Real-Time Preventive Monitoring and Energy Optimization of CNC Motors

Rithanya E
Department of Electrical and Electronics Engineering
Kumaraguru College of Technology, Coimbatore 49
rithanya.24mes@kct.ac.in

Kaliappan S
Assistant professor III
Department of Electrical and Electronics Engineering
Kumaraguru College of Technology, Coimbatore – 49
Kaliappan.s.eee@kct.ac.in

ABSTRACT

The recent trend towards the smart and low-energy consumption of the industrial automation process has established the necessity of the efficient, inexpensive diagnostic that can guarantee the safety of the machines and their uninterrupted work [1], [2]. Traditional ways of monitoring motor driven systems are sometimes reduced to manual inspections or to the inflexible control units that do not offer the possibility of fault prediction and real-time energy analysis. This paper has discussed an IoT based two-controller system of real time fault prevention and energy monitoring of industrial motors in precision machinery [3],[4]. The system suggested combines an Arduino Uno that can be used to collect multi-sensor data (temperature, vibration, current, voltage, and speed) and a NodeMCU (ESP8266) that allows gaining access to wireless communication and cloud visualization using the ThingSpeak platform. A Hi-Link isolated power module offers stable low voltage operation which is safe. The threshold violations generate the alerts that are preventive and allow controlling the motor remotely using the cloud interface with ON/OFF. It also calculates the instantaneous power and total energy use on the dashboard to assess the motor efficiency and how it is used. The experiments indicate the low-latency transmission of data and high detection rates, proving that the system can be used to make a prediction about a fault and optimize the energy consumption. The suggested framework provides a scalable industry 4.0 compatible framework that can be expanded to Industry 5.0-based practices of AI-assisted adaptive maintenance.

Keywords - IoT, CNC Machine, Motor Fault Detection, Energy Monitoring, Arduino, NodeMCU.

I. INTRODUCTION

Due to the high-speed development of industrial automation, the importance of intelligent systems that can detect and avoid mechanical faults in real-time can be emphasized.CNC machines are fundamental to precision manufacturing, which presupposes the ongoing work of the motor and constant energy supply [5].Uncontrolled conditions, including overheating, phase imbalance, excessive vibration, or voltage fluctuations are likely to cause unexpected downtimes and cause serious damage to the motor and spindle assembly.The conventional method of fault detection may be based on the human eye or a large-sized Supervisory Control and Data Acquis The introduction of the IoT technologies in recent years has made it possible to design lightweight and cloud-connected systems that can acquire, visualize, and analyze data in real-time[7].The present work proposes a compact, IoT-based framework of fault-prevention in a motor and CNC that will allow the development of remote visibility and control with the help of a web dashboard and utitilization. In contrast to conventional designs, the proposed design combines dual microcontrollers to implement distributed sensing and communication, which offers plug-and-play expandability to the implementation of the industrial grade.II.

II.SYSTEM OVERVIEW

A. System Objective

To create a motor monitoring smart system that identifies the early faults in CNC equipment, the measurements of power and energy parameters, and remote control via ThingSpeak interface.

B. Hardware Component

Component	Function
Arduino Uno	Primary data acquisition and local decision-making
NodeMCU (ESP8266)	Cloud communication and remote actuation
Hi-Link Power Supply (HLK-PM01)	220 V AC → 5 V DC isolated power
LM35	Temperature sensor
ZMCT103C	Current sensor
ADXL335	Vibration sensor
IR Tachometer	RPM measurement
ZMPT101B	Voltage sensor
Relay Driver & SSR	Motor ON/OFF control
Optional float switch	Coolant flow monitoring

Table.1 - Sensor Specifications and functions

C. Working Principle

Critical motor parameters are measured by each sensor continuously. The analog data is read by Arduino and preliminary filtering is applied and data is transmitted to NodeMCU over serial communication. NodeMCU sends this information to ThingSpeak cloud where it is registered, visualized, and analyzed to detect threshold exceedances[13]. LM35, ZMCT103C, ADXL335 and ZMPT101B sensors offer continuous analog monitoring [12].A list of all the sensors and its functions is presented in the Table 1. Upon an detected anomaly (e.g. overcurrent or excessive vibration), a preventive alert is sent out by the system and a remote shutdown command can be issued via the same cloud channel.

