UCF Senior Design I

Title: Machine Learning Applications in

Power System Fault Location with ADMS and AMI

Department of Electrical Engineering and Computer Science

University of Central Florida

Dr. Lei Wei

Group 11

Michael Mederos	Computer Engineer
Julio Barreto Tannoux	Electrical Engineer
David Silvieus	Electrical Engineer
Yuejun Guan	Electrical Engineer

Reviewers:

Dr. Deepal Rodrigo

Dr. Zhihua Qu

Dr. Aleksandar Dimitrovski

Dr. Mike Borowczak

Acknowledgement

We would like to express our deepest gratitude to all the individuals and organizations who have contributed to the successful completion of this thesis project. Without their unwavering support, guidance, and encouragement, this project would not have been possible.

First and foremost, we extend our heartfelt thanks to Dr. Zhihua Qu, for their invaluable mentorship and continuous guidance throughout the project. We would also like to thank him for providing us with essential equipment crucial for the successful completion of this thesis project.

We would like to acknowledge the mentorship provided by Dr. Deepal Rodrigo from FPL, which played a crucial role in the execution of this project. Their generous support allowed us to access necessary data, making this endeavor possible.

We are also immensely grateful to Dr. Aleksandar Dimitrovski and his PHD student Kwasi Opoku. Their expertise, patience, and insightful feedback regarding power system modeling, protection and control have been instrumental in shaping the direction of this work.

We extend our heartfelt gratitude to Kenneth McDonald for his invaluable assistance and expertise in the field of computer engineering. His guidance has been instrumental in shaping the success of this project.

Everyone’s support has been pivotal in shaping this work and has enriched our growth.

Thank you all.

Table of Content

List of Figures

List of Tables

1 Executive Summary

The objective of this project is to enhance the overall stability and reliability of the power system through the implementation of an advanced machine learning model. By leveraging machine learning techniques and analyzing large amounts of real-time data to identify patterns and anomalies associated with power grid faults, a robust fault locator solution can effectively narrow down parameters, thereby reducing restoration time and improving the overall efficiency of the utility industry. Being able to locate and isolate faults in a timely manner is critical for power distribution systems because it helps improve the reliability of the power distribution grid and ensuring uninterrupted power supply to customers in the vicinity, safeguarding them from power outages. The improved isolation time also prevents further equipment failure which could lead to extended blackouts in certain conditions.

With real-time and on-demand measurements from Advanced Distribution Management System (ADMS) and Advanced Metering Infrastructure (AMI), these measurements serve as vital inputs to the machine learning algorithms. ADMS provides outage data at the feeder level, including but not limited to the affected feeder, fault duration, and the number of affected customers. In this project, data from ADMS is being used to rapidly narrow down the feeder where the fault occurred, enabling quick isolation of the feeder and exploration of alternative routes to restore customer connectivity. AMI is a sophisticated metering infrastructure capable of providing data at the equipment level. The machine learning algorithms process the data efficiently and reconsolidate the system topology, enabling accurate identification of fault locations by measuring impedance through controlled input pulses from the CPU. Analyzing the reaction from all input pulses allows for the detection of anomalies and precise localization relative to the pulse injection sites. This enhanced fault location capability results in expedited response times, minimizing customer downtime, and improving overall grid performance.

The project will be implemented on a 12-volt, single-phase microgrid representation, exemplifying a comprehensive understanding of fault detection and isolation methodologies. Although the project demonstration is a low voltage single phase system, the developed process can be seamlessly scaled up and applied to three-phase distribution systems. By intentionally inducing a ground fault scenario using a switch connect from the line or load to the ground, high impedance from one terminal of the system to another terminal indicates a fault. Once the high overall impedance values are extracted from the certain area using the pulse’s output, they would be used to form a geographical matrix representation to calculate the location of the fault. The design aims to effectively demonstrate the fault location algorithm's capabilities within this microgrid system based on its practical applicability in real-world scenarios raised by the utility industry.

By combining the rising trend of machine learning technology with the dynamic operation of power system grids, this project’s impact extends beyond the development of a robust fault location solution using machine learning. It has the potential to aid further machine

learning applications on the power grid. By introducing a novel machine learning fault identification process that leverages existing data from smart infrastructure and industry-standard applications, this project aims to minimize restoration time and optimize power system performance. Thus, ultimately enhancing the reliability of power distribution and ensuring uninterrupted power supply to customers. Through these transformative efforts, the project aims to develop existing algorithms in the utility industry and serve as a base for further advancements in the intersection of machine learning and power systems engineering.

2 Project Description

In this chapter, we will be providing a high-level description of our project as well as a basic description of the various components that will be used along with our reason for using it. We will also be discussing our inspiration for the project along with our project requirement and specifications, a block diagram detailing the different components present. Finally, we will also discuss our project budget and financing, as well as our milestones. A house of quality is also included to summarize everything visually.

2.1 Project Background

This project aims to improve the stability of the distribution system by implementing a machine learning model utilizing existing data and applications to locate and isolate the fault efficiently. Despite significant advancements in our interconnection's stability, including self-healing capabilities during outage incidents, power outages can still occur due to factors such as equipment failure, weather events, and overloading. These outages exert a substantial impact on our daily lives, affecting crucial aspects such as employment and healthcare. The improved fault location process will lead to shorter restoration times and enhanced overall reliability, ensuring a more resilient power grid. By identifying the fault location as quickly as possible, operators can promptly take corrective measures, such as isolating the fault using fuses and automatic reclosers. This enables them to direct the power flow in a way that minimizes the number of affected customers. The crew members can quickly address the issue by confirming the fault's precise location. This enables them to swiftly initiate the necessary repairs and restore the power supply efficiently.

Machine learning can observe and analyze the pattern of a dataset to identify any anomality. It can process large amounts of real-time and offline measurements produced by wide-area measurement systems such as ADMS, the AMI system, and other intelligent electronic devices in the power grid. The data usually is high quality because multiple measurement points taken along the same line ensure the low-quality data can be easily filtered. This creates a perfect environment for machine learning to help with identifying outage patterns, prioritizing restoration efforts, and improving overall grid reliability. Manually monitoring can be slow and costly even with the help of smart devices and contingency alarms, machine learning enables real-time monitoring and predictive analysis, allowing grid operators to quickly locate the source of the fault and take proactive measures to prevent widespread outages.

This project idea is provided by Florida Power and Light, they are providing the necessary application and data associated with the project's successful completion. They would also jointly guide this project. This project idea came from the utility industry as they noticed the growing trend of machine learning applications in the field of power grid operation, aiding in many areas such as outage forecasting, stability assessment, control, and restoration. This project focuses on the stability assessment aspect, aiming to evaluate transient stability, short-term voltage stability, and contingency screening. They want an applicable machine learning algorithm capable of identifying and isolating faults down to individual meter levels, the restoration time can be shortened if the lineman knows the exact location of the fault instead of manually going through all the areas. This not only saves time and labor in terms of outage restoration, but the shorter restoration time will also help ensure customer satisfaction.

The two elements that will go into this project will be the machine learning algorithm and the hardware simulation demonstration. The algorithms will include data preprocessing, extracting, and learning features with the appropriate machine learning model, task class classification to detect and locate the fault, and alerting the system operator once the fault has been located. With the machine learning algorithm, we hope to be able to search the parameter for faults by comparing overall impedances from different injection sites to compute a geographical representation of the fault location. For three or more injection sites, multiple 2-dimensional matrices are needed to compute its exact location. High overall impedance indicates that the fault location is far away, and vice versa.

The hardware is a single phase one radical line small-scale distribution grid backed with real-world data supported by our PCB for demonstration purposes. It will use the power supplied in the GE-FPL microgrid control lab to power a series of passive components and switches. Protective devices such as fuses will be added for protection purposes and to help with measuring the sent-out voltage. By shorting one of the switches to ground, we will simulate a fault in an area. Once the machine learning algorithm is deployed it should be able to locate the feeder, the possible parameters within the feeder, and then the exact fault area. This design will simulate a one-line radical distribution grid, the second goal for this project is to be able to demonstrate a few branches to ensure the machine learning model is trained for different issues relating to various kinds of power grid design.

There are some equivalent products in terms of stability assessment with machine learning, in general, this idea is still new so many of the approaches have not been tested. A lot of useful data is being wasted because of the lack of fast data processing ability. And because we will be working with applications and data provided by the sponsor, we have the added challenge of converting data type into acceptable matrix form for our chosen machine learning model. The two applications our data come from are ADMS (advance distribution management system) and AMI (advance metering infrastructure). Data from ADMS is SCADA (Supervisory Control and Data Acquisition) based, it will give clarity by gathering real-time data from sensors and devices located throughout the power grid infrastructure for faults in the feeder level. This data updates every 30 seconds to 1 minute. This data includes information on real, reactive, and apparent power, voltage levels, current flow, transformer status, breaker positions, and other operational parameters. This is the most popular telemetry collection and analysis system that provides the operator with the alarm once a contingency is detected at the feeder level. ADMS also provides a map of all nearby substations and power lines so we can gaze far out from where the fault might be located and how it is impacting the adjusting area. AMI is a two-way communication system to collect detailed metering information, typically AMI data normally updates every 15 to 30 minutes and can be requested if needed. AMI measurements extend beyond the feeder level down to the equipment level, they can provide data attached to every customer's meter and solar panels. This data includes information on Cumulative and daily kWh usage, peak kW

demand, load voltage profile, and power factor. AMI data are often underutilized due to their high volume and too much information makes them different to be sorted out.
Our PCB design will consist of an MCU with built-in flash memory. Based on the data we capture from ADMS, we will train our model and then flash it onto the MCU. The PCB will be responsible for determining the geographic location of a fault as well as isolating said fault. To do this, we will have several injection sites along the distribution grid that will measure the impedance across their connected lines, forming a matrix of impedances with their associated weights. This matrix of impedances will be compared to our Machine Learning model to determine the presence of a fault. A visual representation is shown later in Figure 4, take note of the four injection sites and the fault on one of the branches. Locating the fault would consist of taking the weights of all four sites, sites 1 and 2 would have a lower weight due to their proximity. While site 3 and 4 will have higher weights due to their distance. A representation of the impedance matrix is shown in the figure below.

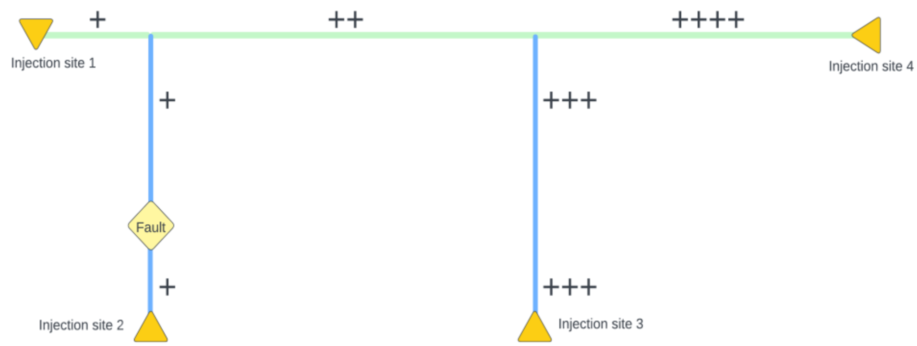


Figure 1 Model system

This project is unique because it is based on a real-world issue that could potentially impact the whole interconnection. We are motivated to work on a project that utilizes machine learning to locate faults on power lines and reduce restoration time because we believe in the transformative potential of advanced technologies in improving the reliability and efficiency of power grids. Every home and commercial and industrial center in America is backed by the power grid, by applying machine learning algorithms to analyze data and identify fault locations accurately we hope to improve the overall stability of the power grid and shorter restoration time. UCF's Power and Renewable Energy Program boasts a highly dedicated and diverse faculty with exceptional backgrounds, all deeply passionate about power systems and committed to student success. Moreover, this program provides access to state-of-the-art equipment and ample mentoring opportunities, ensuring students have the necessary resources to excel in their studies. With the help of the combination of knowledgeable professors, a supportive environment, and abundant mentorship we hope this project can solve a real-world issue.

2.2 List of Requirements and Specifications

In the table below, we have listed our key requirements and specifications. We believe these specifications are general enough to where our design choices are not inhibited. It is also satisfactory for our customer. Later on in our report, we will go into more detail with regards to each of the requirements, as well as assign more quantitative requirements as we begin to choose our desired component.

Table 1: Requirements and Specifications

No.	Requirement	Specification	Description	Priority
1	Ground Fault Detection	Almost instantaneously	Ground faults are detected almost instantaneously due to all the relay devices throughout the power grid	High
2	High Impedance Fault Detection	Less than 30 minutes	The time the system takes to detect and identify the location of a fault must be within 30 minutes.	High
3	Fault Location Assessment	Within a mile of the actual fault location	Models' detection algorithm must be able to provide a location for the fault with an error of less than a mile	High
4	Water/Weather Resistance	IPX6 and IP5X Water/Particulate rating	Must be able to withstand being outside or buried underground.	High
5	Hardware Communication	.8Mbps or more	PCB must be able to connect to a computer in a reasonable amount of time to retrain or debug machine learning model.	High
6	Impedance Matrix Calculations	Able to calculate with 4 or more signals	The model must be able to accurately process impedance values from at least 4 injection sites and weigh them against each other to determine fault location.	Med

2.3 Project Block Diagram

Below is a project block diagram explaining at a high-level the operational flow of our device. It explains each step the project takes to detect a fault, as well as visualize logic loops. We have also highlighted each design block in differing colors to visualize which team member worked on what section. However, it is important to note that our actual contributions were much more fluid, we have all worked on each block to different degrees. In addition, we included a flowchart of our proposed machine learning model which explains at a high level every logical step it will be taking. Finally, we have included a block diagram that encompasses the whole project, this block diagram shows how the PCB will be interacting with our pseudo power grid, as well as visualizing some of our injection sites, of course, this is just an illustration meant to provide a visualization and as such is not exact in regard to our actual layout.

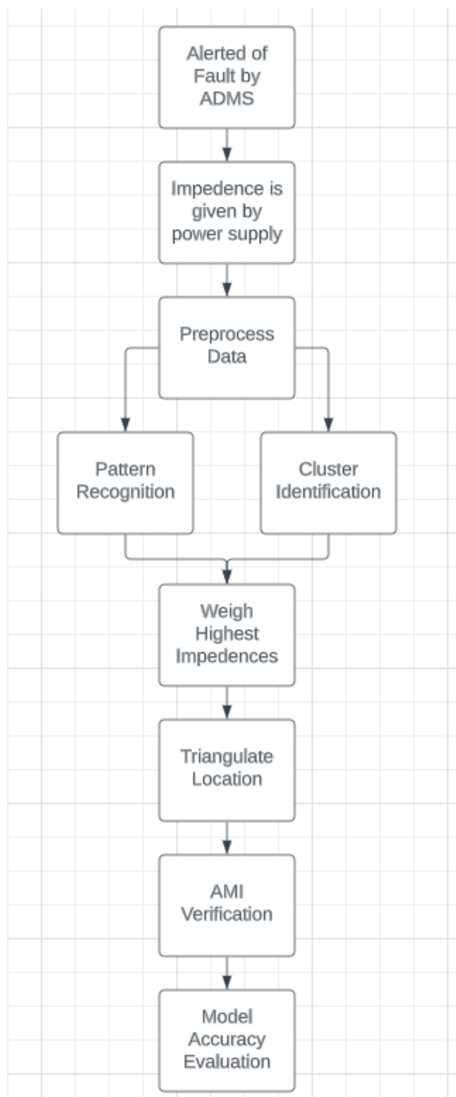


Figure 2: high level overall flowchart

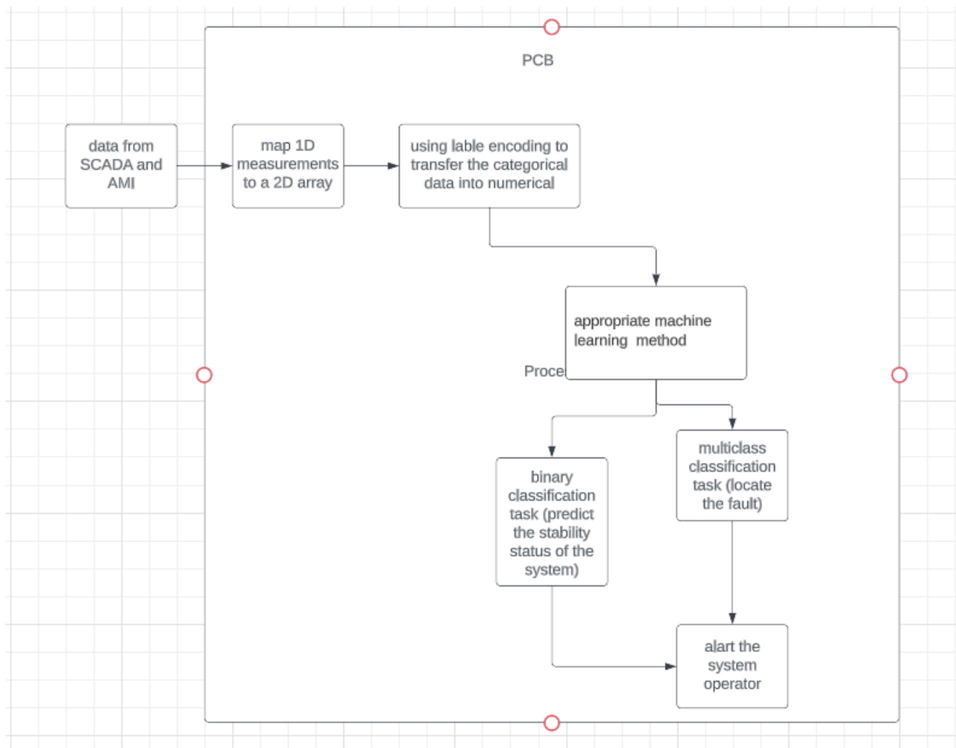


Figure 3: Flowchart of the proposed ML model

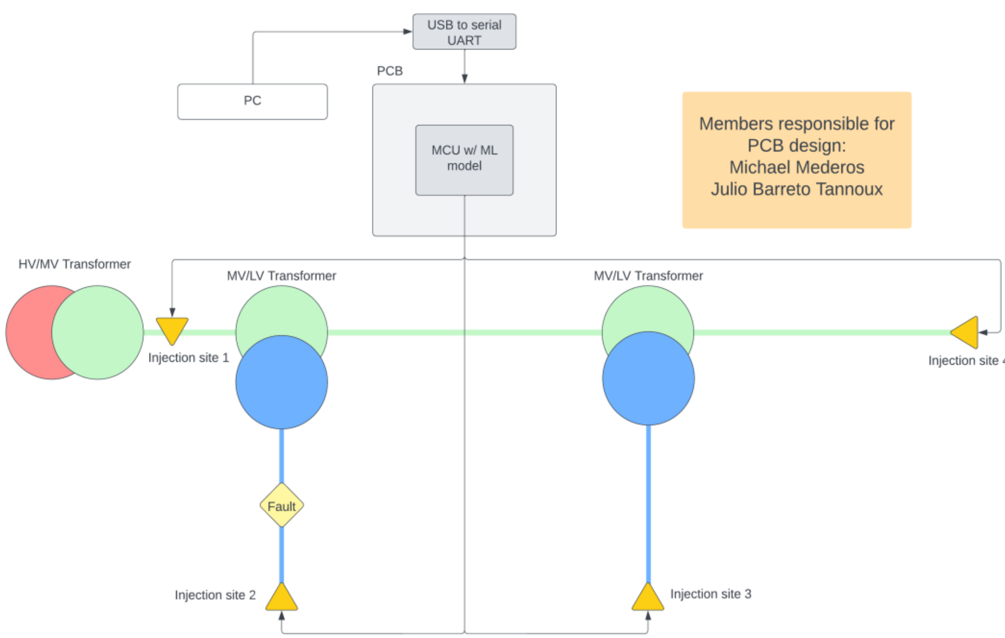
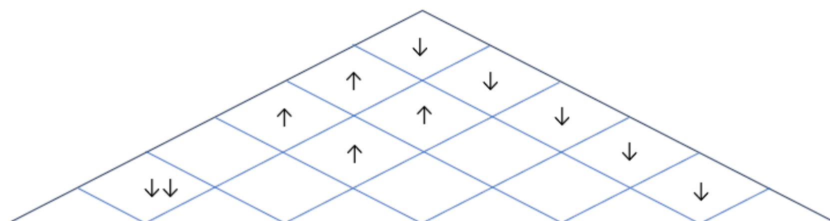


Figure 4: Block diagram of PCB interaction with grid

2.4 House of Quality

In the following chart (figure 5), we were able to visually depict the correlation between the engineer requirements with themselves and with the market requirements. Our project is supposed to be used and installed by professionals, it doesn't have to be very easy to install or understand since those that are going to work with it will possess the necessary skills and knowledge to operate it. Mostly the market requirements are based on how much it improves the process of finding a fault in the electric system and fixing it. The faster and more precise our device, the better. Upward and downward arrows symbolize whether there is a positive or negative correlation respectively. And if there are two arrows, it means that there is a strong correlation.

Figure 6 shows more clearly the engineering requirement trade off, that is also seen in the roof of the House of Quality.



	Fault Detection (-)	Fault Location Assesment (-)	Water / Weather Resistance (+)	PCB Communication (+)	Impedeance Matrix (+)	Cost (-)
Grid Stability & Reliability (+)	↑↑	↑↑			↑	↓
Time to find fault (-)	↑↑	↓		↑	↑	↓
Precise location(+)	↓	↑↑		↑	↑↑	↓
Restoration times (-)	↑	↑	↑		↑	↓
Cost (-)	↓		↓		↑↑	↑↑
Targets	< 30 min	< 1 mile	IPX5 or IPX6	.8Mbps	>= 4 Signals	<= $434.00

Figure 5: House of Quality

	Fault Detection (-)	Fault Location Assesment (-)	Water / Weather Resistance (+)	PCB Communication (+)	Impedeance Matrix (+)	Cost (-)
Fault Detection (-)		↓↓		↑	↑	↓
Fault Location Assesment (-)				↑	↑	↓
Water / Weather Resistance (+)						↓
PCB Communication (+)						↓
Impedeance Matrix (+)						↓
Cost (-)						

Figure 6: Engineering trade off matrix

3 Research and Part Selection

In this chapter, we will be discussing in a technical sense the various components that were discussed at a high-level in chapter 2. For each component, various options were presented, and we will give our critique of each option by comparing them to each other and how well they can serve our needs. After, we present our final choice for each component, and lay out our reasons as to why we chose said component over the other choices. We considered many factors, and in table 4, we highlight the most important ones. While there were more factors, we believe that the ones listed in table 4 are the most crucial and as such have been assigned the highest weights.

