

KATHMANDU UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

PROJECT REPORT



**REFURBISHMENT OF AN INDUCTION MOTOR WITH
CONDITION MONITORING SYSTEM**

A **third year project report** submitted in partial fulfilment
of the requirements for the degree of Bachelor of
Engineering

By:

Saugat Karki (32040)
Bipin Nepal (32045)
Samyam Shrestha (32056)

June 2024

CERTIFICATION

THIRD YEAR PROJECT REPORT ON

REFURBISHMENT OF AN INDUCTION MOTOR WITH

CONDITION MONITORING SYSTEM

By:

Saugat Karki (32040)

Bipin Nepal (32045)

Samyam Shrestha (32056)

Approved by:

1. Project Supervisor

(Signature)

(Name)

(Date)

2. Head of the Department

(Signature)

(Name)

(Date)

ABSTRACT

This project introduces an innovative solution to tackle three-phase induction motor winding issues through two strategic approaches. Firstly, we present simulation-based models for the precise design of induction motors, providing a solid foundation for accurate winding repairs. This ensures innovative motor refurbishment, enhancing efficiency, power density, and power factor. Secondly, our proposal integrates health monitoring during the rewinding process, incorporating sensors at critical locations like windings and the core. Real-time temperature and vibration monitoring enable proactive maintenance, preventing breakdowns and promoting overall motor health. Particularly relevant in Nepal, where traditional motor maintenance practices lack condition monitoring, this holistic model revolutionizes motor design, refurbishment, and maintenance, offering a transformative solution for improved reliability in the industrial landscape.

ACKNOWLEDGMENT

The team would like to express sincere gratitude to the Department of Electrical and Electronics Engineering of Kathmandu University for granting permission and supporting the project. The team would like to thank the project supervisor Dr. Bishal Silwal for proper guidance to start through basics and carry on the project. The team is also grateful to our project co-ordinator Mr. Pramish Shrestha and Mr. Uttam Adhikari for providing necessary assistance and motivation during the project.

Furthermore, we convey regards to all the teachers, non-teaching staff and seniors of the Department of Electrical and Electronics who helped the team directly or indirectly throughout the project.

LIST OF FIGURES

Figure 1: Bearing fault left) and burnt winding due to the stator winding fault (right)	1
Figure 2: Block Diagram of the Project	3
Figure 3: Three SCT-013 Current Sensor clamped to three-phase wire	10
Figure 4: Project Methodology	11
Figure 5: Block Diagram of Alert System.....	13
Figure 6: Calculation of Motor Parameters on Excel.....	15
Figure 7:Calculation of No. of turns per phase.....	15
Figure 8: Calculaiton in Excel	16
Figure 9: Excel Table of Suitable Winding Selection Analysis	17
Figure 10: Entering the motor value as input.	18
Figure 11: Selection of stator slot	18
Figure 12: Stator Slot Design.....	19
Figure 13: Rotor Design	19
Figure 14: Machine with stator and core	20
Figure 15: Stator Slot Shape Dimensions.....	20
Figure 16: Stator Parameters.....	20
Figure 17: User Defined Rotor Slot Design	21
Figure 18: Connection of Coils in SIngle Layer Winding.....	21
Figure 19: Snippet of ANSYS RmXprt showing rated and no load opearion parameters .	22
Figure 20: Ansys RmXprt Stator Winding value.....	23
Figure 21: Plot of Torque and Efficiency respectively with speed.....	23
Figure 22: Plot of Efficiency VS Speed	23
Figure 23: Winding Development Diagram	25
Figure 24: General outline of winding connectins.....	25
Figure 25: First step of Rewinding.....	27
Figure 26: WInder Installation.....	28
Figure 27: Completion of rewinding and installation of fan	29
Figure 28: Painting of the motor	30
Figure 29: Completion of painting of motor.....	31
Figure 30: PT-100 Temperature Sensor	32
Figure 31: PT-100 Temperature Sensor with ESP-32.....	33
Figure 32: Sensor Clibration of PT-100 temperature sensor.....	33
Figure 33: Curve Fitting of temperature data.	34
Figure 34: SCT sensor connection diagram with ESP-32	35
Figure 35: SCT current transformer sensor calibration.....	36
<i>Figure 36: FFT of Current Value measured form the sensor</i>	37
Figure 37: Original Data of Current with respect to time in millisecond.....	37
Figure 38: Three-phase current waveform obtained form the current sensor (live)	38
Figure 39: Display of current value of phase 1 and phase 2.....	38
Figure 40: Current value of third phase.....	39
Figure 41: Display of temperature values.....	39
Figure 42: Live temperature value display of three different phases.....	40
Figure 43: Table obtained from the power analyser while running the motor on no load condition	41
Figure 44: RPM calculation of the motor.....	42
Figure 45: Three-phase supply and power analyser connection to the motor	42
Figure 46: Final Connection	42

SYMBOLS

PTC	Positive Temperature Coefficient
RRT	Rated Response Temperature
RTOS	Real Time Operating System
FFT	Fast Fourier Transform
SSE	Sum of Squared Error
BLDC	Brushless Direct Current
SCT	Split Core Type
RPM	Revolution Per Minute
IoT	Internet of Things
RMSE	Root Mean Squared Error

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENT	iv
LIST OF FIGURES	v
SYMBOLS	vi
CHAPTER I: INTRODUCTION	1
1.1 Background	1
1.2 Motivation	1
1.3 Problem Description	2
1.4 Objectives	2
1.5 Methodology	2
1.6 Organization of report	3
CHAPTER II: LITERATURE REVIEW	4
2.1 Literature Survey	4
2.2 Tools Used	7
2.2.1 Ms-Excel	7
2.2.2 Matlab	7
2.2.3 ANSYS ELECTRONICS DESKTOP	7
2.2.3 ESP-32 WiFi Module	8
2.2.4 ThingSpeak Platform	9
2.2.5 PT-100 Temperature Sensor	9
2.2.6 SCT-013 Current Transformer	10
CHAPTER III: METHODOLOGY OF THE PROJECT	11
3.1 System Block Diagram and Description	11
3.2 Development of Simulation Models	11
3.3 Rewinding the motor and sensor installation	12
CHAPTER IV: RESULT AND CONCLUSION	15
4.1 Work Accomplished	15
4.1.1 Calculation of Induction Motor Parameters on MS-Excel	15
4.1.2 Selection of Winding Type	17
4.1.3 Calculations in Matlab Software	18
4.1.4 Designing in Ansys Electronics Desktop Software.	18
4.1.4 Rewinding and Sensor Installation	24
4.1.5 Rewinding of Three Phase Motor in Steps	27
4.1.6 Sensor Data Collection	32
4.1.6.1 PT-100 Temperature Sensor	32
4.1.6.2 Calibration of PT-100 Sensor	33

4.1.6.3 SCT-013-060 (50A) Non-invasive AC Sensor Split Core Current Transformer	35
4.1.6.4 Sensor Calibration of SCT -013	36
4.1.7 Display on ThingSpeak Online Platform.	38
4.1.7.1 Current Readings	38
4.1.7.2 Temperature Readings	39
4.1.7.3 Some Pictures of Setup and Results	40
CHAPTER V: CONCLUSION	43
5.1 Conclusion	43
5.2 Problems Encountered and Limitations	43
5.3 Mitigation	43
GANTT CHART	44
REFERENCES	45
APPENDIX	46
[A1]. Matlab code for FFT of current signal	46
[A2]. ANSYS Three Machine Design Parameters	47
[A3]. Condition Monitoring Using ThinkSpeak Code	50
[A4]. SCT -013 Current Sensor with ESP-32 Code	53

CHAPTER I: INTRODUCTION

1.1 Background

An induction motor is a self - starting motor. They do not require any external means to initiate rotation; once power is supplied, the rotating magnetic field created by the stator induces currents in the rotor, causing it to start rotating. Almost in every industry, around 90% of the machines apply an induction motor as a prime mover.[1] Preferably three-phase induction motor are used in industries. It can be said that induction motors are the workhorse of the industry. Robust in construction, high starting torque, efficiency and reliability makes the induction motor favorable in industries. The rating of induction motors varies from few watts to tens of Mega-watts that suits to various industrial applications.

Electric motors, especially induction motors are widely used in manufacturing, consume over 45% of global electrical energy. However, these motors often face breakdowns due to factors like dynamic loads and environmental conditions, leading to both electrical and mechanical faults. According to the IEEE, 42% of induction motor faults are related to bearings, 28% to stator windings, and less than 10% to rotor issues. Other faults contribute to 20%. Fig.1 illustrates damage caused by bearing and stator winding faults. When a motor breaks down, industries typically aim for a speedy repair turnaround from workshops to minimize downtime. Addressing these common faults is crucial for improving motor reliability and efficiency in manufacturing processes.



Figure 1: Bearing fault (left) and burnt winding due to the stator winding fault (right)

1.2 Motivation

We have seen in industries and factories that induction motors get damaged due to the faults in stators, bearings, rotors, etc. Our project is focused on the condition monitoring using temperature, current and vibration sensors so that they detect the problem before occurring as we can analyze it from the data and this helps to reduce the downtime period.

1.3 Problem Description

In Nepal, numerous industries grapple with the recurring issue of induction motor malfunctions. The prevailing challenge lies in the fact that technicians may not adhere to established standards when repairing these motors. Consequently, the rewinding performed may fail to withstand specific temperature limits, leading to motor degradation and repeated issues. This underscores the limitations of traditional rewinding methods. Our project aims to address this issue by developing a standardized approach to repairing and maintaining induction motors. This approach involves continuous monitoring using sensors and the analysis of motor parameters such as temperature, current and vibration. Data collected from these sensors can be then employed in predictive analysis to detect early faults or anomalies. By proactively monitoring the motor's condition, we can prevent damages and potentially catastrophic consequences, such as fires or explosions. Additionally, this approach offers substantial economic benefits and enhances productivity in various industries.

1.4 Objectives

The main objectives of the project are:

- i. To create a simulation-based model for designing and refurbishing induction motors, focusing on improving motor performance (efficiency or pf or power density) through rewinding.
- ii. To perform the rewinding of the induction motor based on parameters and data calculated during the modeling phase.
- iii. To implement continuous real-time monitoring of an induction motor to assess its health condition.

1.5 Methodology

A concise block diagram of the project methodology is shown in Figure 2.

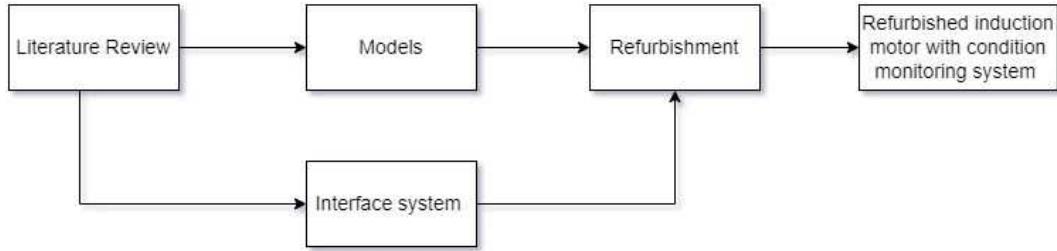


Figure 2: Block Diagram of the Project

At first literature review is done about the induction monitor condition monitoring and refurbishment techniques. Then, the modelling of the machine is done on the simulation software after the calculation of various motor design parameters. Alongside with it, interface system is also built to view the data obtained from the sensors attached to the motor for its monitoring. The rewinding of the motor is done with the analysis form the simulation model developed.

1.6 Organization of report

This project work is organized in the following order.

Chapter one: Chapter one is an introduction to the research/project, problem description and project objectives.

Chapter two: This section consists of literature review of the project.

Chapter three: This chapter deals with the methodology and design of the system. The most important aspect of this chapter is the block diagram of the system. The mathematical analysis on Matlab and the modelling of the motor in Ansys Software is done here.

Chapter four: This section consists of work accomplished and work remaining of the project.

Chapter five: This section consists the conclusion and problems encountered along with possible mitigation measures.

CHAPTER II: LITERATURE REVIEW

2.1 Literature Survey

Hadi Ashraf Raja et.al provides non linear model for fault diagnosis of BLDC motor based on vibration frequency spectrum[2]. They have presented the paper based on the approach on the data acquisition system method using microcontroller boards which are pushed into the cloud and can be monitored in real time. They considered a current signal and made the diagnosis in frequency domain. To detect the faults, they developed a machine learning algorithm. According to their research, among four machine learning algorithms, “Super Vector Control” algorithm gave 97% accuracy.

The literature on proper repair or refurbishment procedures and strategies for electrical machines is limited. Initial discussions on this topic can be traced back to the 1980s. A notable contribution by Armintor et al. (1981) addresses the repair and restoration of large induction motors with a focus on turnaround time [3]. Their paper outlines a procedure for estimating the time and cost associated with repairs and delineates the general steps involved in inspecting and repairing motors. Additionally, it identifies several critical factors that influence decision-making during the repair process.

In the paper by H.A Raja et.al , they have explained about a method to connect electrical machines to a cloud system and detect fault using cloud computing in a communicative IoT system [4]. They have developed a cloud based communication system to transfer the readings from sensors to a frontend in real time. Also they have used trained neural network methods to detect the machine faults in the cloud. The training of models is applied to help switch to predictive maintenance from scheduled maintenance that helps to monitor the machines remotely. Since, real time sensor data is required at high speed, instead of Arduino, they have used Teensy board. Here, they have transferred the converted data to Raspberry Pi through Serial Peripheral Interface (SPI) for high data transfer rate.

Further contributions to this area include the work of Engyong (2013), who explores the refurbishment of an induction motor for scenarios where the input voltage is lower than the motor's nominal voltage [5]. Engyong's study includes general calculations of motor parameters adjusted for the reduced voltage level. The findings suggest that the motor can

still achieve its rated speed after refurbishment by altering the winding parameters. The author also reports validating these theoretical calculations through practical testing.

