Single Phase Transmission Line Fault Detection and Analysis Using an Arduino

Raiyan Wasi Siddiky
21301648
CSE Dept.
BRAC University
Dhaka, Bangladesh
raiyan.wasi.siddiky@g.bracu.ac.bd

Eshtiak Alam Shihab
21301502
CSE Dept.
BRAC University
Dhaka, Bangladesh
eshtiak.alam.shihab@g.bracu.ac.bd

Kazi Israrul Karim
21301509
CSE Dept.
BRAC University
Dhaka, Bangladesh
kazi.israrul.karim@g.bracu.ac.bd

***Abstract*—The primary objective of this project was to develop an easy and effective fault detection system that enhances the reliability and efficiency of single-phase transmission lines. To accomplish this, the project system makes use of a basic arduino and current censor setup with relays to measure the current at two ends of a single phase transmission line to quickly detect any faults in the system, relaying the information to the Arduino IDE(Integrated Development Environment), which allows this system to achieve near real-time fault detection as well as control from everywhere using the server.**

***Index Terms*—Single Phase, Fault Detection, Arduino, Current**

I. Introduction

The dependable and uninterrupted supply of electrical power is a cornerstone of modern society, essential for industries, commerce, and everyday life. Ensuring the reliability and resilience of power transmission systems is essential to minimize disruptions and maintain the quality of service. Single-phase transmission lines play a pivotal role in these systems. However, these critical components are susceptible to a variety of faults and even theft that can compromise system stability and service continuity.

A. Problem Statement

Electricity transmission lines play a pivotal role in supplying power from generation sources to end-users. The smooth and uninterrupted operation of these transmission lines is essential for ensuring the reliability of electrical grids and other systems. However, transmission lines are susceptible to various faults, including short circuits and line breaks, which can lead to service disruptions, equipment damage, and even safety hazards. Timely detection of these faults are paramount for maintaining grid stability and minimizing downtime. Traditional fault detection systems are often costly and lack real-time capabilities, making them impractical for smaller utilities and emerging applications. Thus, there is a pressing need for a cost-effective, real-time fault detection and analysis system that can enhance the reliability of single-phase transmission lines.

B. Objective

The objective of this project was to create a cost-effective, easy to access, assemble and use system that could accurately predict any faults in single phase transmission lines, including theft of power from a line. we had to design and implement a hardware system that used sensors to obtain current values and use a microcontroller arduino alongside algorithms or code to be able to attain real time results as well as be able to pinpoint the location of the fault.

C. Motivation

The motivation behind this project arises from the critical importance of addressing the challenges posed by faults in single-phase transmission lines as any disruption due to transmission line faults can have cascading effects on industries, households, and essential services. We wanted to implement modern technology in fault detection to increase the efficiency of the power sector and show the benefits of moving towards data driven decision making, and that making this shift can be cost effective even for large scale systems like transmission lines.

II. Ease of Use

This project was made with simplicity, understanding and usability in mind. One of the key advantages of the system is its straightforward hardware setup. We have meticulously designed the hardware components to be readily available and easy to assemble. The core component, the Arduino microcontroller, is widely accessible and comes with extensive documentation and online resources for beginners.

Units and symbols

- Resistance(R) - Ω
- Resistivity(ρ) - Ωm
- Current(I) - mA
- Voltage(V) - V
- Area of Conductor(A) - m^2
- Length of Conductor(L) - m

III. Components

We have used a Variety of components to build this hardware system.

A. Arduino MEGA 2560

The most important part of the project that drives the whole system. This is what allows the near real time feedback on faults. Its pins are connected to receive current from the relays, as well as connected

B. ACS712 Current Censor

The current censor senses current and sends them through the relay to the Arduino.

C. Relay

The relay acts as a switch, which when a fault is detected, opens the circuit so that current no longer flows. It does this by demagnetizing the magnets in the relay which occurs after the arduino senses a fault and sends to a signal yo the relays to demagnetize.