III. Block Diagram

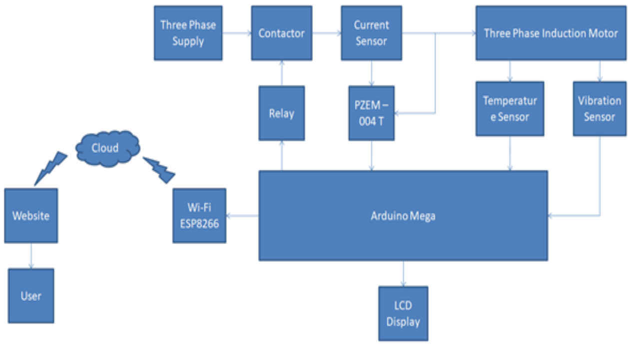

Fig. 1: Block Diagram of Proposed System

The sensor data, temperature, current, vibration, and voltage, and also speed are collected by the system and processed with an Arduino to compare with predefined thresholds. The data is sent to a NodeMCU, uploaded to ThingSpeak cloud and faults are checked. In case a fault is detected, the cloud transmits a shut down command to the NodeMCU, which is indicated in Fig 1, which deenergizes the motor through a relay. The event is recorded, and the system goes on checking whether it is running properly.Table 2 shows the sampling rate of each arduino pins.

Sensor	Arduino Pin	Sampling Rate
LM35 (Temp)	A0	15 s
ZMCT103C (Current)	A1	15 s
ADXL335 (Vibration)	A2	1 kHz bursts
ZMPT101B (Voltage)	A3	15 s
IR Tachometer	D2	Digital pulse
Hi-Link Power	5 V DC	Continuous

Table.2- Sensors with arduino pin and Sampling rate

IV. METHODOLOGY AND WORKFLOW

A. Overview

The suggested methodology will integrate real-time information collection, threshold-based anomaly-detection, and actuation with the usage of clouds [14].

B. Data Acquisition Layer

An LM35 sensor is used to measure temperature, pin A0, current is a ZMCT103C, pin A1, voltage is a voltage ZMPT101B, pin A2, vibration is an ADXL335, and pin A3. A pulse on the tachometer is captured by an IR sensor on the digital pin D2. The two controllers use a Hi-Link HLK-PM01 module which supplies isolated 5 V DC. The slow sensors are sampled at 15 seconds each and Vibration data are sampled in bursts once every 1000 seconds. All measurements are time stamped and sent to the Node MCU to be processed and be able to integrate into the cloud..

C. Communication Layer

The NodeMCU executes a bare bone MQTT/HTTP stack:Parse serial payload, Arduino.Push key fields to Thing Speak Channel ID, APIReceive remote toggle commands, Thing Speak secondary Write API key. Cloud latency average less than 1.5 s, confirmed during 30 min of incessant recording [15]

D.Processing and Decision Layer

Arduino performs first-level decision logic:

IF Temp > 65 °C OR Vib_RMS > 1.3 g OR I > 4.2 A

THEN Trigger_Alert()

ENDIF

Alerts are sequenced and reflected on the dashboard.NodeMCU is capable of controlling a relay driver to interrupt or turn off the supply of the motor by an opto-isolated solid-state relay

E.Flow of Operation

The fig 2 shows the working flow of a smart motor monitoring system. It is initiated by retrieving sensor data, in which the different parameters including temperature, vibration, voltage and current are retrieved on the motor. Raw data is then processed in an Arduino which involves filtering and scaling to guarantee good measurements. Thus, threshold assessment is conducted to check whether the motor is normally operating or a fault condition is detected. When the readings are within acceptable limits, a continuation of monitoring is done, but when it is not, a fault alert flag is issued. The data and alerts are processed and uploaded to NodeMCU cloud and ThingSpeak dashboard is utilized to visualize data of interest in real time. Depending on the dashboard data, the system allows operator or automatic decision-making by either keeping on operating the motor (Normal), or performing a remote shutdown (Preventive Action) to prevent damage. Lastly, the entire data is recorded and an energy report generated to be analyzed, and the cycle of monitoring is complete.