3.1 Technology Comparison

One of our main goals in selecting our senior design project was to make sure that every team member had exposure to something related to their major. We all wanted to be able to see how the concepts we have learned throughout our undergraduate teachings would be applied into creating a final product. Our project can be broken into various subsystems, the main three being the PCB subsystem, the software subsystem, and the electrical grid subsystem. From there, these can be broken down even more. For example, within the PCB subsystem, we can further divide it by considering the microcontroller itself. In terms of workload, everything was divided up so that each team member can focus on his/her area of specialty, with the ability of using each other and our review board as consultants. In some areas, however, we work together as they were too complex in scope for one person to manage in our time frame. In this chapter, we will discuss the various technologies we considered and explain how we arrived at our final choices through weighing the pros and cons of each technological choice for the subsystems in our project.

3.1.1 Microcontrollers

The most important part of our project is that there are certain requirements it must meet to be useful for our purposes. Most importantly, the microcontroller must come with on board memory that can be flashed with our trained machine model and the code to execute it properly. Of high importance as well, our microcontroller must have a high-quality analog to digital converter with high precision and operating ranges. We will also require a high-resolution comparator to compare our resultant values with our threshold values. On the other hand, since our pulses will be every couple of seconds, a highly accurate pulse generator circuit is not a priority, and we believe that any of the boards in the table below have sufficiently accurate timers for our purposes. Similarly, we will require digital I/O pins to send these pulses. But this is also not of high importance, as all the boards in the table below have enough digital I/O pins for our purposes.

Furthermore, cost is a big concern as well, mostly due to the teams' desire to replicate standard industry procedures where cost is a very important factor to consider. Lastly, this project is new territory for the team, and as such, we wanted something with extensive documentation and support infrastructure. This allows us to reverse engineer existing solutions or consult the internet for any problems. With that said, we will consider each of the microcontrollers defined in the table below, and highlight their pros and cons.

Table 2: Microcontrollers under consideration

Microcontroller	Onboard Memory	ADC precision	Timers	Digital I/O Pins	Cost
ATMega328p	32Kb	10-bit	Two 8-bit and One 16-bit timers	23	$3.03
MSP430FR6989IPZ	128Kb	12-bit	Five 16-bit timers	11	$9.34
MSP430G2553IN20	16Kb	10-bit	Two 16-bit timers	24	$3.02

3.1.1.1 ATMega328p

Our leading candidate, the ATMega328p by Atmel is best known as the microcontroller that is present in the ever-popular Arduino boards. For a third of the price of a MSP430FR6989IPZ, this board features 32K bytes of in-system self-programmable flash memory. This is more than enough onboard memory for our purposes, our program will tentatively consist of our trained model along with any associated logic to use said model.

The ATMega328p features 10-bit precision on its analog-to-digital converter. While this is less precise than that of the MSP430FR6989IPZ, we believe 10-bits, which provides 1024 unique levels, is more than precise for the voltage levels we will be working at. We will be providing establishing fault thresholds that will not be challenging the resolution of this 10-bit analog-to-digital converter. It also provides temperature measurement, as of now we have no intention of using it, but it may prove useful as a secondary measurement criterion to determine a fault, as temperature spikes are a common symptom of a power grid fault.

Figure 7: ATMega328p

Regarding the singular 16-bit timer present on the ATMega328p, since we will be sending pulses every couple of seconds in our project. The 16-bit timer on the ATMega328p is more than adequate for our purposes, the two 8-bit timers may also prove to be useful if needed and have enough resolution for any foreseeable use case. Thankfully, the ATMega328p also comes with 23 digital I/O pins, which may be used in our project to connect peripherals or most importantly, additional injection lines to detect faults at other locations. Also, the ATMega328p comes at a third of the price of the MSP430FR6989IPZ, which makes it a much more attractive proposition, and while the MSP430FR6989IPZ has more accurate peripherals, as stated before, the level of precision it provides is not needed for the purposes of our project. Finally, as stated before, we wanted something with robust third-part documentation as well as heavy documentation by the company itself. The ATMega328p may just be the most documented microcontroller in production, with a wealth of information on the internet for just about anything.

3.1.1.2 MSP430FR6989IPZ

Under heavy consideration, the MSP430FR6989IPZ by Texas Instruments is a microcontroller that sees plenty of industry usage due to its reliability, price to features ratio, and heavily documented by Texas Instruments and third parties. This microcontroller features 64kB of memory, which is considerably more than what we will need. Most importantly however, it features an Analog to Digital Converter with 12-bit accuracy, however we don't believe that this level of accuracy is needed for our purposes.

Figure 8: MSP430FR6989IPZ

The MSP430FR6989IPZ has a couple of strengths and weaknesses over the ATMega328p and the MSP430G2553IN20. As stated above, it has considerably more memory and a higher resolution Analog to Digital Converter. However, its lower amount of GPIO pins is not a desirable property for the purposes of our project, as a lower amount of GPIO handicaps how many injection sites we can interact with. Its price is also three times higher than the ATMega328p and the MSP430G2553IN20, which also does not make it a lucrative choice. Developing a PCB to fit the MSP430FR6989IPZ would also be considerably more difficult due to its microprocessor-like design. Overall, the MSP430FR6989IPZ is a great chip, but we will not be needing the capabilities it provides, therefore that is why we chose the ATMega328p.

3.1.1.3 MSP430G2553IN20

Our third candidate for our controller is the MSP430G2553IN20, this one is the most comparable in peripherals and capability to the ATMega328p. It comes with two 16-bit timers instead of one 16-bit timer like that of the ATMega328p. However, we really only need one very accurate timer for our purposes. On the other hand, it only brings 16kB of memory, which while it might be sufficient, for close to the same price the ATMega328p comes with double the memory at 32kB. We prefer to not be handicapped by memory, something that we think will be critical considering we will be uploading the trained model onto the micro controller itself. Its other specifications are comparable, it comes with a 10-bit Analog to Digital converter like that of the ATMega328p, and it also comes with one extra I/O port. However, that is not nearly as important to use as double the memory so therefore it makes more difference in our choice. Overall, it is a solid micro controller but like we stated in the previous section, we believe the ATMega328p will serve us better with its heavy documentation by first and third parties.

Figure 9: MSP430G2553IN20

3.1.2 Fault Location Techniques for Distribution Systems

Fault location and detection techniques in the distribution system are the processes to identify faults so that operators can take appropriate actions to remediate the negative impact. Different fault location techniques are used in the distribution system based on the availability of measurements from the infrastructure within the regions and the application's capability. These measurements typically include power flow analysis of various electrical parameters such as voltage, current, real and apparent power. Then, appropriate mathematical algorithms are applied to analyze the characteristics of the outputs to determine the presence of a fault and its location. The methods can be divided into three categories: methods that are based on traveling waves, methods that use high frequency components of currents and voltages and methods that use the fundamental frequency voltages and currents measured at the terminals of a line. [1] Review papers provide a great overview of the existing fault location techniques and their performance analysis in distribution systems with and without distributed generation. They also compare different types of techniques for their advantages and limitations in search of a better approach. A reasonably accurate fault location indication would expedite the restoration process.

Given the low voltage and single-phase nature of our hardware design, we faced numerous limitations when it comes to choosing the right fault location techniques. Most of the popular fault location techniques, including the one used by ADMS, are designed for more complex 3-phase systems with intricate topology. Given our goal of creating a design that can be scaled up, we must choose or come up with more accommodating detection techniques that will work with the PCBs we discussed above. Fault location and detection techniques are important components of this project because they determine the approach of our machine learning algorithm and PCB design.

Table 2: Fault location techniques for distribution systems samples

Method	advantages	disadvantages	Used for
Distribution Management System (DMS) Based Fault Location.	Excellent environment for further processing.	limited to short circuit faults.	Most utility use this in DMS.
Fault Location Using Tapped Load.	Not affected if the fault current at a fault locator is not in phase with the current at the fault.	for heavily tapped feeders, the accuracy may degrade toward the end of the feeder.	Lateral with many customers.
Fault Locator for Radial Sub transmission and Distribution Lines.	Optimize for radial line because it considers their unique characteristics such as linear structure and one-directional power flow.	Limited capability to loop system.	Radial distribution line.
Fault Location Method for MV Cable Network.	Take into Phase-phase fault-loop and Phase-ground faulted loop.	More complex.	n/a.

3.1.2.1 Distribution Management System (DMS) Based Fault Location

The DMS-based method utilizes Supervisory Control and Data Acquisition (SCADA) to obtain measurements from sensors deployed all around the distribution grid. By choosing to rely on real-time data from sensors throughout the distribution grid and preforming offline engineering analysis based on the network, this method avoids the need for developing a network model and does not rely on special devices at the network level. By comparing the calculated and measured fault current from microprocessor-based relays, it can pinpoint the basis of a fault location with real-time topology information. The triggered relays can be linked to the incoming bay of the substation, and an algorithm then is applied to identify the specific feeder connected to the substation where the fault is located A feeder is a mainline that connects the substation to an area, DMS will show the feeder on which the fault is located but it would not show the lateral or loop beyond the immediate area.

The DMS based a well-tested and trusted method within the utility industry due to its good accuracy and reported reduction in outage duration. Because the output is not explicit during the processing stage on DMS, it provides an excellent environment for further processing, the fuzzy logic can be applied after. It is limited to short-circuit faults. However, it is important to acknowledge that the DMS-based method is primarily suitable for identifying short-circuit faults and may have limitations in detecting other types of faults.

3.1.2.2 Fault Location Using Tapped Load

Tapped loads are electrical loads that are connected to the distribution network at specific points, often through transformers. These loads can significantly influence fault detection and estimation due to their connection configuration. Because tapped load impedance is larger than the feeder impedance, grouping them helps to compensate for the missing load representation on the model. This method calculates the load impedance using pre-fault and fault voltage and current, which gives it immunity to the effects of load current and fault resistance. By leveraging the information obtained from the tapped loads, the fault location technique can provide valuable insights into the distribution system's fault characteristics.

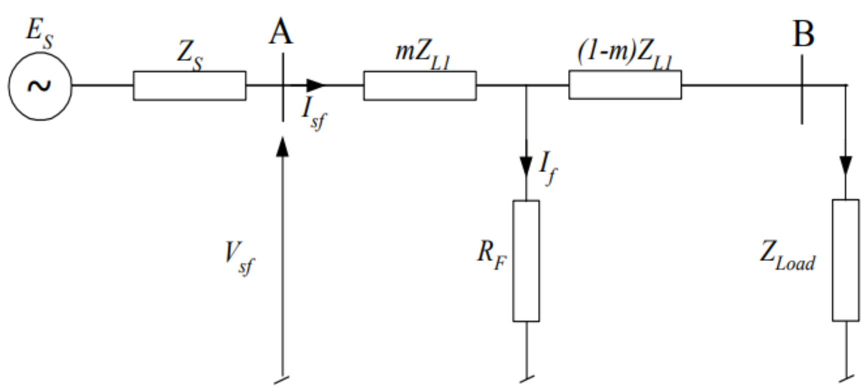


Figure 10: Equivalent scheme of the distribution feeder

The advantages of the tapped load method are fault location and resistance estimation algorithms are adjusted for additional tapped load which leads to the accuracy and effectiveness of fault analysis.

3.1.2.3 Fault Locator for Radial Sub transmission and Distribution Lines

The fundamental frequency voltages and currents measured at a line terminal before and during the fault can be used for fault location solution. Measurements of voltage and current obtained from monitoring devices installed at various points along the sub-transmission and distribution lines serve as inputs to the fault locator system. The type of fault and phasors of the sequence voltages and currents used can help narrow down the parameter. Then the lateral beyond the fault parameter can be reconsolidated for ease of calculation. The distance to fault can be expressed via the mathematical model assisted with this method as a fraction of the length from the parameter.

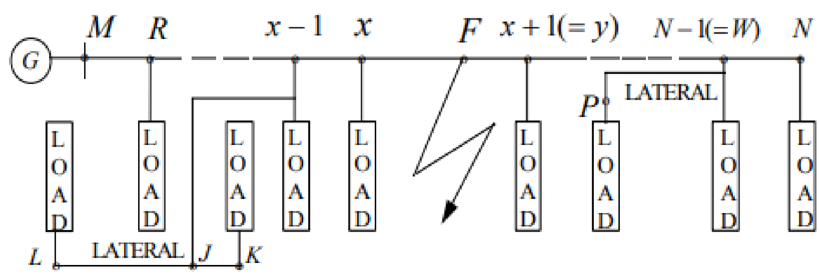


Figure 11: The single line diagram of a radial line experiencing a fault at F

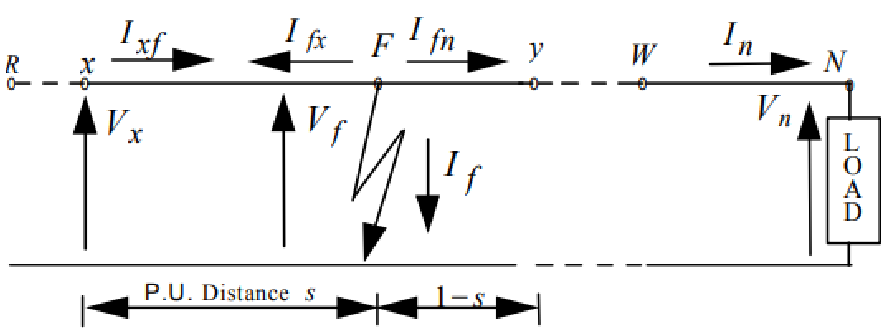


Figure 12: Fault voltages and currents at nodes F and N

This technique might provide multiple fault locations depending on the presence of laterals. At this point, the feeder had already been identified, but the lateral can be narrowed down by analyzing the fault indicators downstream.

The advantage of this method is it is optimized for radical lines because it considers their unique characteristics such as linear structure and one-directional power flow. This property also limits its applicability, especially in networks with complex topologies. And this method becomes too measurement-dependent when multiple laterals are presented. However, it is essential to acknowledge the limitations of this method, particularly in networks with highly complex topologies. The method's effectiveness can be compromised when dealing with networks that exhibit intricate interconnections and multiple laterals.

3.1.2.4 Fault Location Method for MV Cable Network

This algorithm uses the fundamental frequency voltages and currents measured at a line terminal before and during the fault. First, the fault-loop impedance is calculated the fault-loop impedance by utilizing the measured voltages and currents obtained before and during the fault. Second, the impedance along the feeder is calculated by assuming the faults at each successive section. By comparing the measured impedance with the calculated feeder impedance, an indication of the fault location can be obtained. [1] The positive sequence fault loop impedance can give us insight into the phase-to-phase faulty loop and phase-to-ground faulty loop. In the next step of the algorithm, the impedance along the feeder is calculated by assuming faults at each successive section. By comparing the measured impedance with the calculated feeder impedance, valuable indications regarding the fault location can be derived. This comparison helps identify discrepancies between the actual measured impedance and the expected impedance based on the assumed fault locations, thereby providing insights into the precise fault location within the distribution system.

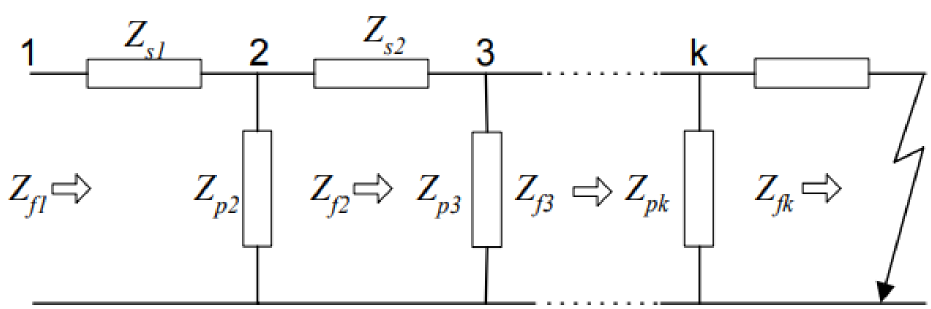


Figure 13: Diagram of the network: measurements are taken in the faulty feeder.

This method is the most versatile out of all the methods we had to seem so far, because it can adapt to a radical MV system with intermediate load taps and non-homogeneity sections.

3.1.3 PCB Design Software

One of the main aspects of our project is our PCB. On it, we will have all the circuitry that will allow us to send pulses, get data, analyze it, and give an answer. It will also have different circuits that will protect all the physical components within the board. Some examples are the voltage and current regulators. For this reason, we must choose the best and most efficient PCB software. We also must consider that there is a limited time to finish this project, and we need something that can have a good balance between its functionalities and ease of use.

Table 4: PCB design software

Software	Features	Compatibility	Price	Simulation
Altium Designer	Advanced range of capabilities and tools. More complicated to use.	Windows	Free (student license)	Advanced simulations
Eagle	Great range of capabilities and tools Ease to use.	Windows, Linux, MacOS	Free (student license)	Advanced simulations
KiCad	Open source Simple capabilities and limited tools	Windows, Linux, MacOS	Free	Basic simulations

3.1.3.1 Altium Designer

Altium is probably the most powerful software out of all three, it is the one with most features for the 3d modeling, which can be very helpful when you have a limited size available only, and you need to be extra careful on the dimensions, this can even help you to route your wires in a more personalized way. It costs $355.00 /mo., but with their student licenses, we can get it for free. It has a built-in SPICE simulation that allows you to access different analysis tools, including circuit performance, signal integrity, and DC power systems. Altium is only compatible with Windows, and it has a bigger learning curve because of its complexity. This fact is critical because we already have experience in other software, and learning to use it will delay us and our time is very limited. Although it is the most complete software, it is also the most complex out of all three mentioned, and we can get the features that we are looking for in other simpler programs which can be more efficient.

3.1.3.2 Eagle

One of the most important things for this project is efficiency because we need to do as much as possible in the shortest time possible. Eagle can provide that because it is a complete PCB design software as well as being user friendly. Also, our team already has experience with it, which will facilitate its use. It costs $70.00 /mo., but we can use the free student license. Circuits can be tested and simulated with the SPICE simulator, and it has many features that facilitate the work, such as real-time design synchronization, intuitive alignment tools, obstacle-avoiding routing, and many others. Eagle is available for Windows, MacOS, and Linux. Another advantage is that it directly connects online when doing the Bill of Materials (BOM), and you can see prices and availability updated for all components. Eagle is more efficient compared to other software. It can help us reduce the time allocated to building the PCB while providing all the tools necessary.

3.1.3.3 KiCad

KiCad is an open-source and entry-level software with limited availability of tools. You can install it on any device for free. Compared to other options, KiCad is very basic. You can design your schematic and PCB design as in the other, but it has fewer libraries and features. KiCad's schematic editor is very user-friendly, and the SPICE simulator can verify your schematic and check for electrical rules. You can create a PCB and a 3D model, but it is harder to use due to its lack of features like auto-routing. Also, it is more difficult to create a BOM because it doesn't connect with sellers as well as having a non-intuitive user interface. Overall, it is great software, but it lacks some features to make the PCB designing process more efficient.

3.1.4 Machine Learning Methodology

Processing our model properly is a very important part of our project. It is how we will determine where the fault is. We plan on having the trained model uploaded to the controller. This makes having a well-trained model a critical priority. In the section below, we go over our choices for our model as well as why we decided to go with the model highlighted in the chart below. We wanted to especially focus on how simple the methodology is, as we are worried of getting lost in something we just are not comfortable with. However, due to our ability to reach out to experts, we believe we can push for something more complex and consult the experts should we need any help. Thankfully for our project, the model itself is not complicated.

Table 5: Machine learning methodologies

Model	Features
Linear Regression	• Simpler. • harder to apply to dynamic systems
Neural Network	• More complex. • Works well with dynamic systems

3.1.4.2 Neural Network

We chose to use a Neural Network based model for our project due to our system being a dynamic system. Neural Networks are "considered to be well-suited at solving a wide variety of problems" [2], They are also heavily documented, "with plenty of user-friendly software frameworks" [2]. As with everything, there are always pros and cons. We identified several pros which were crucial to our project's success, as well as some cons that we considered which ultimately do not hamper the project significantly.

Our leading architecture that we plan on using – DynoNet, is based on a Neural Network system. We will be getting into much more detail in section 3.2, however, at a high level. It is a model specially made to model dynamical systems, which is exactly what the power grid is. Of note as well, one of the experts we consult has experience using DynoNet and has told us that it is the perfect model for the purposes of our project.

Neural networks have become the go-to standard for machine learning solutions, they are well suited for handling nonlinear data whereas linear regression cannot unless you apply feature engineering on the nonlinear variables in the data set. This is perfect for us since our data set, being a slice of the power grid, is not a linear system due to the ambiguity of the load. This was the critical requirement we needed, so therefore that is why we chose it. They are also able to learn from the dataset given it is of sufficient quality. Given our small dataset, we do not expect our model to do a lot of self-learning, however, it may possibly be able to, this is yet to be known and will be clarified once we have our working model and can parse through the set.

Neural networks do have some downsides however, modeling a biological system in a digital system lends itself to very complex structures and code. Therefore, Neural Networks are significantly harder to use as a solution, and require more knowledge and time spent learning it. Another downside of using Neural Networks like DynoNet is that they require real-time data and provide a real-time output, meaning our data would need to be a time variable rather than a singular value. Thankfully, we have experts whom we can consult on these topics. Below is an image depicting a simplified representation of a one-dimensional Neural Network, one can see the heavy abstraction that needs to happen in the middle layer thus making Neural Networks a “black box” like structure, which can complicate our fundamental understanding of how our system works.

