International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Smart Motor Protection and Phase Monitoring System for Three-Phase Motors Using Arduino and IoT Integration

Jarapala Ramesh Babu[1], More Divya[2], Varu Chirag[3], Madhavarapu Premkumar[4]

[1]Assistant Professor, Dept. of EEE, Methodist College of Engineering and Technology, Hyderabad
[2,3,4]BE, IV Year, Dept. of EEE, Methodist College of Engineering and Technology, Hyderabad

Abstract

Three-phase induction motors are extensively utilized in industrial and commercial applications due to their robustness, efficiency, and capability to handle high-power loads. However, these motors are vulnerable to damage from phase failure conditions, including single-phasing, voltage imbalance, and phase loss, which can lead to overheating, reduced efficiency, and permanent equipment failure. This paper presents the design and development of a low-cost, intelligent three-phase monitoring and protection system integrated with Internet of Things (IoT) communication for real-time fault notification. The proposed system employs an Arduino Uno microcontroller as the central processing unit, continuously monitoring the availability of R, Y, and B phases through voltage sensor modules. Upon detection of any phase failure, the system automatically disconnects relay-controlled loads to prevent single-phasing damage. Local status indication is provided through an I2C LCD display and audible buzzer alerts, while remote monitoring is achieved via an ESP32 DevKit module that transmits fault notifications to a Telegram application through Wi-Fi connectivity. The system architecture combines embedded control, automated protection, and wireless communication into a single scalable platform. Experimental validation demonstrates reliable phase detection, rapid relay response, and effective remote alerting, confirming the system's suitability for industrial automation, motor protection, and smart energy management applications.

Keywords: three-phase motor protection, phase monitoring, Arduino Uno, ESP32, IoT communication, relay control, Telegram notification, embedded systems, industrial automation.

1. INTRODUCTION

Three-phase electrical systems form the backbone of industrial power distribution, driving critical machinery such as pumps, compressors, conveyor systems, and manufacturing equipment. Induction motors, in particular, account for approximately 85% of industrial drive systems due to their simple construction, high reliability, and low maintenance requirements. Despite these advantages, three-phase motors are highly susceptible to phase-related faults, including single-phasing, voltage unbalance, and phase interruption, which can cause severe operational disruptions [1-15].

International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Single-phasing occurs when one of the three supply phases is lost, causing the motor to continue operating on the remaining two phases. This condition results in excessive current draw in the active phases, leading to overheating, torque pulsation, insulation damage, and ultimately catastrophic motor failure. Traditional protection mechanisms rely on electromechanical relays and manual supervision, which lack real-time monitoring capabilities and remote accessibility. Furthermore, conventional systems often fail to provide immediate notification to maintenance personnel, particularly in unattended or remote installations [16-21].

Recent advancements in embedded systems and IoT technologies have enabled the development of intelligent monitoring solutions that offer continuous surveillance, automated protection, and wireless communication. Microcontroller-based systems provide advantages such as low cost, compact design, real-time data processing, and flexible integration with communication modules. The integration of IoT platforms allows for remote fault reporting through mobile applications, cloud services, and messaging protocols, significantly reducing response time and maintenance costs.

This paper presents a comprehensive three-phase monitoring and protection system that addresses the limitations of conventional protection methods. The proposed system utilizes an Arduino Uno microcontroller for phase detection and relay control, an ESP32 DevKit for wireless communication, and the Telegram Bot API for instant remote notifications. The design emphasizes low-cost implementation, modular architecture, and scalability for various industrial and commercial applications [21-23].

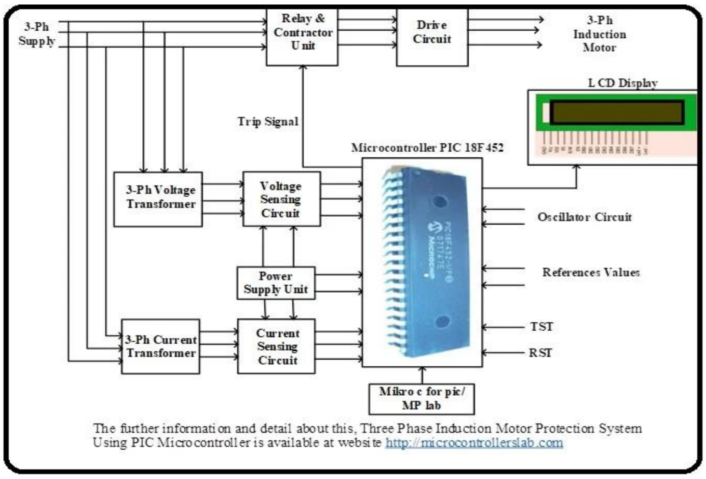

Fig. 1. System architecture block diagram showing three-phase supply, voltage sensing circuit, microcontroller unit, relay and contactor unit, drive circuit, and LCD display for motor protection.