D. LED and Piezo Buzzer

The LEDs turn on when there is a current passing through them, and subsequently turn off when there is a fault due to the relays opening the circuit. At the same time the piezo buzzer goes off at the fault after a signal from the arduino.

IV. Methodology

The primary purpose of the system is to obtain current values through the use of current sensors and then send it the arduino through a relay. An initial model of a transmission line was created using styrofoam to hold the lines which is then fed into a transformer to bring the voltages down to a usable value. In this case, we use a step down transformer that brings the voltage down from 220V to 12V. We have two lines, one main transmission line and one ground line. Both are fed through the transformer to two current censors(ACS712) which are then fed through a relay to the arduino(Arduino MEGA 2560). The arduino then processes the data to detect faults, and passes it to a piezo buzzer which buzzes in the case of a fault. The job of recognizing the faults is done by the arduino. It does this by taking in the current values and calculating the RMS(Root mean square), which it then compares with a experimentally calculated threshold voltage called the pickup voltage. If the voltage is less than the threshold, then a fault is shown. Additionally, we have also used LEDS to indicate the flow of current. In the case that there is a fault, these LEDs will turn off.

V. Experimental Setup

A. Circuit Diagram

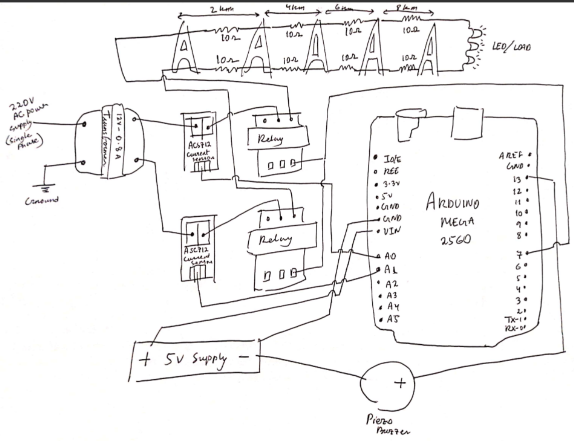

Fig. 1. Circuit Diagram

B. Equations

The equation

$$\text{result1} = \frac{(\text{maxValue1} - \text{minValue1}) \cdot 5.0}{1024.0}$$

in the code is used to convert the raw analog reading from the current sensor into a voltage value.

- $(\text{maxValue1} - \text{minValue1})$: This calculates the range of values that the sensor measured during the sampling period. The difference between the maximum and minimum values gives the span of the sensor's response.
- $\times 5.0$: This scales the sensor's response to a 5V range. The analog pins of an Arduino accept input voltages between 0V and 5V, so this step maps the sensor's span to the full input range of the analog pin.
- $/1024.0$: This divides the scaled value by the total number of possible analog values (1024) that the analog-to-digital converter (ADC) of the Arduino can represent. This step converts the scaled value into a voltage value within the 0V to 5V range.

The final result, result1, is the voltage value that corresponds to the current sensor's output during the sampling period.The current sensor outputs a voltage proportional to the measured current. By converting this voltage to a digital value between 0 and 1023 (which corresponds to the analog-to-digital converter's resolution), we're able to work with the sensor's data in a standardized and easily understandable format.

The scaling factor of 5.0 is used because it corresponds to the reference voltage (V_{ref}) of most Arduino boards.

1) Conversion of Voltage Value to RMS Current in Milliamperes:

$$\text{VRMS1} = \frac{(\text{Voltage1}/2.0) \cdot 0.707}{} \tag{1}$$

$$\text{AmpsRMS1} = \frac{\text{VRMS1} \cdot 1000}{\text{mVperAmp}} \tag{2}$$

These two equations are used to convert the voltage value obtained from the current sensor into an equivalent root mean square (RMS) current value in milliamperes (mA).