These early works lay the foundation for understanding the complexities involved in motor repair and refurbishment, highlighting essential considerations such as time, cost, and technical adjustments required to maintain motor performance under varying operational conditions.

It is generally assumed that a rewound motor is not as efficient as the original motor. This is, on one hand, due to the aforementioned emphasis on the turnaround time, and on the other hand, due to the inaccurate estimation methods after refurbishment. Al-Badri et al (2014) proposes an algorithm to estimate the efficiency of the refurbished motor by the use of no-load tests [6]. The author claims that the proposed procedure can be used in any electrical motor repair workshop and the efficiency can be calculated by over 99% accuracy.

The research work of Anouar Belaheen and H.A. Raja focuses on condition monitoring techniques in wind turbines using an IoT system [7]. The system collects data from various sensors including vibration, wind speed, temperature and voltage at high frequency. This data is then transferred to the cloud for further analysis. The goal is to use machine learning and pattern recognition algorithms to predict faults before they occur, allowing for preventive maintenance and reducing downtime for wind turbines. This approach is more efficient than traditional scheduled maintenance, which can be time consuming and expensive, especially for offshore turbines.

Their research implements an IoT system for condition monitoring in wind turbines. Sensors attached to the turbine capture vital data on vibration, wind speed, temperature, and voltage. A Teensy card converts these analog signals from the sensors into a digital format for processing by a Raspberry Pi. This Raspberry Pi acts as a hub, collecting data from multiple sensors and storing a local backup before transmitting everything to cloud storage. This setup offers a cost-effective and compact data acquisition system compared to traditional methods. Notably, their system can collect data at a high frequency of around 3400 samples per second, allowing for the capture of even minor fluctuations in measurements. This high-resolution data collection is crucial for the later stages of the research, where machine

learning and pattern recognition algorithms will be applied to predict faults before they occur.

Tavakoli, Mohammadreza, et al discusses the application of motor current signature analysis (MCSA) for condition monitoring and fault diagnosis of induction motors. MCSA involves analyzing the stator current to detect faults such as broken rotor bars, eccentricity, and other abnormalities [8].

Mohanraj, Thillaigovindan, and Subbiah provides an overview of vibration analysis techniques employed in condition monitoring of induction motors. It covers methodologies, signal processing techniques, and case studies on detecting faults like bearing defects and rotor faults [9].

Patel, Mital A., et al presents a method for online condition monitoring of induction motors using current signature analysis (CSA) combined with artificial neural networks (ANNs). CSA is employed to detect faults, while ANNs are used for fault classification and prediction [10].

Petchakup, Feradje, et al explores various techniques under electrical signature analysis (ESA) for condition monitoring of induction motors. It covers methodologies such as current, voltage, and power factor analysis, highlighting advancements and challenges [11].

In this project we plan to solve the aforementioned problem in two folds. First, by developing simulation based models for the design of induction motors, which will help to repair the motor windings based on the technically validated model. Second, by enabling health monitoring of the motor by installing multiple sensors during rewinding of the motor, which will help to prevent winding breakdown. The model will be a very generic model. It will have motor dimensions and some of the specifications as the input parameters. The desired performance of the motor after refurbishment, for example, the efficiency or the power factor or the power density, etc. can be calculated from the model. The model can be used for all kinds of cage induction motors. The enabling of the condition monitoring system with the help of temperature, current and vibration sensors will aid to limit the future breakdowns, hence making the refurbishment robust. Unlike the usual methods used in

repair shops and local workshops, our proposed model for motor refurbishment aims to improve key performance indicators like efficiency, power density, and power factor. The use of multiple sensors to monitor the motor's health in real-time is innovative for Nepal, where most industries do not currently use condition monitoring for motor maintenance.

2.2 Tools Used

2.2.1 Ms-Excel

In Excel, all the motors parameters are calculated. Also, the selection for the suitable winding configuration is done through analysis based on parameters like winding type, number of parallel branches, efficiency, loss, etc. Excel is a reliable and suitable tool for calculating the motor parameters value and analyse them effectively.

2.2.2 Matlab

MATLAB, is a high-performance programming language and environment primarily used for numerical computing, data analysis, and visualization. Developed by MathWorks, MATLAB provides a comprehensive set of tools and functions for various mathematical tasks, algorithm development, simulation, and modeling. It is widely used in engineering, science, and academia for tasks such as signal processing, image processing, control system design, machine learning, and more. MATLAB's syntax is designed to be intuitive and easy to use, making it accessible for engineers, scientists, and researchers who may not have extensive programming backgrounds. The platform also includes a graphical user interface (GUI) for certain tasks and supports the creation of custom user interfaces. Additionally, MATLAB allows integration with other programming languages and tools, making it a versatile tool for a wide range of applications. In our project, we are using MATLAB as a programming language to calibrate the temperature and current sensor by linear curve fitting.

2.2.3 ANSYS ELECTRONICS DESKTOP

ANSYS Electronics Desktop is a software suite within the ANSYS simulation platform that focuses on electromagnetic field simulation and analysis. It is designed for engineers and researchers working on the design and analysis of electronic components and systems. In this project, we are using ANSYS to design the model of the induction motor.

2.2.3 ESP-32 WiFi Module

The ESP32 is a chip designed to provide Wi-Fi and, in some models, Bluetooth connectivity for embedded devices, commonly referred to as IoT devices[12]. While the ESP32 chip itself is a standalone component, the term "ESP32" is often used to describe the modules and development boards that incorporate this chip. The original ESP32 chip featured a single-core Tensilica Xtensa LX6 microprocessor with a clock rate exceeding 240 MHz, offering high data processing speeds. Typically, the ESP32 uses a dual-core Tensilica Xtensa 32-bit LX6 microprocessor, except for the ESP32-S0WD model, which utilizes a single-core system. It operates at a clock frequency of up to 240MHz and can achieve performance levels of up to 600 DMIPS. The chip's low power consumption allows for efficient analog-to-digital conversions and computations even in deep sleep mode.

The ESP32 provides robust wireless connectivity with integrated Wi-Fi supporting 802.11 b/g/n/e/i standards and Bluetooth v4.2 BR/EDR, including Bluetooth Low Energy (BLE). Its internal memory includes 448 KB ROM for booting and core functions, 520 KB SRAM for data and instructions, 8 KB RTC fast SRAM for boot data storage, and 8 KB RTC slow SRAM for co-processor access during sleep. Additionally, it features 1 KiBit eFuse for system configuration and customer applications, with some variants like the ESP32-D2WD and ESP32-PICO-D4 having internal flash memory. The ESP32 also supports up to four 16 MB external QSPI flashes and SRAMs, with hardware encryption based on AES for secure data protection. Its security features adhere to all IEEE 802.11 standards, including WFA, WPA/WPA2, and WAPI, and it includes secure boot and flash encryption capabilities.

The ESP32's versatile functions make it ideal for a wide range of IoT applications. It facilitates networking through its Wi-Fi antenna and dual-core processor, enabling embedded devices to connect to routers and transmit data. It processes inputs from analog and digital sensors and performs complex calculations using an RTOS or non-OS software development kit (SDK). Additionally, the ESP32 supports peer-to-peer (P2P) connectivity, allowing direct communication between different ESPs and other devices, and can function as a web server, providing access to HTML pages and other development languages. Common applications of ESP32 modules include smart industrial devices like programmable logic controllers (PLCs), smart medical devices such as wearable health

monitors, smart energy devices including HVAC systems and thermostats, and smart security devices like surveillance cameras and smart locks.

2.2.4 ThingSpeak Platform

To display the sensor live readings, we have used thinkspeak platform. ThingSpeak is an IoT cloud platform operated by MathWorks that allows users to send sensor data to the cloud, analyze, and visualize it using MATLAB or other software [13]. To use ThingSpeak, users need a MathWorks Account. The platform is free for small non-commercial projects and includes a Web Service (REST API) for collecting and storing sensor data, enabling the development of IoT applications. It supports Arduino, Raspberry Pi, and MATLAB with premade libraries and APIs but can work with any programming language via its REST API and HTTP.

2.2.5 PT-100 Temperature Sensor

A PT100 sensor is a type of platinum resistance thermometer commonly used to measure temperature. "Pt" denotes that the sensor is made from platinum, and "100" indicates that the sensor has a resistance of 100 ohms (Ω) at 0°C. These sensors are often referred to as RTDs (resistance temperature detectors) and are known for their accuracy and stability. The PT100 sensor operates on the principle that the electrical resistance of platinum changes predictably with temperature variations; as the temperature increases, the resistance of the platinum element also increases. This change in resistance is measured and converted into temperature readings using specific algorithms or conversion tables.

Calibrating a PT100 sensor involves comparing its output against a known standard across a range of temperatures. This process typically requires specialized equipment, such as temperature baths or ovens, to create precise temperature conditions. The sensor's readings are compared against reference standards, and adjustments are made if necessary to ensure accurate temperature measurement across the desired range. This calibration ensures that the PT100 sensor provides reliable and accurate temperature readings in various applications.

2.2.6 SCT-013 Current Transformer

This sensor is non invasive. We have used three split-core-type (SCT) current transformer to measure the current of three-phase induction motor. Three current sensors are clamped to the three-phases of the induction motor to measure the phase current. The current sensors can be seen in the figure below.



Figure 3: Three SCT-013 Current Sensor clamped to three-phase wire

CHAPTER III: METHODOLOGY OF THE PROJECT

3.1 System Block Diagram and Description

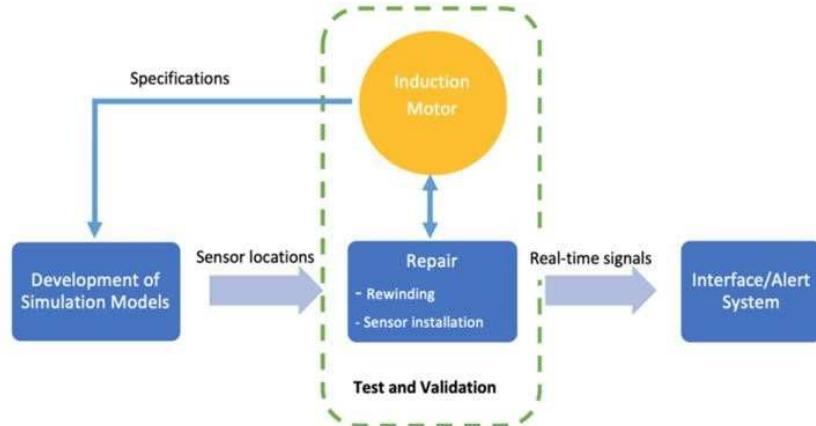


Figure 4: Project Methodology

At first, literature review for the condition monitoring and refurbishment of the induction motor is done. Then, with the help of machine's nameplate, we collect its specification and helpful data needed for calculations. The calculations for the motor parameters will be done. Then, we develop the simulation models using those data. The next step is the repairment of the motor which includes rewinding of the motor. Here, the refurbishment process is followed after the design of the motor model. Appropriate sensors are selected and placed on the motor to get the data for monitoring.

The project methodologies are shown in the block diagram in the above figure. Each block is explained in details in the following paragraph.

3.2 Development of Simulation Models

To initiate the model creation, we begin by inputting essential details extracted from the machine's nameplate and the major geometrical dimensions. These critical parameters from the nameplate encompass the rated speed, output power, load type, operating frequency, and the winding connection type. The major geometrical dimensions include the outer and inner diameters, as well as number of slots and number of poles for the machine. Winding configuration, wire size, and coil pitch are the main parameters providing a tailored

approach to meet specific requirements and optimize performance parameters. With these parameters defined, the Ansys software undertakes the complex task of processing the data to generate a comprehensive model. The resulting simulation offers a wealth of insights into the motor's behavior, presenting a multifaceted analysis of key performance indicators. These indicators include but are not limited to efficiency, torque characteristics, and speed characteristics.

Graphical analysis becomes an invaluable tool in our exploration of the simulation results. Through the examination of various graphs, we seek to identify patterns and trends that lead us to the optimal combination of parameters for maximum output and efficiency. This iterative process allows us to fine-tune the model, ensuring that it aligns with the desired performance goals. In addition to the electrical performance analysis, our methodology extends to the exploration of the magnetic fields generated by the motor. Leveraging Maxwell 2D, we delve into the visualization of crucial parameters such as flux density and air gap flux density. This deeper exploration contributes to a more profound understanding of the motor's magnetic behavior, enriching our overall comprehension of its operational dynamics.

In essence, our research methodology goes beyond the mere construction of a simulation model; it constitutes a systematic and holistic approach to uncovering the intricacies of electric motor design, allowing us to make informed decisions in the pursuit of enhanced efficiency and performance. Through this methodological framework, we aim to contribute to the broader discourse on electric motor optimization and advance the state of the art in this vital field of study.

3.3 Rewinding the motor and sensor installation

This section mainly comprises two distinct parts; rewinding and sensor installation. The former begins with the dismantling of the induction motor. Initially, the enclosure is opened, the bearings are separated from the axis and the old stator winding is removed and the induction motor rests at its barest form. The enclosure is treated with methods like gentle scouring to remove any rust or unwanted stains. The stator coils are prepared according to the winding data calculated in the modeling phase. This step includes choosing the right thickness of the wire and grade of the insulation material of the wire in order to withstand the temperature range while the motor is in operation. Likewise, the stator slots

are isolated with lamination material usually paper and the stator coils are placed in the slots. 5 positive temperature coefficient thermistors are also placed in different parts of the stator coils during the winding phase which are responsible for collection of temperature data. The coils are connected in accordance to the winding diagram and each end of the coil wire is fed into the conduit box for extra isolation. The motor is then fully reassembled and a piezo vibration sensor is attached on the enclosure. Piezo vibration sensors are essentially accelerometers which are quite impervious to moisture, dust and smoke which makes them ideal for industrial application [14]. The installation of sensors is done keeping in mind that it does not affect the mechanical balance of the motor. All these steps are then documented in an illustrious manner in a handbook which can be accessed easily by technicians or any interested party.