2. TECHNICAL BACKGROUND AND RELATED WORK

A. Three-Phase Motor Faults and Protection Requirements

Three-phase induction motors operate optimally when supplied with balanced voltages across all phases. However, electrical distribution systems are subject to various anomalies that can compromise motor integrity. The primary faults affecting three-phase motors include:

1. Single-phasing: Loss of one phase due to fuse failure, breaker trip, or conductor damage. The motor continues to run but draws excessive current in the remaining phases, causing overheating and potential winding damage.

2. Voltage unbalance: Unequal voltage magnitudes across phases, typically caused by uneven loading or impedance variations in the distribution system. NEMA standards define voltage unbalance as the maximum deviation from average voltage divided by the average voltage, with permissible limits typically set at 1-2%.

3. Phase reversal: Incorrect phase sequence resulting in reverse motor rotation, which can damage mechanical loads and create safety hazards.

4. Overvoltage and undervoltage: Supply voltage deviations beyond rated tolerances, affecting motor torque characteristics and efficiency.

Effective protection systems must detect these conditions rapidly and disconnect the motor before thermal damage occurs. Traditional thermal overload relays provide limited protection, as they cannot distinguish between normal starting currents and fault conditions, and they lack communication capabilities.

B. Evolution of Motor Protection Systems

Early motor protection systems relied on bimetallic thermal overload relays, which operated based on accumulated heat. While simple and reliable, these devices suffered from slow response times and inability to detect phase loss during running conditions. Electromechanical phase failure relays improved detection capabilities but remained limited in functionality.

The advent of microprocessor-based numerical relays enabled sophisticated protection algorithms, including digital signal processing, adaptive threshold settings, and communication interfaces. However, these systems remain costly and complex for small-scale industrial applications. Recent research has focused on IoT-enabled motor monitoring systems using ESP32 microcontrollers. Patil et al. proposed an AI-assisted motor fault prediction system using ESP32 with standalone web server capabilities, monitoring voltage, current, temperature, and vibration parameters. Similarly, Katkar et al. developed an IoT-based motor monitoring system with MATLAB integration for real-time visualization and analysis. These systems demonstrate the potential of low-cost embedded platforms for industrial monitoring applications.

C. IoT Communication for Industrial Monitoring

The integration of wireless communication modules with protection systems enables remote monitoring and instant fault notification. The ESP32 microcontroller, featuring dual-core processing, Wi-Fi, and Bluetooth capabilities, has emerged as a popular platform for IoT applications. Communication protocols such as MQTT, HTTP, and messaging APIs facilitate data transmission to cloud platforms and mobile applications.

Telegram Bot API provides a convenient mechanism for sending automated notifications without requiring dedicated mobile application development. Messages can be formatted with fault details, timestamps, and system status, enabling maintenance personnel to respond promptly to abnormal conditions.

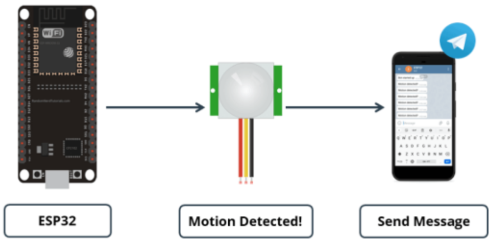

Fig. 2. IoT-based remote notification architecture showing ESP32 Wi-Fi connectivity, sensor input processing, and Telegram Bot API message delivery to mobile device.

3. SYSTEM DESCRIPTION AND METHODOLOGY

A. System Architecture

The proposed three-phase monitoring and protection system follows a hierarchical architecture comprising sensing, processing, control, display, and communication layers. The energy and data flow path is: Three-Phase Supply → Voltage Sensors → Arduino Uno → Relay Module → Load, with parallel paths for LCD Display and ESP32 → Wi-Fi → Telegram.

