration level at various frequencies is a function of the source of excitation (bearings, for example) [21], [35], [37]. Changes in the bands RMS will indicate a general fault condition. Since the KPI data is output at 5 Hz, the FFT was performed over a 0.2-second window of the accelerometer data. Three bands are then pulled out and assigned as variables. The frequency bands are defined as:

- Low frequency band from 10 to 200 Hz

- Middle frequency band from 200 to 2000 Hz
- High frequency band greater than 2000 Hz

The bands were selected based on preliminary work that was done with motor fault detection with Converteam Naval Systems [2]. With the three bands extracted from the FFT data, a RMS calculation is performed in the frequency domain to determine each of the three bands. Each vibration signal is processed through the algorithm as some fault vibration signatures can be more prevalent either axial or radial.

In addition to the vibration RMS, the magnitudes of specific vibration frequencies associated with fault conditions were located by spectral analyses. As discussed in Section 3.1, the *vibharmmech2* functions will output the magnitude of the frequencies within the accelerometer signals. Vibration frequencies at one, two, and six times the electrical frequencies were monitored. Twice the electrical frequency (Torque Pulse Frequency) vibration will increase when the air gap is not symmetrical between the stator and rotor. This condition results an unbalanced magnetic pull being greater in the direction of the smaller air gap. The stator is pulled in one direction, while the rotor gets pulled in the opposite direction to the side with the minimum air gap causing higher shaft vibration. [38]. In addition, problems in a motor such as a rub, loose parts, a bent shaft extension or elliptical bearing journals can cause vibration at 2 times rotational frequency [20]. One times the electrical frequency vibration can exist, although it is not nearly as prevalent as twice the electrical frequency vibration. This line frequency vibration is normally very small or non-existent, but if the stator or rotor system has a resonance at, or near, line frequency, the vibration may be large. The sixth harmonic is an indication that there is looseness in machine parts and can also be generated by a loose rotor.

Vibration frequencies at half, one, two, three and four times the rotational frequencies were monitored. An eccentric rotor, which means the rotor core outer diameter is not concentric with the bearing journals, creates a point of minimum air gap which rotates with the rotor at one times rotational frequency. A bent shaft typically produces spectra that have misalignment type characteristics with a higher than normal one and two times the rotational frequency. Rotating frequency harmonics or half, two, three and four times the rotating frequency harmonics at abnormally high amplitudes generally characterizes mechanical looseness, or an improper fit between components [35], [37], [38].

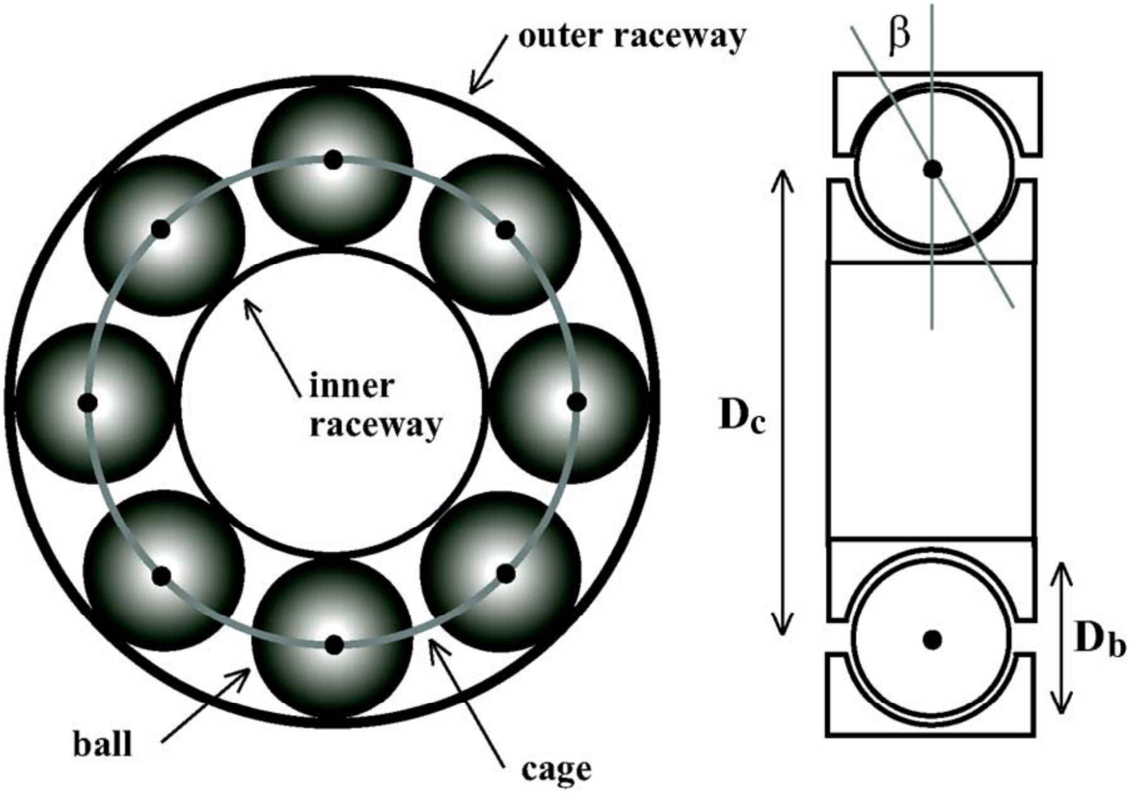


Figure 3.2.4: Geometry of a Rolling-Element Bearing [39]

There are four categories of ball bearing defects: outer raceway defect, inner raceway defect, ball defect, train defect. The faults can be detected at vibration frequencies specified by [39], [40]:

$$\text{Outer Raceway: } f_V = \left(\frac{N}{2}\right) f_r \left(1 - \frac{D_b}{D_C} \cos\beta\right) \tag{3.12}$$

$$\text{Inner Raceway: } f_v = \left(\frac{N}{2}\right) f_r \left(1 + \frac{D_b}{D_c} \cos\beta\right) \tag{3.13}$$

$$\text{Ball Defect: } f_v = \frac{D_c}{D_b} f_r \left(1 - \frac{D_b^2}{D_c^2} \cos^2 \beta\right) \tag{3.14}$$

$$\text{Train Defect: } f_v = \left(\frac{f_r}{2}\right) \left[1 - \frac{D_b}{D_c} \cos\beta\right] \tag{3.15}$$

Where:

- f_v is the vibration frequency in Hz
- N is the number of balls
- D_b is the ball diameter
- D_c is the pitch or cage diameter
- β is the contact angle of the ball with the races

Due to the harmonic frequencies being based on the mechanical speed, the speed is also calculated and output as a KPI. The mechanical speed is a voltage signal input to the *KPIpro2* function as channel 7 as shown by Appendix B. The signal input is taken from a frequency to voltage converter as discussed in Section 4.1. Based on the LM2907 converter (Figure 4.1.11), the *run_speed* function calculates the mechanical speed based on the following formula:

$$f = \frac{V_{in}}{V_{CC} \times C_1 \times R_1 \times k} \tag{3.16}$$

Where:

- f is the mechanical frequency
- V_{in} is the input voltage signal
- V_{CC} is the control voltage (15V)
- C_1 is the capacitor on Pin 2 (0.01µF)
- R_1 is resistor on Pin 3 (100kΩ)
- k is the chip gain (1 typically)

The mechanical speed is output in hertz for use in additional calculations. The code was validated by comparing with the digital display on the fault simulator, handheld meter and oscilloscope measurements of the LM2907 output voltage. When the typical values where used for capacitor, resistor, and gain, the speed was out of calibration by approximately 3Hz. It was determined that the actual values should be used instead of the typical values for capacitor and resistor. Additionally the gain would vary due to a thermal factor. When initially energized the gain would be greater than one, approximately 1.3. After approximately 10mins, the LM2907 reaches a steady state and the gain leveled at 0.9942. These values where used for the variables.

3.3 Electrical Feature Calculations

The electrical features that are used for KPIs are extracted from the voltage and current signals. The features include the electrical frequency, total harmonic distortion, and the magnitude of specific harmonic frequencies in addition to the voltage RMS and current RMS previously discussed in Section 3.1.

The electrical frequency was determined by analyzing one of the phase currents. The *KPIpro2* function locates the zero crossings of the current waveform. The distances between crossings are then used to determine the electrical frequency. During code development, it was observed that zero crossings would occur in between samples. The code would return the nearest sample resulting in errors in frequency. To correct this issue the function *Crossing* was used to locate the index of the zero crossings with interpolation when the zero crossing occurred between samples. This corrected the error of spikes below 60Hz during steady state; however, spikes still occurred reaching to 4000Hz. This occurred significantly during startup. This was determined to be caused by noise that caused the current signal to spike and cross zero. This was corrected by using the MATLAB SP tool to develop an Equiripple Lowpass filter for the current signal. Using the *filtfilt* command applied the filter and corrected the 180° phase shift that occurred when using the *filt* command. The filter was effective in removing the noise spikes from the signal without attenuation that would influence the frequency calculation.

In addition to the RMS of electrical signals, the magnitudes of specific vibration frequencies associated with fault conditions were located within the current spectrum. As discussed in Section 3.1, the function *harmagloc3* will output the magnitude of the frequencies within the current spectrum. The magnitudes of current at 1st, 3rd, 5th, 7th, 9th, 11th, 13th harmonics were monitored by *KPIpro2* due to the increase in variable frequency drives (VFDs). With an increase in VFDs, the use of that power electric motors, the voltages and currents originating from a VFD are strong with harmonic frequency components. Voltage supplied to a motor sets up magnetic fields in the core, which create iron losses in the magnetic frame of the motor. Therefore, higher frequency voltage components produce additional losses in the core of AC motors, which in turn, increase the operating temperature of the core and the windings surrounding in the core damaging the motor [41]. Additionally there is the phenomenon of torsional oscillation of the motor shaft due to harmonics. When a motor is supplied non-sinusoidal voltages and currents, the air gap magnetic fields and the rotor currents contain harmonic frequency components. The interaction between the positive and negative sequence magnetic fields and currents produces torsional oscillations of the motor shaft. These oscillations result in shaft vibrations. If the frequency of oscillations coincides with the natural mechanical frequency of the shaft, the vibrations are amplified and severe damage to the motor shaft may occur. It is important that for large VFD motor installations, harmonic analyses be performed to determine the levels of harmonic distortions and assess their impact on the motor [41].

Based on the impact of the current harmonics on the induction motor, KPIs were also developed for the monitoring of Total Harmonic Distortion (THD). The THD was calculated using both the IEEE and IEC method. Where the IEEE method uses the formula [41]:

$$THD_{IEEE} = \frac{\sqrt{I_2^2 + I_3^2 + I_4^2 + I_5^2 \ldots}}{I_1} \tag{3.17}$$

For coding purposes, this formula was simplified to:

$$THD_{IEEE} = \frac{\sqrt{I_T^2 - I_1^2}}{I_1} \tag{3.18}$$

Where

- I_T is the total measured current
- $I_n$is the magnitude of the current at *n*th harmonic

The IEC method uses the formula [42]:

$$THD_{IEC} = \frac{\sqrt{I_2^2 + I_3^2 + I_4^2 + I_5^2 \ldots}}{\sqrt{I_T^2}} \tag{3.19}$$

Simplified for coding as

$$THD_{IEC} = \frac{\sqrt{I_T^2 - I_1^2}}{\sqrt{I_T^2}} \tag{3.20}$$

Broken rotor bars can be detected by performing spectrum analysis of the motor current. As discussed in Chapter 2, the harmonic frequencies relating to broken rotor bars can be predicted using the formula [43], [44]:

$$f_{br} = \left(1 \pm 2ks\right)f \tag{3.21}$$

Where:

f_{br} = the harmonic frequencies of the broken rotor bars

k = harmonic index (1, 2, 3…)

s = the slip ratio

f = the fundamental frequency of the electrical current

The function *rotorbarsidebands* was developed to calculate the coefficients to be multiplied with the fundamental electrical frequency. The calculated *slip* and *vectorref* are inputs to the algorithm. The algorithm determines the coefficients using only the harmonic indexes of one, two and three as prior research [44] has shown that the magnitudes of the lower and upper sidebands above the third index were minimal. The output is an m × 6 matrix (*rotorbar*) that is an input to the *harmagloc3* function for locating the harmonic frequencies of the broken rotor bars.

Rotating eccentricities can occur due to bearing faults, which can lead to variations in the machine inductances. This produces frequencies in the stator current that can be used to detect bearing faults [39], [40]. The frequencies can be calculated using:

$$f_{bng} = \left| f_1 \pm k f_v \right| \tag{3.22}$$

Where

f_{bng} Vibration frequencies reflected in the current spectrum

f_1 the electrical supply frequency

k = 1, 2, 3…

f_v is the vibration frequency in Hz as discussed in Section

The vibration frequencies are those discussed under Section 3.2. As shown by research in [39] and [45], the bearing faults produce harmonic frequencies in both the current and vibration spectrums.

3.4 Condition Monitoring With KPIs

The greatest difficulty associated with condition monitoring of induction motors is that an operator must possess some degree of expertise in order to distinguish the normal operation from a potential failure. This is because monitored spectral components can be produced from any number of sources, including those related to normal operation. Combine this with the extensive research available makes it difficult to determine a simple and complete set of rules for fault detection. When using hard-and-fast rules, for instance, the developer is likely to omit certain situations out of ignorance or fail to implement rules that deal with the dynamic nature of certain conditions. Without knowing all possible fault conditions and symptoms, the rules based approach to condition monitoring is difficult.

An unsupervised approach has been developed to detect fault conditions, without a set of rules, that learns KPIs that are specific to the system and monitors for changes that may indicate a potential fault condition [2]. The fault detection algorithm is provided the KPIs extracted from collected data as discussed in Sections 3.1-3.3. The relevant information is extracted from the KPI vectors and then the encoded results are compared to a database of healthy features encoded in a similar manner. Any differences indicate that the motor may be faulted. This approach is patterned after the facial recognition scheme presented in [46].

The unsupervised fault detection algorithm is provided with a feature vector of KPIs. As in many pattern-recognition problems, there is a large number of feature vectors of KPIs (186 proposed by this thesis). From the rules-based approach, each feature would be considered for how useful it would be for the detection of one or more faults. This is not necessarily the case as features are not independent. Therefore, classification accuracy does not improve as one includes more features. As discussed in [2], there is actually a diminishing point of returns that stems from the fact that as more features are included a situation develops that can lead to errors. An approach is needed combines features in such a way that it reduces the size of the data set, and reveals trends that best separate various data sets.

The heart of the fault detection algorithm proposed in [2] is the principal component analysis (PCA) block. This algorithm provides information about meaningful trends within the monitored data. The PCA projects high-dimensional data onto a lower dimensional space, thereby performing a transformation that best represents the information in the overall data set. The PCA is essentially providing information about the relevant trends without having to develop a set of rules. The basic approach of the principal-component-based algorithm is to compare each measurement to an expected vector. This expected vector is computed by projecting onto a vector space created using "healthy" features. These healthy values are learned during a training phase in which the motor is assumed to be fault-free. The algorithm computes a health indicator which represents the error between the measured features and the expected value for a healthy motor. If the error is small, the system is operating under normal conditions or is "healthy". If the error grows, a problem may be developing.

3.5 Summary

In summary, the concept of the KPI extraction algorithm for induction motors has been presented. The common types of mechanical and electrical faults associated with induction motors were identified along with the features to detect these faults. The formulas for RMS and FFT are presented as used for KPI development. Additionally the harmonic frequencies within the current and vibration spectrum associated with the common faults are identified. The final output of the KPI extraction algorithm is 187-vector matrix that can be used for condition monitoring.

Additionally the unsupervised approach has been developed [2] to detect fault conditions without a set of rules is briefly discussed. KPI features from a training phase, in which the motor is assumed fault-free, are compared to KPI features that need to be analyzed. The algorithm computes a health indicator which represents the error between the measured features and the expected value for a healthy motor.

The next chapter discusses the experimental setup and results.

4 EXPERIMENTAL METHODOLOGY

During the normal operation of an induction machine, a variety of faults can occur which can lead to a catastrophic failures if undetected. The need to be able to plan equipment maintenance ahead of time, limit the extent of equipment damage and minimize the downtime for repairs has motivated the development of condition monitoring methods to detect machine faults at an earlier stage. The detection of common faults of an induction motor with help of signal processing techniques is the focus of this research. This chapter describes the development of an experimental setup, which allows for the introduction of various faults, individually or jointly, in a controlled environment. The setup includes instrumentation and data acquisition hardware to perform accurate repeated online monitoring of the induction machine. In addition, the testing of the proposed methodology to process the signals into KPIs and train and monitor health indicators for fault identification and diagnosis are discussed.

4.1 Overview of System

The experimental setup centered on a fault simulator designed by SpectraQuest as shown by the block diagram in Figure 4.1.1. Sensors for current, voltage, speed, and vibration measurements were added and wired to a data acquisition system for online monitoring. The collected data was saved as data arrays for processing in MATLAB into KPIs for condition monitoring and fault diagnosis analysis.

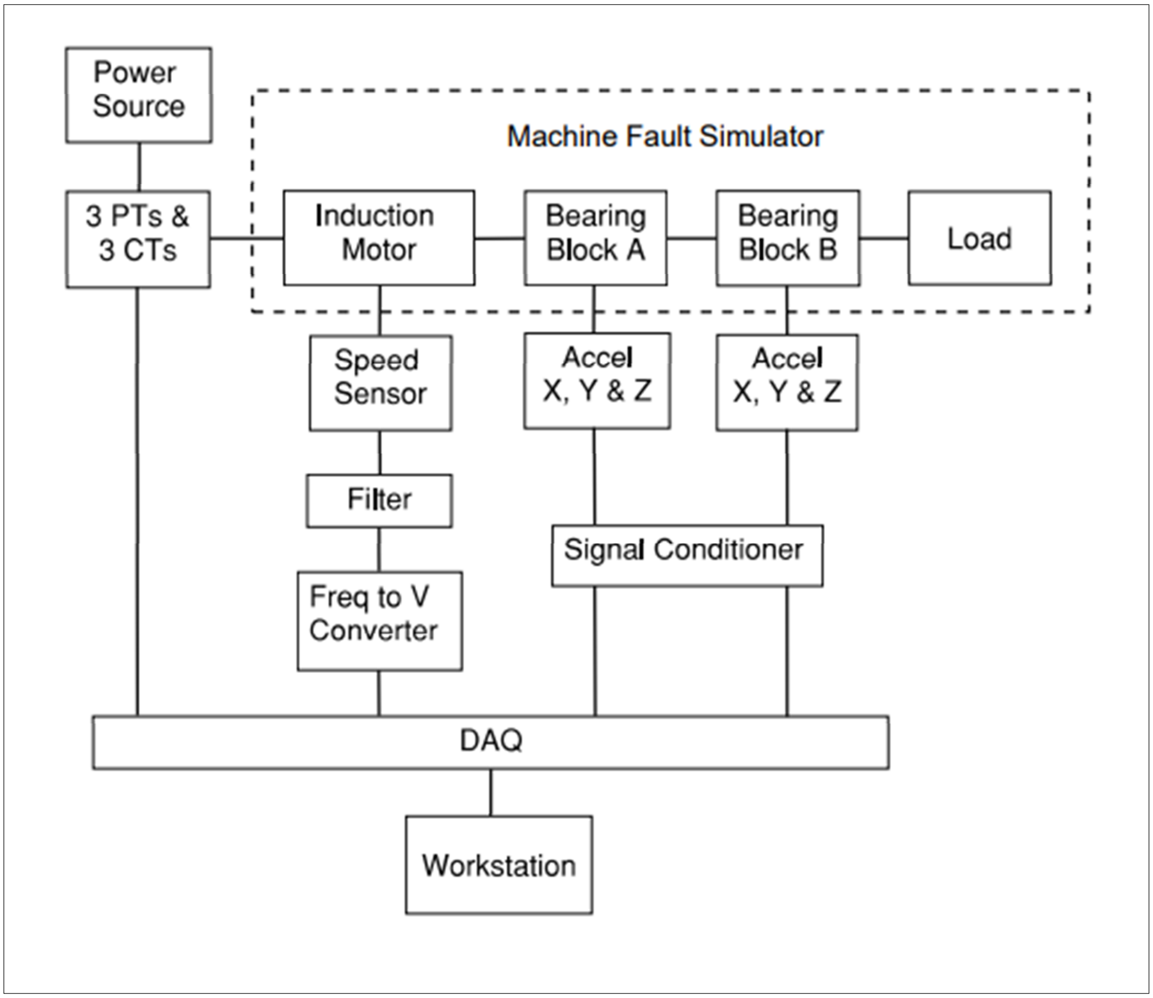


Figure 4.1.1: System Block Diagram

The machine fault simulator was developed for providing a basic setup for performing experiments and learning vibration signatures of different machine faults. It is assembled with a motor, split bracket bearing housing, bearings, shaft, rotor disks with split collar ends, couplings, pulleys, a multiple belt tensioning and a load of either a gearbox or centrifugal pump [47], refer to Figure 4.1.2. The fault simulator includes a machined mounting pad and alignment system with calibrated reference dials and jack bolts that facilitates the quick exchange of parts and aligning the equipment. This enables the introduction of controlled faults while limiting conflicting vibrations from misalignments or equipment vibration due to unreliable mounting.