3.3.1 INTERFACE/ALERT SYSTEM

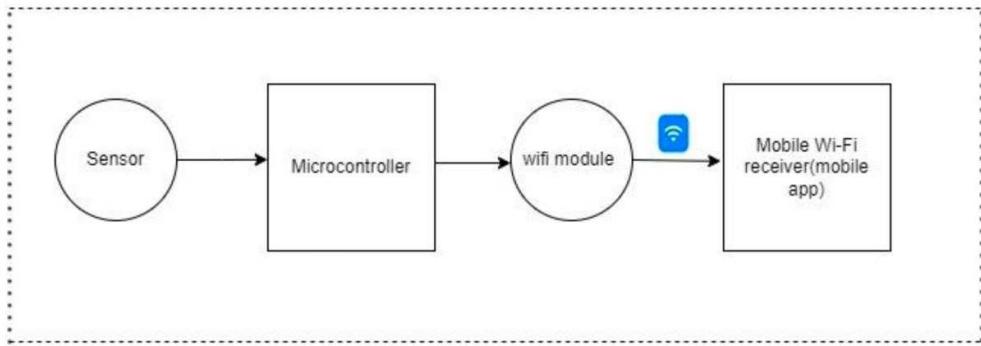


Figure 5: Block Diagram of Alert System

In our system, we rely on a specialized sensor to gather crucial information from various parts of the induction motor, such as vibration, and temperature. This data is then directed to a microcontroller (ESP-32), essentially the brain of our setup, chosen for its robustness in handling vibrations common in industrial settings. A key component is the microprocessor, which gathers sensor data, processes that data, and sends it to a Wi-Fi module. This module transfers the data to another device with a Wi-Fi receiver. Through an application for mobile devices, we can monitor the motor's actions and status from this end. Ensuring that the collected data is comprehensible becomes essential. We create an API server. This server, accessible through our network, efficiently organizes the data. Consequently, when we open the mobile app, we are presented with a user-friendly display, incorporating graphs and figures for easy interpretation.

In the context of an induction motor, specific patterns emerge during normal operation. These include a consistent vibration, current, and temperature when the motor spins at a certain speed. Any deviation from these established patterns may indicate a problem, triggering our alert system. In our alert system with ESP-32, it's like a smart switch. The ESP-32 only activates what is required while sensors are activated. When everything is quiet, it saves energy and swiftly recharges when an important alert is received. It's a power management that works well, guaranteeing efficiency without wasting energy. The alert system acts as the motor's monitoring guardian. It quickly notifies us in case of an abrupt change. This notice alerts us to potential problems with the motor. This notice alerts us to the need to investigate and inspect the motor when it shows up on our mobile app. Using this method enables us to recognize possible problems early on and take appropriate action. We can determine that there may be an issue with the motor if the data fluctuates from the normal range. The alert system gives us the ability to respond quickly, stepping in to resolve a little problem before it becomes a bigger one.

CHAPTER IV: RESULT AND CONCLUSION

4.1 Work Accomplished

4.1.1 Calculation of Induction Motor Parameters on MS-Excel

At first, we calculated the basic motor parameters of the induction motor on MS Excel. Here, the parameters like no. of poles, frequency was entered as user input and the calculation was done based on the formulas created. The calculated parameters are shown in the figure below.

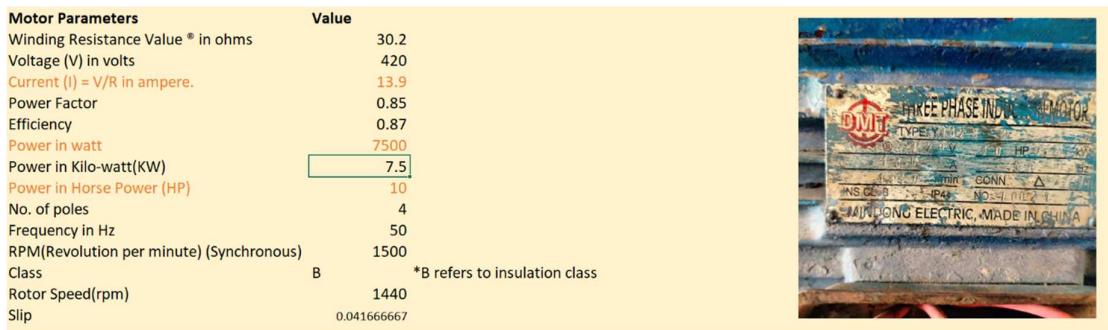


Figure 6: Calculation of Motor Parameters on Excel

Calculation of No of turns per phase	
Eph=	420
kw=	0.954
Bav=	0.52
D=	0.136 m
L=	0.147 m
Pi=	3.14159
P(pole)=	4
flux=	0.008164867 wb
f=	50 Hz
Nph=	485.7679255
Z	80.96132092

$Nph = Eph / 4.44 \times f \times flux \times kw$
Eph = emf per phase (Delta Connection)
frequency = 50Hz
kw (window space factor) = 0.954
flux = Bav X A
Bav = 0.45 - 0.55
*A = PI * D * L / P*
D = Inner Dia of stator
L = width of Stator
P = Pole
Z = Total conductors per slot Slot = 36

Figure 7:Calculation of No. of turns per phase

parameter	value	Independent Variables	
		unit	description
Pout		7.5 kw	Rated output power
Speed		1440 rpm	Rated Speed
Vt		420 V	Line to Line terminal voltage(RMS)
Connection	Delta		Winding Connection Type
fs		50 Hz	Supply Frequency
p		4	No. of poles
deff		87%	<i>Desired efficiency</i>
dPF		0.85	<i>Desired power factor</i>
Bav		0.44563 Tesla	Magnetic Loading (Average)
ac		27 kA/m	electrical Loading
Ns		36	No.of stator slots
Nr		32	No. of rotor slots

Kf		0.68	Slot fill factor
Np		1	<i>No. of parallel paths in winding</i>
Lend		12 mm	End length of conductor before bending
ssq		1	Stator slot skew
N layers	Single layer		No. of layers
Winding Material	Copper		
Jsw		7.1 A/mm ²	Stator windings current density
Jb		6.5 A/mm ²	Rotor bar current density (4<Jb<7)
Je		5.5 A/mm ²	Rotor end ring current density (5<Je<8)
Cage Material	Aluminium		
Bst		1.6 Tesla	Maximum flux density in stator tooth
Bsy		1.4 Tesla	Maximum flux density in stator yoke
Br _t		1.6 Tesla	Maximum flux density in rotor tooth
Bry		1.4 Tesla	Maximum flux density in rotor yoke
Electrical Steel	M350-50A, 50 Hz		Grade of electrical steel of stator and rotor.

Dependent Variables			
parmaeter	value	unit	Description
Pin		8.620689655 kW	Input kW
Sin		10.14198783 kVA	Input kVA
Qin		5.342623607 kW	Input Reactive Power
rpm - s		1500 RPM	Synchronous speed
rps-s		25 RPS	Synchronous speed
wm-s		157 rad/s	Synchronous angular speed
rps		24 rps	Rated speed
wm		150.72 rad/s	Rated angular speed
slip		0.04	Rated slip
fr		2 Hz	Rotor frequency
Tout		49.7611465 Nm	Output Torque
It		13.9420266 A	Terminal Current
Iph_rms		8.049668937 A	Phase Current
Ic_rms		8.049668937 A	Coil Current

Figure 8: Calculaiton in Excel

4.1.2 Selection of Winding Type

Winding Type	Parallel Branches	Conductors/slot	Wire Diameter(mm)	Efficiency(%)	Power Factor	Output Power(kW)	Total Loss(kW)	Torque (Nm)	Speed
Whole coiled	1	40	1.024	92.92	0.903	7.5	0.57109	49.7527	1440.41
Whole coiled	1	60	1.024	92.039	0.91933	3.2	0.28339	21.725	1440
Whole coiled	1	50	1.024	83.87	0.9041	7.5006	1.4419	51.8458	1381.5
Half coiled	1	50	1.024	82.85	0.9	7.5	1.5516	51.9812	1377.81
Half coiled	1	40	1.024	92.65	0.903	7.5	0.59486	49.739	1439.91
Half coiled	1	55	1.024	77.276	0.869	7.5	2.205	54.1362	1322.95
Whole coiled	1	55	1.024	79.024	0.877	7.4996	1.9907	53.7341	1332.77
Half coiled	2	58	1.024	79.91	0.36	7.4994	1.8758	48.127	1467
Whole coiled	2	58	1.024	80.28	0.351	7.4997	1.84	48.8013	1467.52
Whole coiled	2	68	1.024	92.65	0.67	7.5	0.6211	49.1453	1457.31
Half coiled	2	68	1.024	92.1451	0.687	7.5	0.63	49.158	1456.99
Half coiled	1	68	1.024	92.236	0.61	2.5	0.218	16.952	1440
Half coiled	2	90	1.024	90.97	0.91	7.499	0.74	50.4357	1419.92
Whole coiled	1	90	1.024	93	0.91	1.4987	0.1066	9.87	1440
Half coiled	2	80	1.024	92	0.9	7.5	0.9032	49.739	1439.91
Whole coiled	2	80	1.024	92.92	0.90318	7.5	0.57	49.752	1440.41
Whole coiled	2	84	1.024	92.353	0.9124	7.499	0.62	49.962	1433.22
Whole coiled	2	76	1.024	93.3675	0.8816	7.49	0.53	49.49	1446.9
Half coiled	2	76	1.024	93.107	0.882	7.5	0.55	49.5143	1446.48
Half coiled	2	72	1.024	93.086	0.812	7.499	0.557	49.3158	1452.14
Whole coiled	2	72	1.024	93.355	0.8055	7.499	0.53	49.307	1452.48
Half coiled	2	83	1.024	92.21	0.91	7.499	0.63	49.9258	1434.5
Half coiled	2	85	1.024	91.89	0.91	7.499	0.661	50.0605	1430.64

Figure 9: Excel Table of Suitable Winding Selection Analysis

This chart compares the performance characteristics of different winding types in electric motors, specifically focusing on whole coiled and half coiled windings. The data is categorized by winding type, parallel branches, conductors per slot, wire diameter, efficiency, power factor, output power, total loss, torque, and speed. Whole coiled and half coiled windings are evaluated across various configurations with one or two parallel branches and varying numbers of conductors per slot, all using a wire diameter of 1.024 mm. Efficiency ranges significantly from 77.62% to 93.67%, with power factors between 0.81 and 0.92. Output power remains relatively stable around 7.5 kW, while total loss varies from as low as 0.28 kW to as high as 1.99 kW. Torque values are listed between 9.87 Nm and 54.132 Nm, and speed ranges from 1322.77 to 1467.52 RPM.

Highlighted rows indicate configurations of particular interest. For example, the orange-highlighted rows show half coiled windings with a single parallel branch and efficiencies on the lower end of the spectrum, while the green-highlighted rows depict configurations achieving higher efficiencies and lower total losses. Notably, the highest efficiency recorded is 93.36% for a whole coiled winding with two parallel branches and 76 conductors per slot, yielding a total loss of 0.53 kW. This chart illustrates how different winding configurations impact motor performance, emphasizing the trade-offs between efficiency, total loss, and other operational parameters.

4.1.3 Calculations in Matlab Software

Upon calculating the certain parameters on Excel, more other complex parameters related to the motor design was calculated on Matlab Software. The design terms of the machine along with various factors were studied through various books on machine design. The code is listed in the appendix.

4.1.4 Designing in Ansys Electronics Desktop Software.

Now, to analyze the motor design in more detail, we used the help of Ansys Electronics Software. In this software, we entered the motor design parameters value and analyzed the results in graphs. The modeling work done is shown in the following figures.

Name	Value	Unit	Evaluated V..
Name	Setup1		
Enabled	<input checked="" type="checkbox"/>		
Operation Type	Motor		
Load Type	Const Power		
Rated Output Power	7500	W	7500W
Rated Voltage	420	V	420V
Rated Speed	1440	rpm	1440rpm
Operating Temperature	80	cel	80cel

Name	Value	Unit
Winding Connection	Delta	
Frequency	50	Hz

Figure 10: Entering the motor value as input.