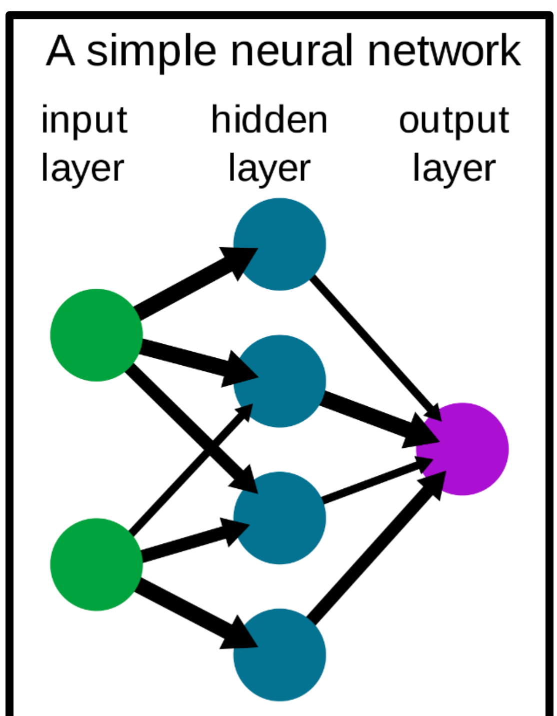


Figure 14: Depiction of a simple neural network

3.1.4.3 Linear Regression

Linear Regression was one of our other leading candidates. However, upon further research we discovered that while Linear Regression models are easier to work with, they do not apply well to dynamic systems without extensive feature engineering. We believe this to be out of our skill scope and would prefer to use Neural Networks which are also highly documented, and while they may be more complex, as stated in section 3.1.4.2, we have an existing architecture that is very well suited to fulfill the demands of our project. In the image below, one can notice why Linear Regression does not apply well to dynamic systems. If one were to imagine datapoints with no linearity, then the line of regression shown below would not provide much meaningful information.

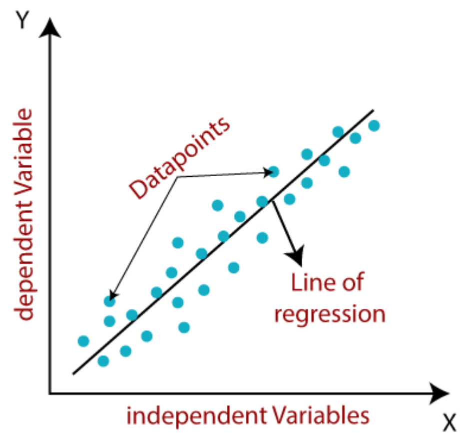


Figure 15: Depiction of linearity of data points

Of note however, our current architecture that we plan on using to train our model does feature properties present in Linear Regression techniques, mainly, "linear dynamical operators are combined with static (i.e., memoryless) non-linearities which can be either elementary activation functions applied channel-wise; fully connected feed-forward neural networks; or other differentiable operators." [2].

3.1.5 AC Power Supply and relay

The Power supply is a crucial part of this project because it provides the necessary power to multiple components of this project. It encompasses the PCB to enable our algorithm, voltage injection pulse generation, and the simulated one-line distribution grid representation. The PCB will receive DC power while the distribution system will need AC power. So, we will be using an AC power source along with a DC to AC converter. The distribution system in the traditional U.S. grid is made up of feeders and laterals, and with power generation units connected to the transmission and then send to the distribution network in different regions. With a keen focus on distribution-related data, our project necessitates a direct connection between our power supply and the distribution mainline, enabling us to analyze and optimize distribution-specific parameters. Normally the voltage level going directly into the feeder is between 11kV and 69kV AC, after a few stepdown transformers the standard residential voltage of 120V that we commonly encounter. While our objective is to establish a faithful representation of the distribution system, we prioritize safety. We will explore different power supplies and incorporate robust protection mechanisms to ensure the well-being of our team members and the integrity of the project's infrastructure.

Table 3: AC power supply and relay options

Power supply	Characteristic	Advantages	Disadvantages	Cost
12V single phase power via function generator	Tektronix AFG3022C Arbitrary/Function Generator is a great option to generate a single-phase 12V sine wave input for the power system. Tektronix 013-0345-00, coupled with a 0.125A fuse can limit the currents to 0.125A.	1) The function generator allows us to control the peak-to-peak voltage level, phase angle, etc. 2) We are experienced with the function generator from the previous class	1) The fuse adapter we will be using to simulate a relay is very expensive 2) We don't have the option to do three-phase hardware modeling	Free (can be found in the senior design lab)
120V three phase wall outlet with relay	Power the system with three 120V three phase wall outlet and protects the system and personal with SEL-551 Overcurrent/Reclosing Relay from Schweitzer Engineering Laboratory	1) The best way to access stable three-phase power, three-phase is more common in the distribution network in real life	1) The high voltage level can be dangerous 2) We need to use the unbalance load techniques to design for three-phase while the scope of the project is to detect one phase ground fault only	Free (can be found in the protection and control lab)
California Instruments MX22.5	12V single phase sine wave signal by California	1) A good single phase low voltage power supply	1) Current can be dangerously high because of the big	Free (can be found in microgrid control lab)

amplifier (12V single phase)	Instruments MX22.5 amplifier	with built protection and a user-friendly interface	potential difference across the amplifier Extremely expensive to replace if we break it. 2) Our testing with modified equipment might interrupt our research project	
California Instruments MX22.5 amplifier (120V three phases)	120V three phase sine wave signals by California Instruments MX22.5 amplifier	1) A good three-phase power supply with built-in protection and a user-friendly interface	1) Too many moving parts could lead to different types of faults which is beyond the scope of this project 2) Extremely expensive to replace if we break it 3) Our testing with modified equipment might interrupt our research project	Free (can be found in the microgrid control lab)
chroma programmable ac source 61704	0-300V 3 phase programmable AC sources	1) It combines synchronized 3 phase with low voltage 2) It is programmable so we can	1) The programmable aspects make it more complicate than the regular signal generator	Free (can be found in the microgrid control lab)

		change the load and phase easily 3) By learning SCPI for this machine we could effectively use most research and industry grade signal generator	2) The peak current can go up to 72A which is super dangerous to handle	

3.1.5.1 12V single phase power via function generator

The reason why we are considering a 12V AC system, despite it not being classified as high voltage, is primarily due to our emphasis on safety. Electric shock at 12V systems is much easier to remediate with appropriate personal protection gear such as insulated gloves and goggles. Additionally, when it comes to short circuits or overloading, it is less likely to spark and cause a fire from excessive current flow at lower voltage. Our team members have undergone comprehensive training and possess a good level of hands-on experience in handling systems at this voltage level, ensuring they are well prepared to address any potential risks.

Distribution grids commonly consist of a combination of three-phase and single-phase systems. Because the 3-phase system provides the most balanced and consistent power, at a reasonable cost. And single phase makes the system simple to manage and, in some cases, helps to transport power over a longer distance because of its spreading out/flexible nature. Single-phase power is adequate for lower loads in most residential applications. The main disadvantage of single-phase power is the possibility of unbalanced loads where one load might vary greatly compared to the other ones on a series, causing voltage fluctuations and inefficient use of electrical energy. Voltage flickering is one of the most common issues with a single phase, it is referred to the small variations or rapid dips and surges in the voltage waveform. This phenomenon is mostly caused by the variation in load, especially in a rural residential area with the radical line where load might change drastically from time to time with very little power remediation plan. Grid disturbance is another cause for voltage flickering, being able to identify false quickly can minimize voltage flickering due to grid management and operation. Another disadvantage is the reduced power factor, the ratio of working power which reflects the efficiency of the system, measured in kilowatts (kW), to apparent power, measured in kilovolt amperes (kVA). A low power factor means the system is inefficient, and if the power demand doesn’t match the power supply it could put strains on the generation control and leads to power loss due to load shedding. Usually, you will find three-phase power in the feeder and more commercial areas, and a single phase can be found in the lateral that’s mostly residential. Based on the fault detection method, this project is more applicable in radials which are in a generally more remote area, so we primarily focus on single-phase power supply demonstrations. While this allows us to implement our algorithm for single-phase ground faults, it should be noted that we will not be able to demonstrate phase-to-phase or three-phase ground faults in this project regardless of our choices of power supply.

Tektronix AFG3022C Arbitrary/Function Generator is a great option to generate a single-phase 12V sine wave input for the power system. This single-phase 12V input will be connected to the feeder side of the hardware design to simulate power from the substation traveling down the mainline.

Figure 16: Tektronix AFG3022C Arbitrary/Function Generator

The signal generator should allow us to observe and control the current flowing through the circuit. To enhance equipment protection and ensure safety and to mimic a real-world power system with relays, we came up with two other methods to provide extra protection for the equipment and ourselves, a theoretical relay and a fuse with adaptor attached to the signal generator. A theoretical relay is a specific algorithm that can be used to detect faults at the power supply level and suppress excess current levels. We can use MATLAB to create blocks that detect high current and power levels. A fuse will allow us to implement a current restaurant, the fuse can be connected to the signal generator via a fuse adaptor such as Tektronix 013-0345-00, coupled with a 0.125A fuse can limit the currents to 0.125A, making it a lot safer to handle. This fuse adaptor can be built with available materials within the senior design lab. This is our chosen method because it prioritizes safety and gives me very direct control of our system input.

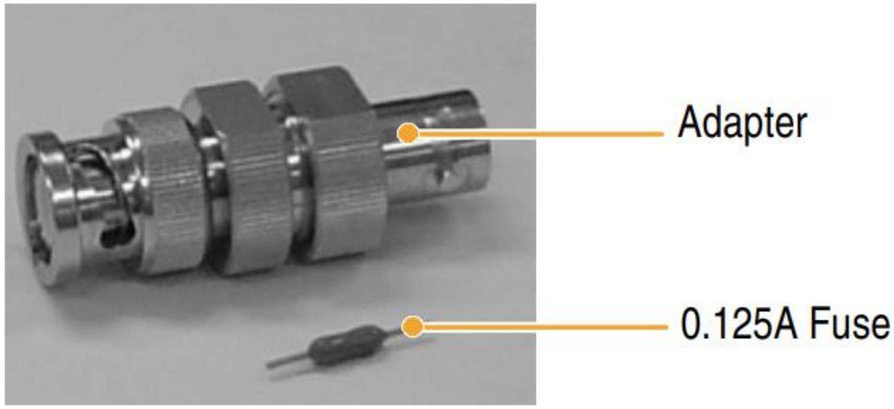


Figure 17: Tektronix 013-0345-00 fuse adapter with 0.125A fuse

3.1.5.2 120V three phase wall outlet with relay

In addition to considering a 12V AC system, we are also considering powering this project with 120V three-phase power from a wall outlet. Because it is an accessible way to have synchronized 3-phase power as our power supply, which closely aligns with the power supply typically found in distribution grids.

The use of 3-phase power in the distribution grid is more common, and there are wider range of fault types associated with it. Phase-to-phase faults and three-phase ground faults are possible with the 3-phase system. However, a more advanced fault detection algorithm is needed for this kind of fault, which would be outside of the scope of this project. Three phase system is usually preferred because of its higher capacity, better balance, and increased efficiency. Although single-phase infrastructure is cheaper per unit, a three-phase system's benefits are much more prominent thus making it a better choice in terms of cost-effectiveness in most situations.

In a 3-phase system, the power is distributed evenly across the three phases due to the 120-degree phase differences, resulting in balanced power delivery to loads. This helps prevent voltage imbalances and reduces the risk of power fluctuations or voltage flickering. The 120-degree phase differences ensure a spread-out rotation of 360 degrees, and they offset each ensuring there is always one phase peaking, ensuring voltage consistency with AC sine function. In theory, more phase means a lower voltage flickering risk. But systems containing more than 3 phases are very expensive to build and the torque within a 3-phase system provides good support to the generator and motors, eliminating oscillating torques that may cause damage. The efficiency of power generation and transmission is also improved in three-phase systems, as the balanced power flow reduces losses and enhances overall system efficiency. The three-phase power generation produces a smoother and more balanced power flow compared to single-phase systems. This results in reduced power losses and improved overall efficiency. Having three phases also means that it will have a higher capacity when transporting power.

Because we will be simulating a one-phase ground fault with the fault location algorithm, we can use the unbalance load techniques to simplify the design and minimize moving parts. In this approach, we can use Thevenin's theorem to calculate the impedance in the phase we intended to simulate the fault at and match the load by using the resistor and inductor in the other two phases shown below.

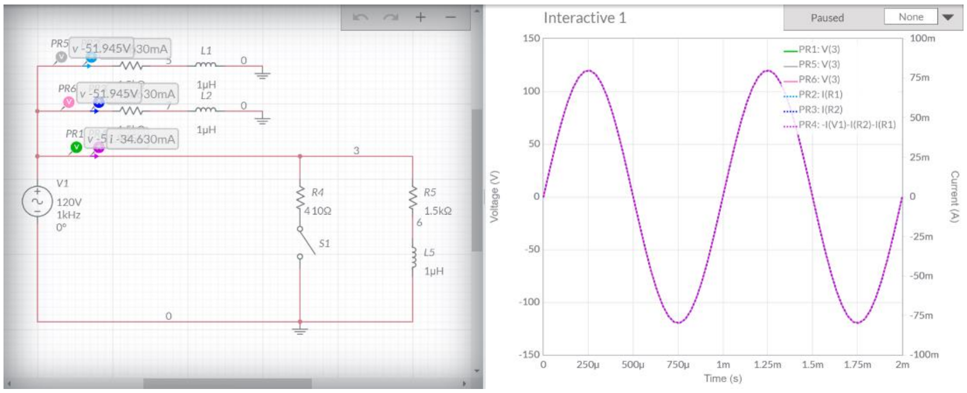


Figure 18: Multisim model with 3 phase 120V normal operation/before fault

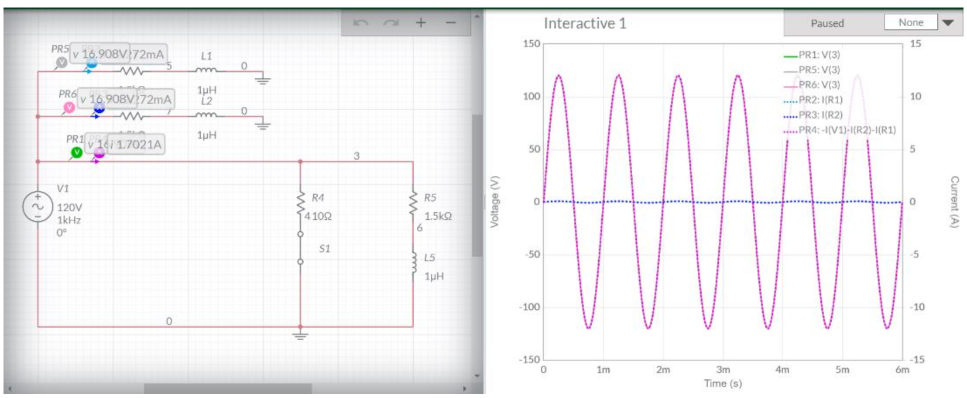


Figure 19: Multisim model with 3 phase 120V at ground fault

To use the 120V three-phase wall outlet we will incorporate a relay module to protect us and the equipment. Protection and control are a big part of power system engineering, it is based on ensuring the safe and reliable operation of power systems. It aims to detect and respond to abnormal conditions, faults, and disturbances in the system to minimize damage, protect equipment, and maintain system stability. Working with 120V can be hazardous, a relay module can help detect abnormal current surges and unplanned faults.

The SEL-551 Overcurrent/Reclosing Relay from Schweitzer Engineering Laboratory can monitor and protect distribution feeders, distribution buses, transformers, capacitors, or circuit breakers. It also can communicate with an automatic recloser to shut down the system if overcurrent is detected. In this project, the relay will protect us and the equipment in all three phases, ensuring our safety and safeguarding the equipment from power surges or faults. This can effectively prevent overheating the elements during the planned fault and electrical fire.

Figure 20: SEL-551 Overcurrent/Reclosing Relay from Schweitzer Engineering Laboratory

3.1.5.3 California Instruments MX22.5 amplifier (12V single phase and 120V three phase options)

The California Instruments MX22.5 amplifier is a versatile and powerful tool that can be utilized for programming power sources. It can deliver up to 22.5 kVA in a single phase. The amplifier can operate in both single-phase and 3-phase modes, offering flexibility in power output configurations. It supports AC, DC, or AC+DC mode, allowing for different types of power delivery. The programmable nature gives sit a big range where can stay within low voltage for our demonstration purposes. In this project, we can utilize it to produce either 12V single-phase or 120V three-phase power, depending on the specific requirements of our system. This is a popular model for research purposes because of its regenerative nature. Automatic crossover between source and sink power modes offers regenerative capabilities in AC or DC mode. The power sources can regenerate up to 85% of the rated output power back to the utility grid during sink mode operation when equipped with the -SNK or -SNK-DC option.

A MX22.5 amplifier is currently connected to the UCF microgrid, which runs on 480V phase to phase to the R1-154 microgrid control lab. The software interface package provided with the amplifier allows us to set the voltage and activate the amplifier. In the Stead State control tab, we can enter the line to neutral voltage for any of the phases or all three phases. We can also define the waveform shape, for this project, we will stick with the unclipped sine wave. In the same interface, we can also set the current limit, this can act as a relay device. This can protect our circuit and the power sources, especially in the 12V Scenario as big voltage drop can lead to a dangerously high current.

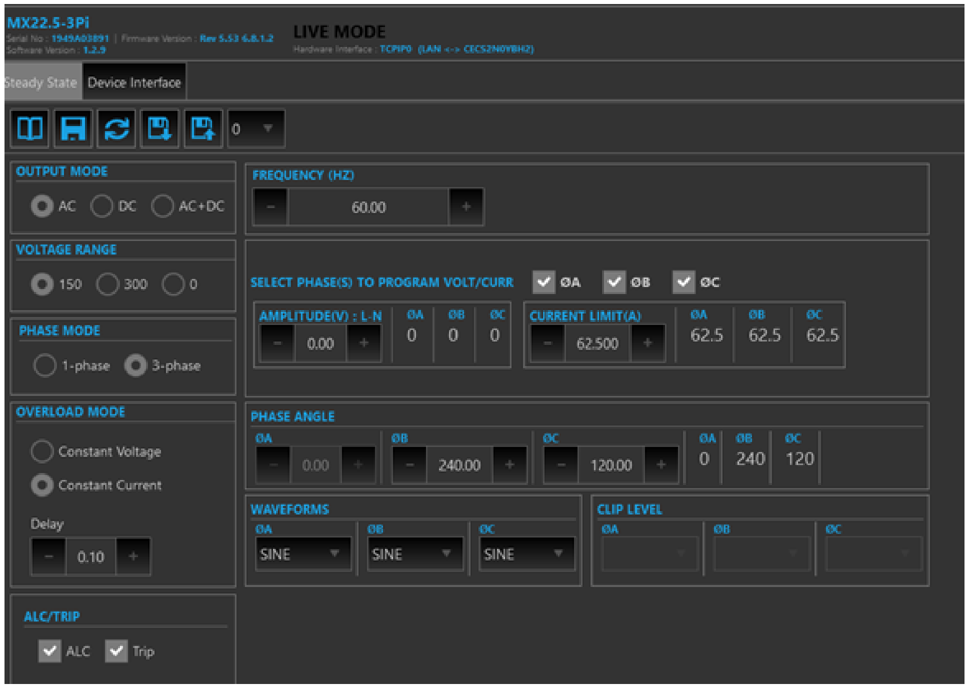


Figure 21: Steady State tab in MX22.5 control interface

The MX22.5 amplifier supports SCPI (Standard Commands for Programmable Instruments) is a programming language for controlling instrument functions over the IEEE-488. SCPI commands are frequently used for control when testing with the MX22.5 to bypass the interface. For example, we can use a simple command to identify and reset the source to reset voltage and current.

5.4 *IDN?

This query requests the source to identify itself. It returns the data in four fields separated by commas.

Query Syntax	*IDN?	
Returned Parameters	<AARD>	
	Field	Information
	California Instruments	Manufacturer
	xxxxxx	Model number and letter
	nnnnnn	Serial number or 0
	Rev. xx.xx	Revision level of firmware
Example Series I	"CALIFORNIA INSTRUMENTS,MX45-3, 12345, Rev 1.0"	
Example Series II	"CALIFORNIA INSTRUMENTS,MX45-3, 12345, Rev 4.0"	

Figure 22: Example SCPI commend to requests the sources to identify itself.

One of the notable advantages of the MX22.5 amplifier is this device responds extremely fast. Once activated, the wave type and voltage level update instantaneously. This is a big advantage because we only have 15 mins to demonstrate this project. as Many high-power input sources typically require a long time to ramp up, but the MX22.5 amplifier provides instantaneous adjustments, ensuring a smooth testing and demonstration environment.

Because this is a commercial product design for research and development, the interface is user-friendly and there are comprehensive documentations for SCPI commands.

However, it is important to acknowledge some of the challenges associated with using the MX22.5 amplifier. One of the biggest disadvantages is the complication of having 480 V input can pose risks when working with such voltage levels, even with the safety features provided by the amplifier. We will need to explore alternative input sources, then we reconfigure the hardware already attached to the lab equipment. This includes but is not limited to cutting the sleeved cable and replacing it with a cable for lower voltages and attaching it to more adequate power sources. This might also interrupt other power system research projects on the UCF microgrid.

Overall, the California Instruments MX22.5 amplifier offers significant benefits in terms of power generation and control. Its programmable nature, regenerative capabilities, and fast response time make it a valuable asset for our project. However, careful attention must be given to safety considerations and potential modifications required to ensure compatibility with lower voltage levels and other ongoing research projects.

3.1.5.4 Chroma programmable AC source 61704

Chroma programmable AC source 61704 is a power generation equipment that can generate a synchronized 3-phase signal. Different from the other 3 phase sources we mentioned above, this one is the most programmable, because it is mostly used for testing different scenarios such as unbalanced phase or unbalanced load. Compared to other programmable sources, this one can deliver pure 5 wire, 3-phase power at a lower cost. This is because it provides a lower voltage range in the context of power system testing with a range of 0-300V. This is an advantage for us because we would like to work with lower voltage for safety reasons. However, this will not be a part of our project budget because we will have access to it in the Research 1 lab.

Figure 23: Chroma programmable AC source 61704

There is a system interface for programming the output voltage and current, users can also set the phase angles ranging from 0-360 degrees. It also has a wide frequency output from 15 to 1200Hz, the most common one being 50Hz or 60Hz. Similar to the California Instruments MX22.5 amplifier we discussed above, the Chroma source also has the GPIB interface taking SCPI command, which allows easier and more precise control.