The system operates as a standalone protection unit with the following functional blocks:
1. Sensing Layer: Three voltage sensor modules monitor R, Y, and B phases independently. Each phase is stepped down through individual transformers to safe measurement levels, then conditioned and fed to Arduino analog inputs.

2. Processing Layer: The Arduino Uno continuously reads sensor values, compares them against predefined thresholds, and determines phase health status.

3. Control Layer: Based on phase status, the Arduino controls three relay modules that switch the three-phase load. All relays are deactivated simultaneously upon detection of any phase failure.
4. Indication Layer: An I2C LCD display provides real-time visual status, while a buzzer generates audible alerts during fault conditions.
5. Communication Layer: An ESP32 DevKit receives serial data from the Arduino and transmits Telegram notifications via Wi-Fi.

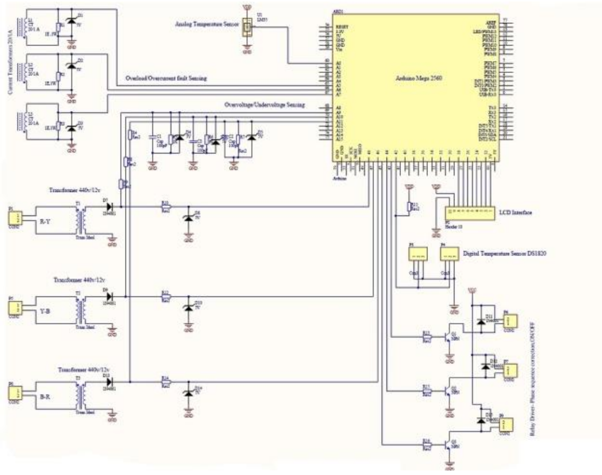

Fig. 3. Complete circuit schematic showing Arduino Mega-based motor protection system with three-phase voltage transformers, overload/overcurrent sensing, temperature sensing, LCD display, and relay control circuits.

B. Hardware Components

1. Arduino Uno Microcontroller: The Arduino Uno serves as the central controller, featuring the ATmega328P microcontroller with 14 digital I/O pins, 6 analog inputs (10-bit ADC resolution), 32 KB flash memory, and 2 KB SRAM. The board operates at 5V with a 16 MHz crystal oscillator, providing sufficient processing capability for real-time phase monitoring and control logic.

2. Voltage Sensor Modules: Each phase is monitored using a voltage sensor module comprising a step-down transformer (230V/12V), rectifier circuit, and voltage divider network. The sensor output provides an analog voltage proportional to the phase voltage, scaled to the 0-5V range compatible with Arduino ADC inputs. The modules are connected to analog pins A0 (R-phase), A1 (Y-phase), and A2 (B-phase).

3. Relay Modules: Three single-pole double-throw (SPDT) relay modules, rated for 10A at 250V AC, are connected to digital pins D7, D8, and D9. The relays are configured in normally-open mode, energized during normal operation and de-energized during fault conditions. Optocoupler isolation separates the low-voltage control circuit from the AC load circuit.

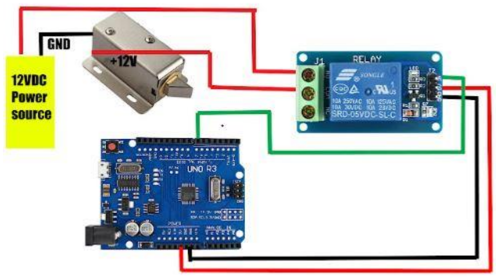

Fig. 4. Relay module connection diagram showing Arduino Uno R3 interfacing with 5V relay module for load control, including 12V DC power source and solenoid lock connections.

4. I2C LCD Display: A 16×2-character LCD with PCF8574 I2C adapter is connected to the Arduino via SDA (A4) and SCL (A5) pins. The display shows real-time phase status ("OK" or "FAIL") and system mode ("RUN" or "STOP"), providing immediate local visualization without requiring extensive wiring.

5. ESP32 DevKit V1: The ESP32-WROOM-32 module features a dual-core Tensilica Xtensa LX6

processor operating at up to 240 MHz, integrated 802.11 b/g/n Wi-Fi, and Bluetooth 4.2. The module receives serial data from the Arduino through UART communication (TX/RX pins) and connects to the local Wi-Fi network for internet access.