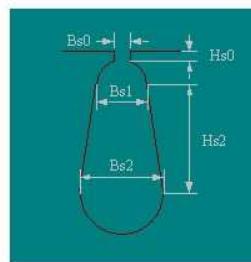


Figure 11: Selection of stator slot

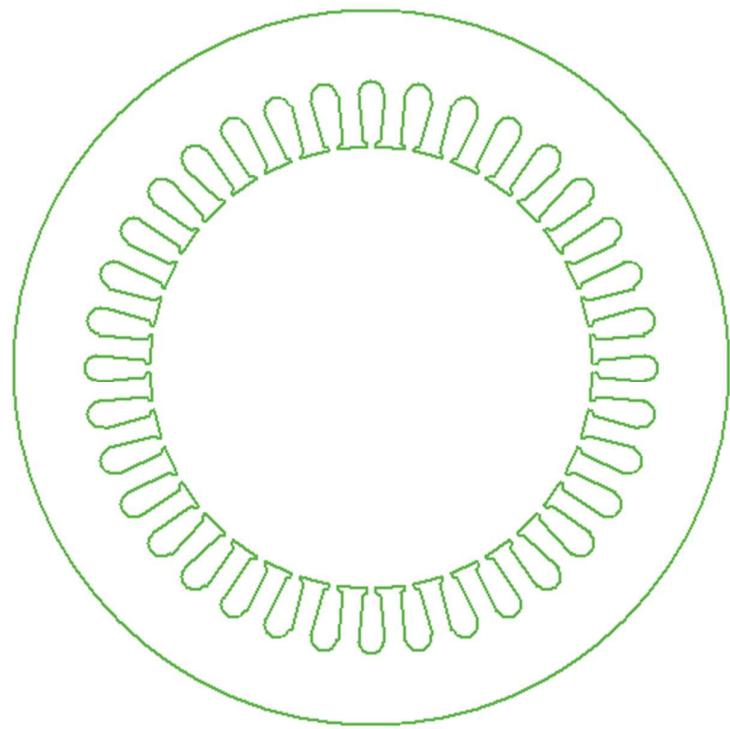


Figure 12: Stator Slot Design

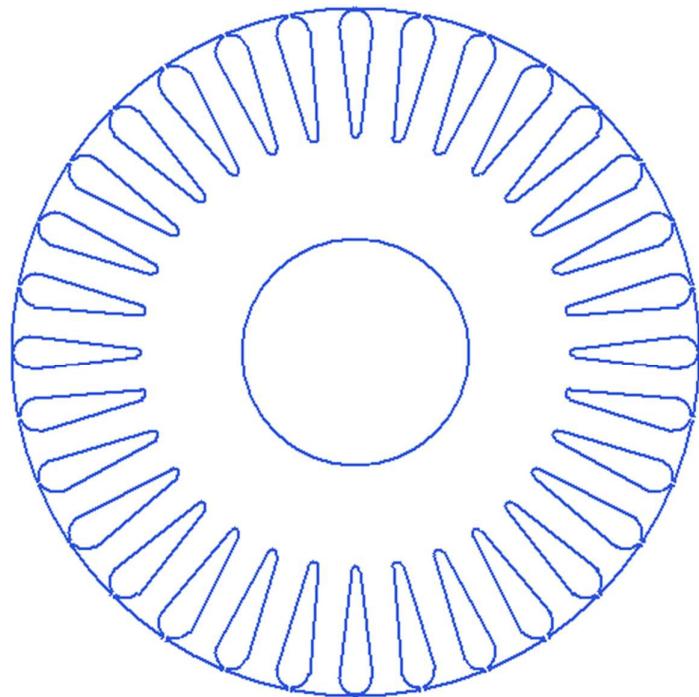


Figure 13: Rotor Design

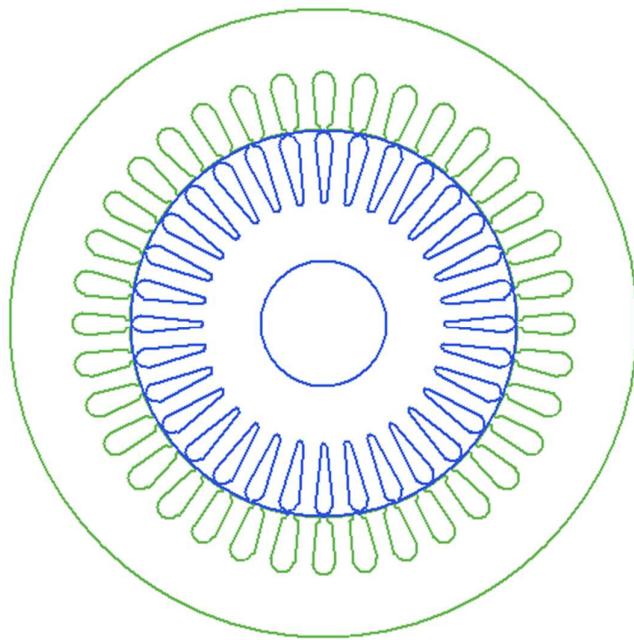


Figure 14: Machine with stator and core

Stator			
	Name	Value	Unit
	Outer Diameter	220	mm
	Inner Diameter	136	mm
	Length	147	mm
	Stacking Factor	0.95	
	Steel Type	User Material	
	Number of Slots	36	
	Slot Type	2	
	Lamination Sectors	0	
	Press Board Thickness	0	mm
	Skew Width	0	

Figure 16: Stator Parameters

	Name	Value	Unit
	Auto Design	<input type="checkbox"/>	
	Parallel To...	<input type="checkbox"/>	
	Hs0	0.85	mm
	Hs1	0.85	mm
	Hs2	14.3	mm
	Bs0	3	mm
	Bs1	5.8	mm
	Bs2	8.5	mm

Figure 15: Stator Slot Shape Dimensions

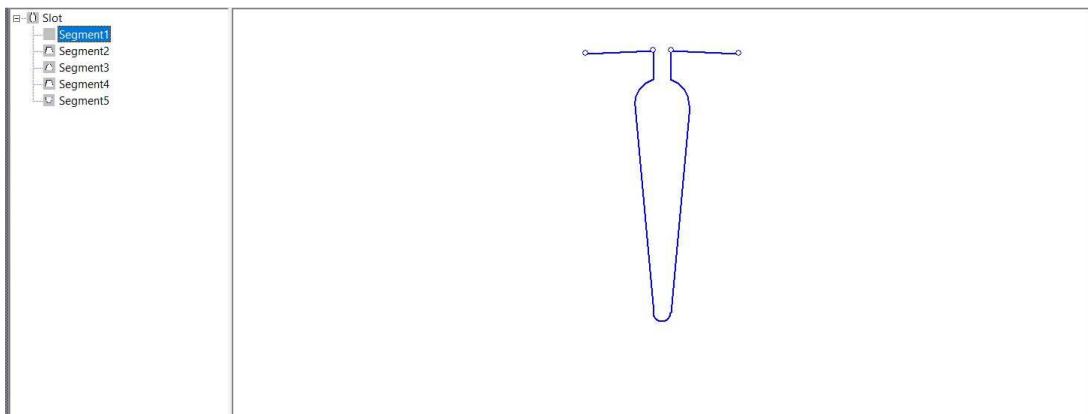


Figure 17: User Defined Rotor Slot Design

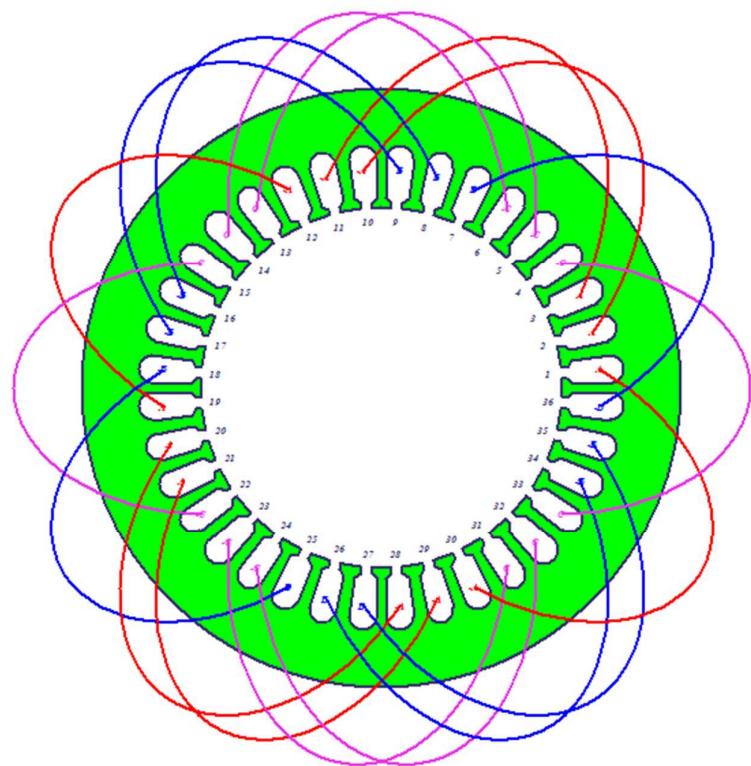


Figure 18: Connection of Coils in Single Layer Winding

Data: Rated Performance

	Name	Value	Units	Description
1	Stator Ohmic Loss	337658	mW	
2	Rotor Ohmic Loss	242677	mW	
3	Iron-Core Loss	4.64637	mW	
4	Frictional and Windage Loss	0	mW	
5	Stray Loss	0	mW	
6	Total Loss	580339	mW	
7	Output Power	7499490	mW	
8	Input Power	8079830	mW	
9	Efficiency	92.8174	%	
10	Power Factor	0.858015		
11	Rated Torque	49.2882	NewtonMeter	
12	Rated Speed	1452.98	rpm	
13	Rated Slip	0.0313448		

	Name	Value	Units	Description
1	Stator Phase Current	7.4737	A	
2	Magnetizing Current	3148.45	mA	
3	Iron-Core Loss Current	0.00391452	mA	
4	Rotor Phase Current	6541.83	mA	
5	Armature Thermal Load	133.284	A^2/mm^3	
6	Specific Electric Loading	23299.8	A_per_meter	
7	Armature Current Density	5720410	A_per_m2	
8	Rotor Bar Current Density	2347970	A_per_m2	
9	Rotor Ring Current Density	11908100	A_per_m2	

Data: No-Load Operation

	Name	Value	Units	Description
1	No-Load Stator Phase Current	3.2626	A	
2	No-Load Iron-Core Loss	4.98841	mW	
3	No-Load Input Power	66394.5	mW	
4	No-Load Power Factor	0.0161509		
5	No-Load Slip	7.65254e-06		
6	No-Load Shaft Speed	1499.99	rpm	
7	Stator Resistance	2.01503	ohm	
8	Stator Leakage Reactance	3.04877	ohm	
9	Rotor Resistance	1.88988	ohm	
10	Rotor Leakage Reactance	4.74219	ohm	

Figure 19: Snippet of ANSYS RmXprt showing rated and no load operation parameters

Data: Stator Winding

	Name	Value	Units	Description	
1	Number of Conductors per Slot	74			
2	Number of Strands	1			
3	Wire Diameter	0.912	mm		
4	Wire Wrap	0	mm		
5	Slot Fill Factor	47.1216	%		
6	Winding Factor	0.959795			
7	Half-Turn Length	268.906	mm		

Figure 20: Ansys RmXprt Stator Winding value

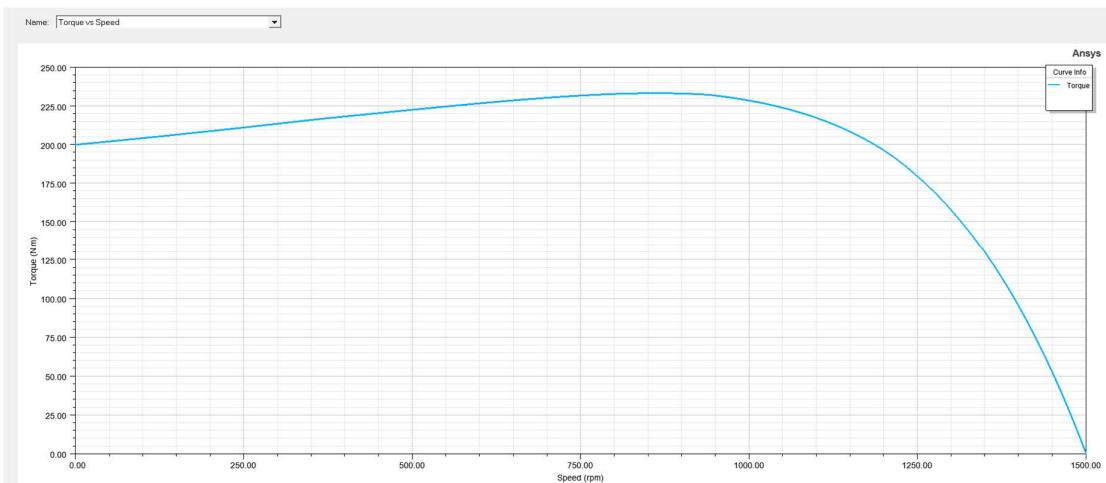


Figure 21: Plot of Torque and Efficiency respectively with speed

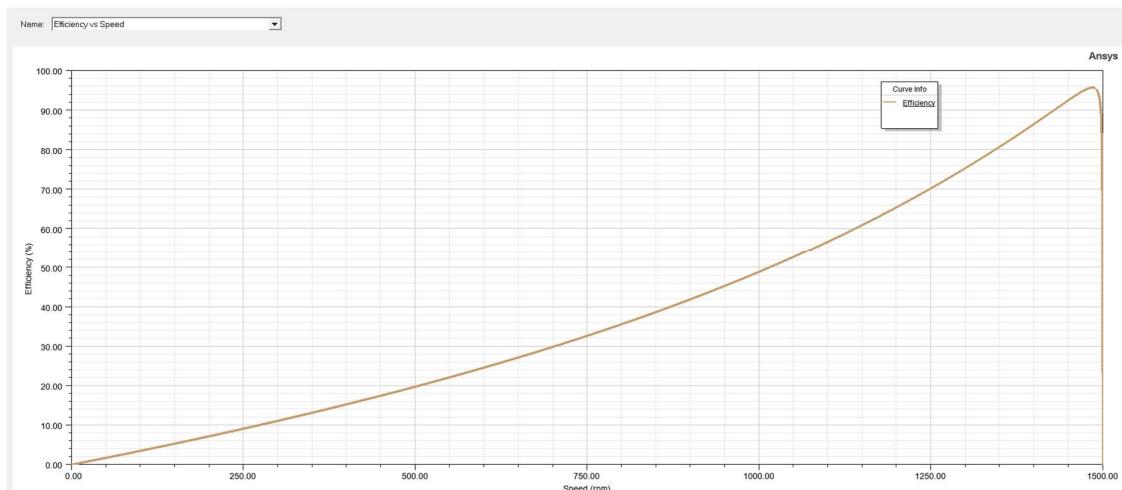


Figure 22: Plot of Efficiency VS Speed

4.1.4 Rewinding and Sensor Installation

At the very beginning, the motor was analyzed through its nameplate. Information of various parameters of the three-phase induction motor was displayed in the nameplate. After that, the motor was disassembled where rotor was separated from the stator.

