Intelligent Load Distribution for Transformers Using Arduino

Swapna God
Assistant Professor,
Electrical Engineering
SKN Sinhgad College of Engineering
Korti, Pandharpur, India

Pradip Vyavahare
Assistant Professor,
Electrical Engineering
SKN Sinhgad College of Engineering
Korti, Pandharpur, India

Arjun Masal
Assistant Professor,
Electrical Engineering
SKN Sinhgad College of Engineering
Korti, Pandharpur, India

Anil Tekale
PhD Scholar,
Electrical Engineering
JSPM University
Wagholi, Pune, India

Abstract—Transformer overloads are frequent in power distribution systems, resulting in overheating, decreased efficiency, and potential failure. This paper presents a system for automatic load sharing between two transformers using an Arduino microcontroller to prevent transformer damage during overloaded conditions. The proposed system employs a current sensor, relay switching, and a real-time monitoring algorithm to balance the load between the primary and secondary transformers dynamically. The system ensures that both transformers operate within safe thermal limits, minimizing downtime and maintenance costs while enhancing the power system's reliability.

Keywords—Transformer protection, automatic load sharing, Arduino, relay control, overloading, power system reliability, current sensor, load balancing.

I. Introduction

Transformers are essential components in electrical power distribution systems, converting voltage to accommodate varying load demands. However, continuous operation under excessive load can cause overheating, insulation degradation, and ultimately, transformer failure. Overloaded transformers require considerable downtime for repairs and result in high operational costs. Therefore, the necessity for an efficient and automated system for transformer protection is critical.

The objective of this study is to design and implement an automatic load-sharing mechanism between a primary transformer and a secondary transformer, controlled by an Arduino microcontroller. The system aims to detect transformer overload conditions in real time and dynamically distribute the load between the two transformers, preventing damage from overheating and enhancing the overall reliability of the power distribution system.

II. Problem Statement

The traditional approach to transformer protection involves manually monitoring the transformer load and initiating corrective actions when overload conditions are detected. However, this process is inefficient, error-prone, and can result in substantial downtime. Moreover, failure to address overloads quickly can lead to catastrophic transformer damage, resulting in power outages and high repair costs. The goal of this research is to automate the load distribution process to ensure that transformers do not exceed their rated load capacity and remain within safe operating temperatures.

III. System Design and Architecture

A. Overview

The proposed system consists of two transformers, a master (primary) transformer and a slave (secondary) transformer. The system uses an Arduino microcontroller to monitor the primary transformer's load in real time, utilizing a current sensor. If the current exceeds a set threshold, a relay is used to automatically engage the secondary transformer, thus distributing the load. The system continuously monitors and adjusts the load-sharing process to prevent overheating and ensure optimal performance of both transformers.

B. Block Diagram

The following components and their interconnections from the system:

- Primary Transformer: The main transformer that provides power to the load.
- Current Sensor (CT): This measures the current in the primary transformer, which is fed to the Arduino for processing.
- Arduino Microcontroller: Monitors the input from the current sensor, compares it to a predefined overload threshold, and controls the relay based on the readings.
- Relay Module: An electromechanical switch that connects the secondary transformer when the overload threshold is exceeded.

- Secondary Transformer: Activated by the relay when the primary transformer is overloaded, allowing load-sharing between both transformers as shown in Fig. 2.

C. Power Supply

The power supply block diagram in Fig.1 shows the components necessary to provide power to the system and ensure its stable operation.

- AC Power Source: The system operates on a standard 220V AC supply, which powers the main transformer and the Arduino system.
- AC-DC Converter: The 220V AC is converted to DC through a rectifier circuit. A filter is used to smooth out the DC signal.
- 5V DC Regulator: The Arduino requires a stable 5V DC power source for operation, which is provided by a voltage regulator.