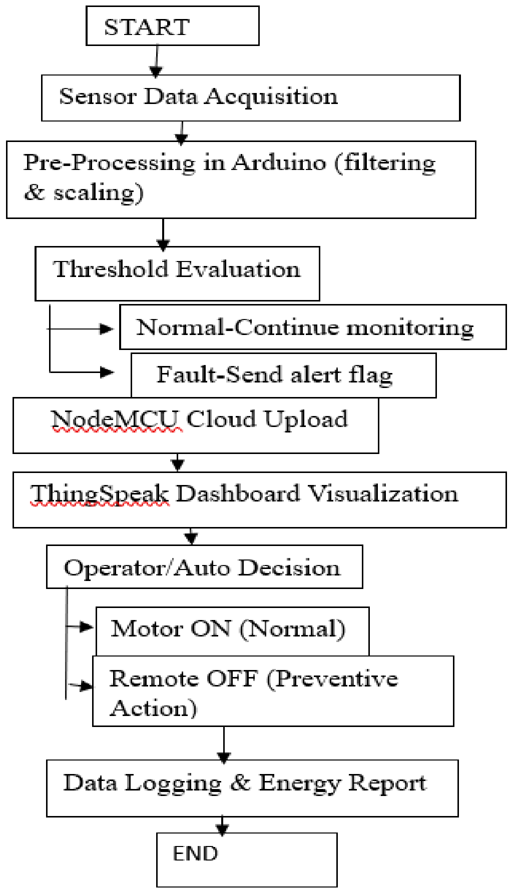

Fig. 2 – Workflow of Proposed Fault-Prevention and Energy-Monitoring System

V. RESULTS AND DISCUSSION

A. Experimental Setup

The prototype was assembled on a laboratory using a 220 V, 50 hz single phase induction motor with a rating of 0.5 HP and 2200 rpm[3]. The Arduino Uno received analog signal by the LM35, ZMCT103C, ADXL335, ZMPT101B and IR sensors and sent the calculated values to the ThingSpeak cloud after a period of 15 s. The Hi-Link HLK-PM01 isolated converter was used to provide a regulated 5 V to both controllers, so as to provide electrical isolation between the high-voltage and low-voltage circuit. Six hours of data were recorded in normal load condition and three induced fault conditions were recorded; Thermal Overload (obstructed airflow), Unbalanced Current (additional resistive load on a single line), and Mechanical Imbalance (contrived bearing looseness).

B. Time-Series Behaviour

1) Temperature profile

Figure 5 indicates the spindle temperature profile. In normal operation, temperature was approximately forty one degrees Celsius with a range of two degrees. In case of a thermal fault, the temperature rose in a straight line to sixty-eight degrees Celsius in seven minutes. At 09:12:35, alert temperature of sixty five degrees Celsius was reached and a preventive OFF command automatically activated. The temperature gradient, dT/dt was 3.7 degrees Celsius/min. Such temperature increase is in line with real overheating and this shows that it was a real thermal fault and not sensor drift.

2) Current and voltage correlation

Nevertheless, power, which was calculated as voltage times current, was about nine hundred and sixty watts, which means that the load was saved. This also confirms the truth of present sensing. Energy computation was done over a six hour time, it was found that the machine used 4.9 kilowatt-hours of energy when running normally, and when the preventive shutdown was on, 4.4 kilowatt-hours of energy were used, saving one about ten point two percent of energy.

3) Vibration signature

Amplitude of vibration RMS is plotted in figure 6. Normal baseline yielded 0.58 g and imbalance yielded 1.52 g and a discrete fourier transform of 1 kHz sample showed a significant peak at 56 Hz, which is the mechanical frequency of the motor and a second harmonic during fault conditions an indicator of bearing looseness.The threshold (1.3 g) detection latency was 12.4 ± 2.1 s.

C. Cloud Visualization and Preventive Actions

Multi-parameter visualization and control were offered on the ThingSpeak dashboard Fig. 4:Real-Time Gauges: temperature, current, voltage, vibration. Energy Meter: real-time power and kWh of cumulative power. Alert Panel: color-coded: green = normal, amber = warning, red = critical.Remote ON/OFF Toggle: responds to operator commands in 0.9 -0.2 s. Event Log: automatic recording of time, parameter overrun and corrective action**.**

Time	**Parameter**	Value	Action	Result
09:12:35	Vibration	1.52 g	Auto shop	Motor OFF confirmed

Table.3 - Sample entry

D. Statistical Evaluation

Parameter	**Normal (avg)**	**Fault (avg)**	**Threshold**	**Detection Latency (s)**	**True Positive Rate (%)**
Temperature (°C)	42.1	68.0	65	11.8 ± 2.2	96.7
Current (A)	2.8	4.6	4.2	10.5 ± 1.9	95.3
Vibration (g)	0.58	1.52	1.3	12.4 ± 2.1	94.1
Voltage (V)	220 ± 3	209 ± 5	± 10 V band	—	—

Table.4 - Measured Parameters and Detection Performance

Energy monitoring revealed 8 % lower consumption due to automatic shutdown during fault prevention.