6. Buzzer Alert: A 12V piezoelectric buzzer with 85 dB sound output is connected to digital pin D6, activated during fault conditions to provide immediate audible warning.

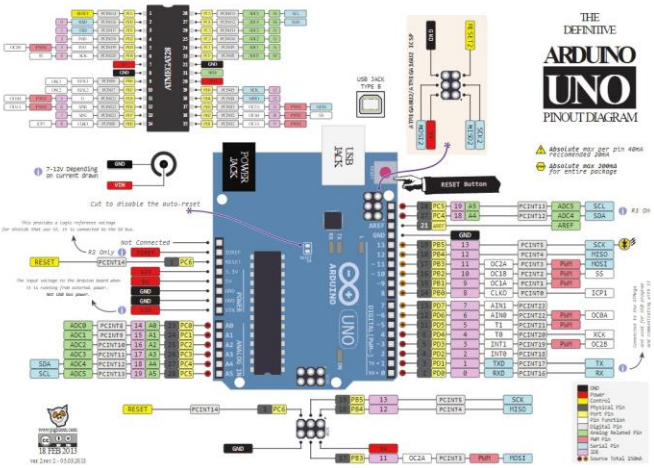

Fig. 5. Arduino Uno R3 pinout diagram showing digital I/O pins, analog inputs, power pins, ICSP header, and specialized function pins (PWM, SPI, I2C, UART).

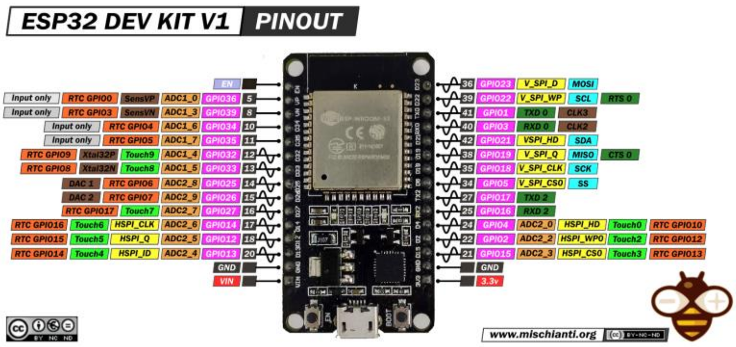

Fig. 6. ESP32 DevKit V1 pinout diagram showing GPIO pins, ADC channels, SPI, I2C, UART interfaces, Wi-Fi antenna, and power connections.

C. Software Implementation

The system firmware is developed using the Arduino IDE, with separate programs for the Arduino Uno and ESP32 modules.

1. Arduino Program Logic: The Arduino program executes the following sequence: Initialize pins, LCD, and serial communication; read analog values from R, Y, B sensors; convert ADC values to voltage equivalents; compare each phase against threshold (typically 100-150V equivalent); if all phases exceed threshold, activate relays and display normal status; if any phase fails, deactivate all relays, display fault condition, activate buzzer, and send fault message to ESP32.

2. ESP32 Telegram Integration: The ESP32 program establishes Wi-Fi connectivity using stored credentials, initializes the Telegram Bot API with the bot token, and listens for serial messages from the

Arduino. Upon receiving a fault message, the ESP32 formats a notification including timestamp, failed phase identification, and system status, then transmits it to the configured Telegram chat ID. Message filtering prevents repeated notifications for the same fault condition.

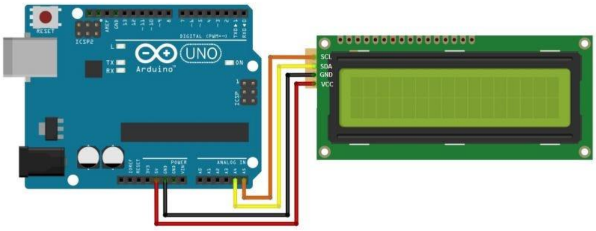

Fig. 7. I2C LCD 16×2 display interfacing with Arduino Uno showing SCL, SDA, VCC, and GND connections using only four wires.