- $\text{VRMS1} = \frac{\text{Voltage1}/2.0 \cdot 0.707}{\text{AmpsRMS1}}$: The voltage value obtained from the current sensor, stored in the variable Voltage1, is in the peak voltage format. Peak voltage is the maximum voltage value reached in a sinusoidal waveform. To convert this peak voltage to the equivalent RMS voltage, we multiply it by the factor 0.707. The factor 0.707 is the RMS value of a sinusoidal waveform with a peak value of 1. This is a common conversion factor for AC waveforms, specifically sinusoidal ones. Dividing by 2.0 is necessary because RMS voltage is related to peak voltage by a factor of the square root of 2 ($\sqrt{2}$) for a sinusoidal waveform.
- $\text{AmpsRMS1} = \frac{\text{VRMS1} \cdot 1000}{\text{mVperAmp}}$: VRMS1 is the RMS voltage calculated in the previous step. 1000 is used to convert the value from Amperes to milliamperes (mA), since mVperAmp is defined as 100. mVperAmp represents the millivolts per Ampere conversion factor for your current sensor. This factor allows us to convert the sensor's output voltage to actual current in milliamperes.

The final result, AmpsRMS1, represents the calculated RMS current in milliamperes based on the voltage output from the current sensor. This conversion is important when dealing with AC currents, as the RMS value gives us a meaningful measurement of the current's magnitude that's consistent with other measurements and standards.

VI. Results and Discussion

Our system takes the current from the transmission line via current censors through relays to the arduino, which then processes the information to check for any faults. A fault is detected when the voltage in the system goes below a certain threshold, in this case being around 0.20 volts. This occurs when there is either a short in the system, which is what we primarily provide via shorting two lines, or via a line to ground connection. Since our system only has two lines, a volatage line and a ground line, a short fault and a ground fault is interpreted as the same. This entire process is carried out by the code in the arduino IDE, which we provide via a USB(Universal Serial Bus) connection via the arduino port. The fault detection is also physically known through the use of the piezo buzzer, which buzzes when there is a fault. The code also relays the point at which the fault has occurred, or rather the line between two poles where the fault occurs.

While working on the project, we as a group learnt alot about our shortcomings. We initially started the project with the intention to create a system for fault detection in three phase transmission lines, but quickly realized that access to such a line was not only hard to obtain for university students, but also dangerous. We also looked into creating circuitry to simulate a three phase current using capacitors and resistors but eventually decided to simplify the project to focus on finding faults within single phase transmission lines, which is still significant as it has great impact on regular households.

VII. Conclusion

The development of a single-phase transmission line fault detection and analysis system using an Arduino microcontroller is a significant step towards enhancing the reliability and efficiency of electrical circuitry in households. This cost-effective solution provides real-time monitoring and fault detection ultimately leading to reduced downtime and improved management.

As technology continues to advance, the integration of such systems into larger smart frameworks holds great promise for the future of electrical power distribution.

Acknowledgment

The completion of this research on Single Phase Transmission Line Fault Detection and Analysis was made possible through the contributions and support of various individuals and organizations, for which we extend our heartfelt gratitude.

First and foremost, we would like to express our deepest appreciation to our research advisors and mentors who provided invaluable guidance and expertise throughout this project.

Our gratitude extends to BRAC University for providing the necessary resources and facilities for conducting experiments and data analysis.

Last but not least, we wish to thank our families and friends for their continuous encouragement and understanding during the research process. Their unwavering support and patience were essential in sustaining our motivation.

References

[1] PROJECTRONICS. (2022, February 18). Power theft detection in transmission line using Arduino! Best Electrical engineering project [Video]. YouTube. https://www.youtube.com/watch?v=4jR24GpKsMA

[2] McFadyen, S. (n.d.). Fault calculations - Introduction. https://myelectrical.com/notes/entryid/192/fault-calculations-introduction

[3] AJ Electric. (2020, January 9). Three Phase Transmission line fault detection and analysis system in urdu/Hindi [Video].

[4] Smart Monitoring and Fault Detection System for Power Transmission Lines Using Digital Sensors.pdf. (n.d.). Google Docs. https://drive.google.com/file/d/1oSFjL4oxDCjnmjZBLNqqqXmpoL0vjqUF/view