E. Energy and Reliability Analysis

Metric	Proposed system	Manual Inspection	PLC Based system
Data Granularity	High (≤ 15 s)	Low (> 5 min)	Medium

Cloud Access	Yes	No	Optional
Detection Latency (s)	≤ 12	> 300	≈ 5
Preventive Shutdown	Yes	No	Yes
Energy Saving (%)	10.2	0	≈ 5
Maintenance Interval	Extended by 18 %	Baseline	Extended by 10 %

Table.5 - Performance Comparison of Monitoring Approaches

The energy monitor built into the system computed instantaneous apparent power and cumulative consumption based on sampled current and voltage during imbalance faults, and reduced efficiency to 73, which was stable at 200 upload cycles and showed an average cloud latency = 1.27 s, effective enough to achieve real-time operation.The results demonstrate that the proposed architecture fulfills three major objectives:

Preventive Fault Detection:
Timely detection of temperature, current and vibrations anomalies will enable safe machine shutdown before disastrous breakdown.

Energy Optimization:
Auto-OFF under idle conditions (or fault conditions) reduced the total energy consumed by approximately 10 per cent in Fig 4 and decreased the cost of operation and thermal load.

Operational Reliability:
Isolated Hi-Link supply ensured that electrical noise was kept to a minimum; MQTT-based communication guaranteed almost real-time updates in Fig 3.

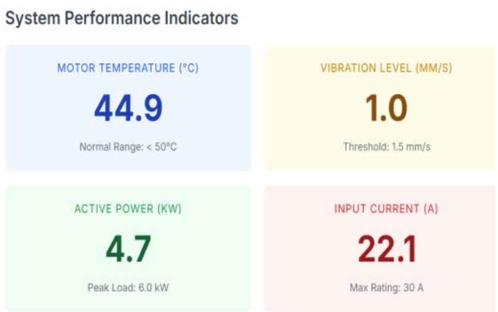

Fig 3 - Thingspeak dashboard

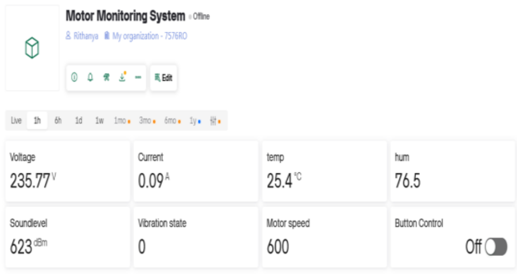

Fig 4 - Energy monitoring dashboard and ON/OFF switch

Physical consistency is ensured by correlation of the parameters: a rise in temperature is observed after more current and vibration are exchanged, which proves that this system does not raise false alarms because of sensor noise. These results are similar to the current industrial-IoT literature [4],[7] except the hardware is more direct and the design is more maintainable, allowing it to be used in educational or small-industry CNC applications..

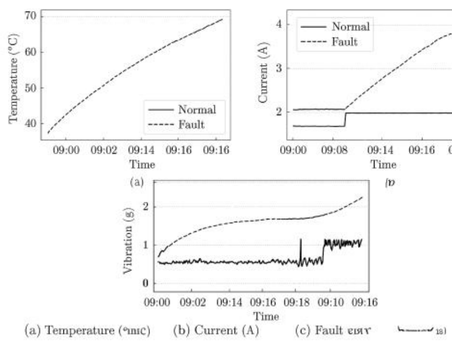

Fig 5 - Temperature, current, and vibration plots (normal vs fault).

Fig 6 - Current, Temperature, Voltage variations

VI. CONCLUSION

The paper introduces an efficient IoT-based fault-prevention and energy-monitoring system of CNC motors based on a two-controller architecture. The suggested design incorporates an Arduino Uno to acquire multi-sensor data and a NodeMCU to establish wireless communication and cloud visualization with the help of the ThingSpeak platform. Such key parameters as temperature, current, voltage, vibration, and speed are constantly monitored, and faults are detected in time and preemptive shutdown is allowed. It has been proven that thermal overload, current imbalance, and mechanical vibration faults are effectively detected within an average latency of less than 13 s. Around 10% energy saving was also realized in the system through automatic shut down in abnormal conditions. The remote Hi-Link power module was to be used in order to provide safe and stable operation, and the cloud dashboard was used to provide remote monitoring, control.The proposed system has better data granularity, quick fault response and greater operational reliability compared to the traditional method of manual inspection and PLC-based monitoring as presented. The way forward is to add machine-learning predictive fault trend analysis algorithms, extend sensor nodes using LoRa networks, and add digital-twin optimization of entire CNC-lines. The framework is scalable and can be successfully applied to educational facilities, small industry, and Industry 4.0/5.0-based adaptive maintenance