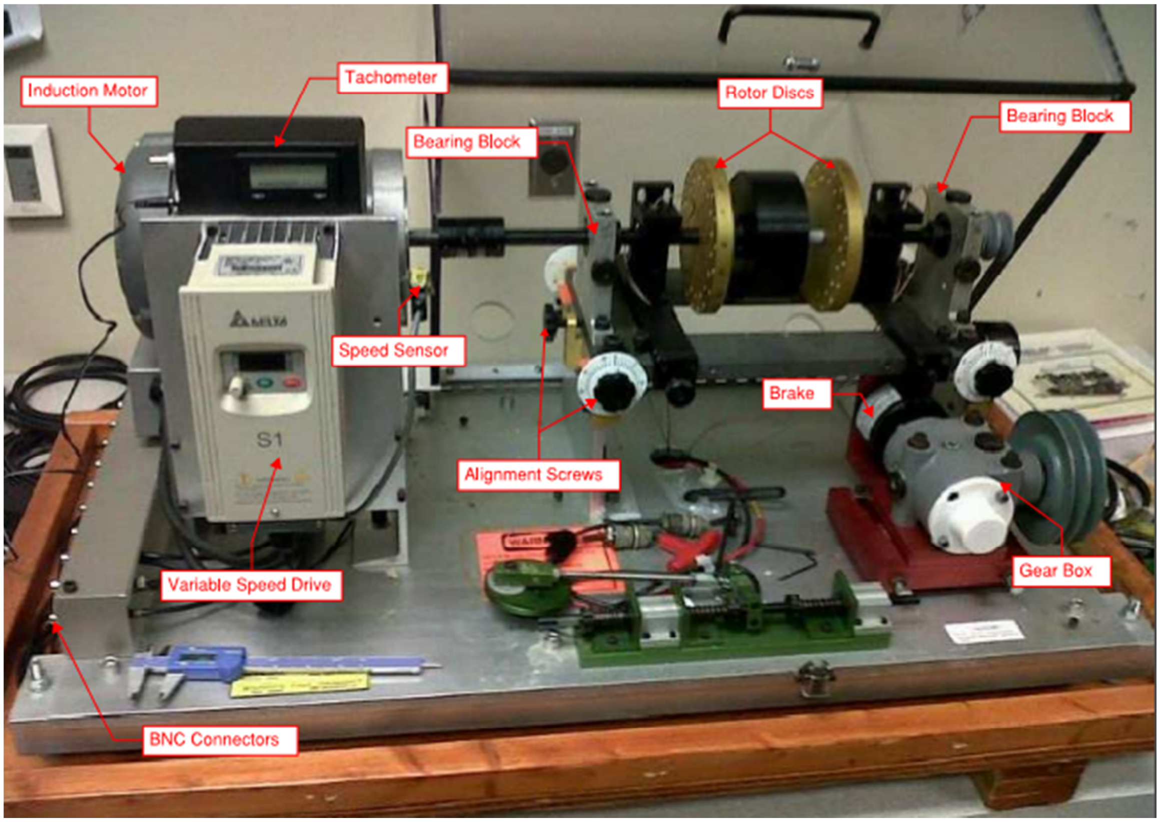


Figure 4.1.2: Machine Fault Simulator

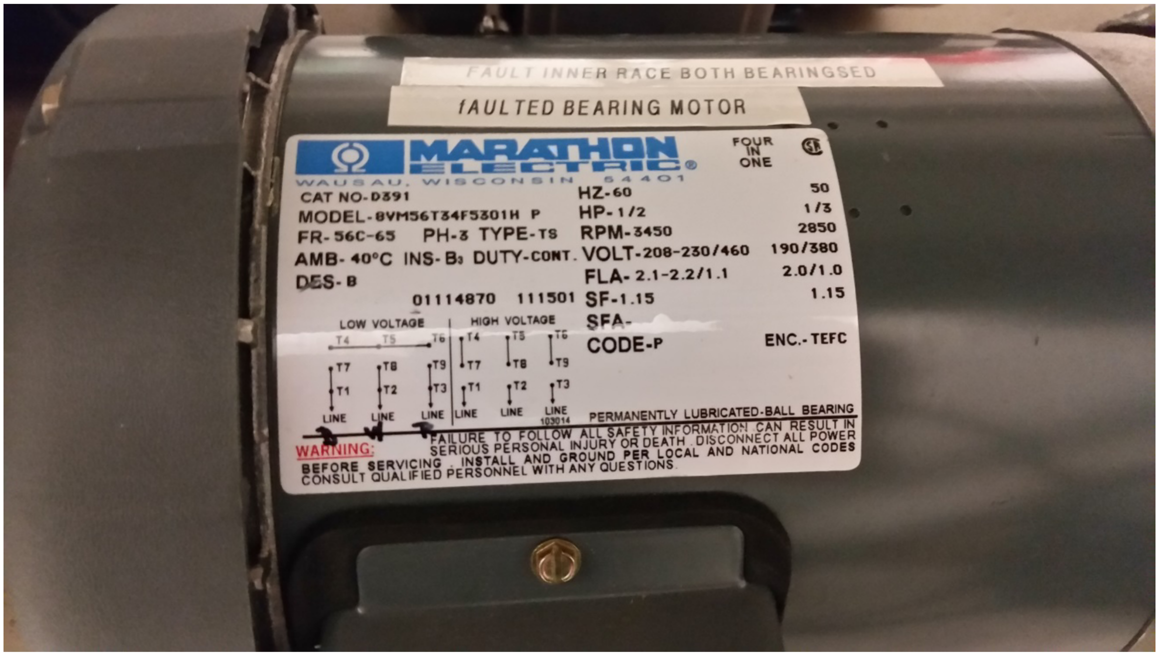


Figure 4.1.3: Induction Motor Nameplate

The experiments performed on the fault simulator involved a three-phase, half horsepower (HP) induction motor; refer to nameplate displayed in Figure 4.1.3. Six motors with identical specifications, as listed in Table 4.1.1, were used in testing. One motor was healthy while the other five each had a built-in fault. The built-in faults included:

- A motor with an inner race defect in both bearings
- A motor with ten percent of rotor bars broken
- A motor with an unbalanced rotor
- A motor with bowed armature shaft, 0.0085 thousandths TIR
- A motor with an adjustable air gap with misalignment

The normal configuration of the fault simulator would have the motor powered by the 1 HP variable frequency AC drive. However, connections were made to allow the motor to be plugged in and powered off the wall, simulating a common industrial setup.

Table 4.1.1: Typical Motor Performance Data [48]

Parameter	Value
Output Hp	0.5 Hp
Full Load Efficiency	66%
Full Load Power Factor	69.7%
KVA Code	P
Output kW	0.37 kW
Full Load Amps	2-2.1A
No Load Amps	1.4A
Locked Rotor Amps	16A
Full Load Torque	0.76 lb-ft
Poles	2
Drive End Bearing	Ball (6203)
Rotor WK^2	1.5 lb-ft^2

Two different load types were used to load the induction motor: a centrifugal pump and a gearbox. The pump, as shown in Figure 4.1.4, was connected to a 35-gallon tank containing water. Valves on the input and output of the tank were used for throttling during experiments. The pump could be exchanged with another worn centrifugal pump that had a built in fault of simulated cavitation damage to the head and impeller. The gearbox was loaded by a magnetic clutch (brake) mounted at the rear of the gearbox. The Precision Tork magnetic clutch, MC4 series, could be adjusted to provide 0.5 to 10 lb.-in of torque. These load types were used to simulate the common loads perceived by induction motors.

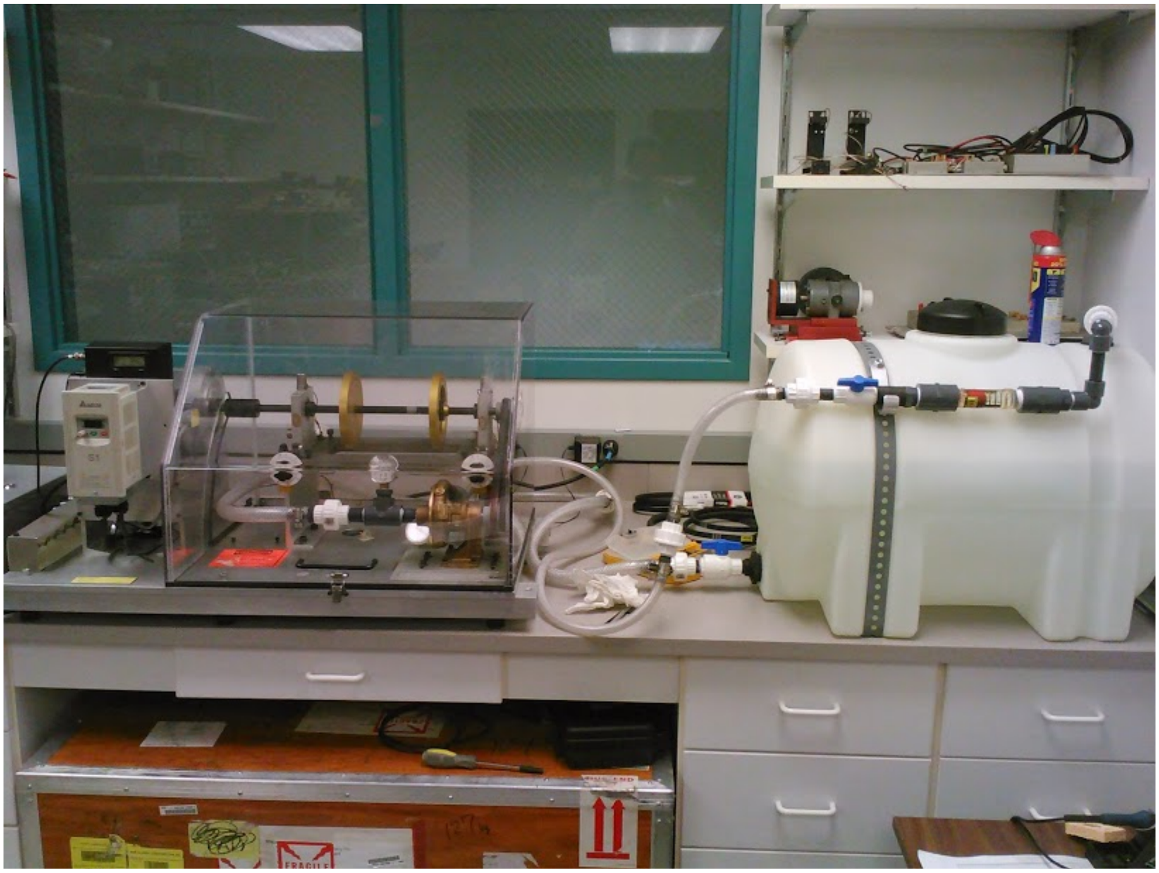


Figure 4.1.4: Fault Simulator with Pump Load

The drive train system comprised of a three-quarters shaft, bearings, sheaves and v-belts to the loads, refer to Figure 4.1.1. The shaft is a TGP straight steel shaft and can be used to simulate a fault by replacing with a shaft that is centrally bent. Two split-bracket bearing housings with bearings are used to support the shaft. The bearings are rolling element ball bearings with eight balls of 0.3125-inch diameter and a 1.3175-inch ball pitch in the bearing [49]. The bearings can be replaced with bearings with built-in faults: ball spin fault; inner ring defect; outer ring defect; combination defect. For additional faults on the drive train, two balanced rotor disks can be placed on the shaft as shown in Figure 4.1.2 . These rotors have two rows of tapped holes that can be used to create an imbalance by placing weight on the rotor at different locations, demonstrated by Figure 4.1.5. The rotors can also be replaced with faulted rotors; eccentric rotor, cocked rotor.

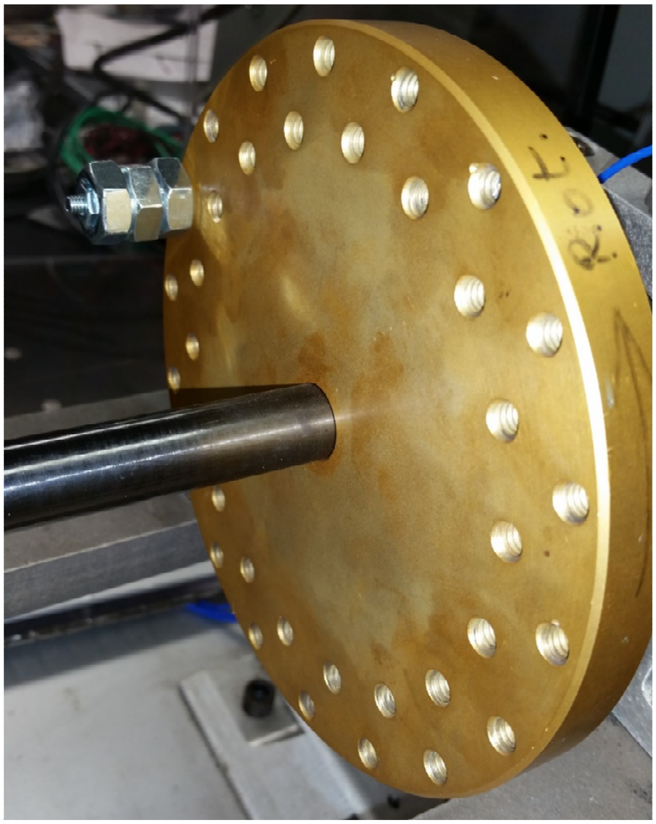


Figure 4.1.5: Rotor Disk with Imbalance

The measuring instrumentation along with signal conditioning hardware, data acquisition hardware and software required for collecting the relevant signals; current, voltage, vibration, and speed, was designed and added to the fault simulator. The fault simulator includes a speed sensor with digital display. However, the setup does not allow for current measurement required for motor current signature analysis, as discussed in the previous chapter. To understand the motor current signatures attributed to different faults, a method of collecting the current and voltage seen at the induction motor is needed.

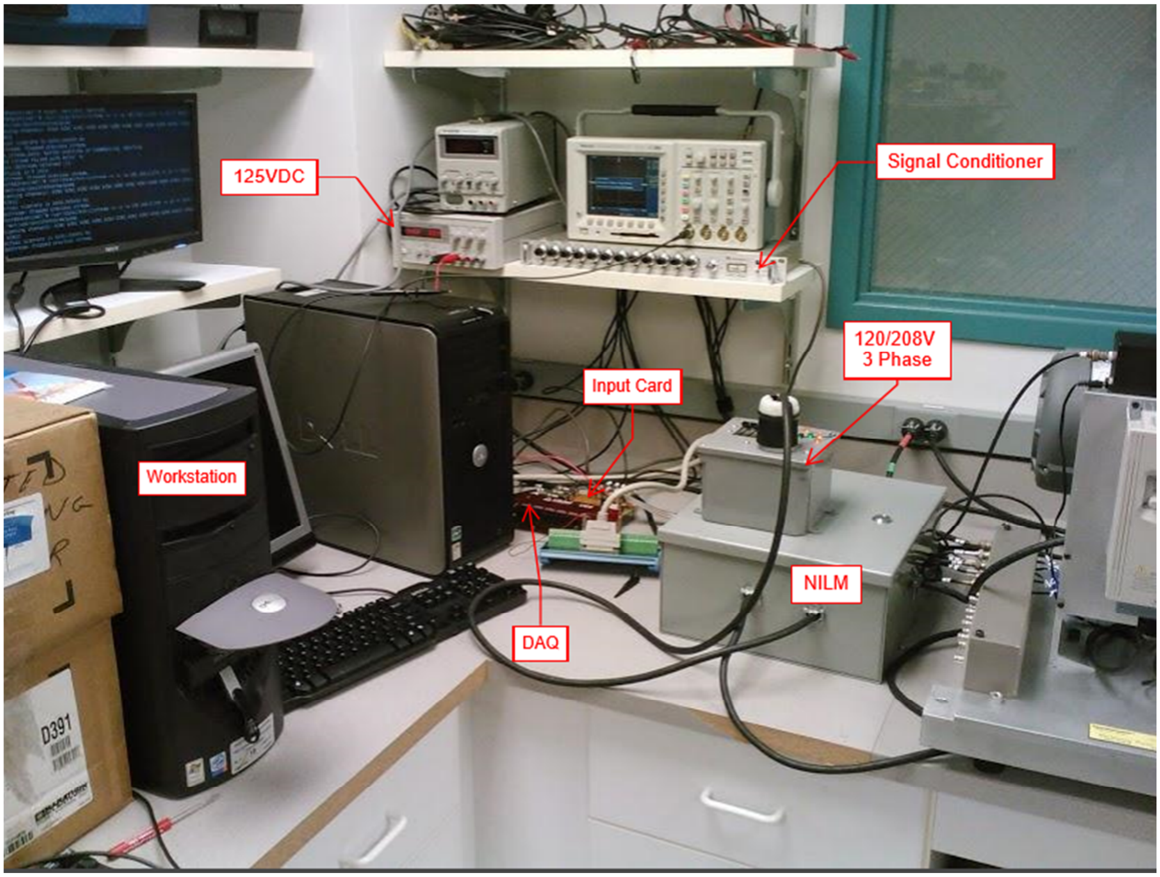


Figure 4.1.6: Data Collection Setup

The current and voltage of the motor are measured using a circuit board design used in other experiments for Non-Intrusive Load Monitoring (NILM). LEM modules were used for the measurements, a LA 55-P current transducer is used for current measurement and the LV 20-P voltage transducer is used for voltage, refer to Figure 4.1.8 for NILM enclosure internals. Isolation from the power circuit is provided by using these types of transducers. The current (LA 55-P) and voltage (LV 20-P) transducers were selected due to the following features: good range of linear operation with linearity better than 0.2%, the response time for the sensors is less than 1µs and 40µs for current [50] and voltage [51] transducer respectively, which provides the response time necessary to acquire transients. The frequency bandwidth, 0 to 200 kHz, of the current transducer was good for applications for inverter-fed motors to allow for the high switching frequencies. The locked rotor current of the induction motors can be in the order of 500 to 800% of the rated current. Motor performance data (Table 4.1.1) for the fault simulator motor shows the locked rotor current is 16A, approximately 728% of Full Load Amps (FLA), which is well within the current transducers primary current measuring range (0 to ±50A [50]). The voltage transducer has a measuring range of 10 to 500 V; therefore, it is capable of measuring the supply voltage of the induction motor as indicated in Table 4.1.1. The LEM modules are connected as shown in Figure 4.1.7. The measuring resistors (R_M) were selected as 110Ω for both types of transducers based on datasheet. The primary resistor (R_1) was sized to be 100kΩ to provide the desired primary nominal current. The output of the transducers was connected to the DAQ system to record the three line current measurements and the three line-to-line voltage measurements.