For the rewinding process, we found that the damaged winding had delta configuration previously. After that, the total number of conductors per slot were calculated by cutting the winding at one end. It was found to be 78 conductors per slot. The type of wire used was copper wire of 19 gauge. Also, the motor had single layer winding structure.

The next step was to actually remove the old damaged winding form the stator slots. After all the windings were removed, with the help of hammer and chisels, the part of the stator was cleaned through blast of air so that dust particles present gets removed. For more accuracy, the stator's outer diameter and inner diameter was again measured.

To give insulation, PVC slot paper was used. The dimension of the slot paper was made according to the stack length of the stator slots. The slot paper was molded in light U- shape in two equal halves and it was inserted in all the 36 slots of the motor. The specific reason for molding the slot paper in U-shape was to create maximum space for the conductors so that they can be inserted as in required number.

For our motor, we chose single layer winding. The stator has 36 slots, 4 poles and 18 coils.

We have to o winding of the motor for 4 poles at 1440 rpm. The pitch is calculated as $\frac{Slot}{Pole}$.

We have pitch = $\frac{36}{4} = 9$ i.e. [1-9]. We have to do slight modification in this. From pitch of [1-9], we cannot do single layer winding. We need even numbers for that. So we do the adjustment as [1-9] ± 1 . So the pitch becomes [1-8] or [1-10]. We chose pitch of [1-8] so as to save quantity of copper required for single layer winding. In single layer, while placing the coils, we leave one slot space. Also, coil group = phase * pole = $3*4 = 12$. But we have 18 coil group.

So $18/12 = 1.5$. We can't have 1.5 coil. So we have made one single coil and other two double coil that means we have alternate single and double coil. Our target is to make 12 coil group. Each phase should get 6 coils each.

The development diagram is shown in the figure below.

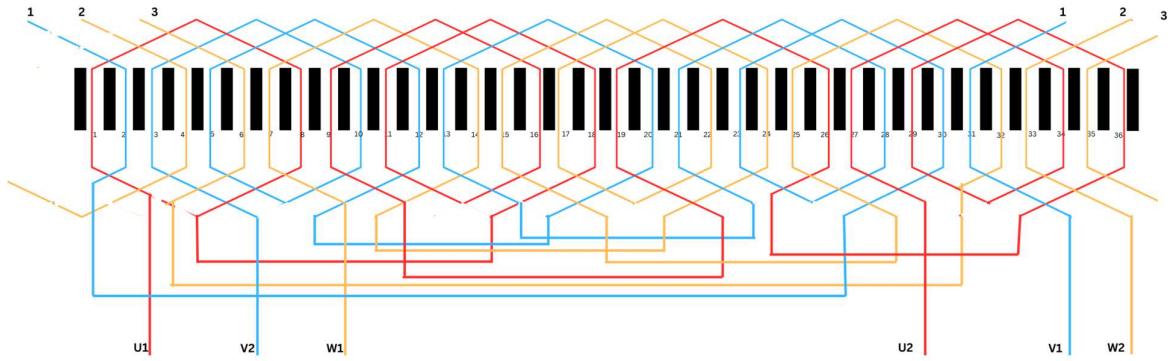


Figure 23: Winding Development Diagram

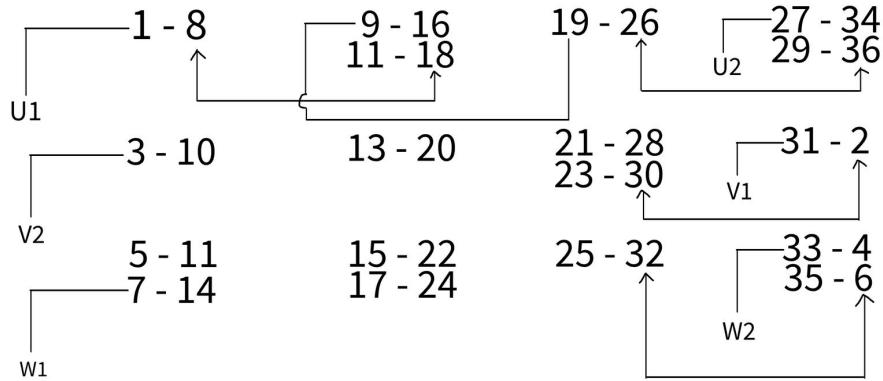


Figure 24: General outline of winding connectins

In the diagram, we can see the first slot is number 1 and the coil is placed at slot number 8. From slot 8, the coil is placed to slot number 18 and hence other slots are also placed accordingly. Here, we have taken one single coil and two double coil in order to match the coil group of 12. So for 18 coil groups, the winding is done such that, it fits to the coil group of 12. So, we alternate one single and two double coil in the single layer winding. Similarly, the end connections of U1, W1, V1, and U2, V2, W2 are taken accordingly as shown in the figure.

We have chosen SWG of 19, so its diameter = 0.912 mm and area(A) = 0.6534mm²

For our winding, we have placed 74 total conductors per slot. In 18 coil group, all the coils are not of equal size. For the single coil, the standard wire gauge no. 19 was taken and for double coil, the standard wire gauge of no. 19.5 was taken. The next step was to take the approximate size and shape of a coil. For this, a single coil was inserted in the empty slot of the motor and then taken out as shown in figure (a). The framing of the coil was done in the wooden shape structure as shown in figure (b).



Figure(a):Estimating the shape of a single coil



Figure (b): Framing the coil shape

4.1.5 Rewinding of Three Phase Motor in Steps



Figure 25: First step of Rewinding

In figure 25 , figure 1 represents the bare stator with 36 slots. Figure 2 represents the insulation paper (PVC paper) that is being cut according to the dimensions of the stator slots. They are usually used for insulations. The insulations is done between motor winding wires and the stator laminations. The length of the paper is of 17 cm and width is of 4 cm. Figure 4 represents this cut dimensions. Figure 3 represents putting the insulation paper on

the stator slots. Figure 5 is the filled stator slots with this insulation paper that is ready for the copper windings to put in. Figure 6 is the cutted plywood to fit the winding size that is to be put on the stator slots. Figure 7 is the windings or copper coils made from that piece of plywood. After the coil is made, the windings are placed in the stator as shown in figure 8. The coils is going from slot number 1 to the slot number 8 as represented in figure 8. Figure 9 shows the adjustment of the coil and covering it again with the insulation paper from above.

And finally, the copper wires were placed in the stator accordingly with development diagram shown in the figure (8). Three PT-100 temperature sensors were placed in three different phases to take the reading of stator winding's temperature.

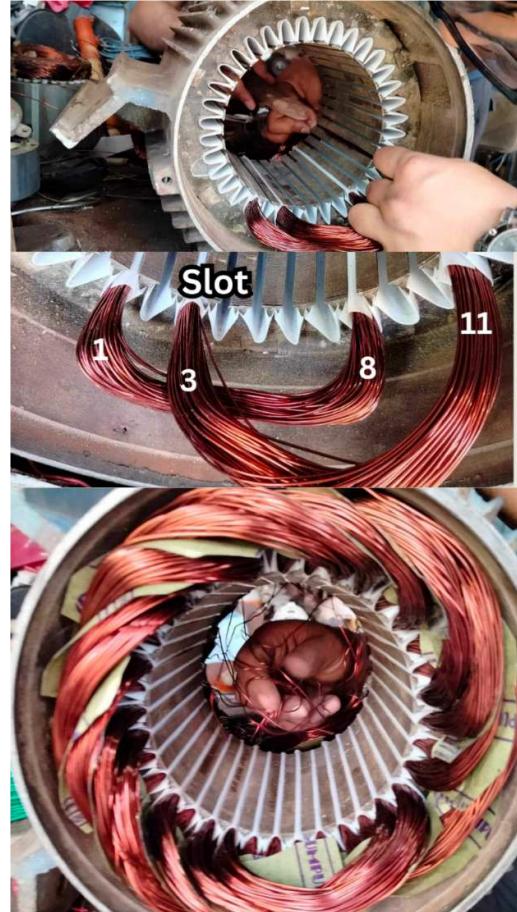


Figure 26: Winding Installation

The first figure shows the windings going from 1 to 8 and 3 to 11 respectively both with the difference of 7 slots. The second figure shows the coverings of the stator slots with

insulation paper from the top completing the works on those stators. The last figure shows the completed windings. The paper that is used between the coils for their insulations is the vertex paper.



Figure 27: Completion of rewinding and installation of fan

Here, the first figure shows the placement of PT-100 sensor in different slots in different phases. The slots as we can see in the figure are slot number 1, 7 and 31 respectively. The sensor is placed in order to get the heat data from different phases connected stator windings. The sleeve pipes are used in order to protect the sensor's wire that is coming to the connecting box. The second figure shows the tightening of the coils so that they are fixed at a single place even when the induction motor has to be moved so that windings

won't get affected easily. The varnish is being poured in the windings for the proper insulation even when the temperature gets a little high. The varnish is used for the protection of the coils from being burnt easily. Also they help to tighten the copper windings together and makes them rugged and connected together. The fourth figure shows the connecting box. The connection is given in delta. The fifth figure shows the completed stator with tightening of the windings along with the successful placement of the PT-100 sensors. Figure 7 represents the changed new fan.



Figure 28: Painting of the motor

The above figure in general represents the painting process of the induction motor. Figure (1) shows the induction motor before the painting process. Figure (2) represents the terpine oil that is to be used mixing with the enamel paints. Figure (3) represents the sand paper that is used in order to smoothen the iron or steel surface of the induction motor as shown in figure (4). The sand paper is useful in removing the external rust, previous paintings and rough surface so that enamel paints get well placed in the surface. Figure (5)

shows the completed work from the sand paper and the motor is ready to be painted. Figure (6) is the enamel paint with mixed with little terpentine oil.



Figure 29: Completion of painting of motor

Above figure simply shows the painting done along with the name plate of the induction motor.

4.1.6 Sensor Data Collection

4.1.6.1 PT-100 Temperature Sensor

In this sensor, platinum is used. This is so because, it gives linearity and stability of any other material. This sensor is sandwiched inside the windings of the induction motor. We have placed three of its kind in three different phases for measuring the temperature. Here, the value of resistance will start to change depending on the ambient temperature. Before using it, the calibration of the sensor is done. The value of the temperature is determined according to the resistance value. From the PT 100 resistance table, the equivalent temperature is determined. At 0 degree Celsius, there is a resistance of 100 ohm.



Figure 30: PT-100 Temperature Sensor

In our sensor, there is a three wire terminal for more accurate measurement. The circuit connection is shown in the figure 31. There are two white and one red wire. The two white wire are internally connected and the resistance of 100 ohm exists between a red and a white wire. Now, since this sensor gives the value in resistance, we need to convert it into voltage so that the microcontroller can read the data. For this, we have used a voltage divider circuit, where we have used a resistance of 330 ohm. The converted analog signal is given to ESP-32 microcontroller which has inbuilt analog to voltage converter. The pin number 36, 39 and 34 of the ESP-32 microcontroller is used for this purpose.

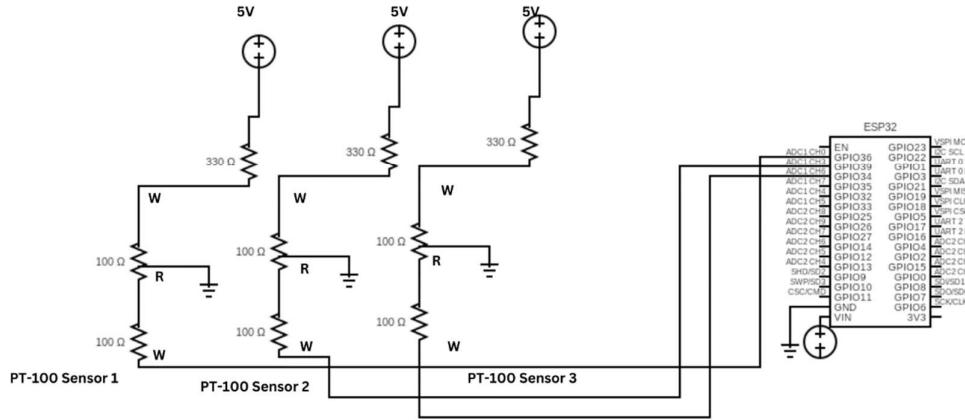


Figure 31: PT-100 Temperature Sensor with ESP-32

4.1.6.2 Calibration of PT-100 Sensor

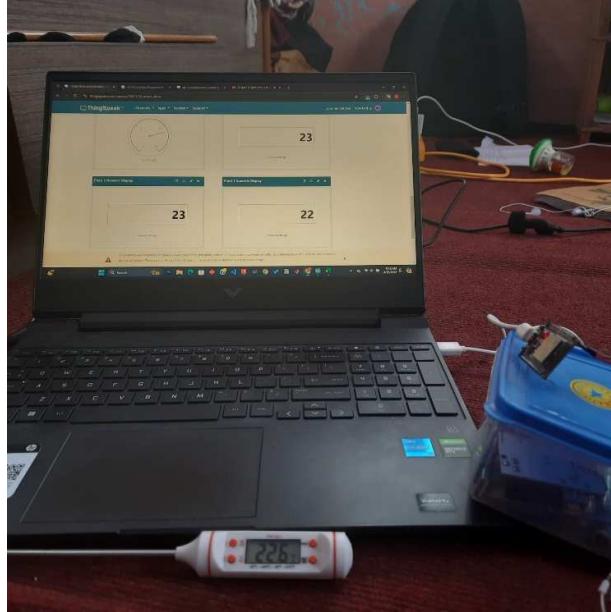


Figure 32: Sensor Clibration of PT-100 temperature sensor

The calibration of PT-100 sensor is done with the help of a thermometer. Here, the temperature is measured both by the sensor and the thermometer. The sensor value obtained is seen in the ThinkSpeak platform and the temperature measured by thermometer is seen in the display of the thermometer itself. Here, we obtained the range of values for temperature and curve fitted it linearly in MATLAB software.