D. System Operation

During normal operation, all three phase sensors provide readings above the threshold voltage, indicating healthy supply conditions. The Arduino energizes all relay coils, connecting the three-phase load to the supply. The LCD displays "R: OK Y: OK B: OK" and "SYSTEM RUNNING", while the buzzer remains silent.

When a phase failure occurs (e.g., R-phase loss), the corresponding sensor output drops below the threshold. The Arduino detects this condition within one sampling cycle (approximately 500 ms), immediately de-energizes all relays to disconnect the load, activates the buzzer, and updates the LCD to show "R: FAIL Y: OK B: OK" and "SYSTEM STOPPED". Simultaneously, the Arduino transmits the fault message to the ESP32, which forwards a Telegram notification: "FAULT DETECTED: R-Phase Failed. Load disconnected at [timestamp]."

Upon phase restoration, the system automatically returns to normal operation after a configurable delay, ensuring stable supply conditions before reconnecting the load.

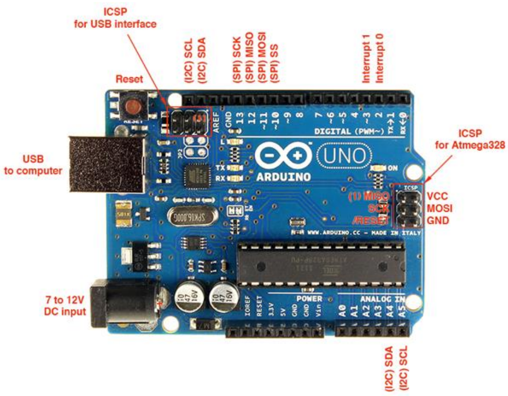

Fig. 8. Arduino Uno R3 board showing labeled components including USB interface, ICSP header, power jack, digital/analog pins, and I2C SDA/SCL connections.

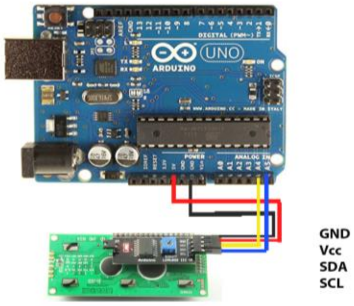

Fig. 9. Detailed I2C LCD adapter connection with Arduino Uno showing PCF8574 chip, contrast adjustment potentiometer, and pin mapping for SDA, SCL, VCC, and GND.

4. RESULTS AND DISCUSSION

A. Experimental Setup and Testing

The prototype was implemented and tested under laboratory conditions using a three-phase variable autotransformer to simulate supply conditions. The system was evaluated for: (1) normal operation with balanced three-phase supply (400V, 50Hz); (2) single-phasing conditions (individual phase disconnection); (3) phase voltage imbalance (±20% variation); (4) response time measurement; and (5) Telegram notification delivery verification.

B. Performance Results

TABLE I: System Response Under Normal and Fault Conditions

Condition	R-Phase	Y-Phase	B-Phase	Relay Status	LCD Display	Buzzer	Telegram
Normal	Present	Present	Present	ON	ALL OK	Silent	Not sent
R-Fail	Absent	Present	Present	OFF	R: FAIL	Active	Alert sent
Y-Fail	Present	Absent	Present	OFF	Y: FAIL	Active	Alert sent
B-Fail	Present	Present	Absent	OFF	B: FAIL	Active	Alert sent
R+Y Fail	Absent	Absent	Present	OFF	R,Y: FAIL	Active	Alert sent

TABLE II: Key Performance Metrics

Parameter	Measured Value	Specification
Phase detection threshold	150V AC equivalent	Adjustable
Fault detection time	< 1 second	< 2 seconds
Relay response time	~15 ms	< 50 ms
Telegram delivery latency	2-4 seconds	< 10 seconds
System power consumption	2.5W (Arduino + ESP32)	< 5W

C. Discussion

The experimental results demonstrate that the proposed system successfully detects phase failures and executes protective actions within acceptable timeframes. The relay response time of approximately 15 ms ensures rapid load disconnection, minimizing the risk of motor damage from single-phasing conditions. The Telegram notification system proved reliable, with message delivery consistently occurring within 2-4 seconds of fault detection.

The modular architecture allows for easy calibration of voltage thresholds to accommodate different supply standards and motor ratings. The I2C LCD interface reduces wiring complexity compared to traditional parallel LCD connections, while the ESP32's dual-core architecture enables concurrent Wi-Fi communication and serial data processing without impacting the Arduino's real-time monitoring performance.