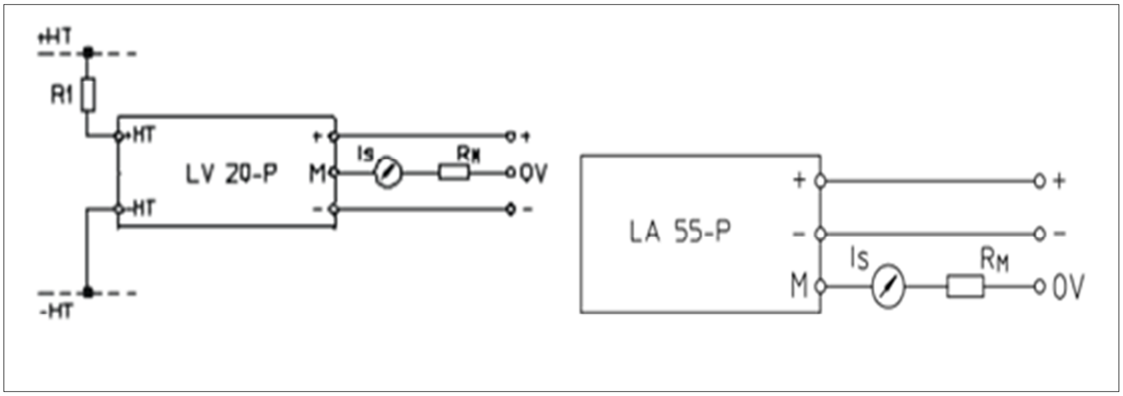


Figure 4.1.7: LV 20 and LA 55 Transducer Connections

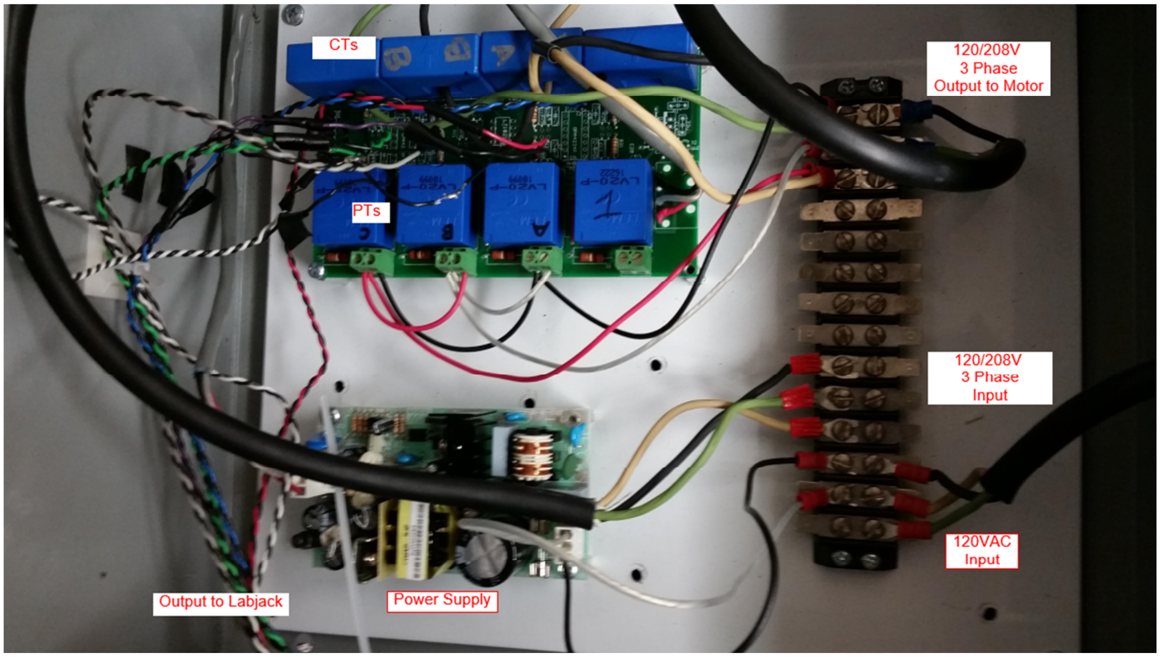


Figure 4.1.8: Non-Intrusive Load Monitoring

Accelerometers were selected to measure vibration for the reasons discussed in Chapter 2. Vibration measurements at the bearing blocks are obtained by using Integrated Circuit Piezoelectric (ICP) accelerometers. Three accelerometers were placed on each bearing block (Figure 4.1.9) to obtain vibration acceleration in three directions X (tangential), Y (radial) and Z (axial) as described in ISO standards for mechanical vibration [37]. The Model 308B accelerometers were used for the X and Y-axis while Model 353B33 accelerometers were used for the Z-Axis. Ideally, all six of the accelerometers would have been identical models; however, six identical models were not available in the lab. For that reason, the models used for the Z-axis were different but have similar characteristics, refer to Table 4.1.2. The model with the higher resonant frequency is used for the tangential and radial measurements to avoid distorted measurements. The bearing blocks were drilled and tapped for stud mounting of the accelerometers, as this method is ideal for obtaining high frequencies as discussed in Reference [52] and [37].

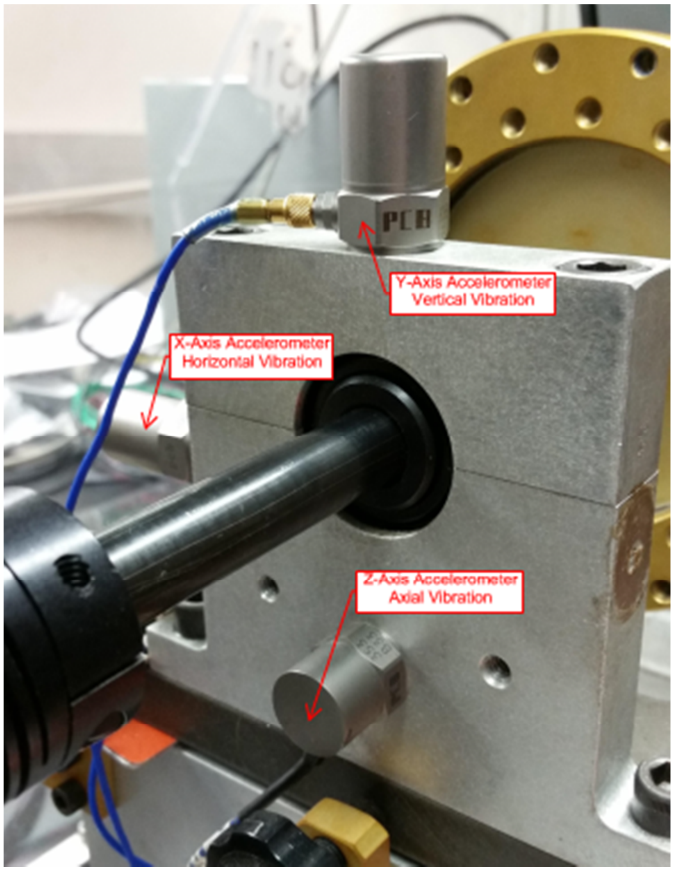


Figure 4.1.9: Accelerometer Mounting Locations

The ICP sensors used require a constant current excitation for proper operation. For this reason, the model 483A07 amplifying power unit was used as the constant current source for the accelerometers. The coaxial cables are routed from the sensor to the amplifier and then to the DAQ system.

Table 4.1.2: Accelerometer Specifications

Parameter	Model 308B [53]	Model 353B33 [52]
Resonant Frequency	25kHz	22kHz
Sensitivity	100mV/g	100mV/g
Measurement Range	±50 g pk	±50 g pk
Frequency Range	0.7 to 6000 Hz	0.7 to 6500 Hz

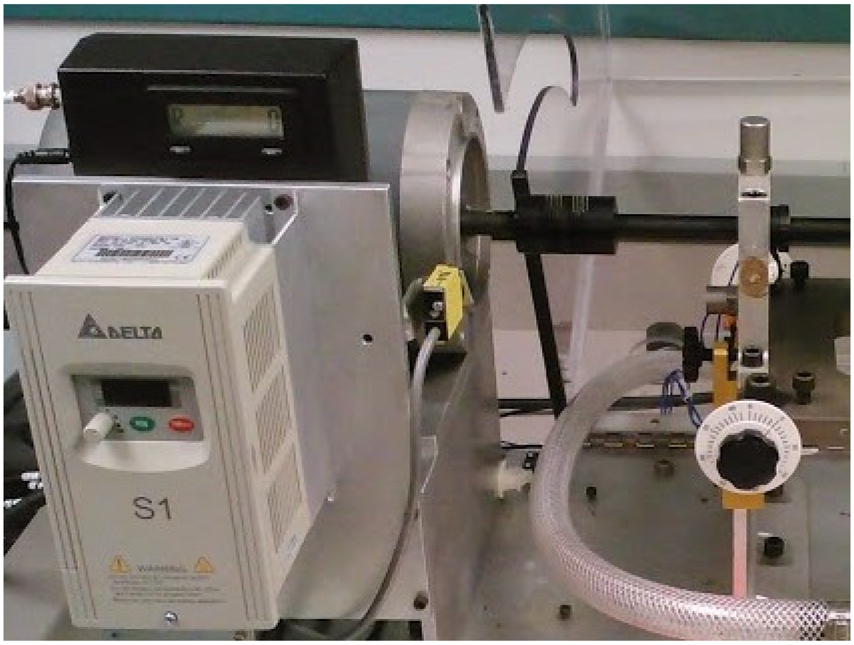


Figure 4.1.10: Speed Sensor

The fault simulator includes a built-in tachometer with LCD display and one pulse per revolution analog output (Figure 4.1.10). A LM2907N frequency to voltage converter was added to the output to convert the frequency of the pulses in the signal to a voltage that is linear with input frequency (Figure 4.1.11). The voltage output can then be fed into the DAQ.

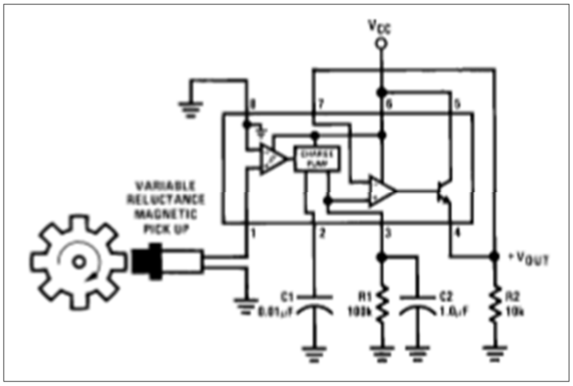


Figure 4.1.11: Frequency to Voltage Converter

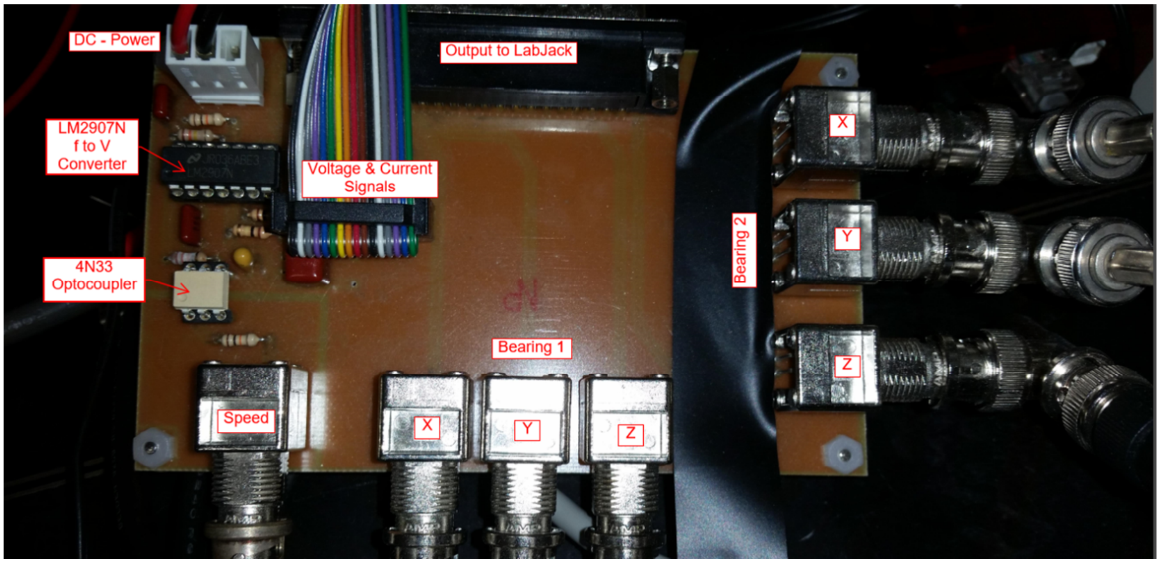


Figure 4.1.12: DAQ Input Connection Card

The thirteen sensor signals were input into the DAQ for recording and processing, refer to for list of signal input connections. The DAQ used for the system, UE9 LabJack, has a DB37 connector for input. Therefore, a Printed Circuit Board (PCB) board, Figure 4.1.12, was made to interface all signals from the sensors with the DAQ. The ribbon cable is used for the signals from the NILM, while BNC connectors are used for the speed and accelerometer signals. Refer to Appendix B for a list of signal connections to LabJack analog inputs. The PC board is also used for the mechanical speed LM2907 circuit. During preliminary testing and calibration of the system, significant noise was observed in the speed signal as demonstrated in Figure 4.1.13. A 4N33 optocoupler was added to the input of the LM 2907 to provide isolation between the speed sensor and the DAQ; as a result, the noise observed in the signal was reduced. Figure 4.1.14 displays the speed signal with a reduction in noise after the optocoupler was added to the circuit.

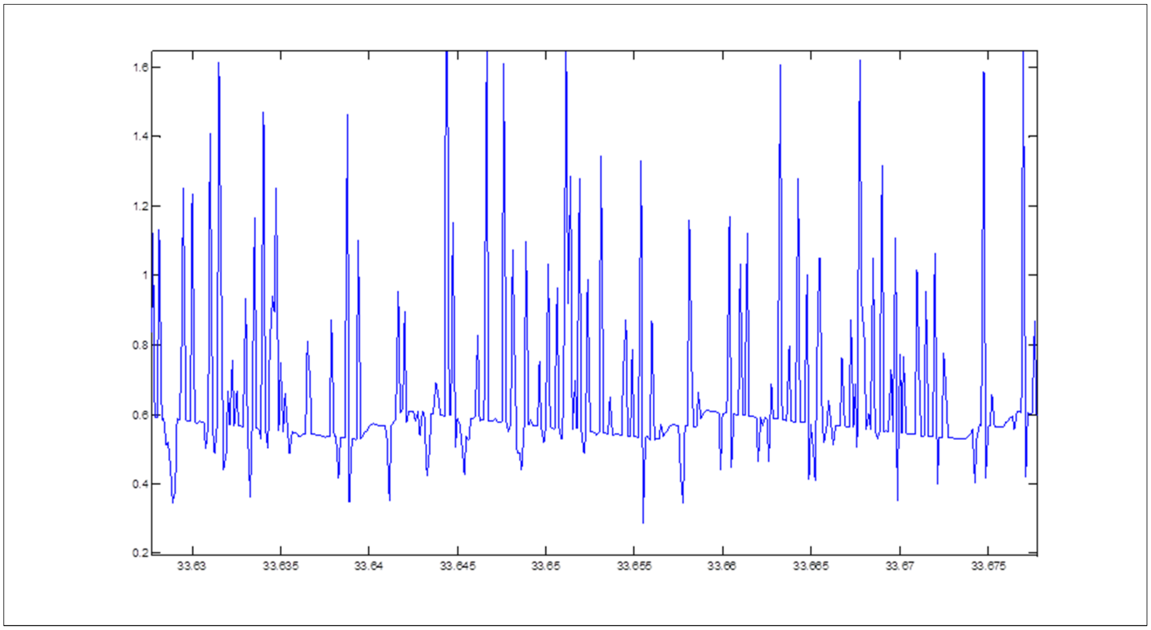


Figure 4.1.13: Noise in the Speed Signal

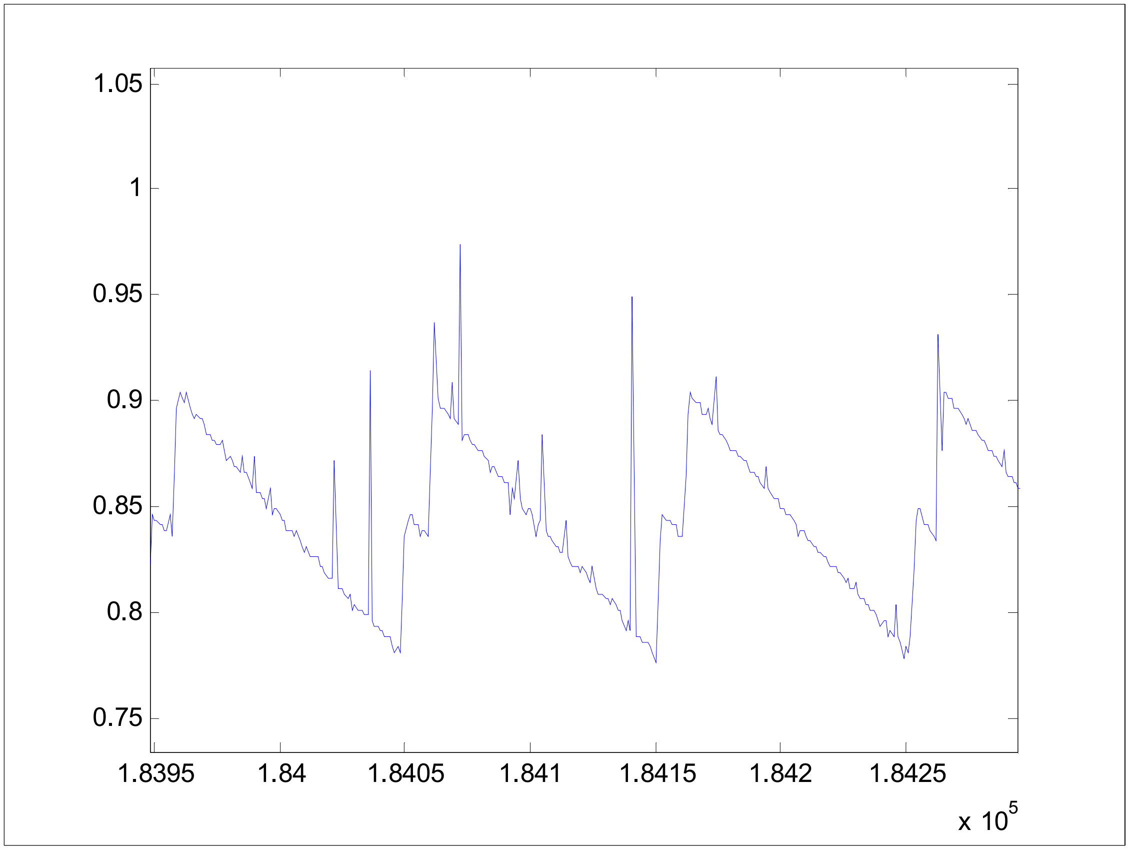


Figure 4.1.14: Speed Signal with Optocoupler

4.2 Results

Several experiments similar to typical motor operating conditions have been considered. This section describes the results of those tests.