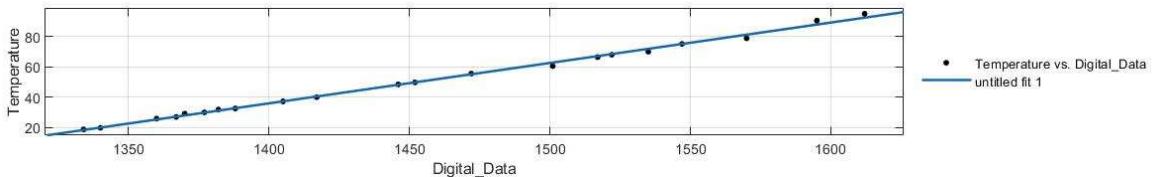


Figure 33: Curve Fitting of temperature data.

Linear model Poly1:

$$f(x) = p1*x + p2$$

Coefficients (with 95% confidence bounds) :

$$\begin{aligned} p1 &= 0.266 \quad (0.259, 0.2729) \\ p2 &= -336.4 \quad (-346.5, -326.2) \end{aligned}$$

Goodness of fit:

SSE: 32.54

R-square: 0.997

Adjusted R-square: 0.9969

RMSE: 1.309

When current flows through motor windings, resistance and other losses generate heat, which can degrade motor components over time. The insulation system protecting the windings is particularly affected by this heat. To address the potential for failure of the insulation due to heat, the NEMA MG-1 standard establishes four classes of motor insulation, which define the insulation system's ability to endure a specified temperature while providing a specific operating life.

The four classes of motor are A, B, F and H. Form the nameplate of our motor, the insulation class is B. So for class B insulation, the maximum allowable temperature is 130 degree Celsius. The ambient temperature is 40 degree Celsius. Considering this factor, the temperature sensor's reading is adjusted to show three different led indicators on Think Speak Platform. For temperature value up to 50 degree Celsius, it indicates green color, meaning it is safe. For up to 120 degree Celsius, we set it to indicate orange color, meaning it is under consideration. And for temperature above 120 degree Celsius, red led is indicated meaning, the special attention is now needed so as to not to overcome the motor insulation class ratings.

4.1.6.3 SCT-013-060 (50A) Non-invasive AC Sensor Split Core Current Transformer

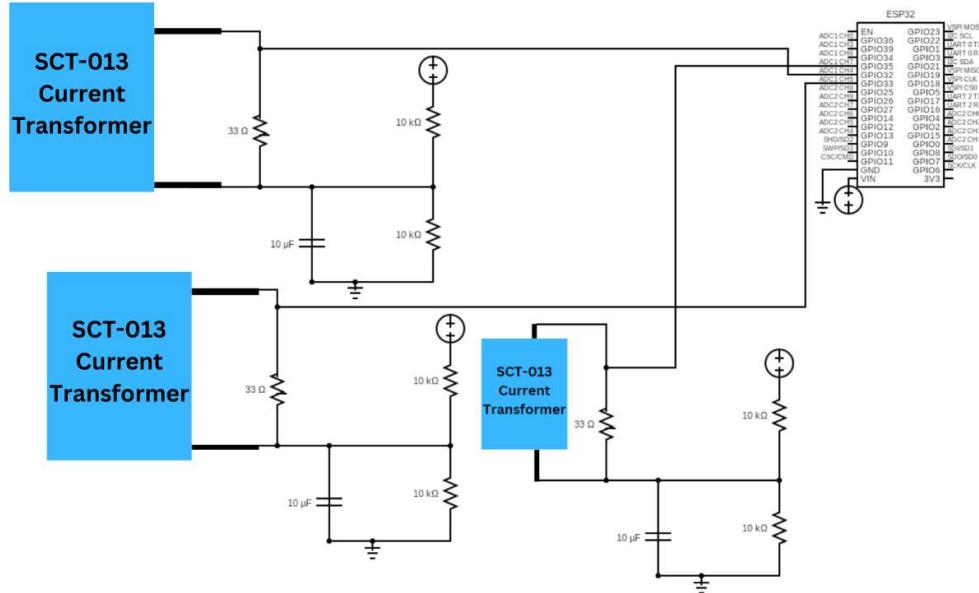


Figure 34: SCT sensor connection diagram with ESP-32

In order to measure the three-phase current, three SCT-013 current sensor are placed in three different phase wire of the motor. This sensor has the sensitivity of 30mV/Ampere. The connection of the sensor is shown in the figure 34 above. Current sensing transformer act as current sources and requires a load. A current source shouldn't be left open because it can cause an infinite voltage. The burden resistor of 33 ohm is kept so as to convert the current to a limited voltage. This can also be used to discharge a capacitor when the circuit is no longer powered. The capacitor of 10 microfarad is used to reduce the unwanted noise.

The voltage divider circuit is used to convert the signal into voltage form so that the ESP-32 microcontroller can read the value. For this, we have used two resistance of 10K ohm in series as shown in figure 34. We have use pin number 32, 33 and 35 of ESP-32 to read the sensor value through the voltage divider circuit.

4.1.6.4 Sensor Calibration of SCT -013

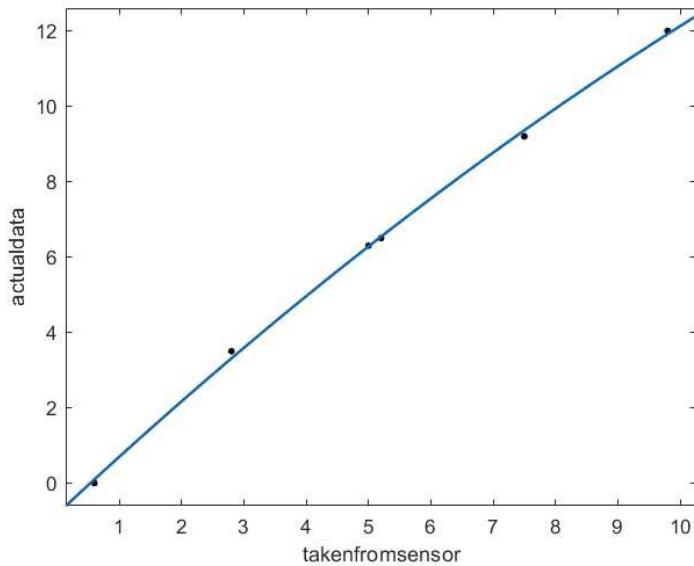


Figure 35: SCT current transformer sensor calibration

Linear model Poly2:

$$f(x) = p1*x^2 + p2*x + p3$$

Coefficients (with 95% confidence bounds):

$$\begin{aligned} p1 &= -0.02515 \quad (-0.04883, -0.001473) \\ p2 &= 1.546 \quad (1.29, 1.803) \\ p3 &= -0.8228 \quad (-1.427, -0.2183) \end{aligned}$$

Goodness of fit:

SSE: 0.07993

R-square: 0.9991

Adjusted R-square: 0.9985

RMSE: 0.1632

This figure shows the curve fitting of sensor value with the measured value. In X-axis, current measured in ampere form the sensor is shown whereas in the Y-axis, the actual current data measured form the multi-meter is shown. The curve is fitted in linear model in MATLAB software. Here, SSE is low which means there are small prediction errors. Also, the RMSE is low, which indicates that the predictions are very close to the actual value.

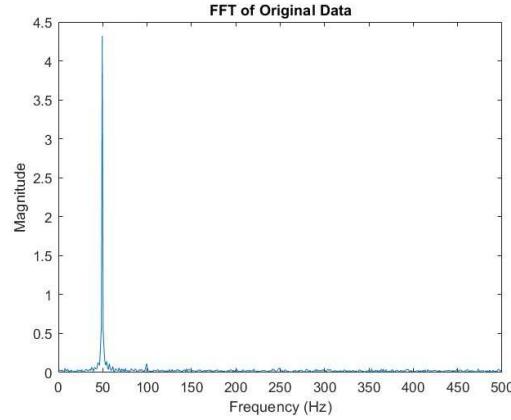


Figure 36: FFT of Current Value measured form the sensor

The figure 36 represents the Fast Fourier Transform of the current value. Since we have operated the system in 50 Hz, there is a very high magnitude spike at 50 Hz as seen in the above figure. The output waveform is shown in the figure (XX) below.

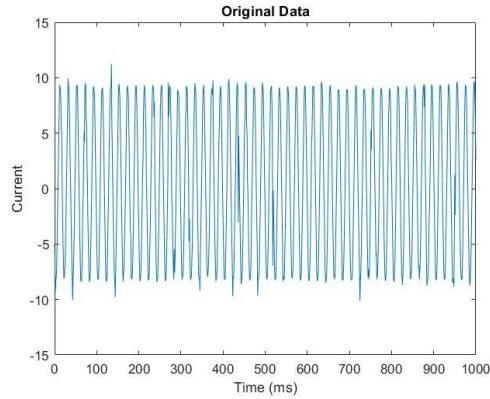


Figure 37: Original Data of Current with respect to time in millisecond

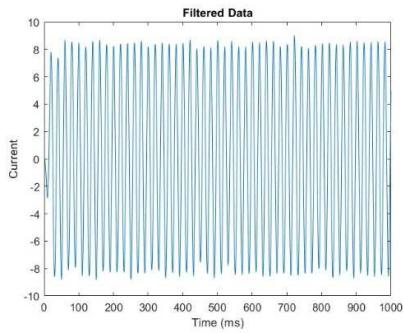


Figure (a): Filtered data

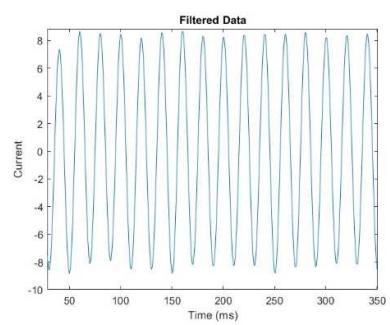


Figure (b) Zoomed filtered data

4.1.7 Display on ThingSpeak Online Platform.

To display the live data from the sensor, we are using ThingSpeak platform. Here, we installed ThingSpeak library on Arduino IDE. Since we are using ESP-32 as a microcontroller, to get live data, RTOS for ESP-32 is used. The code for this is mentioned in appendix section. Below are the figures for live data collected. The three-phase current waveform is visualized on ARDUINO Serial Plotter which can be seen in the figure below.

4.1.7.1 Current Readings

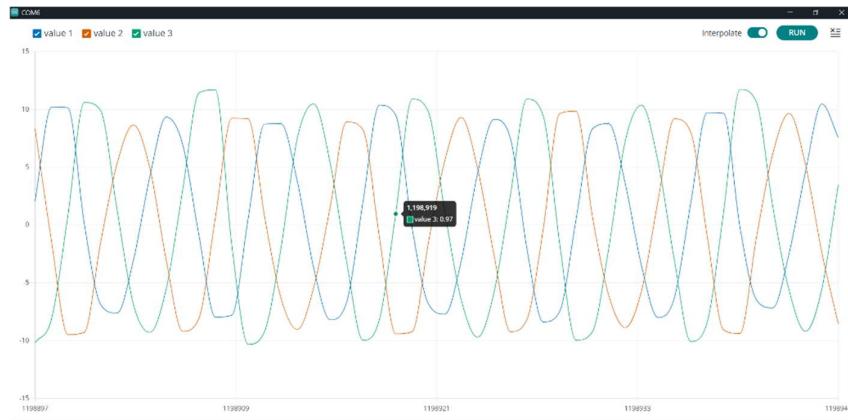


Figure 38: Three-phase current waveform obtained from the current sensor (live)

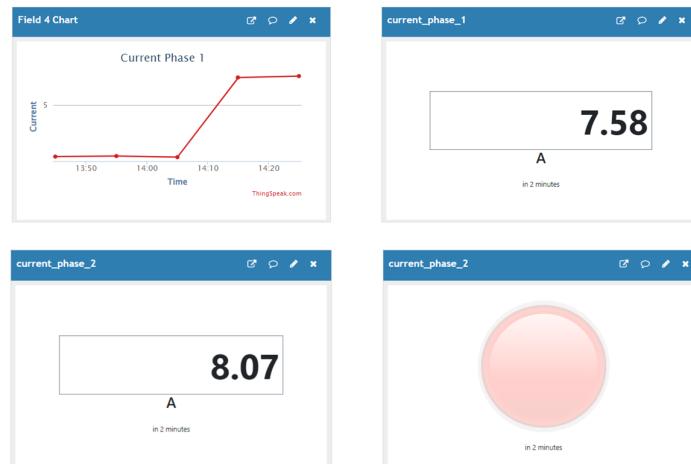


Figure 39: Display of current value of phase 1 and phase 2

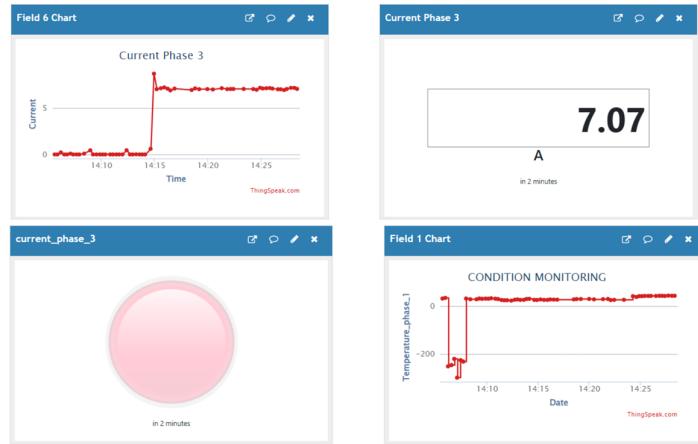


Figure 40: Current value of third phase

Here, our phase current calculated on ANSYS Modeling is 7.47 ampere. The obtained values for the phase current are in the desired range. Hence the pink colour is indicated as healthy condition for the current. In ThingSpeak, not only the live numerical values, but also the graph of the obtained current values with respect to time can be visualized as seen in the figure above.