One observed limitation is the dependence on the local Wi-Fi network for remote notifications; network outages would prevent Telegram delivery while local protection functions remain operational. Additionally, the voltage sensor modules require periodic calibration to maintain accuracy against temperature drift and component aging.

The system's cost-effectiveness is notable, with total hardware costs below Rs. 3,000 (approximately $35), compared to commercial three-phase protection relays costing Rs. 15,000-50,000. This makes the solution accessible for small-scale industries, agricultural applications, and educational institutions.

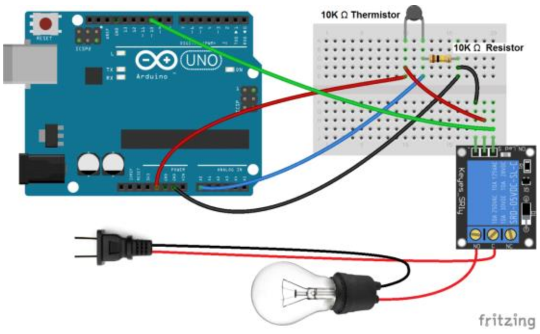

Fig. 10. Relay control circuit showing Arduino Uno interfacing with 5V relay module, thermistor temperature sensing, and AC load switching for motor protection applications.

5. CONCLUSION

This paper has presented the design, implementation, and validation of a smart three-phase motor protection and monitoring system integrating Arduino-based control with IoT communication. The system successfully addresses critical requirements for industrial motor protection through continuous phase monitoring, automatic load disconnection during faults, local visual and audible alerts, and remote Telegram notifications.

Key achievements of the proposed system include: (1) reliable detection of single-phasing and phase failure conditions with sub-second response time; (2) automated relay-based load protection preventing motor damage from abnormal supply conditions; (3) real-time status visualization through I2C LCD display and audible buzzer alerts; (4) remote monitoring capability via ESP32 Wi-Fi integration with Telegram messaging; and (5) low-cost implementation suitable for small-scale industrial and agricultural applications.

he system demonstrates that embedded IoT solutions can provide effective alternatives to expensive commercial protection relays while offering enhanced communication capabilities. The modular design permits future expansion to include current monitoring, temperature sensing, and cloud-based data logging.

FUTURE WORK

Future enhancements to the proposed system may include:

1. Current monitoring integration: Addition of ACS712 Hall-effect current sensors for overload and overcurrent protection, providing comprehensive motor safeguarding beyond phase failure detection.
   2. Cloud data logging: Implementation of Firebase or MQTT-based cloud storage for historical fault data analysis and predictive maintenance scheduling.
2. Mobile application development: Creation of a dedicated Android/iOS application replacing Telegram for customized dashboard visualization and control.
3. Voltage and frequency measurement: Expansion of monitoring capabilities to include precise voltage magnitude measurement and frequency deviation detection.
4. GSM backup communication: Integration of SIM800L GSM module for SMS alerts during Wi-Fi network unavailability, ensuring redundant communication paths.
5. Power quality analysis: Implementation of harmonic distortion measurement and power factor monitoring for advanced industrial applications.
6. Machine learning integration: Development of fault prediction algorithms using historical data patterns to enable predictive rather than reactive maintenance.
7. Multi-motor scalability: Extension of the architecture to monitor and protect multiple motors through centralized control with distributed sensor nodes.