4.2.1 Introducing faults in an off-line pump/motor setup

Many power plant systems use off-line motors to drive centrifugal turbo machines such as pumps and fans. Examples include Reactor Coolant Pumps (RCP), High Pressure Injection (HPI) pumps or Auxiliary Feedwater (AFW) pumps. To simulate these common systems, the motor was coupled to a centrifugal pump as shown in Figure 4.1.4. The pump drives fluid throughout a small distribution system that includes a large reservoir and several throttling valves.

Several different faults were inserted into the system as an initial test condition. The faults included the following:

- Bearing with inner-race damage (BPFI)
- Bearing with outer-race damage (BPFO)
- Bearing with a ball-spin fault (BSF)
- Bearing with a combination of the above faults (COMB)
- An unbalanced mechanical load
- An eccentric rotor disk
- A cocked rotor disk
- A faulted pump (with rubbing)
- A centrally bent shaft
- Motor with a broken rotor bar
- Motor with an unbalanced rotor
- Motor with a faulted machine bearing
- Motor with a bowed armature
- Motor with highly eccentric air gap

Figure 4.2.1, Figure 4.2.2, and Figure 4.2.3 shows the results of various tests. Each plot shows the Health Indicator (HI) value as a function of time for both normal and faulted conditions. The indicator value generally increases by a relatively large amount for each of these relatively early-stage fault conditions.

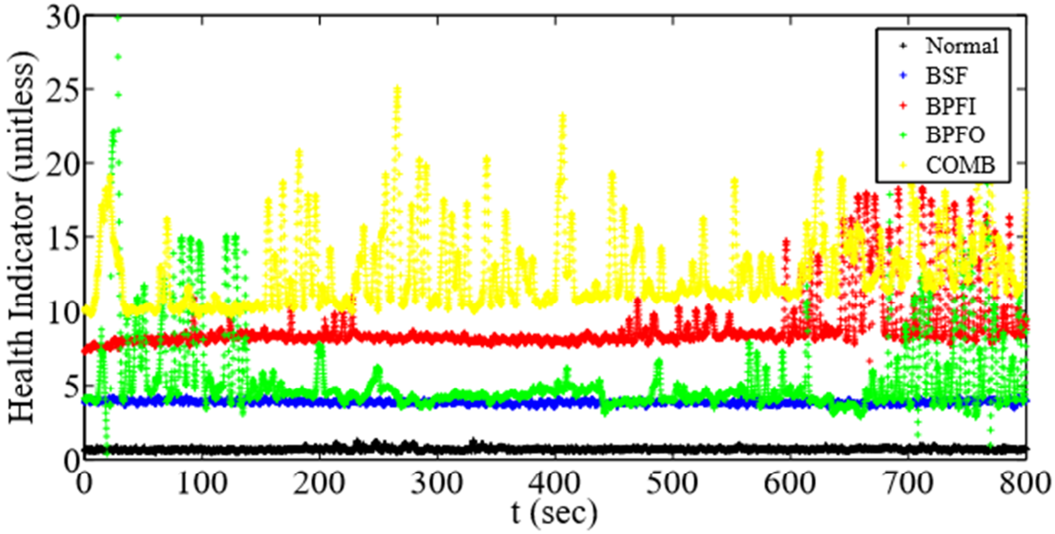


Figure 4.2.1: HI vs. time for bearing block faults

Under the proposed condition monitoring, data is recorded and collected by a power generation facilities DCS during normal operation. The DCS would perform the KPI extraction algorithm in real time. The condition monitoring algorithm initially learns an appropriate set of vectors once the motor has reached a steady-state operating condition. This condition can be determined by using the motors thermal time constant [54]. In general, most induction motors are operating continuously and are in a steady state over most of their operating lifetime. Once a fault develops, various KPI values will change, and the HI value will change as well. However, the HI value indicates only the existence of a fault and does not diagnosis the fault condition. Once the existence of a fault condition is identified, analysis of individual KPIs can be performed to diagnosis the fault condition. This step can be performed automatically or manually. During experimentation, it was found to be extremely helpful manually analyzing the data. With the data saved within the DCS network, operators, engineers or maintenance personnel could analyze the data at any workstation.

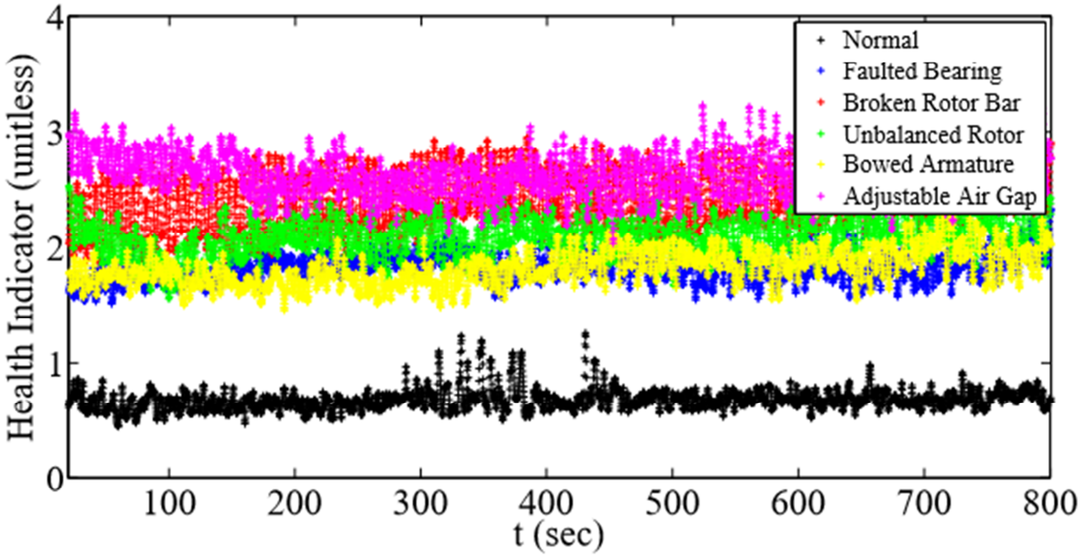


Figure 4.2.2: HI value vs time for faulted motors

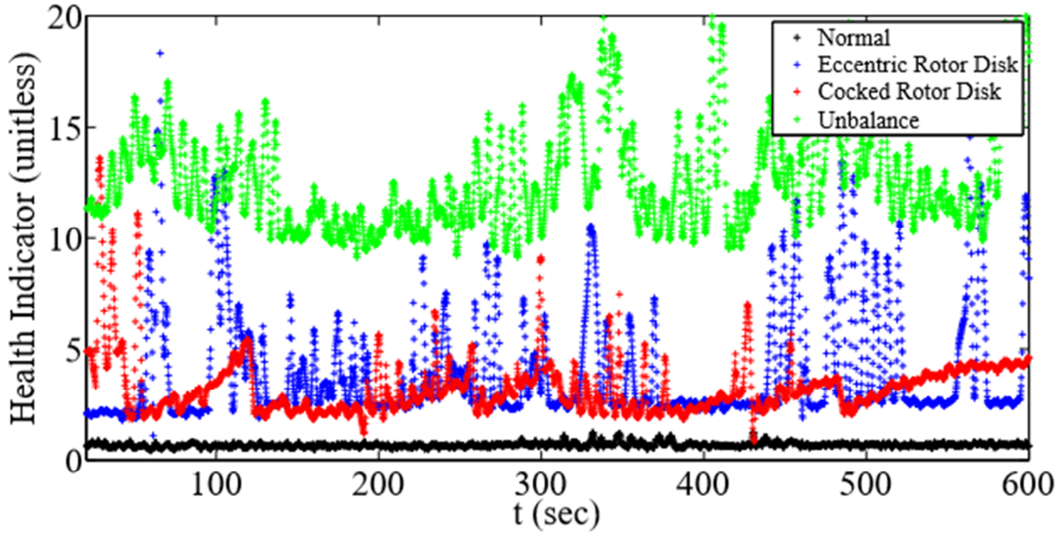


Figure 4.2.3: HI value vs time for faulted rotor disks

4.2.2 Effects of loading in the off-line pump/motor setup

In many systems, a pump will be operated at one speed while the flow rate is controlled by throttling a cutoff valve. This scenario was simulated in the setup by throttling the valves from fully open to 50% or 75% closed.

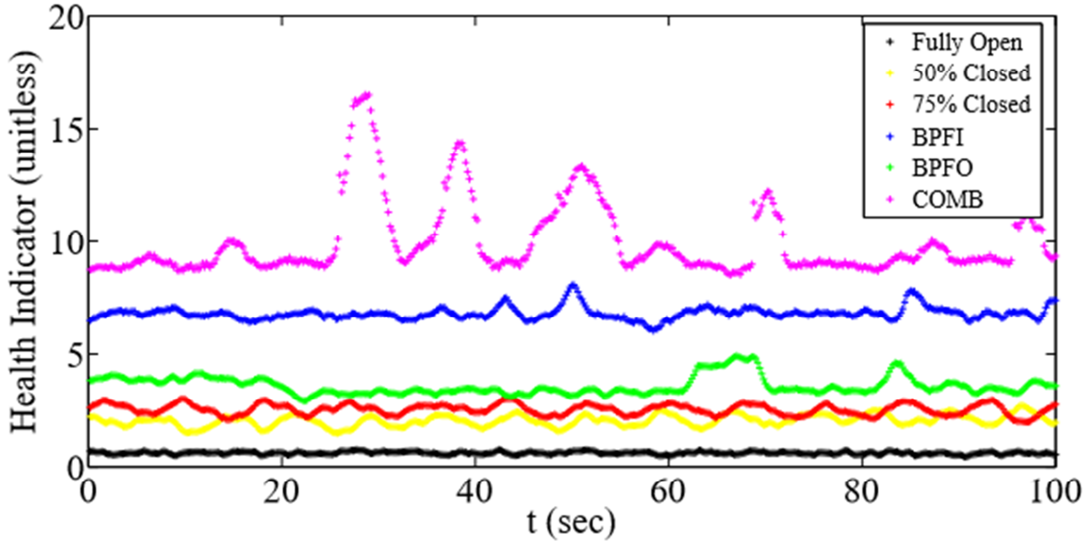


Figure 4.2.4: HI value vs. time for valve setting no fault and faults with valve open

As the valve was throttled, the mechanical speed KPI increased from 3448.2 RPM (fully open), to 3462.6 RPM (50% closed), to 3466.2 RPM (75% closed). Figure 4.2.4 shows the Health Indicator for a healthy system with the valve throttled and the system with faults and the valve opened. At first observation, it appears that the Health Indicator was indicating potential faults when the valve was throttled. The Health Indicator value clearly increases as the valve is being throttled. According to a power-law relationship [55], [2], the motor torque τ_L is related to flow rate Q:

$$\tau_L = \beta_1 Q^2 \qquad (4.1)$$

Where β_1 is an empirical constant. So as the valve is throttled, the flow rate drops and so does the required torque. As a result, the motor speed increases closer to synchronous and the motor sees a change in loading. Even though the Health Indicator value is lower during throttling than when faults are introduced, this scenario demonstrates the difficulty in distinguishing load changes from early-stage faults. The condition monitoring algorithm can be retrained if it is known that the load has changed. In this case, the change in speed is a clear indicator that the load setting has changed and that a re-training is needed. Such a retraining is relatively easy to perform on-the-fly and has been tested in real-time.

Another option that can be used in the field instead of retraining would be a long-term observation of the HI. If the load were known to vary, closer observations over time would show the HI would have a maximum range based on the load change. On the other hand, the HI would be steadily increasing as the fault condition worsened. The fault condition still could be distinguished from a load change and be corrected before significant failure of equipment occurred.

Even though this scenario demonstrates a difficulty with the algorithm, it does not indicate rules based approach is better. It is expected that individual KPI values also change with loading. Therefore, even a rules-based monitoring scheme would need to account for such activities if it were to be appropriately robust [2]. The proposed algorithm uses a single indicator to track changes and is thus potentially better in this regard.

4.3 Faults Detected

During the course of testing, a couple unexpected instances occurred that showed the potential of the proposed KPI and condition monitoring algorithms.

The first instance developed accidentally during the course of laboratory tests. Over the years of use, the bearing set screw had been overtightened and caused the shaft to bur up. As the configuration of the setup was being changed, one of the "healthy" bearings was stuck on the shaft and could not easily be removed. It was taken to the machine shop lab to try to remove the bearing without causing damage. It was placed such that even pressure could be applied to the bearing while force was used to remove the shaft. After the bearing was removed, a laboratory assistant dropped the bearing on the shop floor. No visible damage was detected, and the bearing seemed completely normal. As the bearing was spun, no indication of damage could be felt. However, during the subsequent test, which was believed to be a healthy, the Health Indicator jumped as shown in Figure 4.3.1. At first indication of the fault, it was believed that it might have been due to misalignment during the setup. The system was thoroughly checked for alignment and all bolts tightened. Additional tests were then run as shown in Figure 4.3.1.

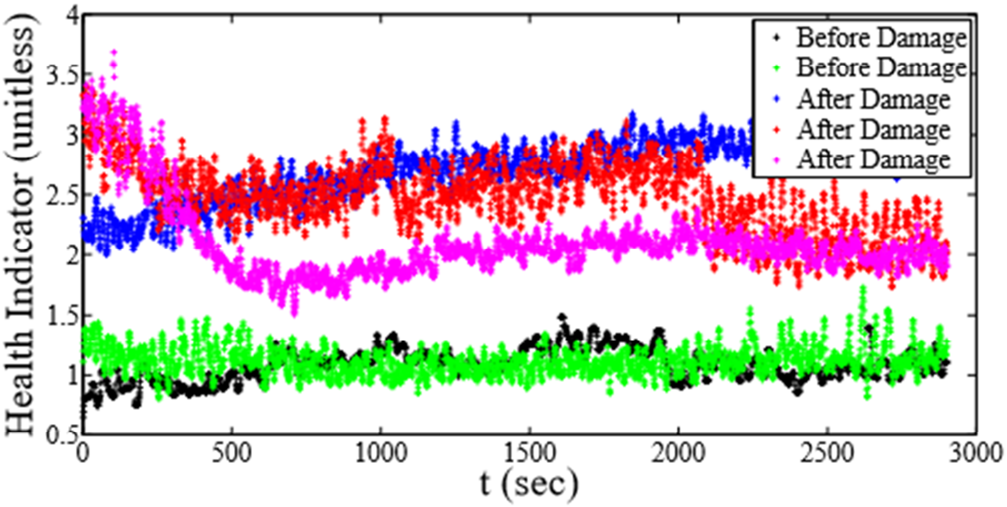


Figure 4.3.1: HI value versus time for the normal bearing before and after it was damaged in the laboratory

A comprehensive examination of the KPI values recorded before and after this fault provides strong support for the use of the proposed algorithm over a rules-based approach. Figure 4.3.2 shows the KPI for the velocity signal recorded from the y-axis accelerometer on the bearing housing. The plot clearly shows that a change has occurred in the bearings condition.

An examination of other fault indicators demonstrates how difficult the fault-detection process can be and how powerful the proposed method is for isolating changes. Figure 4.3.3 shows two examples, the amplitude of the ball-defect signal recorded by the y-axis accelerometer and the amplitude of the inner-race signal recorded by the x-axis accelerometer, respectively. In the case of the former, there is a clear change in mean; in the case of the latter, all of the data appears to be about the same. A change in steady-state speed was also observed before and after the maintenance was performed, and this is likely due to increased rolling friction [2]. The flow rate, however, was unchanged.

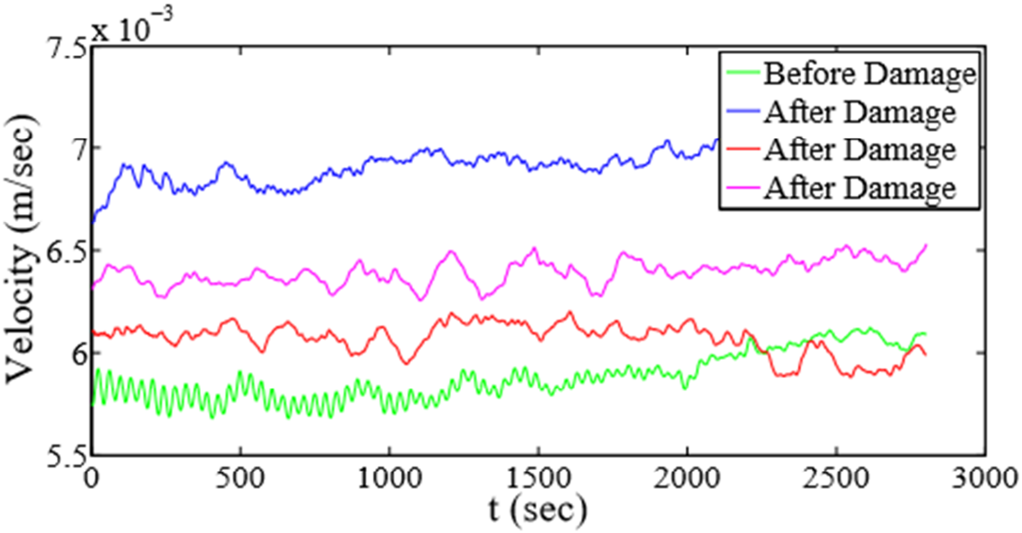


Figure 4.3.2: Velocity versus time for the normal bearing before and after it was damaged in the laboratory.

This example demonstrates the potential benefits of the proposed fault-detection algorithm. The changes in the various fault indicators were subtle and it is unlikely that a set of automated rules would have noticed. From the figure, it is clear that changes have occurred in the KPIs. However, in a rules based system it is unlikely that the changes are significant enough to be observed as a fault especially if data is collected by simple handheld meters. However, the proposed method using HI shows that there has been a change in the bearings condition. This HI could easily alert facility personnel of the need to perform a more detailed investigation of the individual KPIs.

This occurrence resembles a plant outage scenario. Prior to the plant outage, normal KPIs would be recorded. Post outage, the equipment can be operated and the HI would show if any change in condition has occurred the health of the equipment.