4.1.7.2 Temperature Readings

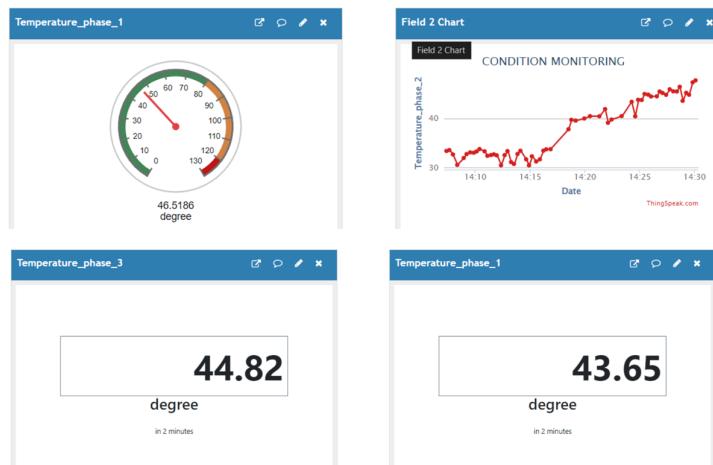


Figure 41: Display of temperature values

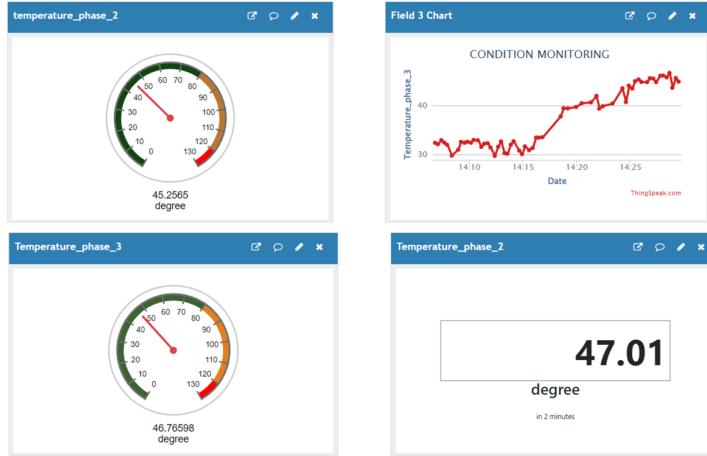


Figure 42: Live temperature value display of three different phases.

Here, for temperature readings, three PT-100 temperature sensors are placed on three different phases at the stator. The obtained values for the temperature can be seen within the normal range since, we are operating the motor in no load condition. Up to 80 degree Celsius temperature, the indicator points green values, meaning safe operation. Above 80 degree Celsius, and up to 120 degree Celsius, as the temperature rises, the indicator will point to orange colour, indicating the slight attention to the rise in temperature in the stator windings. Above 120 degree Celsius, the indicator moves to red color, indicating there is a high attention needed as the rising temperature is reaching to the insulation limit of 130 degree Celsius. Hence in this way, condition monitoring is done for the three-phase Induction Motor with respect to its current and temperature values.

4.1.7.3 Some Pictures of Setup and Results

The setup of three-phase induction motor along with sensor installation is shown in the figures in the coming pages. For, this at first, we have used power analyser to measure the motor's data while running on no load. Since the motor is operated on no load, there current is not running on rated condition, as there is no load for higher current draw. The obtained data from the power analyser is shown in the table below.

Parameter	Phase 1	Phase 2	Phase 3
Voltage (V)	378.9 V	378.9 V	378.8 V
Line Voltage (V)	217.9 V	216.4 V	218.2 V
Current (A)	7.100 A	8.076 A	7.251 A
Active Power (KW)	0.239 KW	0.173 KW	0.058 KW
Apparent Power (KVA)	1.558 KVA	1.747 KVA	1.582 KVA
Reactive Power (KVAR)	1.508 KVAR	1.737 KVAR	1.581 KVAR
Power Factor (PF)	0.15	0.10	0.04
Total Active Power (KW)	0.471 KW		
Total Apparent Power (KVA)	4.880 KVA		
Total Reactive Power (KVAR)	4.857 KVAR		
Total Power Factor (PF)	0.10		
Frequency (Hz)	50.0 Hz		

Figure 43: Table obtained from the power analyser while running the motor on no load condition

Here, since there is no load given and no capacitor bank was available, the power factor obtained was very poor due to high reactive power. The supply was given on 50 Hz frequency. The phase currents were in accordance to the calculated value of 7.4 Ampere. Since, the motor is connected in delta configuration, the line voltage is equal to phase voltage. For our motor, the rated voltage was for 420V. The obtained value for the phase voltage is around 380V. This shows that voltage values across the phases are similar as it should be in delta connection.

We achieved to obtain the no load synchronous speed of the motor of 1500 rpm. The speed was measured with laser tachometer which came to be 1495 rpm. This is shown in the figure below. Here, a laser tachometer is used to measure the rpm of the motor. The laser is pointer to the rotating shaft of the rotor and the rpm value was obtained in the tachometer display.



Figure 44: RPM calculation of the motor



Figure 45: Three-phase supply and power analyser connection to the motor



Figure 46: Final Connection

CHAPTER V: CONCLUSION

5.1 Conclusion

In this project, we took a motor with damaged winding, noted it's nameplate, calculated the motor's parameters, developed and analysed the motor's model in ANSYS software and calculated the desired parameters. We rewinded the motor and refurbished the motor with a new paint. We then condition monitored the motor with it's temperature and current values. Three temperature sensors were inserted in the three different phase of the motor to monitor the temperature of three-phase. Similarly, three current sensor was placed in the three-phase wire to monitor the phase wire current. The motor was run on no load condition and its synchronous speed was verified with the help of tachometer.

5.2 Problems Encountered and Limitations

- Difficulty arises in finding adequate research papers on the rewinding and monitoring of induction motors.
- During literature review, some of the topics were difficult to understand.
- Difficulty in finding some data related to motor parameters.
- Motor was not operated on full load due to unavailability of enough rated current supply in the lab.
- Low power factor of the motor due to no use of capacitor banks.

5.3 Mitigation

- For literature review, we searched the topic on various journals and research papers.
- To obtain the machine data parameters, analytical calculations was done and to remove the error of wrong entry of data, the model data was cross checked with the supervisor.

GANTT CHART

Work Completed 

Task/ Month	Aug	Sept	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul
Literature Review												
Project Proposal												
Analytical Model Development												
Modeling / Simulation												
Mid Term Defense												
Interface System Development												
Rewinding and Sensor Installation												
Implementation and Testing												
Documentation and Final Defense												

REFERENCES

- [1] Swapnil K. Gundewar, Prasad V. Kane, "Condition Monitoring and Fault Diagnosis of Induction Motor", JVET, 2020
- [2] H. A. Raja, H. Raval, T. Vaimann, and A. Kallaste, "Cost-efficient real-time condition monitoring and fault diagnostics system for BLDC motor using IoT and Machine learning," in *2022 International Conference on Diagnostics in Electrical Engineering (Diagnostika)*, Tallinn, Estonia, Sep. 2022, doi: 10.1109/Diagnostika55131.2022.9905102.
- [3] Armintor, Kirk; Mills, Russell W.; Stiffler, Walter G. (1981). *Repair, Restoration, and Revitalization of Large Induction Motors.*, IA-17(6), 0–586.
doi:10.1109/tia.1981.4504008
- [4] H. A. Raja, T. Vaimann, A. Rassõlkin, and A. Kallaste, "Condition Monitoring and Fault Detection for Electrical Machines Using IoT," in *Proceedings of the Future Technologies Conference (FTC) 2022, Volume 2*, Oct. 2022, pp. 162-173, doi: 10.1007/978-3-031-18458-1_12.
- [5] P.M, Enyong. "Refurbishment of a Three-Phase Induction Motor Reflecting Local Voltage Condition." *IOSR Journal of Electrical and Electronics Engineering* 8 (2013): 64-69
- [6] Al-Badri, Maher; Pillay, Pragasesan; Angers, Pierre (2015). *A Novel Algorithm for Estimating Refurbished Three-Phase Induction Motors Efficiency Using Only NoLoad Tests. IEEE Transactions on Energy Conversion*, 30(2), 615–625. doi:10.1109/tec.2014.2361258
- [7] H. A. Raja, T. Vaimann, A. Rassõlkin, A. Kallaste and A. Belahcen, "IoT Based Tools for Data Acquisition in Electrical Machines and Robotics," 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC), Gliwice, Poland, 2021, pp. 737-742, doi: 10.1109/PEMC48073.2021.9432553. keywords: {Cloud computing;Data acquisition;Maintenance engineering;Robot sensing systems;Real-time systems;Wind turbines;Sensors;Internet of Things;Data Acquisition;Condition Monitoring.}
- [8] M. Tavakoli, H. Afshar and M. Ebrahimi, "Condition monitoring and fault diagnosis of induction motors using motor current signature analysis," 2017 IEEE International Conference on Renewable Energy Research and Applications (ICRERA), San Diego, CA, USA, 2017, pp. 1681-1686. doi: 10.1109/ICRERA.2017.8191195.
- [9] T. Mohanraj and V. Subbiah, "A review on condition monitoring of induction motors using vibration analysis," 2016 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), Chennai, India, 2016, pp. 1-6. doi: 10.1109/ICPCSI.2016.7751737.
- [10] M. A. Patel, A. M. Joshi, P. A. Joshi and U. B. Desai, "Online condition monitoring of induction motors using current signature analysis and artificial neural networks," 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Madurai, India, 2015, pp. 1-5. doi: 10.1109/ICCIC.2015.7435756
- [11] F. Petchakup, Y. Chetpattananondh and W. Kongprawechnon, "Comprehensive review on condition monitoring of induction motors using electrical signature analysis," 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, 2017, pp. 1510-1514. doi: 10.1109/IEEM.2017.8290161.
- [12] Nabto, "Guide to IoT: ESP32," [Online]. Available: <https://www.nabto.com/guide-to-iot-esp-32/>. [Accessed: 17-March-2024].
- [13] H.P.Halvorsen, "ThingSpeak," [Online]. Available:<https://www.halvorsen.blog/documents/technology/iot/thingspeak/thingspeak.php>. [Accessed: 18-March-2024].
- [14] Pruftechnik, "AGuide to Vibration Sensors for Motors," Pruftechnik Blog, 2023. [Online]. Available: <https://blog.pruftechnik.com/a-guide-to-vibration-sensors-formotors/>.