It is compatible with the RS-232C interface which is the physical interface and protocol for common use in communication between computers and related devices. This is curial for this project because we will be using internal measurements from the Chroma sources and external measurements by the voltmeter and ammeter for our mathematical model. The data can be transmitted from the PC to an internal or external modem from its Data Terminal Equipment (DTE) interface with a Universal Asynchronous Receiver/Transmitter (UART) chip.

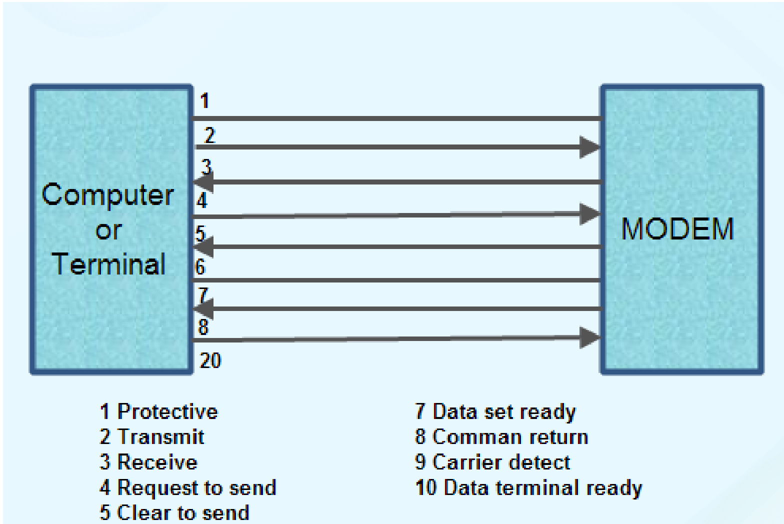


Figure 24: RS-232C connections

The A617001 soft panel software is an optional software interface that provides the ability to synthesize harmonic waveforms and store them in the memory of the AC source.

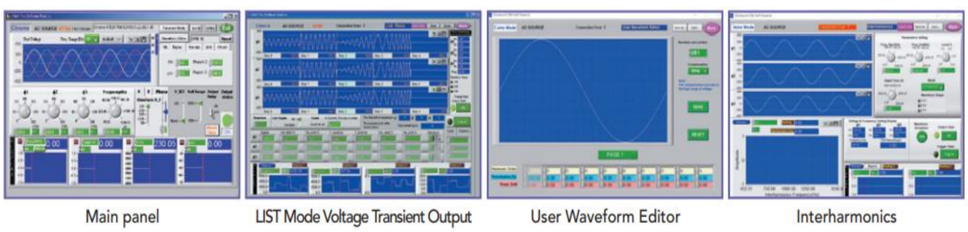


Figure 25: A617001 soft panel software

This machine requires a lot of programming, and we don't have any help to code on, so the learning curve would be very steep, and we will have to take extra precautions to make sure we don't break the AC source. Depending on the voltage level we want to work with we might also need to find the cable of the right size and stripe and reinsulate them. This is an industry-grade machine, by learning how to operate this machine we can effectively use any machine that has the GPIB interface. Although we can impose a current limit on these programmable sources with the SCPI command or its built-in interface, we are considering using the SEL-551 Overcurrent/Reclosing Relay from Schweitzer Engineering Laboratory as a protection and control method.

3.1.6 Comparison to Existing Distribution Technology

To build something that provides value to existing distribution systems, it is important to first compare our design to existing fault detection technology. To provide value to the system, we must recognize what type of technology is currently being used and how our design can provide additional utility to the distribution grid. With our design focusing on fault detection and location triangulation, the important devices to compare it to would be existing fault protection technology that can be found in the distribution and transmission fields. Some examples of fault protection technology include tripsavers, reclosers, and Gridpulse. Tripsavers and reclosers are found mostly in distribution systems, while Gridpulse exists at the transmission level.

Table 4: Existing Distribution Technology comparison

Device	Advantages	Disadvantages	Comparison
Tripsaver II Cutout Mounted Recloser	The Tripsaver II has a variety of voltage levels depending on the situation it is needed for. It is able to isolate parts of the grid and can restore power quickly if the fault was only transient.	Generally, only used for protecting lateral circuits. Cannot provide location of the fault outside of knowing which lateral is affected. Must be manually reset by an operator upon opening for a permanent fault.	The Tripsaver II unit is generally used for fault protection and system isolation where our design will focus more on detecting the exact location of a fault to reduce response time in the case of a permanent fault.
Recloser	Unlike the Tripsaver II, a recloser is often used in the middle of a feeder line rather than to isolate a lateral system. The recloser also has the advantage of being able to return power to a system in the case of a transient fault.	Increased cost compared to other fault protection devices. Reclosers can arc during fault interruption which may cause environmental or fire hazards.	Reclosers are fantastic tools for restoring power in the case of a transient fault, but when it comes to permanent faults the most, they can do is isolate the affected system. Our design is focused more on assisting the response once a fault has already occurred.
Gridpulse	Gridpulse’s versatility is the main strength, the ability to identify a multitude of hazards on the line and communicate to an operator of those hazards	Gridpulse is only able to be used with a transmission system	Gridpulse is focused more on constant monitoring and communication of the conditions of a system, while our design focuses on responding as quickly as possible

	means that any sort of problem on the line can be detected before causing a loss of power		once the issue has occurred.

3.1.6.1 Tripsaver II Cutout Mounted Recloser

A Tripsaver is a device that is attached to distribution poles with the purpose of clearing transient and permanent faults on the grid. The Tripsaver II is a single-phase device that operates similar to a fuse. It will close and reopen rapidly in an attempt to clear a transient fault, and if that doesn't work, the Tripsaver will determine that it is a permanent fault and will open completely. Upon opening completely, the Tripsaver is unable to continue closing itself, and must be manually closed by a line operator. There are two main voltage levels for tripsavers, 15kV and 25kV. The Tripsaver voltage used will depend on the voltage of the distribution line, with 15kV tripsavers commonly used in 13kV lines and 25kV tripsavers used in 23kV lines.

Some important details regarding the Tripsaver II are that because it is a single-phase device, multiples are needed to properly protect a multi-phase line. To protect a three-phase distribution system, three of the Tripsaver II devices are needed, with one Tripsaver attached to each phase in the line. Depending on the phasing of our design, we might need to take a similar approach. For our current single-phase ground fault design, only one device is needed for the line. If our device was ever expanded to check for 3-phase faults, an approach like the Tripsaver II would be taken, with one device on each phase of line. This way, each phase could be individually checked for faults to more accurately describe to the crew what kind of fault they would be dealing with. Another important distinction between the two designs is that the Tripsaver II does not have any sort of fault location detection capability, the device notices a fault, and opens to protect the line. For the crew coming to check for faults, they can see that the Tripsaver has opened and that will give them a general idea of where the fault may be, but because of how distribution systems are setup, the crew may have to analyze miles of distribution line before getting an exact idea of where the fault is. Our design will give the crew a much more precise approximation of where the fault is, reducing the work and time needed for the crew to identify and repair the fault.

Figure 26: The Tripsaver II Cutout Mounting Recloser in the closed position (left) and open position (right)

3.1.6.2 Reclosers

A recloser is a high voltage switch that will automatically shut off power when a problem on the line occurs, such as a short circuit. Reclosers are capable of sensing fault currents and interrupting them, then restoring power once the fault has passed. However, this is only in the case of a transient fault. In the case of a permanent fault, the recloser acts like a Tripsaver, where it will open and reclose multiple times, and if the device still detects fault currents, it will lock open and isolate the faulted area to protect the rest of the grid. A recloser is useful because in the case of a transient fault there will be no need for a crew to come in and restore power, the recloser protects the affected area so that power can be quickly and automatically restored. In the case of a permanent fault, the recloser does not restore power automatically, but it does assist the crew by reducing the size of the affected area since the fault will not travel past the recloser.

The recloser and Tripsaver function quite similarly, and the comparison of these two devices to our design will feature similar points. Although the recloser is great at identifying fault currents, shutting off power, and potentially restoring power if the fault was transient, it does not detect a precise location of the fault. It minimizes the affected area, but the crew will still need to patrol potentially several miles of wire before finding the cause of the fault. The goal of our device is to help reduce this search time by identifying a much more precise location of the fault. Instead of having to travel the distance between two reclosers, the crew would have a much shorter distance they need to patrol before finding the precise fault location and eliminating the problem.

Figure 27: Recloser in a Distribution System

3.1.6.3 Gridpulse

Gridpulse is a transmission line monitoring system that uses a multitude of sensors to monitor a transmission line in real time. It is powered by the transmission line itself and measures a variety of statistics including conductor temperature, current, voltage, wind speed and direction, ambient temperature, and even vibration. Gridpulse takes these measurements to detect sag and potential ice on the line so that it can send a warning to the operators before dangerous situations occur. Gridpulse and our project design are somewhat similar, in the sense that both are used as a system of protection to assist the operator with keeping the system stable. However, Gridpulse is mainly used as a pre-emptive detection device, while our design will focus on assisting the operator with response to an existing problem. Gridpulse focuses on letting the operator know when to be wary of dangerous conditions, while our device will let the operator know where the situation is occurring so that they can send a crew out to get the work done as quickly and as efficiently as possible.

Gridpulse Connect is a real-time dashboard that allows users to monitor the sensors deployed. The dashboard has adjustable widgets for easy adjustments based on the priority of the regions. For example, Florida would prioritize data related to hurricanes such as wind speed, and the northern region might prioritize icing and de-icing status. It updates the operator on the status of the sensor, and it also keeps historical data records which could be used for further data analysis for transmission planning. The data it provides includes the conductor temperature (with alarm status, T>, T>>), ampacity and ampacity prediction, and weather data.

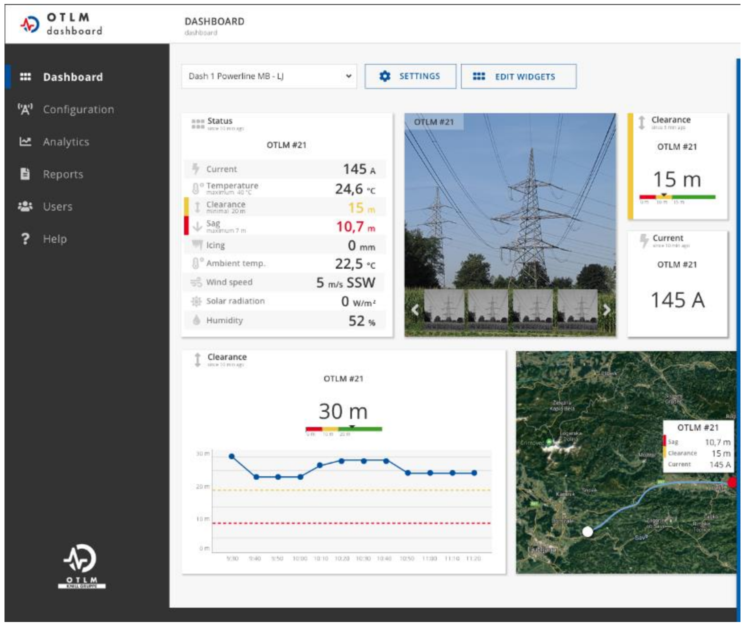


Figure 28: Gridpulse Dashboard

3.1.7 System modeling

System modeling is a crucial approach for building a system when developing complex systems. Having a schematic for simulation can help us gauge the feasibility of the design and can identify areas that require improvement as our project progresses. We also rely on the system model to analyze and optimize the behavior of our design under different conditions. For a project involving a power system, power flow analysis is also needed for a better understanding of the network. In our case, where are trying to solve the uncertainty of fault location, extensive system testing becomes indispensable to enhance the accuracy of our mathematical model. Performing repetitive tests on the physical circuit can be time-consuming and risky. Having a simulation for testing before we test on the physical circuit is a much safer and faster option. Power flow analysis is a crucial aspect of power system modeling, providing a comprehensive understanding of how power propagates within the network. This analysis enables us to determine voltages, currents, and power flows, giving us critical insights into the system's performance and stability. Although we are only working with one feeder simple diagram in this project, in the future we are hoping to be able to utilize this algorithm on a utility-scale, the simulations from this project can be used as a starting point in their testing. Some of the programs below only model the main testing circuit for power analysis, and some include the multi-controller loop as well. This project is made up of a lot of moving parts, we might choose to use more than one program because they are suitable for specific purposes. We also surveyed serval graduate students and professors within the power and renewable engineering track to help determine which modeling applications would be most applicable to us. The choices we made are largely based on the feedback and advice we received from them.

Table 5: System modeling comparison

Modeling application	Application in this project	Advantages	Disadvantages
LTspice	Simulation of whole entire project including Multicontroller, injection control, and one-line grid representation	allows users to design and test various types of analog and digital circuits, including customized parts and input signals. Free	No major disadvantages
Multisim and Multisim live	Simulation of the one-line grid representation with one phase or three phase	Drag-and-drop user friendly interface. We had used this application in our previous lab, but there was no learning curve. Web-based version makes collaboration easy	The premium version costs money Longer simulation time for larger design Too simple sometimes
Powerworld	Simulation of the one-line grid representation with one phase or three phase	One of the most powerful power system modeling and analysis tool Professional level software that's very user-friendly. Combining graphical power system case editor and power system analysis Data can be extracted from CIM and PSS/E Free	No major disadvantages
MATLAB Simulink	Simulation of whole entire project including	Block based diagram works best	It is not free.

	Multicontroller, injection control, and one-line grid representation Control system and fault detection Useful for algorithm for the whole system	for dynamic systems. It can be used for system simulation as well as coding, especially for machine learning.	It has a steep learning curve
Open Distribution System Simulator (OpenDSS)	Simulation of the one-line grid representation with one phase or three phase	Recommended for 3-phase unbalanced load modeling. Would be great for scalability down the road	Steeper learning curve

3.1.7.1 LTSpice

LTSpice® is a powerful, fast, and free SPICE simulator software equipped with an array of features such as schematic capture, and waveform viewer to enhance the simulation of analog circuits. Instead of a traditional oscilloscope, its graphical schematic capture interface allows you to probe schematics and produce simulation results, which can be explored further through the built-in waveform viewer. The simulation allows you to generate simulation results like current or voltage at any given point with a probe. It also allows algebraic expression to calculate impedance and power, which can be extremely helpful in the context of this project because we rely on impedance value for our fault detection algorithm. Although we plan on only using a few impedance values for our hardware implementations, we would like to gather as many details as possible in the simulation stage. By leveraging this feature, we can refine our fault location techniques and ensure the reliability and accuracy of the entire system. The simulation can be in AC sweep, transient responses, DC sweep, and Monte Carlo simulations.

One advantage of LTspice is it allows users to design and test various types of analog and digital circuits, including amplifiers, filters, oscillators, power supplies, etc. This versatility is further enhanced by the software's ability to accommodate custom components. To add specific parts made simple by allowing the user to download datasheet online and add to the library. This works for different types of components including passive active, and switching component, for example, we were able to add the TL084 op-amp and multiplexer model this way. This concept also applies to input signals making them very customizable in terms of amplitude, phase, and waveform. Because we will be working with programable AC sources, we will require additional accommodation which LTspice can supply.

3.1.7.2 Multisim and Multisim live

Multisim and Multisim Live are the two versions of a system modeling and simulation developed by National Instruments. It offers comprehensive electronic circuit design and analysis capabilities. They are commonly used among undergraduate students in the UCF ECE program due to their simple interface and accessibility through Multisim Live's web-based nature makes it easily accessible. Multisim Live is the free version, and provides essential functionality, while Multisim Premium includes features such as unlimited components, additional simulation types, and advanced manufacturer components.

One of the main advantages is its very intuitive design process, it is easier to build and connect nodes on Multisim when it comes to circuitry, and it uses a basic drag-and-drop logic. Couple with its library of electronic components that accommodate most undergraduate level projects, and broad choices of simulation analyses including transient, AC, DC sweep, and parameter sweeps, making it an easy candidate for kick-starting this project. The interactive and real-time responses let us observe active components in both transient and steady states. Unlike the other platform, web-based Multisim makes sharing work and collaboration easy, it allows viewing and editing by other users, this is very useful because our team often works remotely.

One of the disadvantages is the price of the Multisim Premium, while other modeling platforms offer customizable components for free, the advanced features of Multisim Premium come at a price. Although simplicity makes it easy to run circuit analysis it is difficult to simulate the whole project all at once due to its lack of power system simulation. It is also a little too simple for the later stage of the project where we focus on finer details and fine tune the PCB. Handling large-scale simulations may require additional computational power and could potentially lead to longer simulation times.

3.1.7.3 PowerWorld

Different then the modeling platform discussed above, PowerWorld is robust modeling software for power systems. It is mostly used by researchers and power engineers in the industry because of its capability of graphical power system case editor and power system analysis. Commonly known as the “edit mode” and “run mode”. Although this is professional software, it offers an intuitive experience due to its user-friendly interface due to its top Robbin design, like Microsoft Office 365.

In this project, it will be utilized in terms of detailed topology modeling for the seek to visualize the change of impedance, power, and direction of the flow. The multi-controller and pulses switching multiplexer hardware components will not be modeled on this platform. Because PowerWorld's strength lies in its network and power analysis, which closely mirrors the structure found in utility systems. The reason why we rely on this software for the power system simulation is because it is a network and bus-based approach, this is very common in the utility. And for this project to be scalable in the future we need to approach the system from bus topology instead of regular circuit analysis.

PowerWorld also supports data extraction from CIM (Common Information Model), which is the backbone of the EMS model for most utility. By using this platform, we can replicate the system in a utility for grid control purposes the best. Data from PSS/E can also be imported on PowerWorld.

The edit mode allows the users to build the model with the built-in library for power system components, including transmission lines, generators, transformers, loads, etc. It also conveniently has different color options for different voltage levels so the users can change quickly.

The run mode lets the user run different kinds of analysis to calculate the steady-state voltages, currents, and power flows in a power system. The most useful ones are contingency Analysis, Time-Step Simulation, Sensitivity Analysis, Loss Analysis, Fault Analysis, Optimal Power Flow (OPF), PVQV (Photovoltaic-Quadrature Voltage), ATC (Available Transfer Capability), and SCOPF (Security Constrained OPF). Additionally, Transient Stability and Distributed Computing have recently become available. It is capable of efficiently solving systems of up to 100,000 buses. Although we don't plan to model a large system, having the capability will be useful down the line in terms of testing and simulating the algorithm on a larger scale.

The biggest advantage of this power flow analysis is the interactivity. This platform allows us to build and test repeatedly seamlessly in research of the optimized test case of this algorithm, making the testing process quicker.

3.1.7.4 MATLAB Simulink

Simulink is a block diagram environment used to design systems with multidomain models, enabling simulations before hardware implementation, and deploying solutions without the need for writing extensive code. It is very commonly used in dynamic system analysis because it has an intuitive graphical block diagram style interface, making it user-friendly and accessible to both novice and experienced users. However, it does have a bigger learning curve based on my survey among graduate students. Each component is represented by a specific block, and the representation of commonly used components such as generators, transformers, loads, and conductor configurations are pre-built can be found easily. Because everything is configurated so it is also highly customizable, although this project may not necessitate designing components from scratch, it is still a valuable functionality to have, ensuring adaptability to various scenarios. The building block style also allows us to seamlessly implement our algorithms within the platform itself, eliminating the need for external conversion platforms. This streamlines the process and enhances efficiency during the algorithm development phase.

Simulation/analysis can be highly customized on Simulink because it allows a wild range of analyses from load flow analysis to calculating the steady-state voltages, currents, and power flows throughout the network. This application is also ideal for control system engineering, effectively handling dynamic systems and showcasing transient responses to various events. It does a good job of showing the transient responses of various events, such as a single-phase ground fault. In terms of data visualization, it does as well as the rest of the modeling platforms we are comparing it to. It gives us insightful representations of simulation results. One of its disadvantages is Simulink is not free, which could potentially post a constraint on our budget for the project. We had been exploring our options with the university provided version.

In addition to Simulink, MATLAB, a powerful programming language, can complement our understanding of the algorithm and the entire system. MATLAB's machine learning capabilities offer potential for pattern recognition, which can be advantageous in refining fault detection algorithms and further enhancing the accuracy of our fault location approach.

3.1.7.5 Open Distribution System Simulator (OpenDSS) and Multi-Agent OpenDSS

OpenDSS is an electric power distribution system simulator (DSS) designed to support distributed energy resource (DER) grid integration and grid modernization. This is a distribution-specific platform that could be very useful in this project since we are focusing on faults in the distribution system.

OpenDSS does modeling based on nodes and load, and it can handle traditional and advanced distribution assets and their control. Its load allocation features can be extremely useful in this project because we rely on the impedance from the injection pulse for our fault location detection algorithm. We can assign different loads to observe the change in impedances as the fault becomes more apparent. We were planning to do this for hardware testing to ensure our safety. This also represents a transient response ranging from before and after the fault. OpenDSS allows users to assign loads to specific buses based on historical data or load profiles. This feature is essential for a realistic representation of load characteristics in the distribution system. AMI is a newer smart technology, in the scope of this project we will be using manufactured pre-process data from AMI, but in a utility setting the model will have to handle data directly. OpenDSS is well-loved by the utilities because of its ability to model a large three-phase system with an unbalanced load and decentralized energy generation and storage unit. As the popularity of the at-home solar system rises, the load and inertia of the grid change drastically. OpenDSS allows users to model unbalanced loads in detail. It can simulate different types of unbalanced loads, such as single-phase loads, phase imbalances, and variations in load distribution among the phases. This capability is crucial for studying the effects of unbalanced loads on system voltages, currents, and power flows. It can accurately model the three-phase distribution system's network characteristics way better than the other applications on this list. OpenDSS can model an advanced inverter for the development of a new distribution system.

The different types of fault solutions it provides are Unbalanced, multi-phase power flow, Quasi-static time-series (QSTS), Fault analysis, Harmonic analysis, Flicker analysis, Dynamic (electro-mechanical) analysis, Linear and non-linear analysis, Stray voltage/current analysis.

OpenDSS was originally designed for QSTS analysis, it is a power system analysis to study steady-state and time-domine simulation with varying load profiles. This could potentially be helpful for our project if we choose to go with the pattern recognition route, we will need data from different timeframes during and before the abnormality to train the module.

Although undergraduate students are taught how to use OpenDSS in power system related courses, I found this application can be hard to use due to its consolidated representation instead of the full-topology model. It can run a big system, but it does get very culture visually. The different types of fault solutions it provides are Unbalanced, multi-phase power flow, Quasi-static time-series (QSTS), Fault analysis, Harmonic analysis, Flicker analysis, Dynamic (electro-mechanical) analysis, Linear and non-linear analysis, Stray voltage/current analysis.