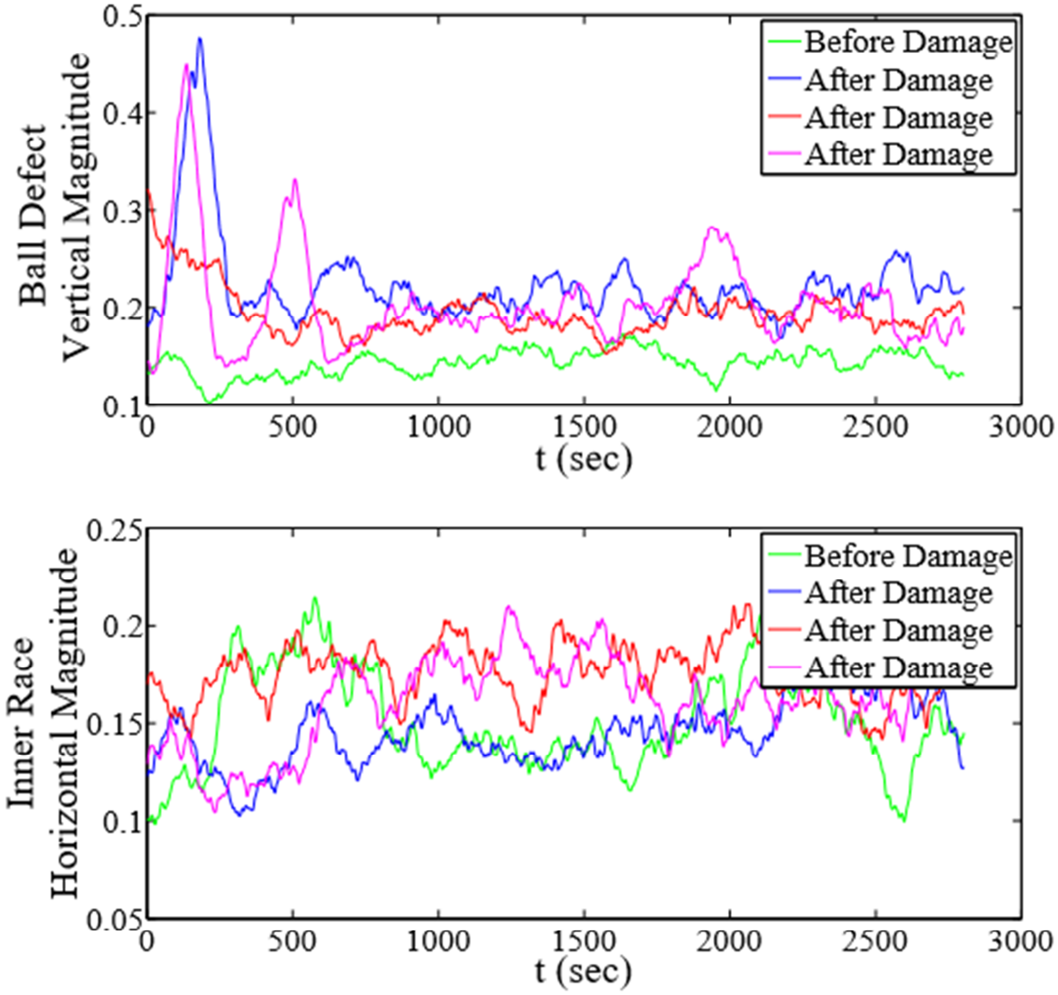


Figure 4.3.3: Amplitude of the ball-defect signal recorded by the vertical accelerometer and the inner-race signal recorded by the horizontal accelerometer before and after the normal bearing was damaged.

The second occurrence was during relocation of the setup. Originally, the setup was located within lab space in the Cameron building on campus. After the Energy Production and Infrastructure Center (EPIC) building was opened, the setup was relocated to the new lab space in EPIC. A short "normal" test was performed to validate the setup. The Figure 4.3.4 displays the THD for the setup when it was located within the Cameron Lab versus the THD after the setup was relocated to EPIC. The relocated THD shows that a change has occurred in the quality of the power supply. Additionally the spikes indicated problems with the setup.

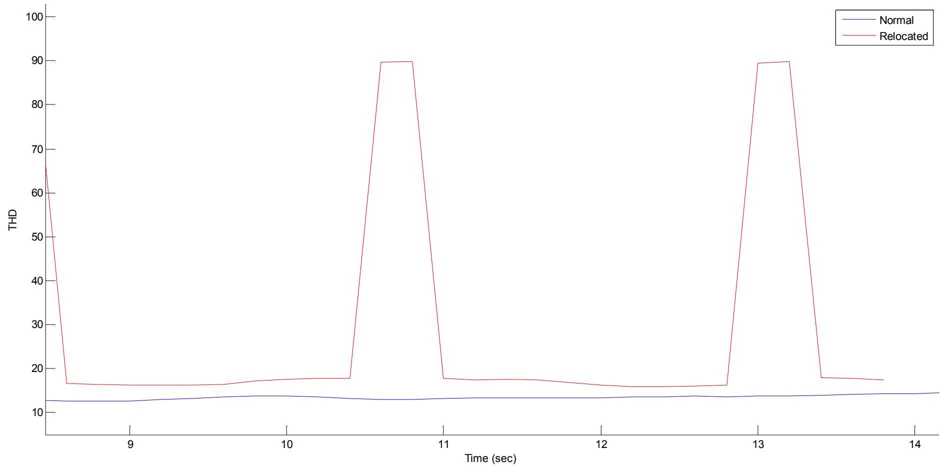


Figure 4.3.4: IEEE THD for Cameron Lab vs. EPIC Lab

A closer look at the no load voltages for each building before processing, Figure 4.3.5, shows a difference in the quality of voltage supplied.

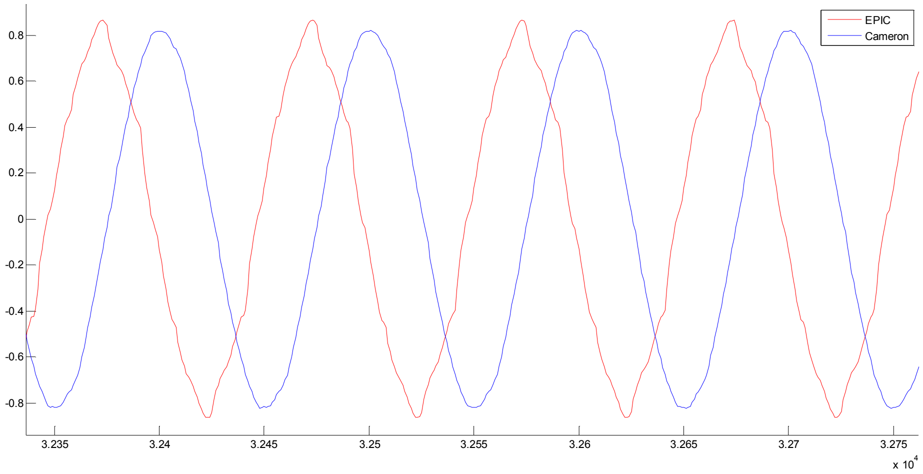


Figure 4.3.5: Unscaled voltage in Cameron Lab vs. EPIC Lab

5 CONCLUSION

5.1 Summary

This thesis proposes an approach for robust condition monitoring for power generation facilities that can detect faults and be used for fault diagnosis. This thesis develops a method for the online measurement of KPI in induction motors that was used in combination with a condition monitoring algorithm. The condition monitoring algorithm was developed based on a concept from facial recognition. The algorithm is used for fault detection in electric motors. Online methods for early stage fault detection will become essential as systems become more heavily dependent on power electronics. This thesis has demonstrated the potential of a robust, automated condition monitoring algorithm for electric motors and their loads. This method is demonstrated to be sensitive to very small changes in load behavior and that those changes can be correlated with the underlying KPI data, as well as maintenance records and other process-variable measurements.

5.2 Potential for Future Work

The approach for condition monitoring proposed in this thesis has several opportunities for future work. Ongoing work focuses on the inclusion of additional features and the completion of more testing for the motor condition monitoring. These tests include building a larger training set of normal motor and drive features. There is particular interest in testing with variations in load and looking into the effects of changes in ambient temperature.

Future testing to improve the fault detection scheme proposed in this thesis is to collect more data across a wide range of operating conditions. This can be done in the laboratory by using the adjustable magnetic brake attached the machinery fault simulator. A larger training set could be recorded to create a better and more extensive feature set with additional features added into the feature set. Some additional measurements include adding more accelerometers mounted onto the motor for additional vibration measurements. Another feature could be added is max volts per hertz to determine if damage is occurring during plant scenarios where a fast bus transfer is implemented. Other possibilities are to work with local companies to obtain small motors that are being replaced. A training set could be recorded for the new motor before installation. Then the old motor could be obtained and tested to see if the fault could be determined.

6 REFERENCES

[1] W.T.Thomson and R.J.Gilmore, "Motor Current Signature Analysis to Detect Faults in Induction Motor Drives - Fundamentalss, Data Interpretation, and Industrial Case Histories," in *Proceedings of the thirty second Turbomachinery Symposium*, 2003.

[2] J.M.Anderson, *Robust Condition Monitoring for Moderrn Power Conversion,* Charlotte: University of North Carolina at Charlotte, 2013.

[3] T.L.Skvarenina, W.E.DeWitt, "Three-Phase Induction Motors," in *Electrical Power and Controls*, New York, Prentice Hall, 2004, p. Chp 7.

[4] P.C.Sen, Principles of Electric Machines and Power Electronics, Kingston: John Wiley and Sons, 1997.

[5] J.Robinson, C.D.Whelan, and N.K.Haggerty, "Trends in Advanced Motor Protection and Monitoring," *IEEE Transaction on Industry Applications,* vol. 40, no. 3, pp. 853-860, 2004.

[6] P.F.Allbrecht, et al., "Assessment of the Reliability of Motors in Utility Applications - Updated," *IEEE Transaction on Energy Conversion,* vol. 1, no. 1, pp. 39-46, 1986.

[7] N.Mehala, *Condition Monitoring and Fault Diagnosis on Induction Motors Using Motor Current Signature Analysis,* Kurukshetra, 2010.

[8] H.Toliyat, Electric Machines Modeling, Condition Monitoring and Fault Diagnosis, New York: CRC Press Taylor and Francis Group, 2012.

[9] P.Zhang, "A Survey of Condition Monitoring and Protection Methods for Medium-Voltage Induction Motors," *IEEE Transactions on Industry Applications,* vol. 47,

no. 1, pp. 34-46, 2011.

[10] A.Siddique, "A Review of Stator Fault Monitoring Techniques of Induction Motors," *IEEE Transactions on Energy Conversion,* vol. 20, no. 1, pp. 106-114, 2005.

[11] J.C.Robertson, "Detecting Stator and Rotor Winding Faults in Three-Phase Induction Machines," ECE Technical Reports, 1995.

[12] K.S.Gaeid, "Fault Diagnosis of Induction Motors Using MCSA and FFT," *Electrical and Electronic Engineering,* pp. 85-92, 2011.

[13] Y.Xie, "Investigation of Broken Rotor Bar Faults in Three-Phase Squirrel-Cage Induction Motors," in *Finite Element Analysis – From Biomedical Applications to Industrial Developments*, Shanghai, Intech, 2012, pp. 477-498.

[14] N.B.Shaikh, "Online Off-Site Condition Monitoring Of Three Phase Induction Motor by Using GSM Technology," *International Journal of Emerging Technology and Advanced Engineering,* pp. 693-696, 2013.

[15] M.Blodt, Condition Monitoring of Mechanical Faults in Variable Speed Induction Motor Drives: Application of Stator Current Time-Frequency Analysis and Parameter Estimation, Toulouse, 2006.

[16] ISO 17359, *Condition monitoring and diagnosis of electric machines- General guidelines,* Switzerland: International Standards Organization, 2011.

[17] T.Assaf, H.Henao, G.Capolino, "Simplified Axial Flux spectrum Method to Detect Incipient Stator inter-Turn Short Circuit in Induction Machines," *IEEE International*

Symposium on Industrial Electronics, pp. 815-819, 2004.

[18] W.Pietrowski, *Application of radial neural network to diagnostics of induction motor stator faults using axial flux,* 2011.

[19] O.I.Okoro, "Steady and Transient States Thermal Analysis of a 7.5-kW SquirrelCage Induction Machine at Rated-Load Operation," *IEEE Transaction On Energy Conversion,* vol. 20, no. 4, 2005.

[20] R.C.Bhavsar, R.A.Patel, "Various Techniques for Condition Moniotirng of Three Phase Induction Motor- A Review," *Internation Journal of Engineering Inventions,* pp. 22-26, 2013.

[21] B.K.N.Rao, Handbook of Condition Monitoring 1st Edition, Oxford: Elsevier Science LTD, 1996.

[22] H.D.Haynes, "Electrical Signature Analysis (ESA) developments at the Oak Ridge diagnosis applied reseach center," Oak Ridge National Laboratories, Tennessee.

[23] P.Phumiphak, "Induction Motor Speed Measurement Using Motor Current Signature Analysis Technique," in *International Conference on Electrical Machines and Systems*, Tokyo, 2009.

[24] V.Wowk, "A Brief Tutorial on Machine Vibration," [Online]. Available: https://www.machinedyn.com/docs/machine_vibration_tutorial.pdf. [Accessed Mar 2016].

[25] SpectraQuest, Applied Vibration Analysis Training Manual & Laboratory Exercises, SpectraQuest.

[26] Meggitt, "Steps to Selecting the Right Accelerometer," Meggitt PLC, Dorset, 2012.

[27] A.V.Oppenheim, R.W.Schafer, Discrete-Time Signal Processing, New Jersey: Prentice-Hall, 1999.

[28] E.W.Kamen, Fundamentals of Signals and Systems Using the Web and Matlab, New Jersey: Prentice Hall, 2000.

[29] A.V.Oppenheim, A.S.Willsky, Signals and Systems, New Jersey: Prentice Hall, 1997.

[30] B.P.Lathi, Z.Ding, Modern Digital and Analog Communication Systems, New York: Oxford University Press, 2009.

[31] G.Didier, H.Razik, O.Caspary, "Rotor Cage Fault Detection in Induction Motor Using Global Modulation Index on the Instantaneous Power Spectrum," *IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives,* vol. 4th, pp. 104-109, 2003.

[32] I.Jaksch, "Rotor Cage Fault Detection in Induction Motors By Motor Current Demodulation Analysis," 2012.

[33] J.Wren, "Converting Acceleration, Velocity & Displacement," 16 Dec 2010. [Online]. Available: http://blog.prosig.com/2010/12/16/methods-of-conversion-between-acceleration-velocity-and-displacement/. [Accessed Apr 2012].

[34] P. Tavner et al., Condition Monitoring of Rotating Electrical Machines, London: The Institution of Engineering and Technology, 2008.

[35] NEMA MG 1, Motors and Generators, Rosslyn: National Electrical Manufacturers

Association, 2006.

[36] EXTECH Instruments, Heavy Duty Vibration Meter User Guide Model 407860, FLIR Company.

[37] ISO 10816-3, "Mechanical Vibration - Evaluation of Machine Vibration by Measurements on Non-Rotating Parts," International Organization for Standardization, Switzerland, 2003.

[38] W.R.Finley, M.M.Hodowanec, W.G.Holter, *An Analytical Approach to Solving Motor Vibration Problems,* IEE/PCIC 1999 Conference Record, 1999.

[39] M.Blodt, P.Granjon, B.Rostaing, "Models for Bearing Damage Detection in Induction Motors Using Stator Current Monitoring," *IEEE Transaction on Industrial Electronics,* vol. 55, no. 4, pp. 1813-1822, 2007.

[40] S.Nadi, H.A.Toliyat, L.Xiaodong, "Condition Monitoring and Fault Diagnosis of Electrical Motors a Review," *IEEE Transaction on Energy Conversion,* vol. 20, no. 4, pp. 719-729, 2005.

[41] IEEE Standard 519, IEEE Recommended Practice and Requirements for Harmonic Control in Electric Power Systems, Institute of Electrical and Electronic Engineers, 2014.

[42] IEC 61000, Electromagnetic Compatibility in General, International Electrotechnical Commission, 2002.

[43] F. Filippetti et al., "AI Techniques in Induction Machines Diagnosis Including the Speed Ripple Effect," *IEEE Transactions on Industry Applications,* vol. 34, no. 1,

pp. 98-108, 1998.

[44] G.B.Kilman et al, "Noninvasive Detection of Broken Rotor Bars In Operating Induction Motors," *IEEE Transaction on Energy Conversion,* vol. 3, no. 4, pp. 873-879, 1988.

[45] R.R.Schoen, T.G.Habetle, F.Kamran, "Motor Bearing Damage Detection Using Stator Current Monitoring," *IEEE Transactions on Industry Applications,* vol. 31, no. 6, pp. 1274-1279, 1995.

[46] M.Turk and A.Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience,* vol. 3, no. 1, pp. 71-86, 1991.

[47] SpectraQuest Inc., "Machinery Fault Simulator," Spectraquest, [Online]. Available: http://spectraquest.com/products/simulators/machinery-fault-simulators/ . [Accessed 2012].

[48] Marathon Motors, "Product Information, Catalog No. D391," [Online]. Available: https://www.regalbeloit.com/Products/Catalog?model=056T34F5301. [Accessed September 2017].

[49] Rexnord Corporation, "Rexnord Bearings," 2012. [Online]. Available: https://www.rexnord.com/Products-Services/Process-Motion-Control/Bearings. [Accessed 18 April 2012].

[50] LEM Components, "Current Transducer LA 55-P," [Online]. Available: http://www.lem.com/docs/products. [Accessed 23 Feb 2012].

[51] LEM Components, "Voltage Transducer LV 20-P," [Online]. Available:

http://www.lem.com/docs/products. [Accessed 23 Feb 2012].

[52] PCB Piezotronics Vibration Division, *Model 353B33 ICP Accelerometer Installation and Operating Manual,* Depew, NY: PCB Group Company, 2002.

[53] PCB Piezotronics, *Voltage Output Accelerometer Specification 308-2010-80,* Depew, NY: PCB Group Company, 1990.

[54] A.E.Fitzgerald, C.Kingsley, S.Umans, Electric Machinery, 6th edition, McGraw-Hill, Inc. , 2002.

[55] R.Isermann, Fault-Diagnosis Systems, Berlin: Springer, 2006.

[56] SpectraQuest Inc., [Online]. [Accessed 2012].