APPENDIX

[A1]. Matlab code for FFT of current signal

```
ata = sct.VarName1;

Fs = 1000;
L = length(data);
t = (0:L-1)/Fs;
freq = (-L/2:L/2-1)*(Fs/L);

figure;
plot(t*1000, data);
title('Original Data');

xlabel('Time (ms)');
ylabel('Current');

f_l = 10;
f_h = 60;
[b,a] = butter(4, [f_l, f_h]/(Fs/2), 'bandpass');

data_filtered = filter(b, a, data);

figure;
plot(t*1000, data_filtered);
title('Filtered Data');
xlabel('Time (ms)');
ylabel('Current');

Y = fftshift(fft(data_filtered, L));
P2 = abs(Y / L);
P1 = P2(L/2+1:end);

% Plot the FFT of the filtered data
figure;
plot(freq(L/2+1:end), P1);
title('FFT of Filtered Data');
xlabel('Frequency (Hz)');
ylabel('Magnitude');

Y_orig = fftshift(fft(data, L));
P2_orig = abs(Y_orig / L);
P1_orig = P2_orig(L/2+1:end);

figure;
plot(freq(L/2+1:end), P1_orig);
title('FFT of Original Data');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```

[A2]. ANSYS Three Machine Design Parameters

Number of Wires per Conductor:	1
Wire Diameter (mm):	0.912
Wire Wrap Thickness (mm):	0
Wedge Thickness (mm):	0
Slot Liner Thickness (mm):	0
Layer Insulation (mm):	0
Slot Area (mm ²):	136.908
Net Slot Area (mm ²):	130.617
Slot Fill Factor (%):	47.1216
Limited Slot Fill Factor (%):	75
Wire Resistivity (ohm.mm ² /m):	0.0217
Conductor Length Adjustment (mm):	0
End Length Correction Factor	1
End Leakage Reactance Correction Factor	1
ROTOR DATA	
Number of Rotor Slots:	32
Air Gap (mm):	0.25
Inner Diameter of Rotor (mm):	45
Type of Rotor Slot:	1
Rotor Slot	
hs0 (mm):	0.5
hs2 (mm):	21
bs0 (mm):	1
bs1 (mm):	6.75
bs2 (mm):	2
Cast Rotor:	Yes
Half Slot:	No
Length of Rotor (mm):	150
Stacking Factor of Rotor Core:	0.95
End Length of Bar (mm):	0
Height of End Ring (mm):	15
Width of End Ring (mm):	3.75
Resistivity of Rotor Bar at 75 Centigrade (ohm.mm ² /m):	0.0263158
Resistivity of Rotor Ring at 75 Centigrade (ohm.mm ² /m):	0.0263158
Magnetic Shaft:	No
MATERIAL CONSUMPTION	
Armature Copper Density (kg/m ³):	8900
Rotor Bar Material Density (kg/m ³):	2689
Rotor Ring Material Density (kg/m ³):	2689
Armature Core Steel Density (kg/m ³):	7650
Rotor Core Steel Density (kg/m ³):	7650
Armature Copper Weight (kg):	4.16489
Rotor Bar Material Weight (kg):	1.4432
Rotor Ring Material Weight (kg):	0.113569
Armature Core Steel Weight (kg):	19.8258
Rotor Core Steel Weight (kg):	10.0855
Total Net Weight (kg):	35.6329
Armature Core Steel Consumption (kg):	37.6074
Rotor Core Steel Consumption (kg):	15.8359
RATED-LOAD OPERATION	
Stator Resistance R1 (ohm):	2.01503
Stator Resistance at 20C (ohm):	1.63122
Stator Leakage Reactance X1 (ohm):	3.02104

Stator Resistance R1 (ohm):	2.01503
Stator Resistance at 20C (ohm):	1.63122
Stator Leakage Reactance X1 (ohm):	3.02104
Slot Leakage Reactance Xs1 (ohm):	1.38453
End Leakage Reactance Xe1 (ohm):	0.693997
Harmonic Leakage Reactance Xd1 (ohm):	0.942513
Rotor Resistance R2 (ohm):	1.8902
Rotor Leakage Reactance X2 (ohm):	4.62218
Resistance Corresponding to Iron-Core Loss Rfe (ohm):	1.01073e+08
Magnetizing Reactance Xm (ohm):	125.666
Stator Phase Current (A):	7.47371
Current Corresponding to Iron-Core Loss (A):	3.91452e-06
Magnetizing Current (A):	3.14845
Rotor Phase Current (A):	6.54183
Copper Loss of Stator Winding (W):	337.658
Copper Loss of Rotor Winding (W):	242.677
Iron-Core Loss (W):	0.00464637
Frictional and Windage Loss (W):	0
Stray Loss (W):	0
Total Loss (W):	580.339
Input Power (kW):	8.07983
Output Power (kW):	7.49949
Mechanical Shaft Torque (N.m):	49.2882
Efficiency (%):	92.8174
Power Factor:	0.858015
Rated Slip:	0.0313448
Rated Shaft Speed (rpm):	1452.98

NO-LOAD OPERATION

No-Load Stator Resistance (ohm):	2.01503
No-Load Stator Leakage Reactance (ohm):	3.04877
No-Load Rotor Resistance (ohm):	1.88988
No-Load Rotor Leakage Reactance (ohm):	4.74219
No-Load Stator Phase Current (A):	3.2626
No-Load Iron-Core Loss (W):	0.00498941
No-Load Input Power (W):	66.3945
No-Load Power Factor:	0.0161509
No-Load Slip:	7.65254e-06
No-Load Shaft Speed (rpm):	1499.99

BREAK-DOWN OPERATION

Break-Down Slip:	0.43
Break-Down Torque (N.m):	233.111
Break-Down Torque Ratio:	4.72955
Break-Down Phase Current (A):	52.9232

LOCKED-ROTOR OPERATION

Locked-Rotor Torque (N.m):	199.806
Locked-Rotor Phase Current (A):	70.5608
Locked-Rotor Torque Ratio:	4.05384
Locked-Rotor Current Ratio:	9.44119
Locked-Rotor Stator Resistance (ohm):	2.01503
Locked-Rotor Stator Leakage Reactance (ohm):	2.12202
Locked-Rotor Rotor Resistance (ohm):	2.1754
Locked-Rotor Rotor Leakage Reactance (ohm):	2.17891

DETAILED DATA AT RATED OPERATION	
Stator Slot Leakage Reactance (ohm):	1.38454
Stator End-Winding Leakage Reactance (ohm):	0.693997
Stator Differential Leakage Reactance (ohm):	0.942527
Rotor Slot Leakage Reactance (ohm):	1.71578
Rotor End-Winding Leakage Reactance (ohm):	0.358464
Rotor Differential Leakage Reactance (ohm):	1.75158
Skewing Leakage Reactance (ohm):	0.796467
Stator Winding Factor:	0.959795
Stator-Teeth Flux Density (Tesla):	1.54077
Rotor-Teeth Flux Density (Tesla):	1.74272
Stator-Yoke Flux Density (Tesla):	1.35972
Rotor-Yoke Flux Density (Tesla):	1.5634
Air-Gap Flux Density (Tesla):	0.773245
Stator-Teeth Ampere Turns (A.T):	25.3688
Rotor-Teeth Ampere Turns (A.T):	159.308
Stator-Yoke Ampere Turns (A.T):	18.2615
Rotor-Yoke Ampere Turns (A.T):	11.7197
Air-Gap Ampere Turns (A.T):	208.366
Correction Factor for Magnetic Circuit Length of Stator Yoke:	0.535154
Correction Factor for Magnetic Circuit Length of Rotor Yoke:	0.250131
Saturation Factor for Teeth:	1.88631
Saturation Factor for Teeth & Yoke:	2.0302
Induced-Voltage Factor:	0.94203
Stator Current Density (A/mm ²):	5.72041
Specific Electric Loading (A/mm):	23.2998
Stator Thermal Load (A ² /mm ³):	133.284
Rotor Bar Current Density (A/mm ²):	2.34797
Rotor Ring Current Density (A/mm ²):	11.9081
Half-Turn Length of Stator Winding (mm):	268.906

WINDING ARRANGEMENT

The 3-phase, 1-layer winding can be arranged in 18 slots as below:

AAAZZZBBBBXXXXCCCCYYYY

Average coil pitch is:	7.66667
------------------------	---------

Angle per slot (elec. degrees):	20
---------------------------------	----

Phase-A axis (elec. degrees):	110
-------------------------------	-----

First slot center (elec. degrees):	0
------------------------------------	---

[A3]. Condition Monitoring Using ThinkSpeak Code

```
#include <WiFi.h>
#include <ThingSpeak.h>

// Wi-Fi and ThingSpeak Configuration
const char* ssid = "Mchinary-Lab"; // Your Wi-Fi SSID
const char* password = "1234doeee"; // Your Wi-Fi Password
WiFiClient client;

unsigned long Channel_ID = 2507153; // ThingSpeak Channel ID
const char* API_key = "QUP5EGV9AXSRXQXE"; // ThingSpeak Write API Key

// Timing and Delay
const unsigned long DELAY_INTERVAL = 10000; // 10 seconds
const unsigned long TEMPERATURE_SAMPLE_INTERVAL = 1000; // 1 second

// Temperature Sensor Pins
int tempSensorPins[] = {34, 36, 39};

// Current Sensor Pins
int currentSensorPins[] = {32, 33, 35};

long lastSample[3] = {0, 0, 0};
long sampleSum[3] = {0, 0, 0};
long sampleCount[3] = {0, 0, 0};

const float vpc = (3.3 / 4095) * 1000;

// Sensor Data Storage
float temperatureData[3];
float currentData[3];

void setup() {
    Serial.begin(115200);
    WiFi.mode(WIFI_STA);
    ThingSpeak.begin(client);

    for (int i = 0; i < 3; i++) {
        pinMode(tempSensorPins[i], INPUT_PULLDOWN);
        pinMode(currentSensorPins[i], INPUT_PULLDOWN);
    }

    xTaskCreatePinnedToCore(readTemperatureSensor1, "ReadTemperatureSensor1", 2048, NULL, 1, NULL, 1);
    xTaskCreatePinnedToCore(readTemperatureSensor2, "ReadTemperatureSensor2", 2048, NULL, 1, NULL, 1);
    xTaskCreatePinnedToCore(readTemperatureSensor3, "ReadTemperatureSensor3", 2048, NULL, 1, NULL, 1);

    xTaskCreatePinnedToCore(readCurrentSensor1, "ReadCurrentSensor1", 2048, NULL, 1, NULL, 1);
    xTaskCreatePinnedToCore(readCurrentSensor2, "ReadCurrentSensor2", 2048, NULL, 1, NULL, 1);
    xTaskCreatePinnedToCore(readCurrentSensor3, "ReadCurrentSensor3", 2048, NULL, 1, NULL, 1);

    xTaskCreatePinnedToCore(uploadData, "UploadData", 4096, NULL, 1, NULL, 1);
}

void loop() {
    // Nothing here, everything is handled by FreeRTOS tasks
}

void readTemperatureSensor1(void * parameter) {
    for (;;) {
        temperatureData[0] = readTemperature(tempSensorPins[0]);
        Serial.print("Temperature sensor 1 - Average: ");
        Serial.println(temperatureData[0]);
        vTaskDelay(TEMPERATURE_SAMPLE_INTERVAL / portTICK_PERIOD_MS);
    }
}

void readTemperatureSensor2(void * parameter) {
    for (;;) {
        temperatureData[1] = readTemperature(tempSensorPins[1]);
        temperatureData[1] = temperatureData[1] - 3;
        Serial.print("Temperature sensor 2 - Average: ");
        Serial.println(temperatureData[1]);
        vTaskDelay(TEMPERATURE_SAMPLE_INTERVAL / portTICK_PERIOD_MS);
    }
}
```

```

        }

void readTemperatureSensor3(void * parameter) {
    for (;;) {
        temperatureData[2] = readTemperature(tempSensorPins[2]);
        Serial.print("Temperature sensor 3 - Average: ");
        Serial.println(temperatureData[2]);
        vTaskDelay(TEMPERATURE_SAMPLE_INTERVAL / portTICK_PERIOD_MS);
    }
}

void readCurrentSensor1(void * parameter) {
    int index = 0;
    for (;;) {
        if (millis() - lastSample[index] > 2) {
            float s = analogRead(currentSensorPins[index]) - 2990;
            float y = s * vpc;
            sampleSum[index] += y * y;
            sampleCount[index]++;
            lastSample[index] = millis();
        }
        if (sampleCount[index] == 1000) {
            calculateCurrent(index);
        }
        vTaskDelay(2 / portTICK_PERIOD_MS);
    }
}

void readCurrentSensor2(void * parameter) {
    int index = 1;
    for (;;) {
        if (millis() - lastSample[index] > 2) {
            float s = analogRead(currentSensorPins[index]) - 2990;
            float y = s * vpc;
            sampleSum[index] += y * y;
            sampleCount[index]++;
            lastSample[index] = millis();
        }
        if (sampleCount[index] == 1000) {
            calculateCurrent(index);
        }
        vTaskDelay(2 / portTICK_PERIOD_MS);
    }
}

void readCurrentSensor3(void * parameter) {
    int index = 2;
    for (;;) {
        if (millis() - lastSample[index] > 2) {
            float s = analogRead(currentSensorPins[index]) - 2990;
            float y = s * vpc;
            sampleSum[index] += y * y;
            sampleCount[index]++;
            lastSample[index] = millis();
        }
        if (sampleCount[index] == 1000) {
            calculateCurrent(index);
        }
        vTaskDelay(2 / portTICK_PERIOD_MS);
    }
}

void calculateCurrent(int index) {
    float mean = sampleSum[index] / sampleCount[index];
    float rms = sqrt(mean);
    float mV = rms / 27;
    // println(mv);
    float y1 = (-0.02515 * mV * mV) + (1.546 * mV) - 0.8228;
    if (y1 <= 1) {
        y1 = 0;
    } else {
        y1 = 1;
    }
    sampleSum[index] = 0;
}

```

```

sampleCount[index] = 0;
currentData[index] = y1;
}

void uploadData(void * parameter) {
    for (;;) {
        if (WiFi.status() != WL_CONNECTED) {
            unsigned long start = millis();
            while (WiFi.status() != WL_CONNECTED && (millis() - start) < 20000) {
                WiFi.begin(ssid, password);
                vTaskDelay(500 / portTICK_PERIOD_MS);
            }
            if (WiFi.status() == WL_CONNECTED) {
                Serial.println("Wi-Fi connected.");
            } else {
                Serial.println("Failed to connect to Wi-Fi.");
                vTaskDelay(10000 / portTICK_PERIOD_MS);
                continue;
            }
        }

        int maxRetries = 3;
        int retryCount = 0;
        int responseCode = -1;

        while (retryCount < maxRetries && responseCode != 200) {
            for (int i = 0; i < 3; i++) {
                ThingSpeak.setField(i + 1, temperatureData[i]);
                ThingSpeak.setField(i + 4, currentData[i]);
            }

            responseCode = ThingSpeak.writeFields(Channel_ID, API_key);

            if (responseCode == 200) {
                Serial.println("Data successfully sent to ThingSpeak.");
            } else {
                Serial.println("Failed to send data to ThingSpeak. Response code: " + String(responseCode));
                retryCount++;
                vTaskDelay(5000 / portTICK_PERIOD_MS);
            }
        }

        if (retryCount == maxRetries) {
            Serial.println("Failed to send data to ThingSpeak after multiple attempts.");
        }

        vTaskDelay(DELAY_INTERVAL / portTICK_PERIOD_MS);
    }
}

float readTemperature(int pin) {
    float totalTemp = 0;
    int numReadings = 300;

    for (int i = 0; i < numReadings; i++) {
        int rawData = analogRead(pin);
        float temp = 0.266 * rawData - 347;
        totalTemp += temp;
        delay(1); // Shorter delay to allow task switching
    }

    float avgTemp = totalTemp / numReadings;
    return avgTemp;
}

```

[A4]. SCT -013 Current Sensor with ESP-32 Code

```
float sensor = 33;
long lastsample = 0;
long Samplesum = 0;
long Samplecount = 0;
float s;
float vpc = (3.3 / 4095) * 1000;

void setup() {
    Serial.begin(115200);
    pinMode(sensor, INPUT);
}

void loop() {
    if (millis() > lastsample) {
        s = analogRead(sensor)-2990;

        float y = s * vpc;
        Samplesum += y * y;
        Samplecount++;
        lastsample = millis();
    }

    if (Samplecount == 1000) {
        float mean = Samplesum / Samplecount;
        float rms = sqrt(mean);
        float mV = rms / 30;

        float y1= (-0.02515 *mV*mV)+( 1.546 *mV)-0.8228 ;// data calibration
        if (y1 <= 0.2) {
            y1 = 0;
        } else {
            y1 -= 0.2;
        }
        Serial.println(y1,1);

        Samplesum = 0;
        Samplecount = 0;
    }
}
```