Multi-Agent OpenDSS, MA-OpenDSS, is built upon OpenDSS. The MA-OpenDSS incorporates an asynchronous, local, possibly varying communication architecture, enables a set of virtual nodes (called virtual leaders) to coordinate local data sharing and actions, includes programmable dynamic blocks for behaviors of distributed energy resources, and implements a host of distributed and cooperative algorithms of estimation, optimization, and control. This is very applicable in the distributed energy sources area, which means it is outside the scope of the project. However, if we choose to use an inverter in this project, this framework can help with dispatching injection signals using investors already in the grid.

3.2 Part Selection

There are various parts that we have considered for our project, in our previous section we discussed each of our selections in a more technical sense, in this section, we will be explaining why we chose a certain part and how it will tie in with the system. Of important note on our methodology for choosing a part, we are choosing on a variety of factors, not just strictly on what is the best. We weigh affordability and ease of use heavily, and we will choose a part that satisfies our requirements and no more than that.

We were able to use some of the more expensive equipment for free in this project because we work closely with the UCF microgrid and protection and control research group. Although affordability is one of our most pressing concerns, we were able to use industry and research grade equipment closely align with the project specification.

3.2.1 Power Supplies

There are two main power supply systems present in our project. One is the AC-DC converter that will power the PCB and all associated components. This consists of a simple AC-DC converter wall wart. Our secondary power supply consists of a function generator that will provide us with 12VAC/60Hz single-phase AC power.

3.2.1.1 AC-DC Converter

For our AC-DC Converter, we will be using a typical wall wart connector. Our wall wart will be converting the 120VAC present in U.S outlets into a desirable 9V/500mA unregulated DC output. Of important note is that it is unregulated output, therefore we will have to include a voltage regular as well. We chose to go with a simple wall wart as our converter as trying to design our own converter is dangerous, mainly because we will have to be handling a hot outlet at an uncomfortable voltage. Also, we believe it is not worth reinventing the wheel, wall warts are a standard choice for projects like these as major companies like Arduino and raspberry pi also use wall warts to provide unregulated power to their boards. More than that, basically every company who makes consumer PCBs uses wall warts to convert outlet AC voltages to usable DC voltages. Below is an example of an AC-DC converter wall wart.

Figure 29: AC-DC converter wall wart

3.2.1.2 12VAC single-phase generation

To generate our 12VAC single-phase for use with our injection points, we will be using a Tektronix AFG3022C Arbitrary/Function Generator to generate clean 12VAC single-phase waveforms at 60Hz. We decided to go with the Tektronix AFG3022C Arbitrary/Function Generator because it is already available for use through our university. Furthermore, all team members have experience with this exact function generator which allows us to not burden ourselves with learning the operation of a new function generator. The Tektronix AFG3022C Arbitrary/Function Generator can create sine waves well within our operating frequency of 60Hz. Of usefulness as well is its ability to allow us to test our designed circuits without having to own another piece of equipment.

Figure 30: Tektronix AFG3022C Arbitrary/Function Generator

Our original plan was to develop our own AC-AC transformer to transform our wall socket voltage (120VAC) into our desired 12VAC single-phase. This is not a complex task, as standard U.S household wall outlets deliver the 120VAC single-phase. However, there would be a lot of complications with developing our own transformer system. Transformers are bulky, and without proper cooling can become very hot, which poses a safety hazard to the team. We also believe that developing our own AC transformer system is out of scope for the purposes of our project, and therefore using an already established technology that can produce highly accurate waveforms like the Tektronix AFG3022C is in our opinion the better path forward.

3.2.2 Passive Component and Switches

This section covers all the components we will be using to control our digital and analog signals being generated. Passive components do not create any energy, and we have covered our power supply generation in the section before this one. There are various requirements from passive components and switches that we will be needing, specifically, we will need to be able to control when our voltage pulses are sent into the system, so we will be needing a switch that can be digitally controlled by our microcontroller. We will also be needing current and voltage meters to be able to calculate the impedance of the circuit and therefore decide on whether there is a fault or not.

3.2.2.1 Multiplexer Switch

Part of what is required from our project is the ability to inject multiple zones in the circuit to measure its impedance and decide if there is a fault at the measured location. However, a user operated switch is not viable as part of our requirements is that the microcontroller should be able to find said fault on its own. Therefore, we require a switch that can control analog signals using digital signals. Thankfully, a multiplexer is exactly what we need for this project. And the Texas Instruments CD4051B CMOS Single 8-Channel Analog Multiplexer or Demultiplexer with Logic-Level Conversion is the perfect component for us to achieve this requirement. It is a 1:8 multiplexer with the ability to be daisy chained should we require more than 8 injection sites; we do not foresee us using more than 8 injection sites, however. Below is a functional diagram of the CD4051B.

Table 6: Multiplexer switches

Component	Analog Signal Levels	1:X Multiplexing
CD4051B	• Less than 20VPP	• 1:8
CD74HC4051-EP	• Less than 10VPP	• 1:8

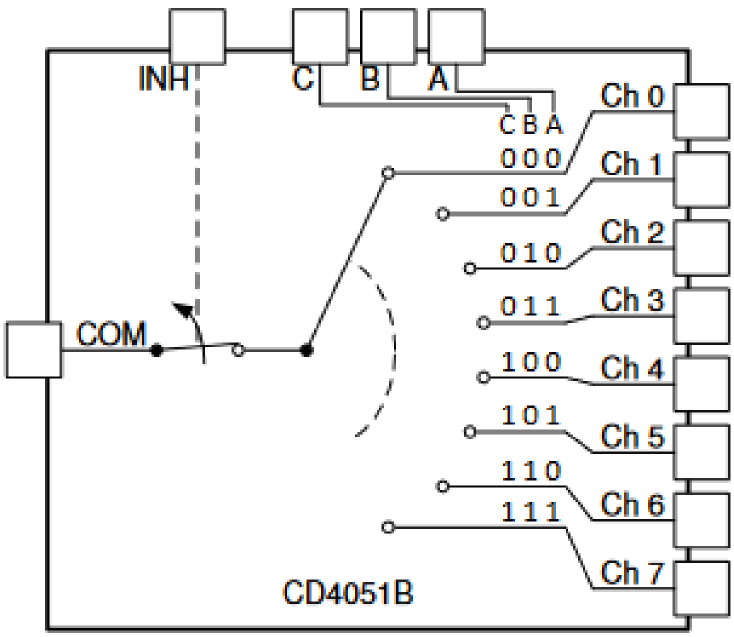


Figure 31: Functional diagram of CD4051B

This component is perfect for our purposes as it can be controlled using a digital signal from between 3V to 20V, and the ATMega328p outputs roughly 4.2V at 85C, this means we are capable of properly using the CD4051B. Furthermore, we spent a good amount of time researching something that can handle the 12VPP output that our function generator will be creating, and this is where the CD74HC4051-EP fell short, we were mainly attracted to it first due to it being cheaper, but upon researching more into it we discovered that its analog signal level limit is less than 10VPP, this of course doesn't satisfy our requirement since our testing signals will be running at 12VPP.

3.2.2.2 Potentiometer

Table 7: Potentiometer choices

Product	**Resistance Range**	**Rated Power**	**Tolerance**	**Price**
PTD901	1k - 1MΩ	50mW	10%	$2.91
P160KN2-0EC15B1MEG	500 - 1MΩ	200mW	20%	$1.33

For this project, we will be using a potentiometer as a representation of a fault on our grid. For this, we will need our steady-state impedance to be multiple factors higher than our simulated fault impedance such that our meters can pick up the difference and therefore trigger an event to our microcontroller. There are several factors to take into consideration however when picking a potentiometer. Our 12VPP function generator has an internal fuse of 0.125A, therefore from Ohm's law, our fault impedance must still be greater than ~100Ω so that the internal fuse of the function generator is not blown. Because of this requirement, we must also have a potentiometer with a tolerance percentage whose bounds do not fall lower than 100Ω. Therefore, we must have a lower bound in the potentiometer that does not fall under ~100Ω, but it also must be as low as possible so that a fault event is very noticeable to our metering components. For this reason, we settled on the TT Electronics P160KN2-0EC15B1MEG. It features an impedance range from 500Ω ±20% to 1MΩ ±20%, this is perfect for us as we will be able to establish a relatively accurate steady-state impedance, and for fault simulation we will be able to drop the impedance to noticeable lower levels all without risking the function generators internal fuse. While the P160KN2 has a noticeable worse tolerance, it is still within limits of our requirements, and its 200mW rated power gives us more headspace when it comes to our 12VPP function generator injection source. The PTD901 with its 50mWof rated power was too close to our margin of safety and we therefore felt it would be inadequate, with a chance of failing us. The P160KN2 is also less than half the price, and since we are buying close to a dozen, it is overall noticeably cheaper. Below is a picture of our chosen potentiometer along with its taper curve. The taper of our chosen potentiometer is the 'B' taper.

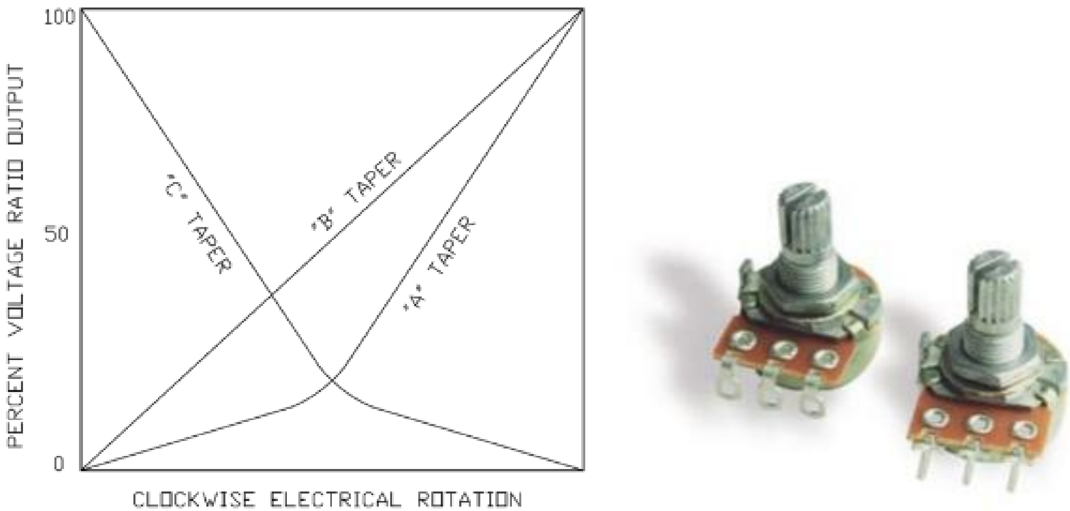


Figure 32 Impedance taper graph and photograph

3.2.2.3 Current meter selection

To find the fault location, we need to calculate the impedance. This will be done by calculating voltage and current at a specific node. It is necessary to find a current meter that can satisfy our needs and that we can incorporate into our design.

Table 8: Current meter comparison

Product	Price	Type	Manufacturer
YHDC non-invasive current sensor	$6.20	Through-hole	YHDC
ACS37002	$8.08	IC	Allegro Microsystems
PZCT - 02	$8.79	Through hole	YueHuam
CS60-010	$4.81	Through hole	Coilcraft

Figure 33: AC current sensor by YHDC

3.2.2.3.1 Non-invasive AC current sensor by YHDC

It is based on current transformers. You can clamp it around the wire, and it will measure the magnetic field produced. This is a great option because you can just put it around the wire and you don't need to put it in series with the wire which is non-ideal because we are trying to simulate that our device is working in power lines, and in real life we wouldn't place part of our device in series with the power line. The hole from where current passes is 13mm x 13 mm which can work on our demonstration. It sends the reading through an auxiliary cable, but we can cut the auxiliary and use the wires alone.

3.2.2.3.2 ACS37002 by Allegro Microsystems

The ACS37002 is an IC current sensor that can measure up to 100 A. It uses 5 V as input and it'll measure the amount of current that passes through it. The advantage is that it is small and can hold a lot of current. But it would be very inefficient in our project because it must be connected in series with the wire who's current, we are measuring. We are trying to create a device that can be added to power lines rather than built within the lines.

3.2.2.3.3 PZCT – 02

This current sensor also works using current transformers which will sense how much current is passing through them and it will send a voltage output according to the current magnitude. It works with up to 100 A, the diameter of the inner hole is 16 mm, and current must have a frequency between 50 to 60 Hz. It also includes a clip on it so it will be easy to use since we can just put it around the wire that we want to measure. The downside is its size, it is 1.97 L x 1.57 W x 1.18 D inches, which could affect our demonstration depending on how much space we have around our wires.

3.2.2.3.4 CS60-010

This current sensor doesn't have a clip, so we would have to install it before connecting the wires so they can pass through it. It can only measure up to 10 A, but it works with frequencies from 50 to 60 Hz. It is very small with a width of only 2 cm, and it could be a great option if our current is not so high.

3.2.2.4 Voltage meter selection

Measuring the voltage of specific nodes will be crucial for our project. We must be able to read the voltage from multiple places to get the impedance. For this, it is considered one fundamental element of our project. To implement it in our design, we are going to be using one of the analog inputs of our ATMega328P. From the datasheet, we were able to get the maximum voltage allowed by these pins, which we can calculate by V supply + 0.5V. In this case, we will be supplying our microprocessor with 5.1V, therefore our analog inputs can work with up to 5.6V. Another factor to consider when using our voltage meter is that our controller can't handle negative voltage.

Since we can't directly connect our PCB to the grid model (due to the high voltage of the grid and the negative values), we can do two things. We could buy an external component that reduces the voltage and eliminates the negative voltage, or we can build our own rectifier with a voltage divider. Next, we will go over both methods.

3.2.2.4.1 zmpt101b voltage sensor

The zmpt101b voltage sensor is an external module that can read up to 250VAC and work with frequencies of 50-60Hz. It uses a transformer with a 1:1 ratio and a couple of operational amplifiers to rectify the signal for our Arduino. They cost around $7.00 per module and are very simple to use. All we must do is power it up with our microprocessor, and then it will send us the data through our analog pin. The negative aspect of it is that we will need to use multiple sensors, and the price will start increasing quickly.

3.2.2.4.2 Self-made voltage meter

Designing our voltage meter will be the better option. It will give us more flexibility to use it since we will only be dealing with wires and not external modules. A self-made voltage meter will be cheaper to implement for multiple voltage readings, and it is not difficult to do. The main idea would be to eliminate all negative voltages by using a diode, and then using a voltage divider to step down the voltage to something less than 5.6V.

3.2.2.5 Power resistor and inductor

We will be using a resistor and inductor to simulate load on the line in our hardware demonstration model. Given we will work with possible medium to high power in our hardware model, we would like to use a power resistor instead of the regular resistor in the senior design lab. Power rating is a characteristic of a resistor, it describes the amount of power flows a resistor can handle to safely dispatch the heat, this rating is given in the datasheet of the resistor in wattage. In high power operation, high heat is expected so in general they have large surface areas or heat sinks to facilitate better heat dissipation. Most common resistors have a power rating between 1/8 watt (0.125W) and 1 watt. Given that we will be using a potential meter with the lowest resistance of 5k ohms, we aim to have resistor values of 50k to 100k ohms. The normal 100k Ohm Carbon Film Resistor has a power rating of 0.25W. We would need something with a higher power rating for the impedance close to the power supply.

The 50k ohms wire wound power resistor from TE Connectivity provides a power rating up to 50W.

Figure 34 wire wound power resistor from TE Connectivity

The Uxcell 3W Power Rating 100K Ohm High Voltage Glass Glaze Film Resistor might also be a good option. From our observation, power resistors tend to have low resistance because they are designed for having high current travel through. This 100K ohm fits all the requirements of this project by having a high resistance to create a contrast to the potential meter and being able to safely handle higher power.

Figure 35 Uxcell 3W Power Rating 100K Ohm High Voltage Glass Glaze Film Resistor

3.2.3 PCB Selection

Table 9 PCB selection

Software	Features	Downsides
Single-Sided PCB	• Easy to design. • Cheap	• Low density circuitry
Multilayer PCB	• Compact • Flexible design	• Expensive • Complex
Double-Sided PCB	• Reduced size • Lower cost than multilayer	• Medium density circuitry

Choosing the right PCB is a very important design choice, as having an inadequate design can leave us with a very poor design or an outright impossible design. We took into heavy consideration the three PCB design choices in the table above. We also took inspiration from existing solutions like the Arduino Uno that implement the ATMega328p.

A single-sided PCB proved inadequate due to the complexity of the parts we will be installing on our PCB. Single sided PCB is suitable for low-density circuits (i.e., LED circuits) with conducting paths that do not intersect. This design requirement proves to be impossible due to the number of pins on the ATMega328p itself, not including the other passive components we will be needing. Electromagnetic interference also proves to be a problem with this PCB design.

A double-sided PCB was heavily considered and may prove suitable for our purposes. But due to our level of expertise, we felt tying ourselves to such a strict requirement of only two conducting layers was an unnecessary and potentially dangerous decision for our project. Double-sided PCBs allow for a more compact circuit, while allowing a higher density due to its two conductive layers. Electromagnetic interference is still a present problem; however, it is much easier to work around.

For our PCB, we intend to use a multi-layer PCB with vias since we will be including components such as a microcontroller, power supplies, transformers, as well as passive components. This design will allow us to minimize electromagnetic interface between our various components, as well as allow for a more compact board. We have taken inspiration from popular boards like the MSP430 and Arduino boards, both of which feature multi-layer PCB designs. Another inspiration from popular boards that we have taken is pouring a ground layer on both the top and bottom layer to provide better protection against interference. This of course is standard industry practice, and is recommended by ATMega, who are responsible for developing the microcontroller we will be using, the ATMega328p. This makes our final PCB more expensive, however there are several advantages going with a multi-layer design. Mainly, it affords us design flexibility using vias. Below is an example image of a multi-layer PCB.

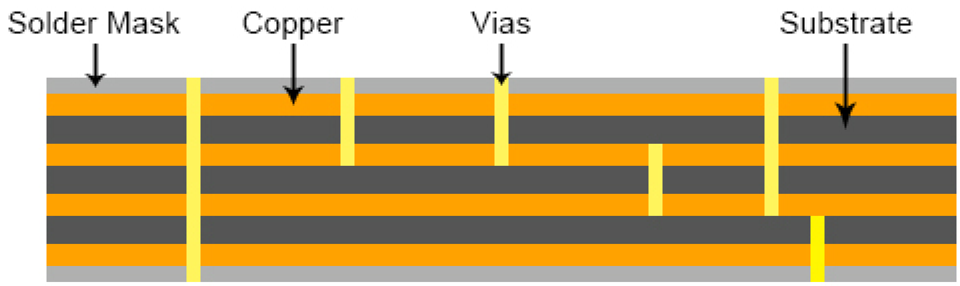


Figure 36: Sideview of a multi-layer PCB with vias

3.2.4 Software Components

This Project comprises a few software components, they are the overall algorithm multi-controller programming, machine learning for pattern recognition, and AC signal generator programming. They are the backbone of the project because we relied on them to simulate and test our fault algorithm. Each component contributes to the accuracy and efficiency of our research, allowing us to develop a robust fault detection system for distribution grids.

3.2.4.1 Machine Learning Model

The machine learning model we plan on using is one called DynoNet, it is a neural network architecture for dynamic systems. Given that our power grid dataset is a dynamic system, we believe this model architecture is well suited for our uses. Also worth mentioning, we have consulted an expert who has experience with machine learning models for power grid systems, and they have explained that DynoNet is exactly what we are looking for, they were also willing to consult us on learning how to use it. For us as students, this was a very important point to consider when making our final choice to use this architecture. This model, while being a neural network, employs concepts found in that of linear regression models such as linear operators. An image of that concept is shown below. As stated in the GitHub repository, “In the DynoNet architecture, linear dynamical operators are combined with static (i.e., memoryless) non-linearities which can be either elementary activation functions applied channel-wise; fully connected feed-forward neural networks; or other differentiable operators.” [2]

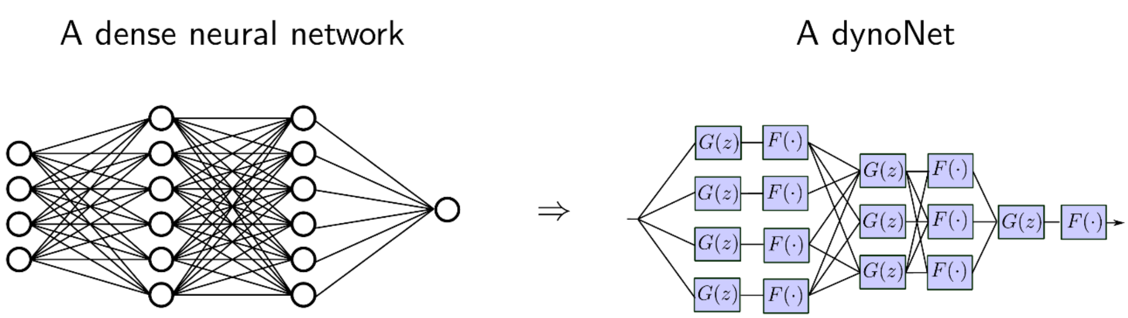


Figure 37: Depiction of DynoNet model

The DynoNet architecture is built on Python and features various examples ranging from circuits as well as other dynamic systems to experiment and learn from. We believe that it being written in Python is another positive point, as Python has libraries available that allow you to flash into the ATMega328p directly thus allowing us to flash our model into our controller.

DynoNet is a mix between traditional recurrent neural network and 1-D Convolutions neural network, it utilizes linear dynamic operators parametrized as rational transfer function as building blocks. It extends the 1D convolution NNs to infinite with impulse response dynamic. Back propagation also makes this architecture more efficient. Recurrent NNs is based on feed forward state space model, this model is very versatile because it can represent most systems. Because this is a feed forward system so there are a lot of limitations such as not being able to run multi things in parallel, and numerical issue in terms of vanishing guidance. With 1-D Convolutions NNs it can be parallelizable and neural training is more well behaved. However, it does have a lower capacity.