APPENDIX A: MATLAB CODE

APPENDIX A.1: Function “KPIpro2”

```
Function [ KPI ] = KPIpro2( rawdata,f_sample )
% KPIpro2 finds the 187 key performance indicators from 13 input signals
%   rawdata is the 13 column matrix of recorded data from induction motor
%   rawdata(:,1) Line AB voltage
%   rawdata(:,2) phase A current
%   rawdata(:,3) Line BC voltage
%   rawdata(:,4) phase B current
%   rawdata(:,5) Line CA voltage
%   rawdata(:,6) phase C current
%   rawdata(:,7) speed
%   rawdata(:,8) block 1-x accelerometer data
%   rawdata(:,9) block 1-y accelerometer data
%   rawdata(:,10) block 1-z accelerometer data
%   rawdata(:,11) block 2-x accelerometer data
%   rawdata(:,12) block 2-y accelerometer data
%   rawdata(:,13) block 2-z accelerometer data
%   f_sample is the sampling frequency used when collecting the data

% Forrest Suter, 2012-04-26

% Copyright (c) Forrest Suter, 2012-2017
% forrestsuter@gmail.com

%%
%Setting up the output KPI matrix
%KPI data is given for every 0.2sec of operation
numsamples = length(rawdata(:,1)); %determining the number of samples taken during testing
KPIsampling = f_sample/5; %defines number of samples during a 0.2 sec period
vectorref = 1:KPIsampling:numsamples; %creates a reference vector for number of samples every 0.2sec
time = 0:0.2:numsamples/f_sample; %creates a time vector for plots of KPI vs time

%pre-allocating the KPI matrix length and writes zeros to all positions
KPI = zeros(length(vectorref),187);
KPI (:,187) = time; %assigns time vector to KPI vector 187
%
%Scaling the raw data from Labjack
scaled = zeros(length(rawdata),13); %initializes scaled matrix length and 13 columns, writes zeros to all positions
```

```
%scale raw line voltage data
scaled(:,1) = ((rawdata(:,1)*(100000/110)*(1/2.5))); %Line AB voltage with Labjack
outputting in voltage
scaled(:,2) = ((rawdata(:,3)*(100000/110)*(1/2.5))); %Line BC voltage with Labjack
outputting in voltage
scaled(:,3) = ((rawdata(:,5)*(100000/110)*(1/2.5))); %Line CA voltage with Labjack
outputting in voltage

%scale raw current data
scaled(:,4) = (1000*(rawdata(:,2))/110); %phase A current with Labjack outputting in
voltage
scaled(:,5) = (1000*(rawdata(:,4))/110); %phase B current with Labjack outputting in
voltage
scaled(:,6) = (1000*(rawdata(:,6))/110); %phase C current with Labjack outputting in
voltage

%scale raw mechanical speed data
scaled(:,7) = (rawdata(:,7)); %scaling not needed if using labjack and use -v -c in code
saving data
%scale raw accelerometer data
scaled(:,8) = rawdata(:,8)*(9.80665/0.1); %converts the block 1-x accelerometer data
from V to m/s^2 based on use of 100mv/g accelerometers
scaled(:,9) = rawdata(:,9)*(9.80665/0.1); %converts the block 1-y accelerometer data
from V to m/s^2 based on use of 100mv/g accelerometers
scaled(:,10) = rawdata(:,10)*(9.80665/0.1); %converts the block 1-z accelerometer data
from V to m/s^2 based on use of 100mv/g accelerometers
scaled(:,11) = rawdata(:,11)*(9.80665/0.1); %converts the block 2-x accelerometer data
from V to m/s^2 based on use of 100mv/g accelerometers
scaled(:,12) = rawdata(:,12)*(9.80665/0.1); %converts the block 2-y accelerometer data
from V to m/s^2 based on use of 100mv/g accelerometers
scaled(:,13) = rawdata(:,13)*(9.80665/0.1); %converts the block 2-z accelerometer data
from V to m/s^2 based on use of 100mv/g accelerometers

%%
%Electrical frequency is determined
%filter current before  finding zero crossing locations
Num = get(FIRfilter3,'Numerator'); %gets filter data from FIR filter design
filtcurrent = filtfilt (Num,1,scaled(:,5)); %filters the noise out of the current signal

%determine electrical frequency by zero crossing
zercrs = crossing(filtcurrent); % finds and returns index of locations of zero crossings

%frequency calculation
frqcal = zeros (1,length(zercrs));
```

```
for d = 1:length (frqcal)
    if d == 1
        frqcal (d) = 0;
    else
        deltazercrs = zercrs(d) - zercrs (d-1); %determine number of samples between zero
crossings
        if deltazercrs == 1
            frqcal(d) = 0;
        else
            frqcal(d)= (1/(deltazercrs/f_sample))/2; %first converts samples to sec, then
period length in sec to frequency
        end
    end
end

%create a frequency vector the length of the scaled vectors
frequency = zeros (length(scaled(:,4)),1); %sets up the size of the vector and writes zeros
to all locations
for m = 1:length (frqcal)
   if m == 1
        frequency (m:zercrs(m)) = 0;
   else
        frequency(zercrs(m-1)+1:zercrs(m))=frqcal(m); %writes the calculated frequency to
frequency vector index matching zercrossing index
   end
end

%%
%RMS voltage calculation using KPIrms2 function
KPI(:,4) = KPIrms2( scaled(:,1),vectorref,f_sample ); %Line AB rms voltage

KPI(:,5) = KPIrms2( scaled(:,2),vectorref,f_sample ); %Line BC rms voltage

KPI(:,6) = KPIrms2( scaled(:,3),vectorref,f_sample ); %Line CA rms voltage

%%
%RMS current calculation using KPIrms2 function
KPI(:,1) = KPIrms2( scaled(:,4),vectorref,f_sample ); %phase A rms current

KPI(:,2) = KPIrms2( scaled(:,5),vectorref,f_sample ); %phase B rms current

KPI(:,3) = KPIrms2( scaled(:,6),vectorref,f_sample ); %phase C rms current

%%
```

```
%time scaling frequency for KPI vector using KPItscaling
KPI(:,34) = KPItscaling( frequency,vectorref);

%%
% rotational speed in HZ using the run_speed function
KPI(:,41) = run_speed(scaled(:,7),f_sample);

%%
%slip frequency calculated using slipratio function
slip  = slipratio( KPI(:,34),KPI(:,41),vectorref );
KPI(:,108) = slip; %slip assigned to KPI vector

%%
%broken rotor bar side band frequency calculated using rotorbarsidebands function
rotorbar = rotorbarsidebands( slip, vectorref );

%%
%bearing fault harmonics in vibration spectrum calculated using bearingfreqs
bearingvibfreq = bearingfreqs( KPI(:,41), vectorref );

%%
% bearing fault harmonics in current spectrum calculated using bearingcurrent
bearingcurfreq = bearingcurrent(KPI(:,34),bearingvibfreq, vectorref);

%%
%Current Spectrum
%Find the magnitude of current at the 1st, 3rd, 5th, 7th, 9th, 11th, 13th Harmonics
%Find the magnitude of the rotor bar side band freq
%Find the magnitude of the bearing faults

%phase A current harmonics
rmsharmonA = scaled(:,4)/sqrt(2); %calculate RMS
[RMSharmonmagA,RMSharmonmagAb,RMSharmonmagAc] =
harmagloc3(rmsharmonA,f_sample,vectorref,KPI(:,34),rotorbar, bearingcurfreq);
KPI(:,9:15) = RMSharmonmagA;% current harmonics
KPI(:,109:114) = RMSharmonmagAb;% Broken rotor bar side band
KPI(:,157:166) = RMSharmonmagAc;% Faulted Bearing

%phase B current harmonics
rmsharmonB = scaled(:,5)/sqrt(2); %calculate RMS
[RMSharmonmagB,RMSharmonmagBb,RMSharmonmagBc] =
harmagloc3(rmsharmonB,f_sample,vectorref,KPI(:,34),rotorbar, bearingcurfreq);
KPI(:,18:24) = RMSharmonmagB; % current harmonics
KPI(:,115:120) = RMSharmonmagBb; % Broken rotor bar side band
KPI(:,167:176) = RMSharmonmagBc; % Faulted Bearing
```

```
%phase C current harmonics
rmsharmonC = scaled(:,6)/sqrt(2); %calculate RMS
[RMSharmonmagC,RMSharmonmagCb,RMSharmonmagCc] =
harmagloc3(rmsharmonC,f_sample,vectorref,KPI(:,34),rotorbar, bearingcurfreq);
KPI(:,27:33) = RMSharmonmagC; % current harmonics
KPI(:,121:126) = RMSharmonmagCb; % Broken rotor bar side band
KPI(:,177:186) = RMSharmonmagCc; % Faulted Bearing

%%
%IEEE total harmonic distortion Phase A
IEEETHDA = zeros (length(vectorref),1);
for ha = 1:length(vectorref)
   IEEETHDA(ha) =(sqrt((KPI(ha,1)^2)-(KPI(ha,9)^2)))/KPI(ha,9);
end
KPI(:,7) = real(IEEETHDA)*100;

%IEEE total harmonic distortion Phase B
IEEETHDB = zeros (length(vectorref),1);
for hb = 1:length(vectorref)
   IEEETHDB(hb) =(sqrt((KPI(hb,2)^2)-(KPI(hb,18)^2)))/KPI(hb,18);
end
KPI(:,16) = real(IEEETHDB)*100;

%IEEE total harmonic distortion Phase C
IEEETHDC = zeros (length(vectorref),1);
for hc = 1:length(vectorref)
   IEEETHDC(hc) =(sqrt((KPI(hc,3)^2)-(KPI(hc,27)^2)))/KPI(hc,27);
end
KPI(:,25) = real(IEEETHDC)*100;

%%
% IEC total harmonic distortion Phase A
IECTHDA = zeros (length(vectorref),1);
for ga = 1:length(vectorref)
   IECTHDA(ga) =(sqrt((KPI(ga,1)^2)-(KPI(ga,9)^2)))/(sqrt(KPI(ga,1)^2));
end
KPI(:,8) = real(IECTHDA)*100;

% IEC total harmonic distortion Phase B
IECTHDB = zeros (length(vectorref),1);
for gb = 1:length(vectorref)
   IECTHDB(gb) =(sqrt((KPI(gb,2)^2)-(KPI(gb,18)^2)))/(sqrt(KPI(gb,2)^2));
end
KPI(:,17) = real(IECTHDB)*100;
```

```
% IEC total harmonic distortion Phase C
IECTHDC = zeros (length(vectorref),1);
for gc = 1:length(vectorref)
   IECTHDC(gc) =(sqrt((KPI(gc,3)^2)-(KPI(gc,27)^2)))/(sqrt(KPI(gc,3)^2));
end
KPI(:,26) = real(IECTHDC)*100;

%%
%velocity calculated with velrms function

%Accelerometer 1
KPI(:,35) = velrms( scaled(:,8),vectorref,f_sample );

%Accelerometer 2
KPI(:,36) = velrms( scaled(:,9),vectorref,f_sample );

%Accelerometer 3
KPI(:,37) = velrms( scaled(:,10),vectorref,f_sample );

%Accelerometer 4
KPI(:,38) = velrms( scaled(:,11),vectorref,f_sample );

%Accelerometer 5
KPI(:,39) = velrms( scaled(:,12),vectorref,f_sample );

%Accelerometer 6
KPI(:,40) = velrms(scaled(:,13),vectorref,f_sample );

%%
%rms band lf,mf,hf
%lf band 10 to 200 hz
%mf band 200 to 2000 hz
%hf band 2000+ hz
KPI(:,42:44) = bands( scaled(:,8),f_sample,vectorref ); %Accelerometer 1
KPI(:,53:55) = bands( scaled(:,9),f_sample,vectorref ); %Accelerometer 2
KPI(:,64:66) = bands( scaled(:,10),f_sample,vectorref ); %Accelerometer 3
KPI(:,75:77) = bands( scaled(:,11),f_sample,vectorref ); %Accelerometer 4
KPI(:,86:88) = bands( scaled(:,12),f_sample,vectorref ); %Accelerometer 5
KPI(:,97:99) = bands( scaled(:,13),f_sample,vectorref ); %Accelerometer 6
%%
%vibration 0.5,1,2,3,4 rotational freq for bearing block 1 x axis
[ vibmag1x,vibelec1x, bearingvib1x] =
Vibharmmech2(scaled(:,8),f_sample,vectorref,KPI(:,41),KPI(:,34),bearingvibfreq);
KPI(:,45:49) = vibmag1x; %vibration harmonic multiples of rotational speed
KPI(:,50:52) = vibelec1x; %vibration harmonic multiples of electrical frequency
KPI(:,127:131) = bearingvib1x; %vibration harmonic at bearing frequencies

%vibration 0.5,1,2,3,4 rotational freq for bearing block 1 y axis
[ vibmag1y,vibelec1y, bearingvib1y] =
Vibharmmech2(scaled(:,9),f_sample,vectorref,KPI(:,41),KPI(:,34),bearingvibfreq);
KPI(:,56:60) = vibmag1y; %vibration harmonic multiples of rotational speed
KPI(:,61:63) = vibelec1y; %vibration harmonic multiples of electrical frequency
KPI(:,132:136) = bearingvib1y; %vibration harmonic at bearing frequencies

%vibration 0.5,1,2,3,4 rotational freq for bearing block 1 z axis
[ vibmag1z,vibelec1z, bearingvib1z] =
Vibharmmech2(scaled(:,10),f_sample,vectorref,KPI(:,41),KPI(:,34),bearingvibfreq);
KPI(:,67:71) = vibmag1z; %vibration harmonic multiples of rotational speed
KPI(:,72:74) = vibelec1z; %vibration harmonic multiples of electrical frequency
KPI(:,137:141) = bearingvib1z; %vibration harmonic at bearing frequencies

%vibration 0.5,1,2,3,4 rotational freq for bearing block 2 x axis
[ vibmag2x,vibelec2x, bearingvib2x] =
Vibharmmech2(scaled(:,11),f_sample,vectorref,KPI(:,41),KPI(:,34),bearingvibfreq);
KPI(:,78:82) = vibmag2x; %vibration harmonic multiples of rotational speed
KPI(:,83:85) = vibelec2x; %vibration harmonic multiples of electrical frequency
KPI(:,142:146) = bearingvib2x; %vibration harmonic at bearing frequencies

%vibration 0.5,1,2,3,4 rotational freq for bearing block 2 y axis
[ vibmag2y,vibelec2y ,bearingvib2y] =
Vibharmmech2(scaled(:,12),f_sample,vectorref,KPI(:,41),KPI(:,34),bearingvibfreq);
KPI(:,89:93) = vibmag2y; %vibration harmonic multiples of rotational speed
KPI(:,94:96) = vibelec2y; %vibration harmonic multiples of electrical frequency
KPI(:,147:151) = bearingvib2y; %vibration harmonic at bearing frequencies

%vibration 0.5,1,2,3,4 rotational freq for bearing block 2 z axis
[ vibmag2z,vibelec2z ,bearingvib2z] =
Vibharmmech2(scaled(:,13),f_sample,vectorref,KPI(:,41),KPI(:,34),bearingvibfreq);
KPI(:,100:104) = vibmag2z; %vibration harmonic multiples of rotational speed
KPI(:,105:107) = vibelec2z; %vibration harmonic multiples of electrical frequency
KPI(:,152:156) = bearingvib2z; %vibration harmonic at bearing frequencies

%%
end
```

APPENDIX A.2: Function “CROSSING”

```
function [ind,t0,s0,t0close,s0close] = crossing(S,t,level,imeth)
% CROSSING find the crossings of a given level of a signal
%   ind = CROSSING(S) returns an index vector ind, the signal
%   S crosses zero at ind or at between ind and ind+1
%   [ind,t0] = CROSSING(S,t) additionally returns a time
%   vector t0 of the zero crossings of the signal S. The crossing
%   times are linearly interpolated between the given times t
%   [ind,t0] = CROSSING(S,t,level) returns the crossings of the
%   given level instead of the zero crossings
%   ind = CROSSING(S,[],level) as above but without time interpolation
%   [ind,t0] = CROSSING(S,t,level,par) allows additional parameters
%   par = {'none'|'linear'}.
%   With interpolation turned off (par = 'none') this function always
%   returns the value left of the zero (the data point thats nearest
%   to the zero AND smaller than the zero crossing).
%
%   [ind,t0,s0] = ... also returns the data vector corresponding to
%   the t0 values.
%
%   [ind,t0,s0,t0close,s0close] additionally returns the data points
%   closest to a zero crossing in the arrays t0close and s0close.
%
%   This version has been revised incorporating the good and valuable
%   bugfixes given by users on Matlabcentral. Special thanks to
%   Howard Fishman, Christian Rothleitner, Jonathan Kellogg, and
%   Zach Lewis for their input.

% Steffen Brueckner, 2002-09-25
% Steffen Brueckner, 2007-08-27     revised version

% Copyright (c) Steffen Brueckner, 2002-2007
% brueckner@sbrs.net

% check the number of input arguments
error(nargchk(1,4,nargin));

% check the time vector input for consistency
if nargin < 2 || isempty(t)
   % if no time vector is given, use the index vector as time
   t = 1:length(S);
elseif length(t) ~= length(S)
   % if S and t are not of the same length, throw an error
```

```
    error('t and S must be of identical length!');
end

% check the level input
if nargin < 3
    % set standard value 0, if level is not given
    level = 0;
end

% check interpolation method input
if nargin < 4
    imeth = 'linear';
end

% make row vectors
t = t(:)';
S = S(:)';

% always search for zeros. So if we want the crossing of
% any other threshold value "level", we subtract it from
% the values and search for zeros.
S   = S - level;

% first look for exact zeros
ind0 = find( S == 0 );

% then look for zero crossings between data points
S1 = S(1:end-1) .* S(2:end);
ind1 = find( S1 < 0 );

% bring exact zeros and "in-between" zeros together
ind = sort([ind0 ind1]);

% and pick the associated time values
t0 = t(ind);
s0 = S(ind);

if strcmp(imeth,'linear')
    % linear interpolation of crossing
    for ii=1:length(t0)
        if abs(S(ind(ii))) > eps(S(ind(ii)))
            % interpolate only when data point is not already zero
            NUM = (t(ind(ii)+1) - t(ind(ii)));
            DEN = (S(ind(ii)+1) - S(ind(ii)));
            DELTA =  NUM / DEN;
            t0(ii) = t0(ii) - S(ind(ii)) * DELTA;
            % I'm a bad person, so I simply set the value to zero
            % instead of calculating the perfect number ;)
            s0(ii) = 0;
        end
    end
end

% Addition:
% Some people like to get the data points closest to the zero crossing,
% so we return these as well
%[CC,II] = min(abs([S(ind-1) ; S(ind) ; S(ind+1)]),[],1);
%ind2 = ind + (II-2); %update indices

%t0close = t(ind2);
%s0close = S(ind2);
```

APPENDIX A.3: Function “KPIrms2”

```
function [ rms_signal ] = KPIrms2( signal,vectorref,f_sample )
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
rms_signal = zeros(length(vectorref),1);
start_ind = f_sample*3 * ones(1,length(vectorref));
temp = find(vectorref < f_sample*3);
start_ind(temp) = vectorref(temp) - 1;

for xx = 1:length(vectorref)
    if xx == 1
        rms_signal(xx) = 0;
    else
        rms_signal (xx,1) = sqrt(sum(signal(vectorref(1,xx)-start_ind(xx):vectorref(1,xx)-
1).^2)/start_ind(xx));
    end
end

end
```

APPENDIX A.4: Function “KPItscaling”

```
function [ KPIscaled ] = KPItscaling( signal,vectorref )
% KPItscaling converts scaled data to a 0.2sec period for KPI output
%   signal is the vector being converted
%   vectorref provides the number of samples per 0.2sec


% Forrest Suter, 2012-04-26

% Copyright (c) Forrest Suter, 2012-2017
% forrestsuter@gmail.com

KPIscaled = zeros(length(vectorref),1);

for p = 1:length(vectorref)

    if p == 1
        KPIscaled (p,1) = 0;
    else
        KPIscaled (p,1) = median(signal(vectorref(p-1):vectorref(p)-1));
    end
end

end
```

APPENDIX A.5: Function “run_speed”

```
Function [ speedmag ] = run_speed( speeddata, f_sample )
% run_speed converts raw speed signal to a rotational frequency in Hz
%   speeddata is the recorded speed signal
%   f_sample the number of samples signal recorded


% Forrest Suter, 2012-04-26

% Copyright (c) Forrest Suter, 2012-2017
% forrestsuter@gmail.com

signal = speeddata;
vectorref = 1:f_sample/5:length(signal);
speed = zeros (length(vectorref),1);
run = zeros (length(vectorref),1);
speedmag = zeros (length(vectorref),1);
start_ind = f_sample*3 * ones(1,length(vectorref));
temp = find(vectorref < f_sample*3);
start_ind(temp) = vectorref(temp) - 1;
for xx = 1:length(vectorref)
  if xx == 1
     speed (xx) = 0;
  else
     run (xx) = mean(signal(vectorref(1,xx)-start_ind(xx):vectorref(xx)-1));
     speed (xx) = run (xx)/(15*(0.0097*10^-6)*99200*0.9942);
  end
  speedmag (xx) = speed (xx);%*60; % speed magnitude in RPM

end

end
```