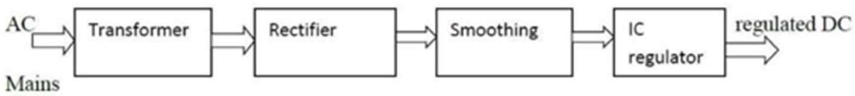

Fig. 1. Power Supply

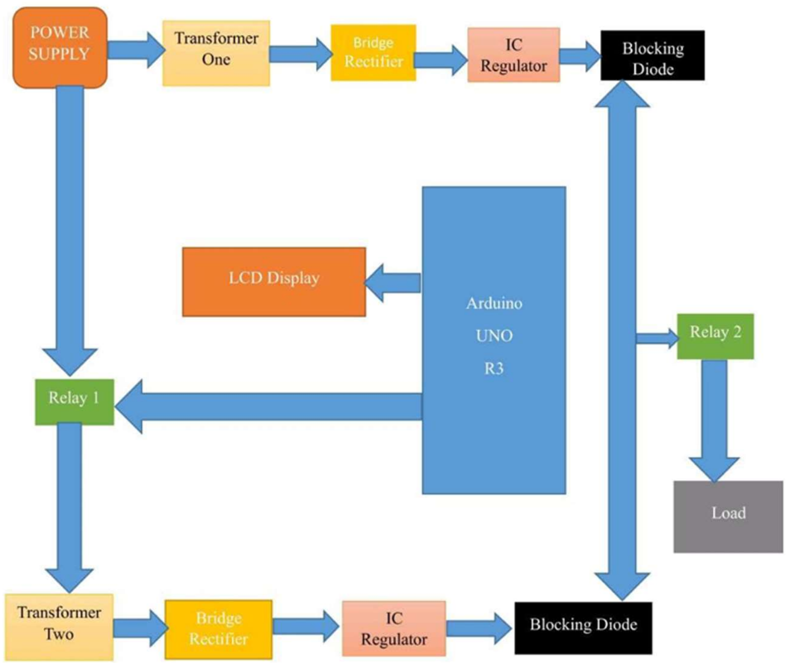

Fig 2. Block diagram of Automatic Load sharing Transformer using Arduino

D. System Flow Diagram

- System Initialization: Upon startup, the system initializes all components, including the Arduino and sensors.
- Current Measurement: The system continuously measures the current flowing through the primary transformer using a current sensor.
- Comparison with Threshold: The Arduino compares the measured current with a preset threshold. If the current exceeds the threshold, it indicates that the primary transformer is overloaded.
- Relay Activation: If the overload condition is met, the system activates the relay to connect the secondary transformer, sharing the load between the two transformers.
- Load Monitoring: The system continuously monitors the load on both transformers, ensuring that the load is shared appropriately and does not exceed safe limits.

The system continuously loops through this process as shown in Fig.3, adjusting the load-sharing conditions based on real-time data from the current sensor.

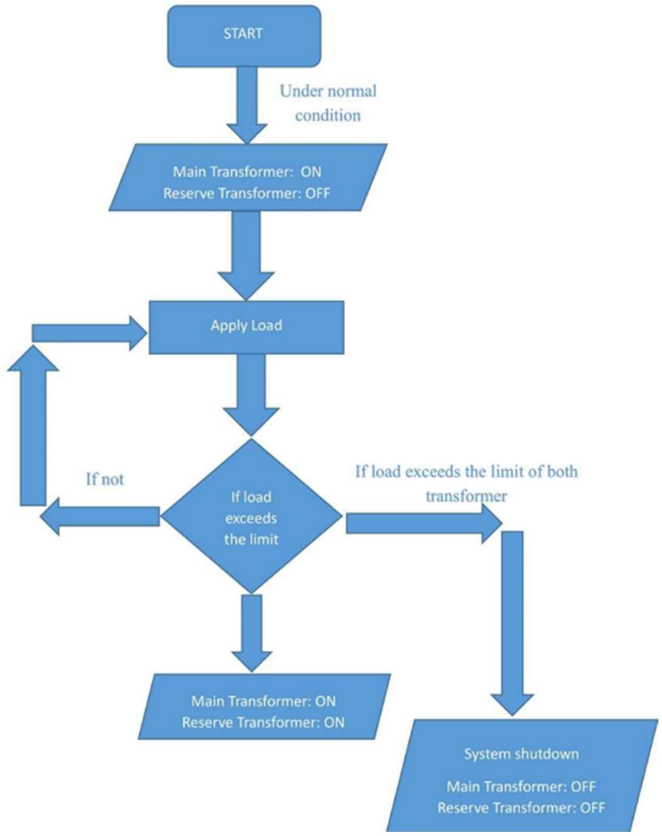

Fig. 3. System Flow Diagram

IV. Methodology

A. *Monitoring Current Using a Current Transformer*

A current transformer (CT) is used to measure the current flowing through the primary transformer. The CT outputs a signal proportional to the current, which is then fed into the Arduino's analog input pin. The Arduino converts this analog signal to a digital value using the Analog-to-Digital Converter (ADC). By calibrating the CT's output signal to represent the transformer's rated current, the Arduino can continuously monitor whether the transformer is operating within safe load limits.