In back-propagation-based training, the user defines a computational graph by interconnecting blocks that individually support back-propagation. This graph will produce a loss and the objective is to minimize the loss L. There are two phases, the forward and the backward pass. In the forward pass, the L is computed. Each block receives an input and transforms into an output by the transfer function, this will help filtering operation. The backward pass is for computing the derivative of L with respect to the training variables are computed. Each operator must be able to "push back" derivatives from its output to its input.

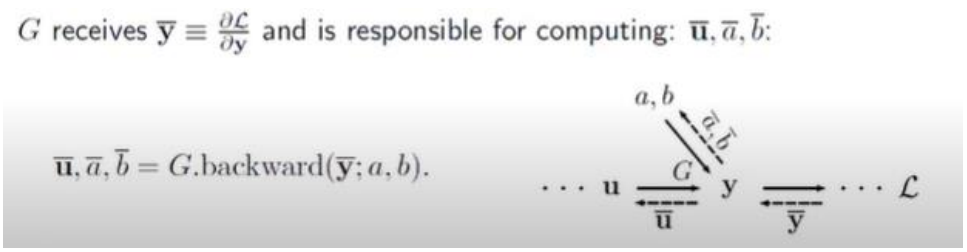


Figure 38: Backward pass of G with respect to the transfer function utilizing chain rule

3.2.4.2 Software and Programming

For this project, we are required to program the ATMega328p as we are required to carry out some logic events that are driven by software interrupts. The software aspect of this project can be classified as relating to embedded systems since we will be programming spatially efficient code onto a microcontroller. Certain precautions must be taken with this in mind, code must be efficiently written and must integrate as little libraries as possible to reduce the spatial cost of it. It must also run close to real time, as we will be doing some analog measurements of aspects like voltage and current.

As a result, the programming language we will be using for this project will be C++ using the Arduino IDE. We chose C++ because it is the language the Arduino IDE is based off; the team also has experience coding in C++ and therefore there is no need to learn another language. However, we may need to interface with Python for the machine learning aspect of our program. DynoNet, the model we will be using, is written in Python. We will therefore have to figure out how to interface with it, or port the code over to C. Furthermore, since we will be using the Arduino IDE, it will make interfacing with the ATMega328p vastly easier. To elaborate, we chose the Arduino IDE as it is specially designed to be used to program Arduino boards, which use the ATMega328p as its microcontroller. The Arduino IDE gives us powerful tools to track memory usage, as well as providing an optimized version of C++ specifically meant for embedded systems.

We will be using a 'raw' ATMega328p, which for clarification is the ATMega328p microcontroller itself without the Arduino board. Since we will be using this approach, we will not have the Arduino bootloader available on our PCB to flash our code onto the microcontroller. Therefore, we will need a way to be able to flash our microcontroller. As a result, we will be using a spare Arduino Uno board as our programmer. To do this, we will have to connect our ATMega328p to our programmer Arduino board using a breadboard. We will also need to take into consideration that our raw ATMega328p will not be running at the same frequency as the clock in the Arduino, we will therefore have to specify the clock rate of our raw ATMega328p, which is 1MHz compared to the Arduino's 8MHz. Thankfully, the ATMega328p includes an internal clock, which means we will not be needing to attach a clock outside on the breadboard.

3.2.4.3 AC source Programming

The AC signal generator is one of the most crucial parts of our hardware demonstration. Depending on which AC power source we end up choosing for this project, we might need to program the input signal to the system. On top of the built-in features with normal AC signals, we might need to design custom waveforms, set voltage levels, and control frequency and phase shifts to simulate real-world grid conditions. Allowing us to test our algorithm under different conditions.

Some signal generators allow users to impose a current limit, which can ensure our safety and the requirements. To do so, we will need to program the power source with the SCPI command via the GPIB interface. SCPI is a commend/script language used for programmable instruments. It is used for communication and control with test and measurement instruments, such as oscilloscopes, power supplies, signal generators, and multimeters. It is possible to set up the AC source with the built-in interface, however, SCPI usually allows more efficient and precise control.

3.2.5 Microcontroller Selection

A fundamental part of our project will be the microcontroller since all the measurements are going to be read by it, and then it will use this data to get information about the fault location. Following we will go over different MCU.

Table 10: Microcontroller selection

Microcontroller	Onboard Memory	ADC precision	Timers	Digital I/O Pins	Cost
ATMega328p	32Kb	10-bit	Two 8-bit and One 16-bit timers	23	$3.03
MSP430FR6989IPZ	128Kb	12-bit	Five 16-bit timers	11	$9.34
MSP430G2553IN20	16Kb	10-bit	Two 16-bit timers	24	$3.02

For our project, we have decided to go with the ATMega328p as the microcontroller of choice to control all logic events. We believe in terms of memory, 32kB is enough space for our needs. For example, the classic introductory task of making a flashing LED with a button, which almost anyone who has interest in embedded systems has tried making, roughly takes up 1kB of memory. That is a fair amount of code in such a small footprint. Furthermore, one of the colleagues that we are consulting on this project has experience successfully creating a machine learning model on an Arduino, which for clarification has an ATMega328p as its microcontroller. Another positive point is that there is heavy documentation regarding creating machine learning models on an Arduino Uno, which also contains an ATMega328p as its microcontroller. We therefore believe that all this available documentation can be used for our own purposes, as the only critical piece that it depends on is the microcontroller being the same.

Of course, while 32kB is enough space for our purposes, it is not comfortably high enough to employ lax coding standards. We will have to be efficient when writing code, this will come in the form of using unsigned integers whenever possible, and explicitly defining them with an 8-bit size (i.e., uint8_t). Of course, this limits our usable integer range, but we don't believe it to be a problem as we also have the option of using the traditional 16-bit integer when needed. Another important thing to keep our wasted memory as low as possible is to avoid costly constants like strings and structs. We don't foresee heavy use of these data structures, but it is well to keep in mind.

One area of concern is the low amount of SRAM available on the ATMega328p with a size of 2kB. Thankfully, for any application that we foresee using, we have enough space. For example, ideally, we'd like to have our impedance calculation variables off-loaded to SRAM as that area of our circuit behaves much like a real-time circuit where instantaneous readings are desired as the state of the circuit has a chance of changing extremely quickly.

Finally, to keep our resource usage low, we will be consulting the AVR assembly manual, this is the documentation and manual for the development environment we will be using. The resulting assembly code is built in what is called the AVR-GCC compiler, which translates the written C code into AVR assembly. We will be designing our C code to map as efficiently as possible to these instructions to keep within our memory budget.

The ATMega328p also features an analog to digital converter with 10-bit accuracy. This is accurate enough for our purposes, and the performance difference between a 12-bit analog to digital converter and the ATMega328p 10-bit analog to digital converter is not great enough to justify the MSP430FR6989IPZ. For example, this 10-bit analog to digital converter has a resolution of 1,024 possible voltage steps. For our 12V AC fault detection, this gives us a resolution of 11.7mV per step. To compare, a 12-bit analog to digital converter has a resolution of 2.92mV per step, considerably lower, but this level of resolution is not required for our purposes.

The timers present on the ATMega328p are two 8-bit timers and one 16-bit timer. We do not foresee having to use more than one timer, and therefore the single available 16-bit timer satisfies our requirements. To illustrate, we expect to do a fault injection every 30 seconds and will be actively watching for the return signal for multiple seconds at a time. The ATMega328p includes a pre scaler that goes up to 1024, which gives us a max interrupt length of around 5 seconds. However, through software-based loops, we can increase this timer to an arbitrarily high amount, or more specifically, our desired 30 second timer length.

The ATMega328p also comes in with a healthy number of I/O pins at 23, this gives us a lot of flexibility with the amount of injection sites we can have, however we do not expect to reach the I/O pin limit in this project. Lastly, the ATMega328p comes in at an attractive price compared to the other two choices, we believe this microcontroller fulfills our requirements at a price suitable to the budget of our project.

4 Design Constraints and Standards

In the following chapter, we will go through some electrical and mechanical aspects that could impact the functionality of our design. Our project primarily focuses on power grids, which involve significant risks. Due to this, we will have to adjust our model to efficiently implement our design while still working with safe levels of power.

We also need to go through some industry standards that will help us follow some instructions our guidelines imposed already by the industry so all our work can be communicated and shared efficiently.

4.1 Low Voltage Single-Phase Demo

Traditionally, power is generated at large power plants and transmitted through transmission lines. These transmission lines are then connected to substations with step-down transformers to supply power to the sub-transmission lines in the distribution grid. Normal sub-transmission feeder lines typically operate at voltages ranging between 26kV to 69kV. This medium voltage range facilitates the efficient transmission of power over long distances from the substations to various areas. Transformers are used to further step down the voltage to 120/240V for residential areas and 480/11kV for commercial and industrial areas.

High-voltage electrical systems pose significant dangers, although we would like to set up the model in distribution-level voltage to match real-world scenarios, we encountered limitations as undergraduate students with limited real-world experiences. We believe some of the risks associated with high voltage systems for us to create the model using a high voltage level. Some of the risks we are concerned about are electric shock, sparks, and equipment damage. High-voltage electric shock can cause severe harm to the human body which can lead to injuries, organ damage, and even death. Improper handling and improper grounding are the most common reasons for electrical shock along with improper handling, wearing the wrong personal protective equipment (PPE) or accidental contact with energized equipment. Equipment damage in a high-voltage environment is typically more expensive to repair and takes longer to replace than equipment used in low voltage system.

One of the design constraints we faced was how to implement a low-voltage power supply for our physical model. We can choose to use a function generator or DC voltage supply to generate 12V to directly power our model and MCU. The other option is to use the 120V wall plug and implement a step-down transformer, which allows for more precise output control but might interfere with our budget. If we choose to power our model with a 120V power supply, we will need to use multiple three-phase inverters as a part of our design to simulate the injection site. These inverters are expensive and take a long time to deliver due to the chip shortage.

In reality, the entire grid is in three-phases because three-phases provides optimal efficiency and stability. Although single phase is used in certain neighborhoods lateral lines or within big commercial buildings. Three-phase systems provide a balanced load on generators, resulting in lower overall losses and increased power transfer capacity. The three-phase system is very balanced because its 120-degree angle differences ensure the power is distributed evenly across the three phases, resulting in a more balanced load. And the waveforms of the three phases are symmetrical and identical in shape, magnitude, and frequency. This symmetry ensures that the power delivered by each phase is equal and cancels out any negative sequence or zero sequence components, thereby minimizing voltage fluctuations and power quality issues.

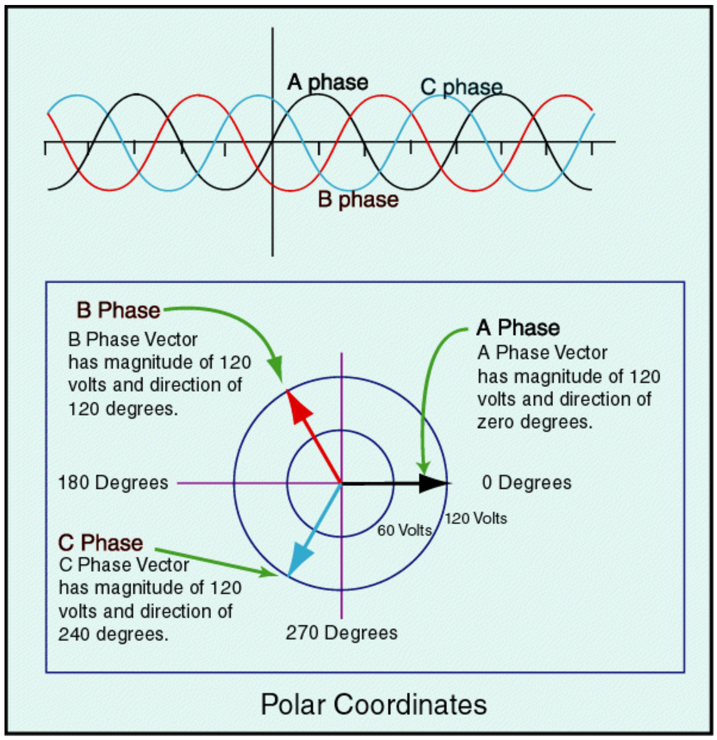


Figure 39: 3 phase power supply waveform and polar coordinates representation

In this project, we are restricted to a single-phase model because we choose to use a 12V power source instead connecting our project to the grid for safety reasons. That means we are also restricted to only using a single-phase converter.

4.2 The Effect of Non-Homogeneity of the Feeder and Lateral Systems

An important design constraint that must be discussed is the non-homogeneity between feeder and lateral systems and how that may affect our approach to design. Although these two types of distribution lines are both an important part of the distribution system, there are important differences and complications that may come up when working on a system of design that will affect both types of lines. Because there is a connected but separate entity, it is important to understand the distinctions between the two. This section will discuss in detail a description of the two types of distribution lines and how their differences may complicate the design process.

4.2.1 A Description of the distinctions between feeder and lateral lines

After exiting the substation, there are two levels of distribution, feeder lines and lateral lines. Both lines are vital to the power delivery system and play an important role in supplying electricity to customers. These two types of lines are similar in structure but do have some important distinctions that need to be discussed in order to properly understand the challenges that may come up when working with these systems.

A feeder is a circuit that comes directly from the substation. Feeder lines are a three-phase power system that can range in voltage levels beginning at 5kV and going as high as 35kV. Feeders can vary in length from a few miles long to tens of miles long. Modern feeder systems will usually have the feeder line travel along a main road and split off into laterals at the location of neighborhoods, side streets, or building developments that may be found away from the main road. Feeder lines will also have additional equipment attached to the line and poles along the path. Some examples of this equipment could be overhead transformers to supply homes or businesses with 120V/240V power, capacitor banks used for power factor correction, and reclosers or switches which will be used as protection systems in case there is a fault or short circuit along the line. Normally open reclosers can also be used to distinguish between two sections of feeder line. With the lines on either side of the recloser being connected to a different feeder.

Lateral lines are offshoots of feeder lines that will travel into neighborhoods, side streets, and other building developments in order to supply those areas with power. On a feeder pole where a lateral line is desired, a Tripsaver or fuse switch will be installed and connected to the feeder wire via jumper wire. Then primary cable will travel in a direction different than the direction of the feeder line. These lateral lines can vary from being a single-phase system, a two-phase system, or even a three-phase system. Generally, the phasing on a lateral will depend on the load of the area its supplying. This is because each fuse at the end of a lateral has a certain load limit associated with it. So lateral lines that supply lots of customers will need to have multiple phases in order to balance the load between the different phases to not overload the fuses.

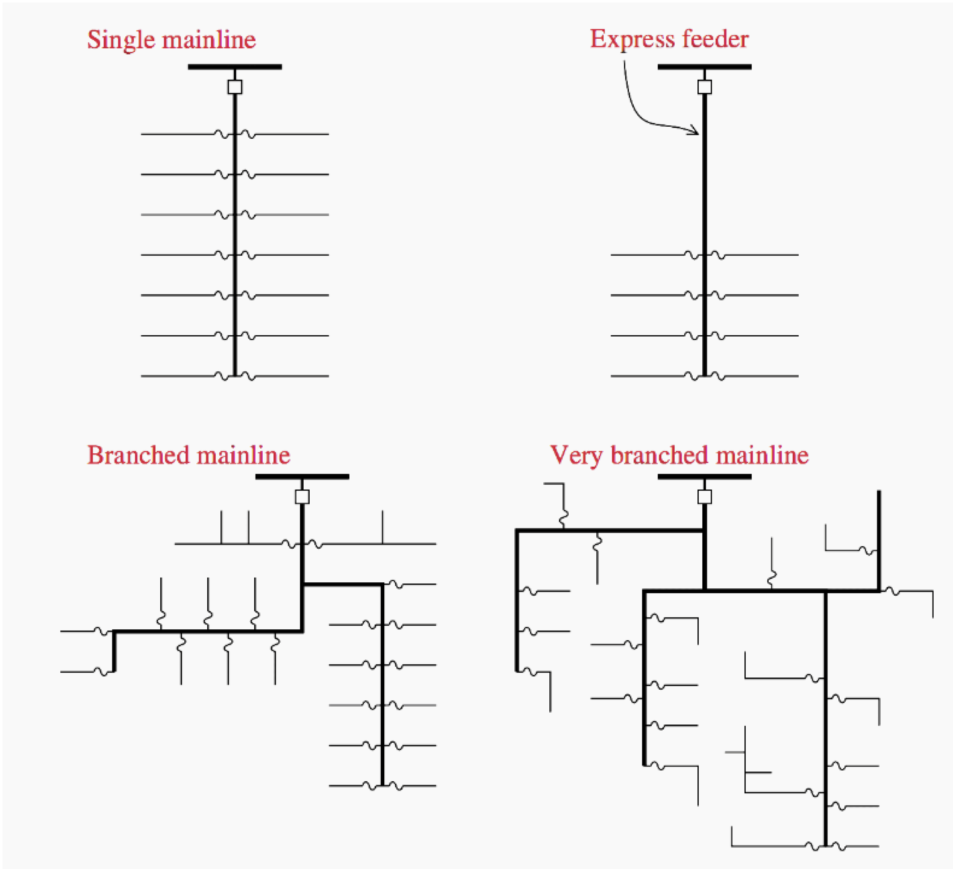


Figure 40:Examples of how Feeder and Lateral lines

4.2.2 Challenges and Design Constraints Associated with These Two Types of Lines

Now the general structure of feeder lines and lateral lines have been discussed. It is important to understand how their structure may create challenges and design constraints that must be respected in order to create a successful design. There are two possible sources for design constraints caused by the non-homogeneity of the lines and those are the fuses that separate the lines from each other as well as the potential change in the number of phases from a feeder to a lateral.

To begin with the fuse problem, every lateral is marked with a fuse attached to the feeder. This is put in place as a protection system. If there is ever a fault in the lateral, the fuse or Tripsaver can open, which protects the general feeder system from the fault. Although this is a great system which protects the feeder line in case of emergencies, it creates a headache for our design. If the fault is along a lateral, then we have to figure out a way to determine the location of the fault while working with an area that is likely completely closed off from the rest of the grid. When the fuse switch opens, power is no longer being supplied to the entire lateral system. This means that there must be an alternate method of determining the fault's location without power being sent to that region. A potential idea for how this could be done is by reading the AMI data around the time of the fault. If there is any sort of abnormality in the time leading up to the fault, it might be possible to determine the location of the fault based on the location of the meters showing abnormal readings.

Another issue associated with the non-homogeneity of the feeder and lateral line is the potential change in the number of phases as a feeder transitions into a lateral line. The design must be able to work with any number of phases seamlessly in order to be able to search both laterals and feeders for any potential faults. If the design is only able to work in a 3-phase system, then many lateral lines will not be able to be checked for the fault location. Therefore, the design must be able to work in a single-phase, two phase, and three phase system while also being able to correctly adjust and compensate for each change. Although being able to work in a three-phase system is all that would be needed for the design to work on the feeder, for lateral implementation the design will need versatility.

4.3 Equipment Along the Feeder Line

Something important to consider when working with a feeder system is the potential equipment already attached to the line. Some examples of this equipment could be transformers, reclosers, capacitor banks, fault indicators, and many more. Keeping all of these different pieces of equipment in mind is important to consider when thinking about how to implement our design along a feeder line. It is important that anything that happens in the grid system does not interfere with the existing equipment and can function properly. If our design somehow interferes with the existing equipment, it could lead to a loss of power for customers or create faults along the line. For example, if our design somehow interferes with the power supply to transformers, then any customers being fed from that transformer could lose power until the issue is resolved. When our goal is to reduce the amount of time for fault response and reduce the time that people are without power, we cannot add to the number of customers losing power. That is why it is important that our design sticks to the feeder lines, and any sort of additional current added to the lines is small enough that it does not overload any associated equipment. Another potential issue that could arise from interacting with equipment on the line is potential issues with capacitor banks. Capacitor banks are incredibly important in keeping the power factor of the line balanced, and if the design interferes with their function, it could drastically lower the power factor of the line. Capacitor banks are used as a way to counter the inductive load caused by transformers and other equipment on the line. Therefore, our design should either be able to have minimal effect on the function of a capacitor bank to the point where it does not have more inductive load to compensate for, or it creates a low enough inductive load to where the capacitor bank is able to compensate for the additional inductance without causing issues to the overall power factor of the feeder system. Ideally, we would like for our design to have as little interaction and interference with the existing equipment as possible, that way the complications and design constraints associated with the existing equipment can be minimized.

4.4 Power Considerations for the PCB

There are several key design constraints that we must adhere to regarding the operation of our PCB. Chief among them is providing our PCB with the correct voltage to ensure operation of the board. Since the ATMega328p is designed to operate between 1.8V to 5.5V, there are various requirements we will be considering that ensures the proper voltage range is being fed. Furthermore, we must also consider any constraints from other passive components such as decoupling capacitors, and other components such as the ac-dc converter.

4.4.1 System Voltage Requirements

Since we will be representing the concept of our project using low voltage, which we define as less than 12V single phase AC, human bodily harm is not a major concern. Therefore, regarding human safety, we will follow the basic concepts of applying insulating materials to any conductive surfaces, as well as separating the user from any electrically conductive components using materials such as plexiglass.

As for our system, we will be operating with low voltage DC power, which originates from a typical home wall outlet (120V AC). Therefore, our design must include an AC-DC converter capable of stepping down 120V AC into 5V DC for operation of the PCB and its included microcontroller. Thankfully, this design is well standardized as it is a common conversion for devices such as cell phones. Our converter will be using this standardized wall wart AC-DC converter, as reinventing the wheel is not a wise use of our time. Of course, these wall warts provide an unregulated power supply, so therefore we will need to include a voltage regulator system in our PCB to provide a regulated voltage. However, including the voltage regulator satisfies our design requirement of a steady 5VDC power source for the operation of the microcontroller and all other components on our PCB. In the figure below is an example of the regulator circuit that is present in the Arduino. The regulator portion is everything to the right of the reverse voltage protection diode D1.