APPENDIX A.6: Function “slipratio”

```
function [ slipr ] = slipratio( elecfreq , mechfreq , vectorref  )
% slipratio calculates the slip ratio from the electrical frequency and the mechanical

% Forrest Suter, 2012-04-26

% Copyright (c) Forrest Suter, 2012-2017
% forrestsuter@gmail.com slipr = zeros(length(vectorref),1);
for yy = 1:length(vectorref)
   if yy == 1
      slipr(yy,:) = 0;
   else
      slipr(yy) = (elecfreq(yy) - mechfreq(yy))/elecfreq(yy);
   end
end

end
```

APPENDIX A.7: Function “rotorbarsidebands”

```
function [ rotorfreq ] = rotorbarsidebands( slip, vectorref )
% rotorbarsidebands calculates the frequencies related to broken rotor bars

% Forrest Suter, 2012-04-26

% Copyright (c) Forrest Suter, 2012-2017
% forrestsuter@gmail.com
rotorfreq = zeros(length(vectorref),6);
for xx = 1:length(vectorref)
        rotorfreq(xx,1) = (1 - 2*3*slip(xx));
        rotorfreq(xx,2) = (1 - 2*2*slip(xx));
        rotorfreq(xx,3) = (1 - 2*1*slip(xx));
        rotorfreq(xx,4) = (1 + 2*1*slip(xx));
        rotorfreq(xx,5) = (1 + 2*2*slip(xx));
        rotorfreq(xx,6) = (1 + 2*3*slip(xx));
    end

for yy = 1:length(vectorref)
    for rr=1:1:6
        if rotorfreq(yy,rr)<0
            rotorfreq(yy,rr) = abs(rotorfreq(yy,rr)) ;
        end
    end
end

end
```

APPENDIX A.8: Function “bearingfreqs”

```
function [ bearingfreq ] = bearingfreqs( Mechspeed, vectorref )
% bearingfreqs calculates the bearing fault harmonics in vibration

% Forrest Suter, 2012-04-26

% Copyright (c) Forrest Suter, 2012-2017
% forrestsuter@gmail.com
bearingfreq = zeros(length(vectorref),5);
for xx = 1:length(vectorref)
        bearingfreq(xx,1) = (0.0332*60*Mechspeed(xx)); %Ball spin bearing fault
        bearingfreq(xx,2) = (0.0063*60*Mechspeed(xx)); %Fundamental Train fault
        bearingfreq(xx,3) = (0.0825*60*Mechspeed(xx)); %Inner ring defect fault
        bearingfreq(xx,4) = (0.0508*60*Mechspeed(xx)); %Outer ring defect fault
        bearingfreq(xx,5) = (0.0663*60*Mechspeed(xx)); %Ball defect fault
        %end
end
 for yy = 1:length(vectorref)
    for rr=1:1:5
        if bearingfreq(yy,rr)<0
            bearingfreq(yy,rr) = abs(bearingfreq(yy,rr)) ;
        end
    end
 end

end
```

APPENDIX A.9: Function “bearingcurrent”

```
function [ fbng ] = bearingcurrent( elecfrq, bearingfreq, vectorref )
% bearingcurrent calculates the bearing fault harmonics in current spectrum

% Forrest Suter, 2012-04-26

% Copyright (c) Forrest Suter, 2012-2017
% forrestsuter@gmail.com
fbng = zeros(length(vectorref),10);
fbnga = zeros(length(vectorref),5);
fbngb = zeros(length(vectorref),5);
for xx = 1:length(vectorref)
    for yy = 1:1:5
        fbnga(xx,yy) = abs(elecfrq(xx) - 1*bearingfreq(xx,yy));
        fbngb(xx,yy) = abs(elecfrq(xx) + 1*bearingfreq(xx,yy));
    end
    fbng(xx,1:5) = fbnga(xx,:);
    fbng(xx,6:10) = fbngb(xx,:);
end

end
```

APPENDIX A.10: Function “harmagloc3”

```
function  [harmonmag] = harmagloc(signal,f_sample,vectorref,KPI)
%harmagloc locates and outputs the Harmonic magnitudes

% Forrest Suter, 2012-04-26

% Copyright (c) Forrest Suter, 2012-2017
% forrestsuter@gmail.com
max_Y = zeros (1,7);
harmonmag = zeros(length(vectorref),7);
start_ind = f_sample*3 * ones(1,length(vectorref));
temp = find(vectorref < f_sample*3);
start_ind(temp) = vectorref(temp) - 1;

for s = 1:length(vectorref)
   if s == 1
      for r=1:2:13
         max_Y((r+1)/2) = 0;
      end
   else if KPI(s) == 0
         for r=1:2:13
            max_Y((r+1)/2) = 0;
         end
   else
      y = (signal(vectorref(1,s)-start_ind(s):vectorref(1,s)-1));
      N=length (y);
      f=[0:1:N/2-1]*(f_sample/N);
      har=4/length(y)*fft(y.*hann(length(y)),N);
      harmag = abs(har);

      for r=1:2:13
         ind_vec = find(f>r*KPI(s)-2 & f<r*KPI(s)+2);
         [~, ind] = max(harmag(ind_vec));
         ind = ind_vec(ind);
         max_Y((r+1)/2) = harmag(ind);
      end
      end
   end
      harmonmag(s,:) = max_Y;

end
end
```

APPENDIX A.11: Function “velrms”

```
function [ KPI ] = velrms( acceldata,vectorref,f_sample )
% velrms calculates the rms velocity of vibration

% Forrest Suter, 2012-04-26

% Copyright (c) Forrest Suter, 2012-2017
% forrestsuter@gmail.com
Num = get(Highpass1,'Numerator');
filtaccel = filtfilt (Num,1,acceldata);
temp1 = fft(filtaccel);
temp1(1) = 0;
integ1 = 1/6000*cumtrapz(real(ifft(temp1)));
KPI = KPIrms2( integ1,vectorref,f_sample );
end
```

APPENDIX A.12: Function “bands”

```
unction [ band_rms ] = bands( signal,f_sample,vectorref )
% bands calculates the rms vibration in bands of frequencies

% Forrest Suter, 2012-04-26

% Copyright (c) Forrest Suter, 2012-2017
% forrestsuter@gmail.com
lfband_rms = zeros(length(vectorref),1);
mfband_rms = zeros(length(vectorref),1);
hfband_rms = zeros(length(vectorref),1);
band_rms = zeros(length(vectorref),3);
%start_ind = zeros(length(vectorref));
start_ind = f_sample*3 * ones(1,length(vectorref));
temp = find(vectorref < f_sample*3);
start_ind(temp) = vectorref(temp) - 1;
for xx = 1:length(vectorref)
    if xx == 1
        lfband_rms(xx) = 0;
        mfband_rms(xx) = 0;
        hfband_rms(xx) = 0;
    else
        y = signal(vectorref(1,xx)-start_ind(xx):vectorref(1,xx)-1);
        N = length (y);
        NumUniquePts = ceil((N+1)/2);
        freq = (0:NumUniquePts-2)*f_sample/(N-1);
        accelfft = fft(y);
        accelmag = abs (accelfft);
        ind_vec = find(freq>10 & freq<200);
        lfband = (accelmag(ind_vec));
        lfband_rms (xx) = sqrt(sum((abs(lfband)/(N)).^2));
        ind_vec2 = find(freq>200 & freq<2000);
        mfband = (accelmag(ind_vec2));
        mfband_rms (xx) = sqrt(sum((abs(mfband)/(N)).^2));
        ind_vec3 = find(freq>2000);
        hfband = (accelmag(ind_vec3));
        hfband_rms (xx) = sqrt(sum((abs(hfband)/(N)).^2));
            end
    band_rms(xx,1)= lfband_rms (xx);
    band_rms(xx,2)= mfband_rms (xx);
    band_rms(xx,3)= hfband_rms (xx);
    end
 end
```

APPENDIX A.13: Function “Vibharmmech2”

```
function [ vibharmonmag,vibharelec,vibharbearing  ] =
Vibharmmech2(signal,f_sample,vectorref,KPI,elecfq,bearingvibfq )
% Vibharmmech2 calculates the magnitude of harmonics in vibration spectrum

% Forrest Suter, 2012-04-26

% Copyright (c) Forrest Suter, 2012-2017
% forrestsuter@gmail.com N=2^16;
f=[0:1:N/2-1]*(f_sample/N);
max_Y = zeros (1,5);
max_YY = zeros (1,5);
max_YX = zeros (1,3);
vibharmonmag = zeros(length(vectorref),5);
vibharbearing = zeros(length(vectorref),5);
vibharelec = zeros(length(vectorref),3);
h = [1,0.5,2,3,4];
hh = [1,2,6];
start_ind = f_sample*3 * ones(1,length(vectorref));
temp = find(vectorref < f_sample*3);
start_ind(temp) = vectorref(temp) - 1;
for s = 1:length(vectorref)
   if s == 1
      for r=1:1:5
         max_Y(r) = 0;
         max_YY(r) = 0;
      end
      for rr=1:1:3
         max_YX(rr) = 0;
      end
   else if KPI(s) == 0
         for r=1:1:5
            max_Y(r) = 0;
            max_YY(r) = 0;
         end
         for rr=1:1:3
            max_Y(rr) = 0;
         end
   else
      y = (signal(vectorref(1,s)-start_ind(s):vectorref(1,s)-1));
      har=4/length(y)*fft(y.*hann(length(y)),N);
      harmag = abs(har);
```

```
        for r=1:1:5
           ind_vec = find(f>h(r)*KPI(s)-0.5 & f<h(r)*KPI(s)+0.5);
           [~, ind] = max(harmag(ind_vec));
           ind = ind_vec(ind);
           max_Y(r) = harmag(ind);
        end
         for rr=1:1:5
           ind_vec = find(f>bearingvibfq(s,rr)-1 & f<bearingvibfq(s,rr)+1);
           [~, ind] = max(harmag(ind_vec));
           ind = ind_vec(ind);
           max_YY(rr) = harmag(ind);
        end
        for rf=1:1:3
           ind_vec = find(f>hh(rf)*elecfq(s)-0.5 & f<hh(rf)*elecfq(s)+0.5);
           [~, ind] = max(harmag(ind_vec));
           ind = ind_vec(ind);
           max_YX(rf) = harmag(ind);
        end
        end
    end

    vibharmonmag(s,:) = max_Y;
    vibharbearing(s,:) = max_YY;
    vibharelec(s,:) = max_YX;

end

end
```

APPENDIX A.14: Function “FIRFilter”

```
function Hd = FIRfilter3
%FIRFILTER3 Returns a discrete-time filter object.

%
% M-File generated by MATLAB(R) 7.9 and the Signal Processing Toolbox 6.12.
%
% Generated on: 07-Mar-2012 16:31:12
%

% Equiripple Lowpass filter designed using the FIRPM function.

% All frequency values are in Hz.
Fs = 6000;  % Sampling Frequency

Fpass = 70;              % Passband Frequency
Fstop = 400;             % Stopband Frequency
Dpass = 0.057501127785;  % Passband Ripple
Dstop = 0.01;            % Stopband Attenuation
dens  = 20;              % Density Factor

% Calculate the order from the parameters using FIRPMORD.
[N, Fo, Ao, W] = firpmord([Fpass, Fstop]/(Fs/2), [1 0], [Dpass, Dstop]);

% Calculate the coefficients using the FIRPM function.
b  = firpm(N, Fo, Ao, W, {dens});
Hd = dfilt.dffir(b);

% [EOF]
```

APPENDIX A.15: Function “Highpass1”

```
function Hd = Highpass1
%HIGHPASS1 Returns a discrete-time filter object.

%
% M-File generated by MATLAB(R) 7.9 and the Signal Processing Toolbox 6.12.
%
% Generated on: 26-Mar-2012 19:04:27
%

% Equiripple Highpass filter designed using the FIRPM function.

% All frequency values are in Hz.
Fs = 6000;  % Sampling Frequency

Fstop = 3;               % Stopband Frequency
Fpass = 55;              % Passband Frequency
Dstop = 1e-007;          % Stopband Attenuation
Dpass = 0.057501127785;  % Passband Ripple
dens  = 20;              % Density Factor

% Calculate the order from the parameters using FIRPMORD.
[N, Fo, Ao, W] = firpmord([Fstop, Fpass]/(Fs/2), [0 1], [Dstop, Dpass]);

% Calculate the coefficients using the FIRPM function.
b  = firpm(N, Fo, Ao, W, {dens});
Hd = dfilt.dffir(b);

% [EOF]
```

APPENDIX B: DAQ INPUT

Channel	DB37 Pin	Labjack Input	Description
1	37	AIN0	Line AB Voltage
2	18	AIN1	Phase A Current
3	36	AIN2	Line BC Voltage
4	17	AIN3	Phase B Current
5	35	AIN4	Line CA Voltage
6	16	AIN5	Phase C Current
7	34	AIN6	Mechanical Speed
8	15	AIN7	Block 1 Accelerometer X
9	33	AIN8	Block 1 Accelerometer Y
10	14	AIN9	Block 1 Accelerometer Z
11	32	AIN10	Block 2 Accelerometer X
12	13	AIN11	Block 2 Accelerometer Y
13	31	AIN12	Block 2 Accelerometer Z

APPENDIX C: KPI PRO OUTPUT KEY

	KPI	Description
1	CurrentA.RMSCalc.Amps	Phase A RMS Current
2	CurrentB.RMSCalc.Amps	Phase B RMS Current
3	CurrentC.RMSCalc.Amps	Phase C RMS Current
4	Voltage_ab.RMSCalc.Volts	Line AB RMS Voltage
5	Voltage_bc.RMSCalc.Volts	Line BC RMS Voltage
6	Voltage_ca.RMSCalc.Volts	Line CA RMS Voltage
7	THD.CurrentA.IEEE	Phase A Current THD using IEEE Method
8	THD.CurrentA.IEC	Phase A Current THD using IEC Method
9	CurharidA.OneE.Mag	Phase A Current 1st harmonic of electrical freq
10	CurharidA.ThreeE.Mag	Phase A Current 3rd harmonic of electrical freq
11	CurharidA.FiveE.Mag	Phase A Current 5th harmonic of electrical freq
12	CurharidA.SevenE.Mag	Phase A Current 7th harmonic of electrical freq
13	CurharidA.NineE.Mag	Phase A Current 9th harmonic of electrical freq
14	CurharidA.ElevenE.Mag	Phase A Current 11th harmonic of electrical freq
15	CurharidA.ThirteenE.Mag	Phase A Current 13th harmonic of electrical freq
16	THD.CurrentB.IEEE	Phase B Current THD using IEEE Method
17	THD.CurrentB.IEC	Phase B Current THD using IEC Method
18	CurharidB.OneE.Mag	Phase B Current 1st harmonic of electrical freq
19	CurharidB.ThreeE.Mag	Phase B Current 3rd harmonic of electrical freq
20	CurharidB.FiveE.Mag	Phase B Current 5th harmonic of electrical freq
21	CurharidB.SevenE.Mag	Phase B Current 7th harmonic of electrical freq
22	CurharidB.NineE.Mag	Phase B Current 9th harmonic of electrical freq
23	CurharidB.ElevenE.Mag	Phase B Current 11th harmonic of electrical freq
24	CurharidB.ThirteenE.Mag	Phase B Current 13th harmonic of electrical freq
25	THD.CurrentC.IEEE	Phase C Current THD using IEEE Method
26	THD.CurrentC.IEC	Phase C Current THD using IEC Method
27	CurharidC.OneE.Mag	Phase C Current 1st harmonic of electrical freq
28	CurharidC.ThreeE.Mag	Phase C Current 3rd harmonic of electrical freq
29	CurharidC.FiveE.Mag	Phase C Current 5th harmonic of electrical freq
30	CurharidC.SevenE.Mag	Phase C Current 7th harmonic of electrical freq
31	CurharidC.NineE.Mag	Phase C Current 9th harmonic of electrical freq
32	CurharidC.ElevenE.Mag	Phase C Current 11th harmonic of electrical freq
33	CurharidC.ThirteenE.Mag	Phase C Current 13th harmonic of electrical freq
34	Run.Speed.Frequency	Electrical Frequency
35	AccMeter_1.VelRMS.Calc	Accelerometer 1 RMS Velocity
36	AccMeter_2.VelRMS.Calc	Accelerometer 2 RMS Velocity
37	AccMeter_3.VelRMS.Calc	Accelerometer 3 RMS Velocity

	KPI	Description
38	AccMeter_4.VelRMS.Calc	Accelerometer 4 RMS Velocity
39	AccMeter_5.VelRMS.Calc	Accelerometer 5 RMS Velocity
40	AccMeter_6.VelRMS.Calc	Accelerometer 6 RMS Velocity
41	Run.Speed.Magnitude	Rotational Speed
42	RMS_Band1.LF.RMS	RMS of Lower freq band 10 to 200 Hz Accel-1
43	RMS_Band1.MF.RMS	RMS of Med freq band 200 to 2000 Hz Accel-1
44	RMS_Band1.HF.RMS	RMS of Higher freq band >2000 Hz Accel-1
45	Vibtonid.OneR1.Mag	Vibration magnitude at 1*rotational speed Accel-1
46	Vibtonid.HalfR1.Mag	Vibration magnitude at .5*rotational speed Accel-1
47	Vibtonid.TwoR1.Mag	Vibration magnitude at 2*rotational speed Accel-1
48	Vibtonid.ThreeR1.Mag	Vibration magnitude at 3*rotational speed Accel-1
49	Vibtonid.FourR1.Mag	Vibration magnitude at 4*rotational speed Accel-1
50	Vibtonid.OneE1.Mag	Vibration magnitude at 1*electrical freq Accel-1
51	Vibtonid.TwoE1.Mag	Vibration magnitude at 2* electrical freq Accel-1
52	Vibtonid.SixE1.Mag	Vibration magnitude at 6* electrical freq Accel-1
53	RMS_Band2.LF.RMS	RMS of Lower freq band 10 to 200 Hz Accel-2
54	RMS_Band2.MF.RMS	RMS of Med freq band 200 to 2000 Hz Accel-2
55	RMS_Band2.HF.RMS	RMS of Higher freq band >2000 Hz Accel-2
56	Vibtonid.OneR2.Mag	Vibration magnitude at 1*rotational speed Accel-2
57	Vibtonid.HalfR2.Mag	Vibration magnitude at .5*rotational speed Accel-2
58	Vibtonid.TwoR2.Mag	Vibration magnitude at 2*rotational speed Accel-2
59	Vibtonid.ThreeR2.Mag	Vibration magnitude at 3*rotational speed Accel-2
60	Vibtonid.FourR2.Mag	Vibration magnitude at 4*rotational speed Accel-2
61	Vibtonid.OneE2.Mag	Vibration magnitude at 1*electrical freq Accel-2
62	Vibtonid.TwoE2.Mag	Vibration magnitude at 2* electrical freq Accel-2
63	Vibtonid.SixE2.Mag	Vibration magnitude at 6* electrical freq Accel-2
64	RMS_Band3.LF.RMS	RMS of Lower freq band 10 to 200 Hz Accel-3
65	RMS_Band3.MF.RMS	RMS of Med freq band 200 to 2000 Hz Accel-3
66	RMS_Band3.HF.RMS	RMS of Higher freq band >2000 Hz Accel-3
67	Vibtonid.OneR3.Mag	Vibration magnitude at 1*rotational speed Accel-3
68	Vibtonid.HalfR3.Mag	Vibration magnitude at .5*rotational speed Accel-3
69	Vibtonid.TwoR3.Mag	Vibration magnitude at 2*rotational speed Accel-3
70	Vibtonid.ThreeR3.Mag	Vibration magnitude at 3*rotational speed Accel-3
71	Vibtonid.FourR3.Mag	Vibration magnitude at 4*rotational speed Accel-3
72	Vibtonid.OneE3.Mag	Vibration magnitude at 1*electrical freq Accel-3
73	Vibtonid.TwoE3.Mag	Vibration magnitude at 2* electrical freq Accel-3
74	Vibtonid.SixE3.Mag	Vibration magnitude at 6* electrical freq Accel-3
75	RMS_Band4.LF.RMS	RMS of Lower freq band 10 to 200 Hz Accel-4
76	RMS_Band4.MF.RMS	RMS of Med freq band 200 to 2000 Hz Accel-4