B. *Defining Overload Conditions*

The threshold current value is calculated based on the transformer's rated load capacity. When the current exceeds this predefined value, it indicates an overload condition. The threshold can be dynamically adjusted for different operational conditions (e.g., seasonal load variations). The overload threshold could be set at a percentage of the rated transformer capacity (e.g., 90% of the rated current).

C. *Relay Control for Load Sharing*

The relay module is interfaced with the Arduino to switch the slave transformer on or off. The relay is triggered through a digital output pin of the Arduino, where a low signal activates the relay and a high signal deactivates it. When the Arduino detects that the current exceeds the threshold, it sends a signal to the relay to connect the slave transformer in parallel with the primary transformer.

In normal operation, the system operates with only the primary transformer running. However, when overload is detected, the slave transformer is automatically activated, sharing the load to prevent overheating.

D. *Algorithm for Load Sharing*

The load sharing algorithm consists of several steps:

- Initial Check: The Arduino reads the current sensor value every second.
- Comparison: The current value is compared with the overload threshold.
- Action on Overload: If the current exceeds the threshold, the system triggers the relay to connect the slave transformer.
- Load Reduction Check: If the load on the primary transformer decreases below the threshold, the system disconnects the slave transformer.

The system uses a hysteresis method to avoid frequent switching of the slave transformer. This ensures that the slave transformer is not engaged unless the load is consistently above the threshold for a set period.

V. Methodology

A. *Simulation Setup*

The system was simulated using Proteus software, with the following components:

- Arduino Uno as the microcontroller.
- Current sensor (e.g., ACS712 or CT sensor).
- Relay module to switch the slave transformer.
- Load to simulate the transformer's load (modeled as a resistive load in the simulation).
- Slave Transformer was simulated as a secondary load to share the power.

B. *Simulation Steps*

- Initial Load Simulation: The primary transformer was initially set to carry a normal load (below the threshold).
- Overload Condition Simulation: The load was gradually increased beyond the predefined threshold (e.g., 90% of rated current).
- Load Sharing Activation: Once the current exceeded the threshold, the Arduino microcontroller activated the relay, engaging the secondary transformer to share the load.
- Load Monitoring: The system continuously monitored the load on both transformers, and when the load on the primary transformer fell below the threshold, the slave transformer was automatically disconnected.

C. *Simulation Results*

- Normal Operation: Initially, the current through the primary transformer remained well below the overload threshold, with only the primary transformer supplying power to the load shown in Fig.4.
- Overload Detection: When the load on the primary transformer exceeded the set threshold, the Arduino detected the overload and activated the relay to engage the slave transformer shown in Fig.5.
- Load Sharing: Upon activation of the relay, the slave transformer shared the load, reducing the current in the primary transformer and preventing it from overheating.
- Disengagement of Slave Transformer: As the load decreased below the threshold, the system automatically disconnected the slave transformer to optimize efficiency.

TABLE I. Simulation Results

Load (Amps)	*Primary Transformer Current (A)*	*Secondary Transformer Status*
0-10 A	5 A	OFF
10-20 A	15 A	OFF
20-30 A	25 A	OFF
30-35 A	35A	ON
35-40 A	38 A	ON
40+ A	45 A	ON