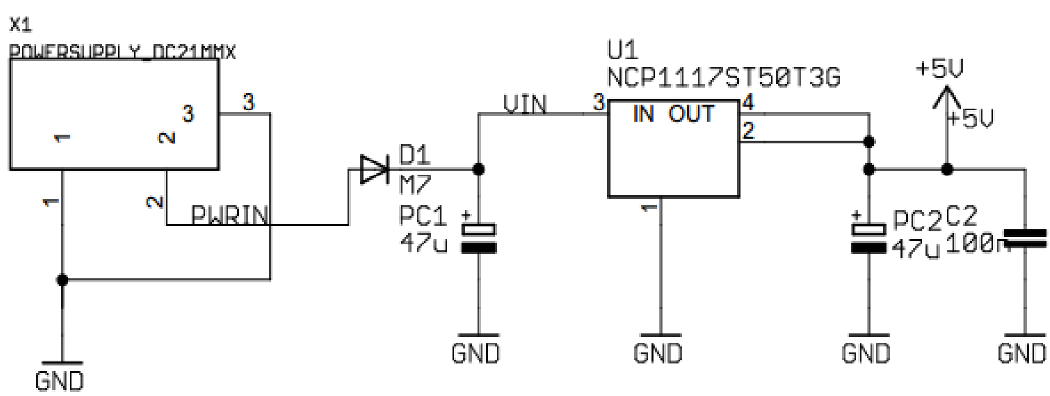


Figure 41: Voltage Regulating circuit of an Arduino.

Regarding our output voltage, we will be using a single-phase 12V AC output at our injection sites. We believe it proves the concept we are aiming to test and is considerably safer than using a typical high voltage power line. It also lowers the financial burden of all parties involved as we can use higher gauge wiring and more affordable off-the-shelf components. There were also administrative limitations to us being able to use high voltages, as we were not allowed to tap into the grid of our institution.

To create these outputs, we will use a Tektronix AFG3022C Arbitrary/Function Generator to generate a 12VAC single-phase signal at 60Hz. From there, we will have a source of 12VAC single-phase which we will then control using the PCB. We can then use a relay to control when to send these 12VAC pulses. With this relay, we can control the operation of the 12VAC power supply digitally, which allows us to only send these 12VAC power pulses when we are interested in locating a fault. Furthermore, this allows us to designate which injection site we want to send this pulse to. The provided function generator will be one that we have sourced from our university. We believe this is a much simpler way of creating a signal source we can use, instead of using an AC-AC step down transformer, which adds unneeded cost and complexity.

4.5 Safety Constraints

Our project's objective is to be able to detect faults and locate them in a power system. That means that we aim to implement it in real-life power systems. That is a problem, given that power lines usually carry up to 380KV in the United States. As college students, we lack experience working with such high-power levels, which can be hazardous if you don't have the necessary knowledge to work with it. Even with sufficient expertise, it would still be considered highly dangerous.

To avoid any possible risks of working under dangerous conditions, we decided to modify our model, so it is more manageable and safer to work with. In our demonstration, we will be using 12VAC as the maximum voltage rating of our power grid model. This constraint will allow us to engage with it without worrying about handling high voltage levels. We will use a power supply that controls the voltage level to guarantee we don't go over it. We are using a 120VAC power supply via a wall outlet, but by using a step-down transformer to lower our voltage level, we can ensure a safe working environment. As engineers, safety is of the utmost importance, and it is not optional to our work. Any way that we as designers can make our system safer is going to be something that we weigh heavily to determine if we need to implement it or not. A 120VAC source will generally have a 1.2mA current. Contact with a current in the 1-2mA threshold could lead to experiencing a painful shock. Any sort of injury hazard in the project is unacceptable, and if our system is still replicable on a safer voltage level, then we have a duty as engineers to ensure that our system works as safely as possible.

Although 1-2mA current is still within the let-go threshold, we do not want any sort of injuries to occur while testing. The let-go threshold is the current level in which a person’s muscles contract due to electric shock. This is dangerous because if someone’s muscles are contracting during electric shock, they will be unable to let go of the equipment shocking them in the first place. This means that someone else must come to their aid and turn off the electrified equipment in order for the victim to walk away from the system. This is a terrifying threshold, and we will do everything in our power to ensure our current levels do not reach this point. This point tends to be 6-10mA current for adults, and 3-5mA for children. Although some adults can withstand up to 22mA, we will not be allowing our system to reach a current that high. It is not a risk worth taking.

This same power supply will also prevent abrupt current rises. Another of the problems with our implementation is that we are going to be dealing with fault simulations. For this, we will short our circuit to ground. Shorting a circuit can make the current increase way above safety levels. That is why a signal generator is so beneficial as our power supply because it will protect us from harmful conditions.

Even though 12VAC is safer to work with, there are still many consequences from poor handling of our model. 12VAC is still a flammable hazard. Therefore, we must be incredibly careful with any potential fire hazards surrounding our electrified equipment. Part of the safety checks for our design will be ensuring there are no unnecessary parts surrounding the test environment. This is to ensure that if there is an error with the testing, there are no lasting safety concerns such as a fire or electrified items that should not be experiencing a charge.

If there is an electrical short, the temperature can increase so high that it can be harmful. For this reason, we will utilize special gloves that will allow us to handle the cables if they are too hot. We will also avoid having any wires exposed to prevent accidental shorts. This means that before going through with a test, we must inspect our equipment to ensure that all wired are properly covered and insulated, so that they are not a burn or temperature hazard to any engineers working in the test environment. These shorts can create sparks that can ignite flammable objects nearby. To prevent this from happening, we will create a hazard zone. The hazard zone will be composed of all the areas surrounding our model. All flammable objects that can cause fire will be prohibited in this area. Also, we will not allow any unauthorized person or someone that doesn't comply with our minimum safety apparel (gloves, protective glasses). This is to ensure that the test environment is only comprised of people who understand the risks associated with remaining in the environment, which will help to raise the safety standard to a higher level and avoid carelessness.

Overall, safety in engineering is one of the most important things to consider in design. If your design is incapable of keeping people safe, then it is not a design worth using. The health of those working on the project is always going to be the number one priority, and our group will be taking many steps to ensure that everyone involved is going to remain healthy and safe.

4.6 PCB Standards

Printed Circuit Boards (PCB) standards have a big importance when developing PCBs because they give us the guidelines and preferences of the industry on how to correctly build them. We need to design our PCB, and using these standards will ensure that our final design will have many benefits. By following them, our design process will be more efficient because we will have best practices and industry proven techniques. It will also be more consistent with other industry products, and this will facilitate the understanding of our design to others in the industry that may want to fix or recreate something like our final product.

We will be following the standards proposed by IPC, which is a non-profit organization founded in 1957, and is one of the leading sources for industry standards and their mission according to their website is to give competitive excellence and financial success to all of those who participate in using their standards. We will follow the Generic Standard on Printed Board Design developed by the IPC-2221 Task Group (D-31b) of the Rigid Printed Board Committee (D-30). In the document they clarify that they provide two types of guidelines, one where it is not necessary to follow but that it should be considered, they use words like "should" or "may" to denote it. And they use the word "shall" for mandatory aspects that have to be followed unless there is enough data to justify this exception.

In our case, the IPC-2221 standard suits us the best because we are working with a simple PCB, but IPC also offers more in-depth standards for more specific PCB characteristics. They offer IPC-2222 for rigid PCBs that are built into a stiff base, our design is also going to be rigid, so we are also going to be following these standards although. The IPC-2223 is made for flexible PCBs like those used for wearables, the IPC-2225 is for multichip modules, and IPC-2226 is for high density interconnect PCBs.

IPC-2221 will help us design our PCB in the most efficient and correct way possible. They offer guidelines on board size and shape, component placement, trace and space, via design, thermal considerations, and implementations.

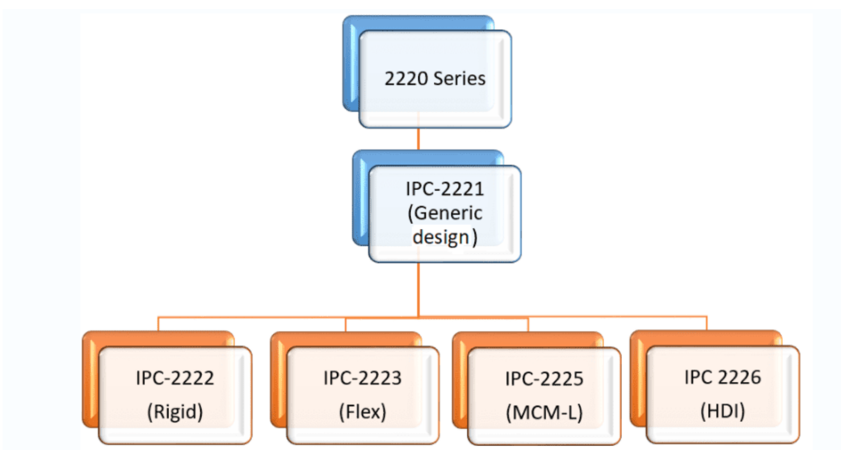


Figure 42: IPC 2220 series family diagram

It is also important to understand the IPC classes and their definitions. These classes will let us know how much quality our product needs. Each class has its own characteristics and guidelines. This way we won't use the same quality standards and degree of inspection for a kid's toy that has a PCB that lasts 1 year, compared to medical equipment that needs as much precision as possible and that it is made to last many years. For class 1, we have general electronic products, these have short life cycles and the lowest quality PCB. They are not made to be fixed or to last for too long, are mostly produced in big quantities, and it can be found in toys or flashlights. Class 2 are dedicated service electronic products, they are expected to have larger life cycles, and it provides better quality than the lower class. There are many standards that are shared between class 2 and 3. But for class 3 we have high-reliability electronic products. They must be as precise as possible, and it is critical that all forms of errors are minimized. Class 3 is the highest quality of PCBs, and it is usually found in medical and aerospace devices. The following table was made by Matric Group, and it compares the most important aspects between IPC classes.

Table 11: IPC Class definitions by Matric Group

	IPC Class 1	IPC Class 2	IPC Class 3
Category	General electronics	Dedicated service electronics	High-reliability electronics
Life Cycle	Short	Long	Very Long
Quality	Cheap	Good	Failproof
Examples	Toys, flashlights, smartphones	Laptops, microwaves, mining equipment	Aerospace, military, and medical applications

We are going to be working with IPC Class 2 PCBs because it is a great balance between quality, cost, and time available. Class would be unnecessary because we don't need that level of accuracy, and class 1 quality is too low for our standards, so we are going to working under IPC class 2 guidelines.

We will be using 1.0 oz/sq ft copper thickness because we don't need much current within the PCB, and it is good enough to dissipate heat from current without getting too hot. According to IPC, this will mean that our minimum copper thickness would be 35 µm.
Component placement guidelines are very important because they help to avoid overlapping or placing wires too close from each other. This may cause electrical interference or physical damage because you can create an unwanted short, or maybe if traces are too close electrons can start to go between the two, causing malfunction in our PCB. There are many guidelines about negative spaces, which are those places where there are no components, but one of the most important to follow is drill holes. IPC standards will help us determine drill holes characteristics such as width-thickness ratio or how far from components they should be from all other components, so it doesn't affect it.

Regarding spacing of wires, our PCB will be considered low voltage because it will be working under 15 V, IPC says that in these cases they recommend having a separation of at least 0.1 mm between any two conductors. And in the case of power conversion devices, it must be 0.13 mm or larger.

4.7 Coding Standards

When coding, it is important to use modern coding standards in order to create a well-organized and clean code. These guidelines and best practices will assist the team in ensuring the creation of high-quality code that is both consistent and readable for others that may look at it or alter it later. One of the standards that may be used in the project is a DynoNet template provided by one of our review board members. This template will be an excellent way to make sure our use of DynoNet is consistent with previous applications of the machine learning technique in a way that anyone who has used DynoNet previously will be able to understand what has been done by our group.

4.6.1 Google C++ Style Guide

Another example of an important standard to be used when we are creating our code will be the Google C++ Style Guide. The C++ style guide is a general guideline that helps ensure the code is readable, easy to work with, flexible, and consistent with modern coding techniques. This guide has important information regarding how certain techniques should be used and things to avoid when writing your code. This comes with the idea that small mistakes or errors may be made more easily when using certain unoptimized techniques. As programmers and engineers, we want to ensure that we are working as efficiently as possible while also working in a way that does not overcomplicate the coding process unnecessarily.

In any software project, maintenance and debugging are a constant process. The Google C++ Style Guide’s commitment to simplicity and readability will reduce the amount of effort needed to maintain and repair any malfunctioning code. This will mean that the engineers in our project will have less bugs they need to fix and the bugs we do have will be easier to modify and repair. This timesaver means that our engineers can take the time we would be spending debugging and work on something else in the project like hardware design or implementing new software features, increasing our efficiency greatly and increasing the quality of our design to new levels.

The Google C++ Style Guide is also incredibly important as a standard that can help our design maintain a scalable foundation. This means that as the scale of our project increases into real-life applications, the code will still be effective and malleable to the situation. As engineers, we want to create something as helpful as possible for the world, and if there is anything that we as designers can do to maintain scalability for real-world applications, then we will work our hardest to make sure it can be achieved. Another part of the Google C++ Style Guide that will assist in future real-life applications is maintaining consistency with industry best practices. By keeping track of how engineers have been approaching coding techniques and application in their years of service at Google, they have been able to create guidelines on common practices within the industry. This will be incredibly helpful if our code is ever picked up by a software engineer in the industry because they will be used to the coding techniques used by our project as they will hopefully line up with common techniques.

Overall, having a coding standard will help create a useful structure to our code while also making the debugging and maintenance process a much smoother and more efficient task. Using industry standards and scalability techniques will also help keep our project relevant for real-life applications, an important factor to consider when wanting to make a design that will be useful to the world. Ultimately, the Google C++ Style Guide will help us maintain the course to create a working project that can continue to be used after this course has ended.

5 Comparison of ChatGPT

Chat GPT and other AI have been drastically increasing in popularity in recent years and have also dramatically increased in the usefulness they can provide. In this section a summary and brief history of Chat GPT will be given as well as discussing the pros and cons of using Chat GPT in a senior design project.

5.1 A Summary of Chat GPT

Chat GPT was released by OpenAI in November of 2022 and has since become an increasingly popular generative AI tool. It is used for many purposes, some of which include essay writing, solving math equations, coding debugging and assistance, writing or proofreading a story, or general help. The use of AI assistance is a somewhat controversial topic, and in this section the potential benefits and drawbacks of using generative AI for assistance will be discussed.

Before discussing the pros and cons of these platforms, it is important to know how generative AI is used and how the AI obtains its information. Chat GPT uses a complex algorithm known as a large language model to pull information to feed to the user. In order to understand how ChatGPT obtains and chooses the information it feeds the user; it is important to understand what a large language model is. A language model is a probability distribution that uses incomplete sequences of text and words and creates a probability of how the text sequence would be completed. This can be an incredibly complex problem to solve based on the complexity of language and how there could be a near infinite number of ways to complete a text sequence. For this to work well on a large scale, artificial intelligence analyzes billions of sets of unlabeled text to get an understanding of common grammar, language patterns, and natural speech patterns. An important thing to note about this is that the AI is unable to create completely new ideas, any information given by the AI is given because the AI has processed that data before in its bank of billions of sets of text. This information may seem to be unique because of the way the language is assembled, but the art of the machine learning is the ability to connect the separate ideas that it has processed thousands of times before and combine them to fit the situation at hand for a complete idea.

Now that it is known how ChatGPT uses large amounts of data to understand language, it is important to know how that information is then given to the user. The user is given a prompt box in which they can ask ChatGPT questions or give it commands. Some examples of this could be something as simple as “What color is the sky?” or as complex as “Write a three-page essay about how particle acceleration is achieved with citations”. Using its extremely large bank of information, ChatGPT will provide the user with a solution to whatever is asked of it. In the next section, we will go further in-depth about the potential benefits that may be gained by using ChatGPT as a resource.

5.2 The Pros of Using ChatGPT for Senior Design

Now that there is a basic groundwork of how ChatGPT and functions and how it could be used, there must be an analysis of the potential pros and cons of using ChatGPT for Senior Design. First the possible benefits of using ChatGPT will be discussed, followed by potential drawbacks and challenges that may arise from using ChatGPT.

One potential benefit of using ChatGPT is increased understanding of a topic. This could happen through a couple of different methods. One example would be simply asking ChatGPT to teach me about something related to our senior design project, a prompt such as "How could machine learning be used to accurate detect power line faults?" will result in a wall of text from the AI with relevant information to the subject and possible solutions. This could be used to kickstart brainstorming or used as a method to find additional solutions and methods that were not originally discussed by the group. Accessing a knowledge pool as large as the one ChatGPT uses means that there could be many ideas and concepts brought up without having to search.

Another potential benefit of using ChatGPT is code debugging. Since our project is a machine learning system for fault detection, there will be code involved. Machine learning models need to be trained before they can be effectively used, and this is often done using Python coding techniques. For our project, the neural network DynoNet will be used. To use the DynoNet architecture effectively, several python files are needed as well as the inputting of our data from ADMS and AMI so that the machine learning model can analyze our data and learn from it. While working through this process, it is likely that a lot of debugging will need to be done due to the complexity of code required for a good machine learning system. Although it is very possible for debugging to be done manually within the group, debugging sometimes works best when it is done by someone who has never seen the code before. This is because the outside source is looking at the code with a clean slate compared to someone who has been working on tuning the code for hours and hours. With this cleaner lens, they may be able to spot possible logical issues or even coding syntax errors that may be missed while the code was being written. ChatGPT can fulfill this role of being the outside source to debug code. Posting our code into the prompt and asking ChatGPT to debug the code will result in ChatGPT updating the code along with posting an explanation of the shortcomings of the previous version. However, debugging with ChatGPT could be done in different ways as well. It could be as simple as asking ChatGPT about the error messages that are coming up and asking for how these error messages are usually caused or common solutions to those error messages. This way the code would be unaltered by the AI and the coder is simply pointed in the correct direction while they search for debugging solutions. ChatGPT could also take the code and figure out which lines in the code are causing the error if it was previously unknown. This could show the coder which lines they need to focus on altering for the code to run smoothly. Ultimately, there are several different ways that ChatGPT could help with the debugging process that would be beneficial to the senior design project outcome.

Another potential benefit of using ChatGPT is the ability of the AI to create flowcharts for our report. ChatGPT, if given the proper instructions, is capable of making graphs and flowcharts. This could be incredibly useful for the design process as both a visual aid during the brainstorming phase of design and as an actual flowchart in our report. Having the ability to plug in a timeline of events into the AI and receiving a complete visual aid in return could make what was once a lengthy and verbose explanation into an easy to digest piece. This could also be used to save time for the user, as making a flowchart themselves could take a lot of time. This is because it may take a lot of tweaking in order to get the flowchart to look aesthetically pleasing as well as nailing down the exact sequence of events. Being able to plug in an already written text and receive an image within seconds is a huge way to save time compared to the 30 minutes to an hour or even longer that might be taken up by manually making the flowchart. This of course depends on the complexity of the flowchart, but it is still worth discussing ChatGPT's potential benefit of creating flowcharts and other imagery with the information written and designed by the user.

In conclusion, the use of ChatGPT could result in several potential benefits and advantages for our senior design group, and it is important to consider when it may be useful to implement AI aid into our design process.

5.3 The Cons of Using ChatGPT for Senior Design

Using ChatGPT is not all benefits though, it is also important to acknowledge potential drawbacks of using ChatGPT in the senior design process. The first drawback of using ChatGPT would be stifling creativity. Although ChatGPT is a useful tool with a lot of information, it is also limited by this same information base. The references that it can provide are only the ones that are currently in its database, meaning that if a source or article has not been added to the database then ChatGPT will not be able to use the information. This means that the user of ChatGPT might be missing out on important bits of technical knowledge or research if they rely on ChatGPT for information. Missing out on this knowledge could then mean that the user is not using the most efficient and effective solution. When working on the design, it is important to look at all possible solutions and techniques so that the most effective solutions and methods are chosen, using a limited knowledge base means that it is very possible to miss out on the best method, and that is why the user must not rely on ChatGPT for information and ideas.

Another potential drawback of using ChatGPT is that the information provided is not always accurate. ChatGPT is known to sometimes make things up entirely. This could have devastating effects on the user if they are using ChatGPT as a source for important technical information. If the AI feeds the user incorrect information, the entire design could suffer. It is vital to the designers that their information is accurate, and using information obtained without a citation is high-risk due to the potential that the information could be wrong. If ChatGPT provides a source that can be checked along with its answer, then the information could be considered valid, but without a source it is important to scrutinize any potential information that is given by the AI.

Ultimately, ChatGPT has potential to be useful in Senior Design, but it is important to be careful with its usage and scrutinize the information provided. The students working on their design must be careful to explore all possible solutions and make sure that their work is still their own, and not a regurgitation of ChatGPT. This way, their design will maintain a creative and original approach while simultaneously ensuring the technical information used in the design is accurate so that the required specifications can be appropriately reached.

6 Hardware and Software Design

In the next section we will go through all the components that we are going to use to design our hardware. There will be information about our schematics, PCBs, testing, and more. The main goal of our hardware will be to be able to collect the data necessary so it can be analyzed later. Different components like regulators, multiplexers, and switches will be used to accomplish this goal.

6.1 Measurement Collection

To find the distance of the fault, first we need to measure the impedance of certain places. To do this we will use Ohm's Law equation and solve for Z. By using $Z = V/I$ we are going to be able to calculate it, all we need to find is current and voltage.

The Atmega328P can handle analog inputs, this means that we can directly measure voltage using this capability. All we need to do is to build a voltage divider because our microcontroller won't work if we input more than 5 V. In our case, we will be working with a maximum voltage of 12 V, so the voltage divider will help us lower that amount of voltage and get the appropriate readings. Our idea will be to solder wires in every node that we need to get measurements from and then connect them all together to a multiplexer. We will choose which wire we want to read from using our PCB to control the multiplexer, and multiplexer will be connected to the analog input of our PCB.

To measure the current, we will use a current sensor that uses a current transformer that converts the current passing through into voltage. As said before, our PCB can measure this voltage. Our current meter can be easily clamped to any wire we want, so it is not necessary to have many of them connected if we have enough to do our presentation.

6.2 Fault Creation

We will simulate our fault by using potentiometers that we can adjust to low resistance if we want a fault. Our potentiometer's lowest resistance is 5K ohms, so our system will be high enough that 5K ohms can drag a lot of currents to simulate a fault. The 5K ohms can simulate a fault to the ground at the fault location if the other load is much higher than 5k ohms. By doing that we will create an open on the line, which we should be able to see with a high impedance. We will start the testing with the potential meters at a higher resistance level, so we don't overwhelm our design.

6.3 Power Supply

The power supply will be made of two main components, the Ac power source, and the DC power source. They will each be responsible for the microgrid representation and the PCB.