	KPI	Description
77	RMS_Band4.HF.RMS	RMS of Higher freq band >2000 Hz Accel-4
78	Vibtonid.OneR4.Mag	Vibration magnitude at 1*rotational speed Accel-4
79	Vibtonid.HalfR4.Mag	Vibration magnitude at .5*rotational speed Accel-4
80	Vibtonid.TwoR4.Mag	Vibration magnitude at 2*rotational speed Accel-4
81	Vibtonid.ThreeR4.Mag	Vibration magnitude at 3*rotational speed Accel-4
82	Vibtonid.FourR4.Mag	Vibration magnitude at 4*rotational speed Accel-4
83	Vibtonid.OneE4.Mag	Vibration magnitude at 1*electrical freq Accel-4
84	Vibtonid.TwoE4.Mag	Vibration magnitude at 2* electrical freq Accel-4
85	Vibtonid.SixE4.Mag	Vibration magnitude at 6* electrical freq Accel-4
86	RMS_Band5.LF.RMS	RMS of Lower freq band 10 to 200 Hz Accel-5
87	RMS_Band5.MF.RMS	RMS of Med freq band 200 to 2000 Hz Accel-5
88	RMS_Band5.HF.RMS	RMS of Higher freq band >2000 Hz Accel-5
89	Vibtonid.OneR5.Mag	Vibration magnitude at 1*rotational speed Accel-5
90	Vibtonid.HalfR5.Mag	Vibration magnitude at .5*rotational speed Accel-5
91	Vibtonid.TwoR5.Mag	Vibration magnitude at 2*rotational speed Accel-5
92	Vibtonid.ThreeR5.Mag	Vibration magnitude at 3*rotational speed Accel-5
93	Vibtonid.FourR5.Mag	Vibration magnitude at 4*rotational speed Accel-5
94	Vibtonid.OneE5.Mag	Vibration magnitude at 1*electrical freq Accel-5
95	Vibtonid.TwoE5.Mag	Vibration magnitude at 2* electrical freq Accel-5
96	Vibtonid.SixE5.Mag	Vibration magnitude at 6* electrical freq Accel-5
97	RMS_Band6.LF.RMS	RMS of Lower freq band 10 to 200 Hz Accel-6
98	RMS_Band6.MF.RMS	RMS of Med freq band 200 to 2000 Hz Accel-6
99	RMS_Band6.HF.RMS	RMS of Higher freq band >2000 Hz Accel-6
100	Vibtonid.OneR6.Mag	Vibration magnitude at 1*rotational speed Accel-6
101	Vibtonid.HalfR6.Mag	Vibration magnitude at .5*rotational speed Accel-6
102	Vibtonid.TwoR6.Mag	Vibration magnitude at 2*rotational speed Accel-6
103	Vibtonid.ThreeR6.Mag	Vibration magnitude at 3*rotational speed Accel-6
104	Vibtonid.FourR6.Mag	Vibration magnitude at 4*rotational speed Accel-6
105	Vibtonid.OneE6.Mag	Vibration magnitude at 1*electrical freq Accel-6
106	Vibtonid.TwoE6.Mag	Vibration magnitude at 2* electrical freq Accel-6
107	Vibtonid.SixE6.Mag	Vibration magnitude at 6* electrical freq Accel-6
108	Slip	Slip
109	Broken rotor bars.A -3*s	Phase A Current rotor bar at -3*slip
110	Broken rotor bars.A -2*s	Phase A Current rotor bar at -2*slip
111	Broken rotor bars.A -1*s	Phase A Current rotor bar at -1*slip
112	Broken rotor bars.A +1*s	Phase A Current rotor bar at 1*slip
113	Broken rotor bars.A +2*s	Phase A Current rotor bar at 2*slip
114	Broken rotor bars.A +3*s	Phase A Current rotor bar at 3*slip
115	Broken rotor bars.B -3*s	Phase B Current rotor bar at -3*slip

	KPI	Description
116	Broken rotor bars.B -2*s	Phase B Current rotor bar at -2*slip
117	Broken rotor bars.B -1*s	Phase B Current rotor bar at -1*slip
118	Broken rotor bars.B +1*s	Phase B Current rotor bar at 1*slip
119	Broken rotor bars.B +2*s	Phase B Current rotor bar at 2*slip
120	Broken rotor bars.B +3*s	Phase B Current rotor bar at 3*slip
121	Broken rotor bars.C -3*s	Phase C Current rotor bar at -3*slip
122	Broken rotor bars.C -2*s	Phase C Current rotor bar at -2*slip
123	Broken rotor bars.C -1*s	Phase C Current rotor bar at -1*slip
124	Broken rotor bars.C +1*s	Phase C Current rotor bar at 1*slip
125	Broken rotor bars.C +2*s	Phase C Current rotor bar at 2*slip
126	Broken rotor bars.C +3*s	Phase C Current rotor bar at 3*slip
127	bearing.vib.Ball spin.1	Ball spin frequency Accel-1
128	bearing.vib.Fund Train.1	Fundamental train frequency Accel-1
129	bearing.vib.Inner ring.1	Inner Ring frequency Accel-1
130	bearing.vib.Outer ring.1	Outer Ring frequency Accel-1
131	bearing.vib.Ball defect.1	Ball Defect frequency Accel-1
132	bearing.vib.Ball spin.2	Ball spin frequency Accel-2
133	bearing.vib.Fund Train.2	Fundamental train frequency Accel-2
134	bearing.vib.Inner ring.2	Inner Ring frequency Accel-2
135	bearing.vib.Outer ring.2	Outer Ring frequency Accel-2
136	bearing.vib.Ball defect.2	Ball Defect frequency Accel-2
137	bearing.vib.Ball spin.3	Ball spin frequency Accel-3
138	bearing.vib.Fund Train.3	Fundamental train frequency Accel-3
139	bearing.vib.Inner ring.3	Inner Ring frequency Accel-3
140	bearing.vib.Outer ring.3	Outer Ring frequency Accel-3
141	bearing.vib.Ball defect.3	Ball Defect frequency Accel-3
142	bearing.vib.Ball spin.4	Ball spin frequency Accel-4
143	bearing.vib.Fund Train.4	Fundamental train frequency Accel-4
144	bearing.vib.Inner ring.4	Inner Ring frequency Accel-4
145	bearing.vib.Outer ring.4	Outer Ring frequency Accel-4
146	bearing.vib.Ball defect.4	Ball Defect frequency Accel-4
147	bearing.vib.Ball spin.5	Ball spin frequency Accel-5
148	bearing.vib.Fund Train.5	Fundamental train frequency Accel-5
149	bearing.vib.Inner ring.5	Inner Ring frequency Accel-5
150	bearing.vib.Outer ring.5	Outer Ring frequency Accel-5
151	bearing.vib.Ball defect.5	Ball Defect frequency Accel-5
152	bearing.vib.Ball spin.6	Ball spin frequency Accel-6
153	bearing.vib.Fund Train.6	Fundamental train frequency Accel-6
154	bearing.vib.Inner ring.6	Inner Ring frequency Accel-6

	KPI	Description
155	bearing.vib.Outer ring.6	Outer Ring frequency Accel-6
156	bearing.vib.Ball defect.6	Ball Defect frequency Accel-6
157	Cur.bearing.-Ball spin.A	Phase A Current Lower Ball Spin freq
158	Cur.bearing.-Fund Train.A	Phase A Current Lower Fundamental Train freq
159	Cur.bearing.-Inner ring.A	Phase A Current Lower Inner Ring freq
160	Cur.bearing.-Outer ring.A	Phase A Current Lower Outer Ring freq
161	Cur.bearing.-Ball defect.A	Phase A Current Lower Ball Defect freq
162	Cur.bearing.+Ball spin.A	Phase A Current Upper Ball Spin freq
163	Cur.bearing.+Fund Train.A	Phase A Current Upper Fundamental Train freq
164	Cur.bearing.+Inner ring.A	Phase A Current Upper Inner Ring freq
165	Cur.bearing.+Outer ring.A	Phase A Current Upper Outer Ring freq
166	Cur.bearing.+Ball defect.A	Phase A Current Upper Ball Defect freq
167	Cur.bearing.-Ball spin.B	Phase B Current Lower Ball Spin freq
168	Cur.bearing.-Fund Train.B	Phase B Current Lower Fundamental Train freq
169	Cur.bearing.-Inner ring.B	Phase B Current Lower Inner Ring freq
170	Cur.bearing.-Outer ring.B	Phase B Current Lower Outer Ring freq
171	Cur.bearing.-Ball defect.B	Phase B Current Lower Ball Defect freq
172	Cur.bearing.+Ball spin.B	Phase B Current Upper Ball Spin freq
173	Cur.bearing.+Fund Train.B	Phase B Current Upper Fundamental Train freq
174	Cur.bearing.+Inner ring.B	Phase B Current Upper Inner Ring freq
175	Cur.bearing.+Outer ring.B	Phase B Current Upper Outer Ring freq
176	Cur.bearing.+Ball defect.B	Phase B Current Upper Ball Defect freq
177	Cur.bearing.-Ball spin.C	Phase C Current Lower Ball Spin freq
178	Cur.bearing.-Fund Train.C	Phase C Current Lower Fundamental Train freq
179	Cur.bearing.-Inner ring.C	Phase C Current Lower Inner Ring freq
180	Cur.bearing.-Outer ring.C	Phase C Current Lower Outer Ring freq
181	Cur.bearing.-Ball defect.C	Phase C Current Lower Ball Defect freq
182	Cur.bearing.+Ball spin.C	Phase C Current Upper Ball Spin freq
183	Cur.bearing.+Fund Train.C	Phase C Current Upper Fundamental Train freq
184	Cur.bearing.+Inner ring.C	Phase C Current Upper Inner Ring freq
185	Cur.bearing.+Outer ring.C	Phase C Current Upper Outer Ring freq
186	Cur.bearing.+Ball defect.C	Phase C Current Upper Ball Defect freq
187	Time	Testing Time in Seconds

APPENDIX D: HEALTH MONITORING ALGORITHM

KPI Classifier Code

```
%’filename’ is used as generic filename in this example of
%the KPI classifier
data = filename(:,:);
[Ureduced ,psi,lambda] = defineEigenSpace2(data ’);
[E_filename , E2_filename] = KPI2E(data28, Ureduced , psi, lambda);
```

Function Used to Calculate Error from KPIs

```
function[E,E2] = KPI2E(data, Ureduced , psi, lambda)
    omegaMatrix = eigenParams2(data ’,0,Ureduced ,psi,lambda);
    data_hat = bsxfun(@plus ,omegaMatrix*Ureduced ’,psi ’);
    e = data-data_hat;
    E = zeros(size(e,1),1);
    E2 = zeros(size(e,1) ,1);
    for ii = 1:size(e,1)
        vec = e(ii ,:);
        E(ii) = norm(vec);
        E2(ii) = norm(vec)/norm(data(ii,:));
    End
End
```

Function Loads or Calculates Eigenspace Basis Vectors, Ureduced, and the Mean Vector

```
% This fuction loads or calculates the eigenspace basis
% vectors, Ureduced , and also the mean vector, psi. Psi
% must be subtracted from evt transients before being
% projected into the eigenspace defined by Ureduced.
%
% If no inputs are given the function will lookfor and
% load Ureduced and psi from a EIGEN_SPACE.mat file in
% the current directory.
%
% [Ureduced,psi,lambda] = defineEigenSpace();
%
% If inputs are given , Ureduced and psi will be
% calculated and then saved to EIGEN_SPACE.mat in the
% current directory. This will write over any existing
% EIGEN_SPACE_2.mat file.
%
%* USE ’eigenParams.m’ for projecting evts into eigenspace.*
%
% function [Ureduced,psi,lambda] = defineEigenSpace2(gammaMatrix)
%
% Inputs:
% gammaMarrix , a the complete set of test sample features
% in a DxM matrix. Where D is the dimensions of the
% feature vectors, and M is the number of sample vectors.
%
% D = number of pts in vector
% M = number of training Vectors
% Ufull , Dx(D or M) matrix of eigenvectors , min(D,M)
% lambda, (D or M)x1 vector of corresponding eigenvalues , min(D,M)
% psi, a Dx1 image mean vector
%
% Nt = number of test vectors
% K = reduced dimensions from D (K<D) % PhiMatrix , DxM zero-mean traing vectors
% Ureduced, DxK matrix of eigenvectors
% omegaMatrix is MxK matrix, K reduced dimensions ,
% projected training vectors => the features for
% each training vector
function [Ureduced ,psi,lambda] = defineEigenSpace2(gammaMatrix)

if ~exist(’gammaMatrix’,’var’)
disp(’Loaded current EIGEN_SPACE_2.mat’)
load(’EIGEN_SPACE_2.mat’); % this loads Ureduced and psi
return
```

```
 end
% Setting Full Space
[Ufull , lambda, psi] = computeFullEigenSpace(gammaMatrix);
PhiMatrix = bsxfun(@minus,gammaMatrix ,psi); % zero mean image g →vectors
[omegaMatrix , Ureduced] = reduceEigenSpace(Ufull ,lambda,PhiMatrix);
%%% note: omegaMatrix = PhiMatrix1 '*Ureduced;
% Saving
save('EIGEN_SPACE_2.mat','Ureduced','psi','lambda');
disp('Saved new EIGEN_SPACE_2.mat')
end
function [Ufull , lambda, psi] = computeFullEigenSpace(gammaMatrix)
% Computes the Eigenspace vectors/values, and the mean
% input vector for a set of M training vectors with D
% dimensions.
%
% [Ufull , lambda, psi] = computeFullEigenSpace(gammaMatrix)
%
% Input ,
% gammaMarrix , a the complete set of test images in a
% DxM matrix.
% Where D is the dimensions of the image vectors, and
% M is the number of image vectors.
%
% Output,
% Ufull , a DxM matrix of eigenvectors of the full eigenspace ( ||g →Ufull(:,i)|| = 1 ).
% lambda, a Mx1 vector of eigenvalues (in descending order).
% psi, a Dx1 image mean vector.
[D,M]=size(gammaMatrix);
% average the image vectors (2nd dim)
psi = (1/M)*sum(gammaMatrix ,2);
% zero mean image vectors
PhiMatrix = bsxfun(@minus,gammaMatrix ,psi);
if D>M
C = (1/M)*PhiMatrix '*PhiMatrix; % if D>M
else
C = (1/M)*PhiMatrix*PhiMatrix '; % do it this way for M>D
end
[vec,d]=eig(C);
[lambda,index] = sort(diag(d),'descend');
% sorting eigenvectors to correspond with sorted eigenvalues
U= vec(:,index);
% if eigenvalue is <= 0 set corresponding eignevector to zero(g →numerical instability)
U(:,lambda <=0)=0;
if D>M
```

```
%if D>M also makes ||Ufull(:,i)|| = 1
Ufull = bsxfun(@times ,((1./(M*lambda)).^0.5) ',PhiMatrix*U);
else
% do it this way for M>D
Ufull = U;
end
end
function [omegaMatrix , Ureduced] = reduceEigenSpace(Ufull ,lambda,g  →PhiMatrix)
% Reduces the Eigenspace dimensions from M to K (K<M<D);
% eigenvectors are D dimensions.
%
% [omegaMatrix , Ureduced] = reduceEigenSpace(Ufull ,lambda,PhiMatrixg  →)
% Input ,
% Ufull , DxM matrix of eigenvectors for the full eigenspace.
% lambda, Mx1 vector of corresponding eigenvalues.
% PhiMatrix , DxM zero-mean traing vectors, M vectors of D dimensiong  →.
%
% Output,
% omegaMatrix , MxK matrix of reduced eigenspace training
% vectors (K coordinates for each training vector M).
% Ureduced, DxK matrix of eigenvectors associated with
% the K dimensional eigenspace.

[D,M] = size(Ufull); avgVal = (1./M).*sum(lambda); Ureduced =
          Ufull(:,lambda>avgVal)
% must have at least 2 dimensions.
if size(Ureduced ,2)<2
Ureduced = Ufull(:,1:2);
end
% projecting each training vector into reduced eigenspace.
omegaMatrix = PhiMatrix '*Ureduced;
end
```

Function Projects Measurements Into Eigenspace Defined by Ureduced and the Mean Vector

```
% This function projects measurements into an eigenspace
% defined by Ureduced and the mean vector psi. Psi is
% subtracted from evt transients to make them zero-mean
% before being projected into Ureduced.
%
%* USE ’defineEigenSpace.m’ to define Ureduced and psi *
%
% If Ureduced and psi not inputted, will load them from
% EIGENSPACE.mat
%
% function omegaMatrix = eigenParams(evtlist,white ,Ureduced ,psi,g  →lambda)
%
% white = 1/0, 1 if you want the eigenvector to undergo
% a whitening transform , this makes each of the
% cordinates to have similar values. Otherwise the
% omegaMatrix would be heavily weighted to the first
% few eigenvectors. If white is not given will be set % to 0.
%
% D = number of pts in vector
% M = number of training Vectors
% Ufull ,Dx(D or M) matrix of eigenvectors , min(D,M)
% lambda,(D or M)x1 vector of corresponding eigenvalues , min(D,M)
% psi, a Dx1 image mean vector
%
% Nt = number of test vectors
% K = reduced dimensions from D (K<D)
% PhiMatrix , DxM zero-mean traing vectors
% Ureduced, DxK matrix of eigenvectors
% omegaMatrix is MxK matrix, K reduced dimensions ,
% projected training vectors => the features
%for each training vector

function omegaMatrix = eigenParams2(gammaMatrix ,white ,Ureduced ,psi,g  →lambda)
if ~exist(’Ureduced’,’var’)
load(’EIGEN_SPACE_2.mat’);
% this loads Ureduced , psi, and lambda end
if ~exist(’white’,’var’)
white = 0;
end
% zero mean image vectors
PhiMatrix = bsxfun(@minus,gammaMatrix ,psi);
omegaMatrix = PhiMatrix ’*Ureduced;
```

```
if white == 1
unweight_lambda = lambda(1:size(omegaMatrix ,2)).^-0.5;
omegaMatrix = bsxfun(@times,omegaMatrix ,unweight_lambda ’);
end
end
```