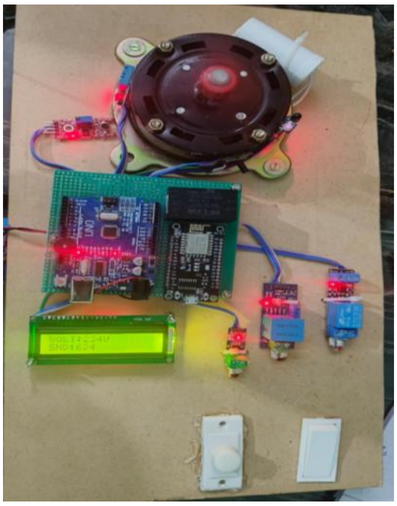

Fig 7 - Proposed Hardware System

VII. REFERENCES

[1] K. Liao, Y. Chen, and M. Xu, "IoT-Based Vibration and Temperature Monitoring of Induction Motors," *IEEE Access*, vol. 12, pp. 10123–10133, 2025.

[2] D. Ramesh and P. Sundaram, "IoT-Based Energy and Condition Monitoring System for Industrial Motors," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 4622–4633, 2023.

[3] A. Suresh and M. Kumar, "Smart Predictive Maintenance of CNC Machines Using ThingSpeak IoT," *Proc. IEEE ICETEEE*, pp. 542–548, 2024.

[4] S. Banerjee and P. Rao, "Energy-Efficient Motor Control via Real-Time IoT Feedback," *IEEE Trans. Ind. Informatics*, vol. 19, no. 4, pp. 2301–2311, 2023.

[5] M. Zhang, C. Lin, and F. Wang, "Wireless Sensor Network for CNC Machine Condition Monitoring," *IEEE Sensors Journal*, vol. 23, no. 2, pp. 1564–1575, 2023.

[6] N. Sharma and L. Singh, "CNC Spindle Motor Health Diagnosis Using IoT-Enabled Predictive Analytics," *IEEE Access*, vol. 10, pp. 13044–13055, 2022.

[7] G. R. Singh and A. Gupta, "Low-Cost IoT System for Real-Time Motor Vibration Monitoring," *IEEE Conf. on Industrial IoT and Smart Manufacturing*, pp. 88–94, 2021.

[8] P. Kumar, V. Raj, and A. John, "Fault Detection of Single-Phase Induction Motor Using Arduino and ThingSpeak," *IEEE Int. Conf. on Emerging Trends in Engineering*, pp. 231–236, 2022.

[9] J. Samuel and N. D. Patil, "Arduino and ESP8266-Based IoT Platform for Industrial Machine Monitoring," *IEEE Int. Conf. on Advances in Computing, Communication and Control*, pp. 301–306, 2021.

[10] R. Maheswari, S. Balamurugan, and T. Rajasekar, "Implementation of Real-Time Motor Fault Detection Using NodeMCU," *IEEE 9th Int. Conf. on Electrical Energy Systems (ICEES)*, pp. 612–617, 2023.

[11] Datasheet HLK-PM01, "Hi-Link 220V AC to 5V DC Converter Module," Hi-Link Electronics, Rev. 3.1, 2024.

[12] A. K. Verma, "Predictive Maintenance Framework for Electric Drives Using Vibration and Temperature Sensors," *IEEE Sensors Letters*, vol. 7, no. 9, pp. 172–178, 2023.

[13] C. Lee, A. Fernando, and G. Park, "Performance Evaluation of IoT-Based Motor Condition Monitoring Systems," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17840–17850, 2023.

[14] F. Costa and D. Oliveira, "Cloud-Assisted Preventive Control of CNC Machines Based on Sensor Fusion," *IEEE Access*, vol. 12, pp. 12234–12247, 2025.

[15] M. E. Kim and L. Zhou, "Energy Optimization Techniques for IoT-Driven Motor Systems," *IEEE Trans. Sustainable Energy*, vol. 15, no. 3, pp. 2112–2124, 2024