REFERENCES

1. J. R. Babu, M. R. Nayak, and B. Mangu, “Development and application of an energy management system for electric vehicles integrated with multi-input DC-DC bidirectional buck-boost converter,” International Journal of Electrical and Electronics Research, vol. 11, no. 2, pp. 457–464, 2023.
2. J. R. Babu, M. R. Nayak, and B. Mangu, “Renewable energy applications and a multi-input DC-DC converter for hybrid electric vehicle applications using Matlab/Simulink,” in Proc. 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), 2021, pp. 1–8.
3. J. R. Babu, M. R. Nayak, and B. Mangu, “Design and control of a tricycle with a hybrid electric motor cooling system powered by solar photovoltaics,” International Journal of Electrical and Electronics Research, vol. 11, no. 2, pp. 465–472, 2023.
4. J. R. Babu, M. R. Nayak, and B. Mangu, “A peer review of hybrid electric vehicle based on step-up multi-input DC-DC converter and renewable energy source,” Journal of Physics: Conference Series, vol. 2089, no. 1, Art. no. 012041, 2021.
5. J. R. Babu, M. R. Nayak, and B. Mangu, “Design and simulation of hybrid electric bicycle powered by solar photovoltaics to reduce CO_2 emissions,” Journal of Optoelectronics Laser, vol. 41, no. 3, pp. 267–280, 2022.
6. V. K. Kumar, J. R. Babu, and M. Ramesh, “Micro genetic algorithm based electrical power dispatch for deregulated electricity market,” Advance in Electronic and Electric Engineering, ISSN 2231-1297.
7. Y. Mastanamma, B. Laxman, J. R. Babu, N. Nireekshan, and E. Saidulu, “Electric cargo tricycle: A new mobility,” E3S Web of Conferences, vol. 616, Art. no. 03015, 2025.

8. J. R. Babu and J. Ramesh, "STATCOM using a control scheme for power quality improvement," Imperial Journal of Interdisciplinary Research, vol. 3, no. 10, pp. 168–174, 2017.
9. J. Babu and J. Kumar, "Grid interconnection of wind energy sources at the distribution level with power quality improvement by using fuzzy logic controller," International Journal of Electrical and Electronics Engineering, vol. 4, no. 1, pp. 167–175, 2013.
10. J. R. Babu, M. K. Murty, D. Rajashekar, and N. Susheela, "ANN-based design of a hybrid renewable energy system for hybrid electric vehicle applications," E3S Web of Conferences, vol. 616, Art. no. 01004, 2025.
11. J. R. Babu, M. Mustafa, M. K. Murty, D. Rajashekar, Y. Mastanamma, et al., "Efficient energy demand-supply forecasting via deep learning," in Proc. 2024 1st International Conference on Sustainability and Technological Advancements, 2024.
12. J. R. Babu, M. R. Nayak, and B. Mangu, "An electric vehicle energy management system design and implementation," in International Conference on Innovations in Clean Energy Technologies, 2023, pp. 29–41.
13. Y. Mastanamma, J. R. Babu, and B. Sireesha, "An electric vehicle by using wind energy: An experimental setup," in International Conference on Innovations in Clean Energy Technologies, 2023, pp. 71–79.
14. B. Pavankumar, J. R. Babu, and V. Saketha, "Analysis of matrix converter using direct transfer function approach," Grenze International Journal of Engineering and Technology, vol. 4, no. 3, 2018.
15. J. R. Babu, M. Mustafa, S. V. T. I. Srikanth, and K. Pavan, "AI-based connected load analysis and energy consumption prediction for Methodist College of Engineering and Technology."
16. J. R. Babu, M. Mustafa, D. N. S. U. B. Kishore, and Y. J. H. C. S. Kumar, "Design and development of a 500-W-class solar tree for Methodist College of Engineering and Technology."
17. J. R. Babu, "Micro genetic algorithm based overload minimization approach for transmission expansion planning."
18. J. R. Babu, "Micro genetic algorithm based optimal power flow."
19. A. K. Azmeera, P. Jadhav, and C. Lande, "Microstructure image-based finite element methodology to design abradable coatings for aero engines," Aerospace, vol. 10, no. 10, Art. no. 873, 2023.
20. A. K. Azmeera, P. Jadhav, and C. Lande, "Effect of change in material properties of the abradable coating on the wear behavior of it—microstructure model-based analysis approach," Lubricants, vol. 13, no. 1, Art. no. 22, 2025.
21. P. Jadhav, G. Yella, A. Azmeera, and C. Lande, "Mode II-delamination modeling in multi-directional laminates using cohesive zone methodology under static and dynamic loading," Materials Today: Proceedings, 2023.
22. P. Jadhav, G. Yella, A. Azmeera, "Ply level strain measurement in composites using digital image correlation," Materials Today: Proceedings, vol. 106, pp. 7–13, 2024.
23. A. A. Kumari, P. Jadhav, and C. Lande, "Simulation of abradable coatings compositions through microstructure image modeling," Metallurgical Research & Technology, vol. 123, no. 1, Art. no. 106.