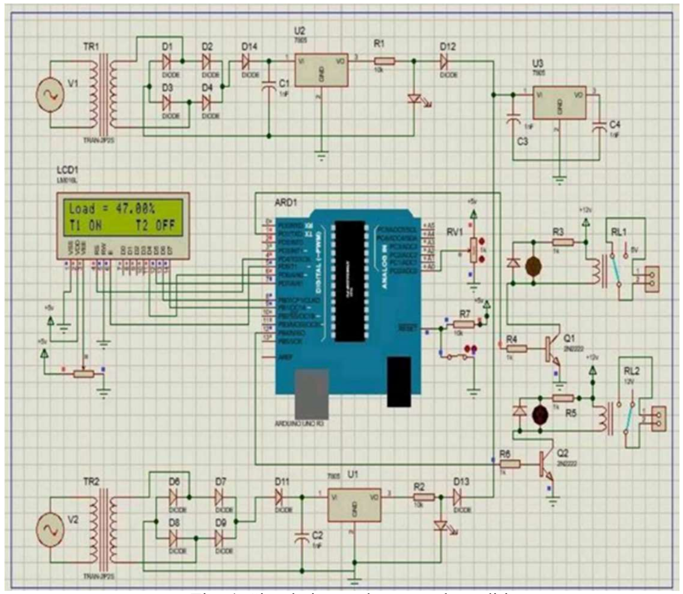

Fig. 4. Simulation under normal condition

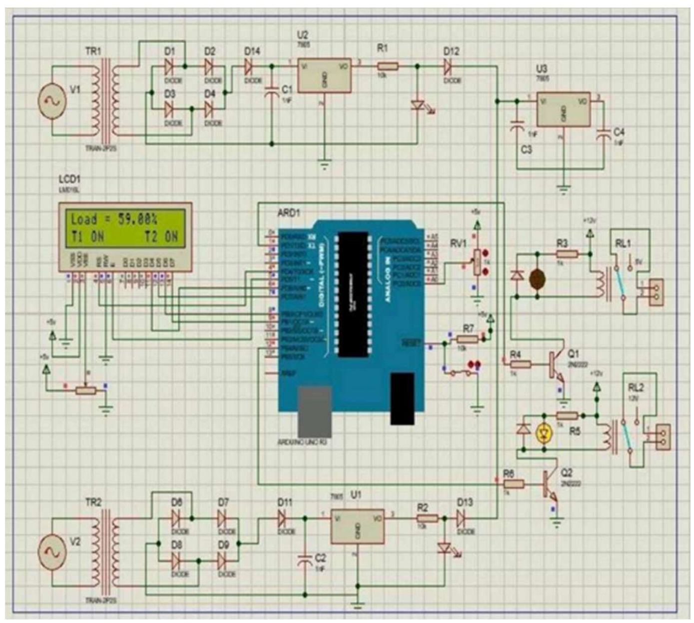

Fig. 5. Simulation under loaded condition

VI. CONCLUSION

This research introduces a cost-effective and automated solution for transformer overload protection through load sharing using an Arduino microcontroller. The system demonstrated high efficiency in managing overload conditions and distributing the load between two transformers. The system was successfully tested through simulation, and the results showed that it could reliably prevent transformer overheating and reduce the risk of transformer failure. Future work can involve hardware testing and the integration.

REFERENCES

[1] Ware, V., Dungahu, G., Arade, T., Tekale, A., & God, S. Amorphous Transformer: Study Paper.

[2] Ahamed, M. A., Nafi, N. S., & Mahfuz, S. M. (2017). "Load sharing and network reconfiguration in distribution system using microcontroller." International Journal of Power Electronics and Drive Systems, 8(1), 123-130.

[3] Kumar, M., & Verma, K. (2019). "A review on applications of microcontrollers in electrical engineering." Journal of Advanced Research in Electrical and Electronics Engineering, 6(1), 19-25

[4] Anil Tekale | Radharaman Saha | Pratik Ghutke | Swapna God "Implementation of Fuzzy Based Grid-Tied PV System for Single Phase Transformer less Inverter with Charge Pump Circuit Concept" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-4, June 2019, pp.163-171, URL: https://www.ijtsrd.c om/papers/ijtsrd23 583.pdf.

[5] Tekale, A., & God, S. (2021, June). Charge Pump Circuit Based Grid-Tied PV System for Single Phase Transformer Less Inverter Using Fuzzy Logic Controller. In Techno-Societal 2020: Proceedings of the 3rd International Conference on Advanced Technologies for Societal Applications—Volume 2 (pp. 79-85). Cham: Springer International Publishing.

[6] God, S. P., Tekale, A., & Mallareddy, C. (2017). Control Scheme Based on Quasi Z-Source Network for Four Switch BLDC Motor. International Journal Research and Development 5.7 (2017): 461, 464.

[7] Tekale, A., Kumar, A., & God, S. (2018). Automatic Fault Detection and Protection of Three Phase Induction Motor.

[8] Wang, P., & Wang, X. (2021). "Real-time load balancing in smart grids using IoT-based control systems." Sensors, 21(12), 4053.

[9] Santos, M., & Almeida, L. (2022). "Affordable energy distribution solutions for rural areas: A survey." Renewable and Sustainable Energy Reviews, 139, 110686.

[10] Dixit, A., & Patel, S. (2022). "Real-time load management systems for smart grids: A comprehensive review." IEEE Access, 10, 14457-14468.

[11] Sadeghian, A., & Fakhar, M. (2018). "Low-cost IoT-based monitoring for transformer load sharing in rural areas." IEEE Internet of Things Journal, 5(5), 3789-3796.

[12] Thomas, A., & Kannan, S. (2023). "Microcontroller-based load balancing system in decentralized energy networks." Electronics, 12(3), 391.