The PCB will be powered by the 120V AC wall outlet with an AC to DC converter, the PCB will be receiving 9V DC after the converter. Because the design is very sensitive, it requires a consistent and regulated power supply, so DC would be best to ensure reliable and accurate performance. We chose this because it is commercially available and simple to use. We can also consider powering it with a battery if we need to move the system.

The main circuit will be powered by a single phase 12V AC signal generated by the Tektronix AFG3022C Arbitrary/Function Generator. This is the most accessible function generator for low voltage single phase AC signal, it also comprises advanced functionality such as peak-to-peak voltage level and phase shifting. Also, everyone on the team had prior experience with the AFG3022C so we can approach the power supply with confidence.

We cannot find any information for the internal current limiter on the AFG3022C, so we decided to implement a current fuss to protect ourselves and our circuit. The fuss adapter we want to purchase is on backorder, so we will be building a fuse adapter with the senior lab manager.

6.4 Schematic

It is crucial to understand how all our devices work together. For this reason, we created a schematic for every component to have an idea of how they communicate with each other to help us achieve our goal of collecting all the measurements. Next, we will examine each component we used and its schematic. Finally, we will be showing our overall schematic. This schematic is composed by putting together every part.

6.4.1 Current-meter schematic

To calculate the current of our main system, we will be using a non-invasive current sensor based on a current transformer. These sensors will use the electric current that passes through its hole to produce an output voltage. Our microprocessor can use this voltage to give us the current passing through it. In this case the maximum voltage that can be input for our processor is 5V, so we used a voltage divider to ensure that the input voltage is never higher than 5V.

The voltage measured will be the one across our burden resistor (R7 in Figure 34), and we can see that it will be connected to one of the multiplexers so we can receive it as an analog input. For our demonstration, it is essential to measure current from six different places. That is why we will have six different current meters connected to a multiplexer.

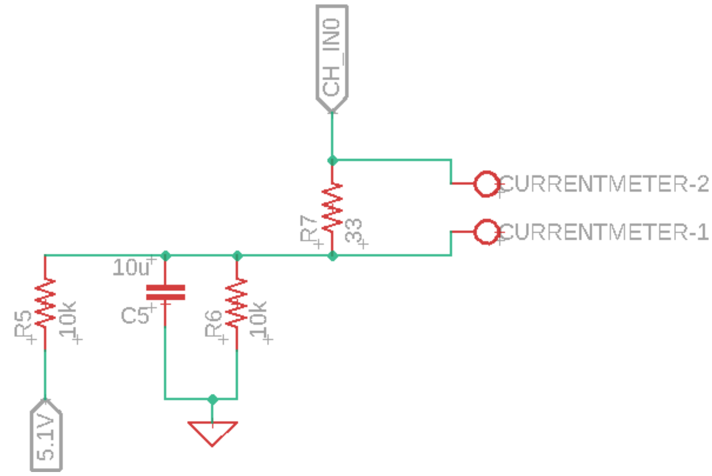


Figure 43: Current-meter Schematic

6.4.2 Voltage-meter schematic

The ATMega328P has multiple analog pins, which means that we don't need external components to measure voltage. All we need to do is to create a voltage divider and connect this output with one of the analog pins (or multiplexers in this case). We added a diode after one of the probes to eliminate negative values. And we are also using a Zener diode with a reverse breakdown of 5.1V so we don't damage our MCU.

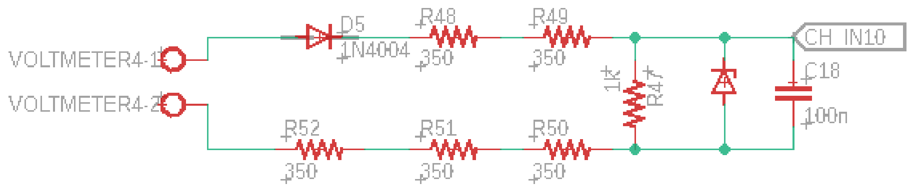


Figure 44: Voltage-meter Schematic

6.4.3 Multiplexer schematic

For our project, we aim to collect either current or voltage data from at least twelve places. This is a problem because our MCU only has six analog inputs. Therefore, to solve this issue, we will use two eight-input multiplexers together to have sixteen different analog inputs. By implementing this method, we can use two analog inputs to read from sixteen places.

Figure 45: Multiplexer Schematic

6.4.4 9V to 5.1V Voltage Regulator Schematic

We will use a wall wart to power our ATMega328P. This wall wart provides 9VDC output and 500mA. Since the maximum input voltage for our processor is 5.5VDC, we built a voltage regulator to obtain our desired value.

We used WEBENCH to design our regulator. WEBENCH is an online tool by Texas Instruments that allows you to develop custom AC-DC and DC-DC regulators. We set our input voltage between 8 - 10V to leave room for any wall wart imperfection. For the regulator output, we specified that we wanted 5.1V with a current of 200mA which is the standard for ATMega328P.

Finally, we added a 2.1mm barrel plug to the input of the regulator, with a switch to turn our device on and off.

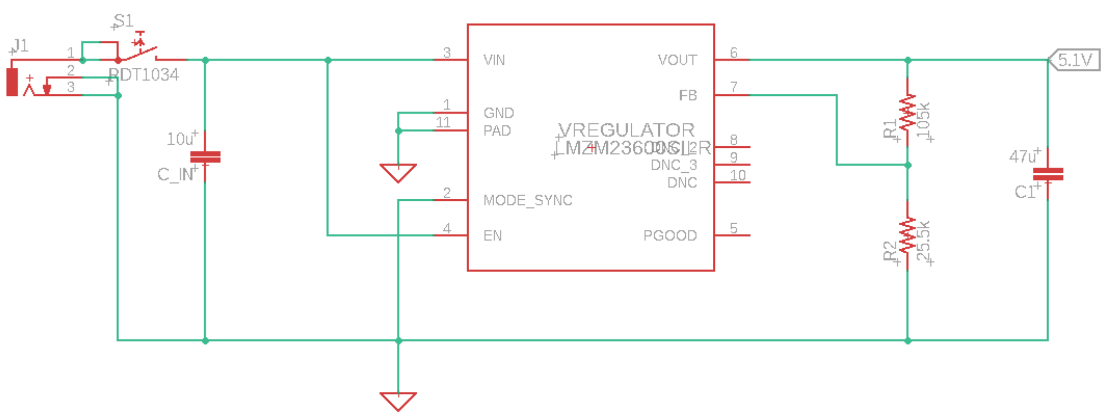


Figure 46: 9V to 5.1V Voltage Regulator Schematic

6.4.5 MCU Schematic

The ATMega328P has many functions, but for our project, we only need its digital pins and its analog input pins. We implemented some basic arrangements for the MCU design.

In this design, we added a 16MHz crystal clock and connected it to our power supply for it to function. Additionally, we are using pins 15, 16, and 17 (figure 38) to program our processor with another Arduino. We connected these pins to a four-pin header for a more convenient and efficient way to connect the Arduino with our MCU.

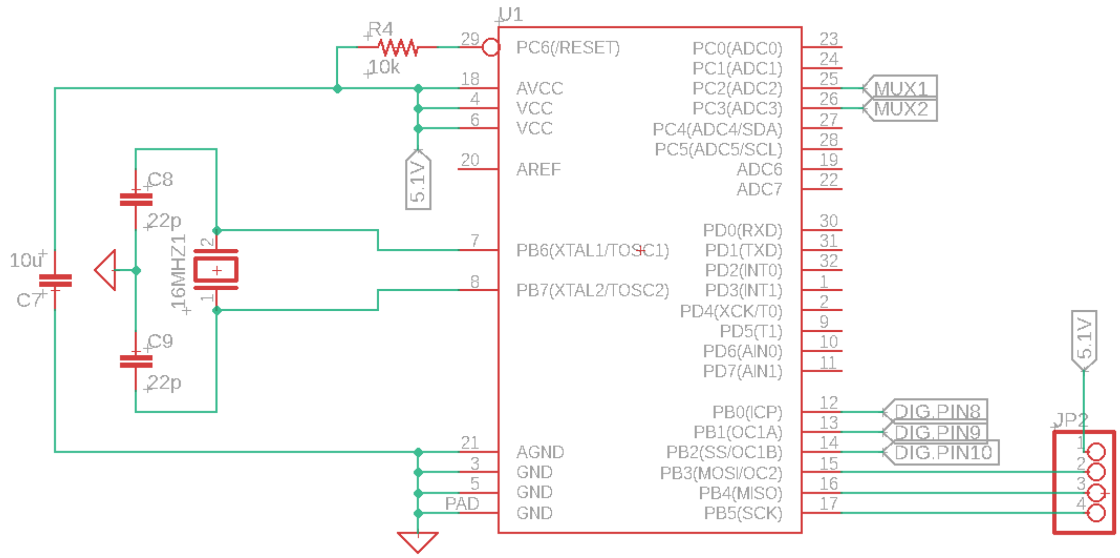


Figure 47: MCU Schematic

6.4.6 Overall Schematic

Figure 39 shows every component we are using. This overall schematic is essential because it provides a great visualization of how all previous schematics are connected.

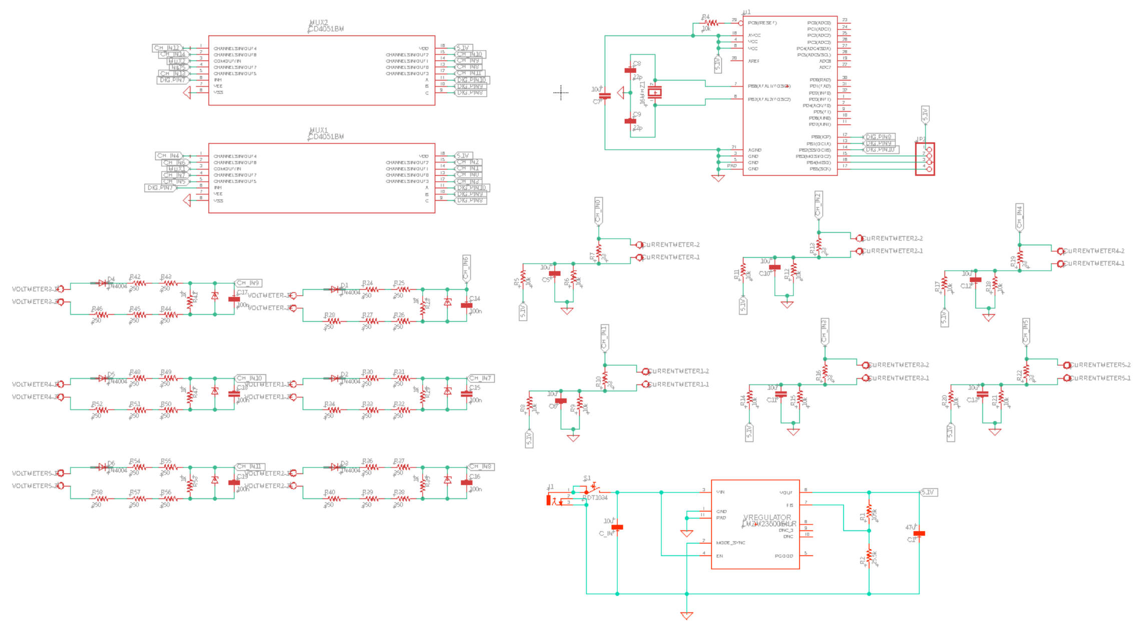


Figure 48: Overall Schematic

6.5 Pulse Injection

We will be injecting 5 V pulses into different parts of our main system to find our measurements. We do this so we can have a consistent response from every point. Signals will be generated from our signal generator. Injections will be square waves.

Our signal generator will be outputting the corresponding signal constantly. We will decide the location of the injection with a multiplexer which will be connected to the PCB and that it will be controlled by our microcontroller. One side of the multiplexer will be connected to the signal generator and the other side will be connected to multiple wires that will be attached to every testing point. So, every time we want to send a pulse to a specific location, all we need to do is choose which path do we want to use.

To have a better understanding of the idea of injections we created a simulation using LTspice. We recreated our main system by using a 12 V sinewave, and we use resistors and inductors to represent loads.

For the pulse, we created another circuit (figure 34) that has a rectangular wave. Here we placed a switch (S1) that will toggle every one cycle of the rectangular wave. When connected, this circuit will send pulses of one period long to a specific node in the main system.

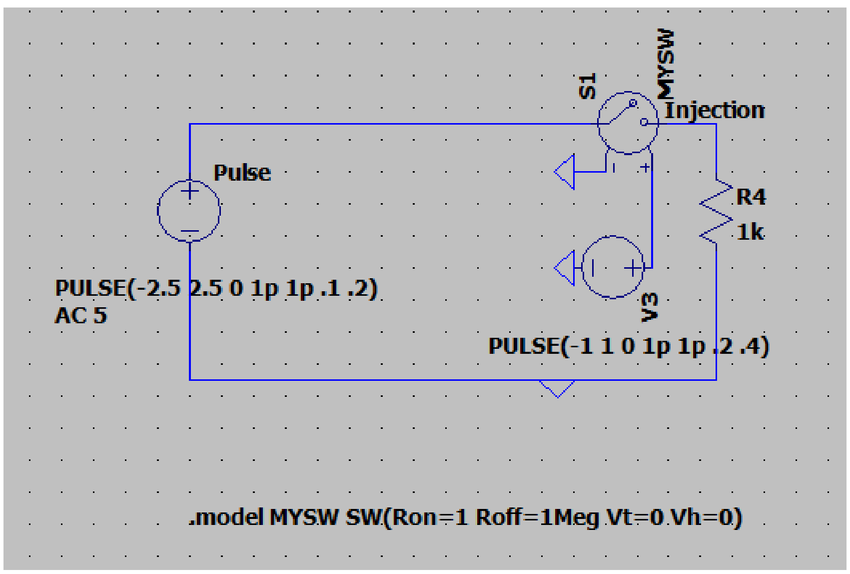


Figure 49: Pulse Circuit simulation

A fault (S3) will be simulated by another switch in the main line (figure 35), which will toggle every two periods of the pulse. When combining all of these, we will get a functional design analysis that, thanks to the tools provided by LTspice, will help us to measure and understand how a real system would behave before and after injecting a pulse.

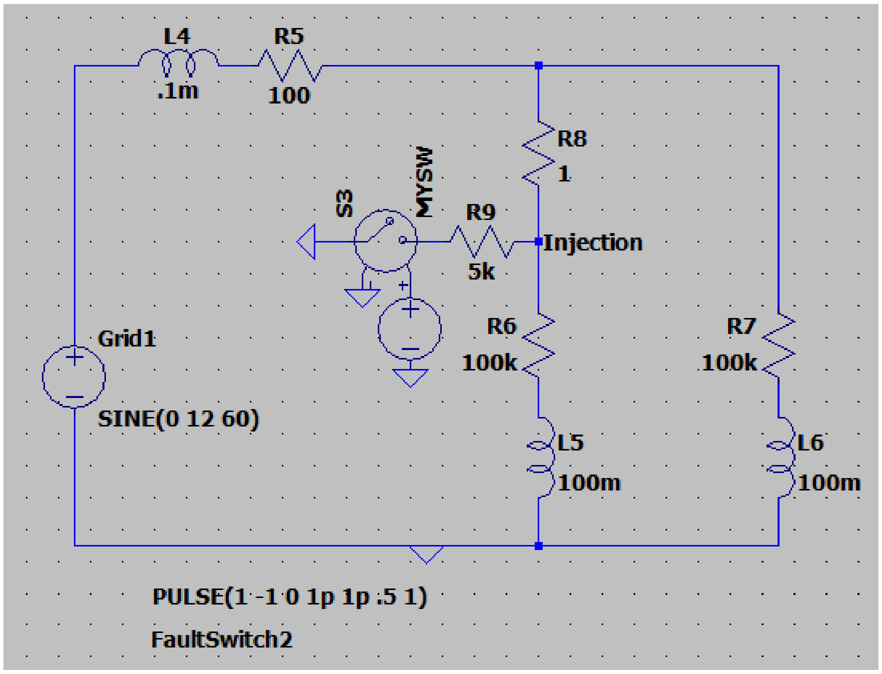


Figure 50: Main circuit simulation

In the next figure we will see how impedance (red) decreases from almost 1M Ohm to almost 0 Ohms when a fault (green) is present.

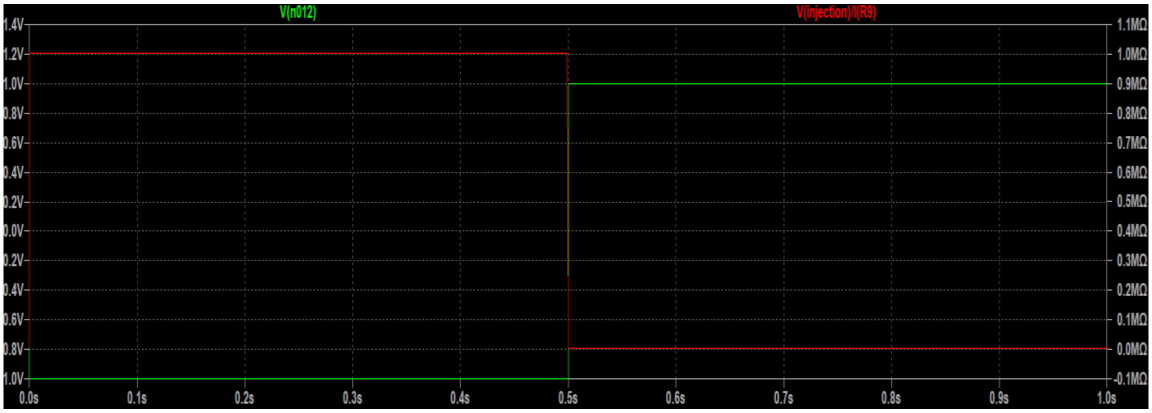


Figure 51: Simulation of impedance change after fault.

6.6 PCB

After finishing all our schematics, we developed our PCB using Eagle. Most components used in it are from Eagle Libraries, although we had to download part of them (like the CD4051B) from Ultra Librarian to get their pinout and 3D model. We organized our PCB so we can have all voltage meters on one side, and the current meter on the other. We also left all power-related components on the left side and used the right side to arrange the MCU and the multiplexers.

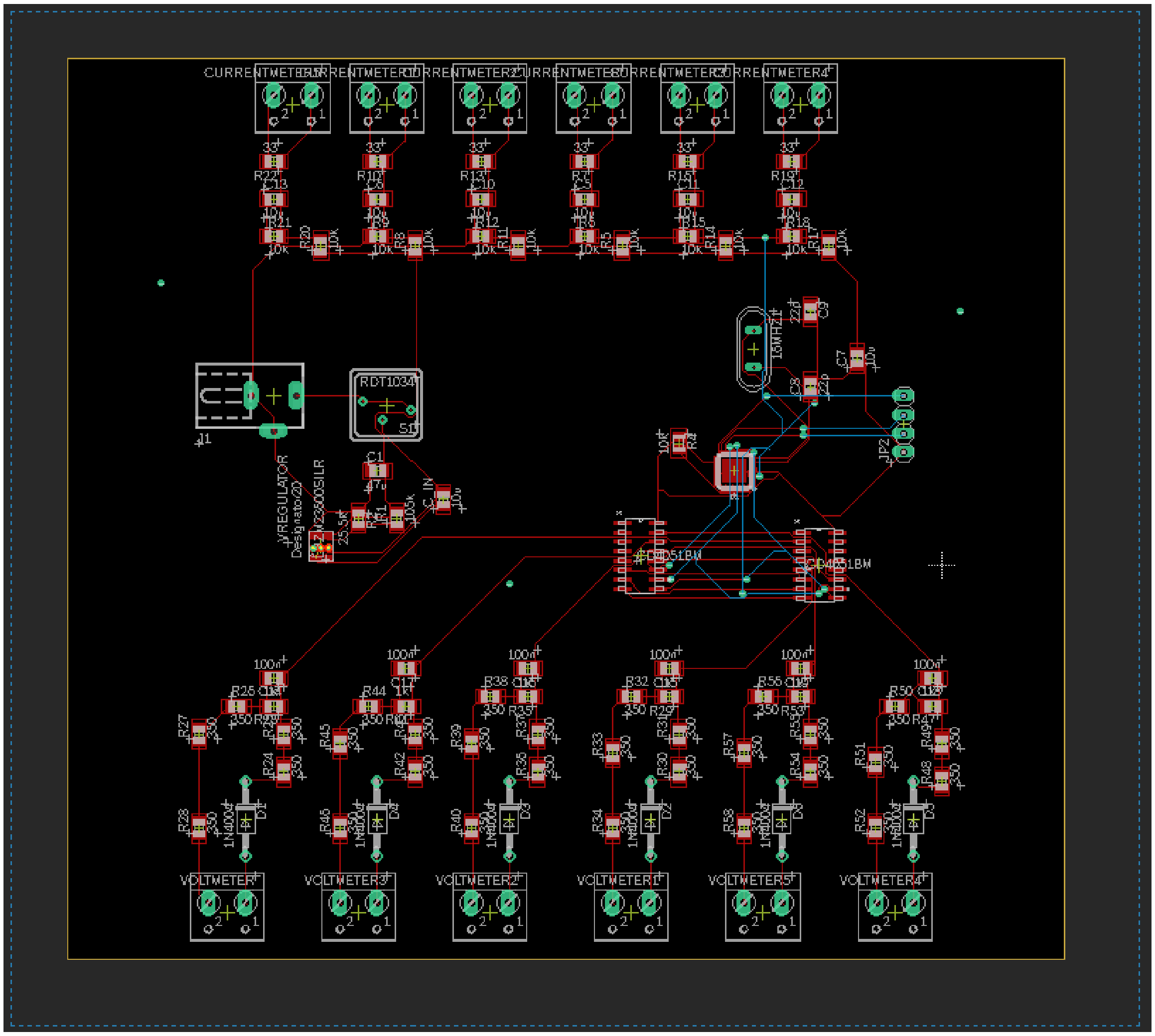


Figure 52: Final PCB design

6.7 Software Design

This section will cover the scope of the software design portion of our project. There are various design areas to be talked about, our microcontroller will need to be programmed using our chosen IDE, the Arduino IDE. We will also need to program and train our machine learning model; this will be done through Python through a DynoNet template that was graciously provided by one of our review board members. We will then need to tie our trained model into our microcontroller and flash our finished code into the ATMega328p. Further on, we will be talking about testing and verifying our code. Below is a high-level diagram of the design flow for the software components of our project.

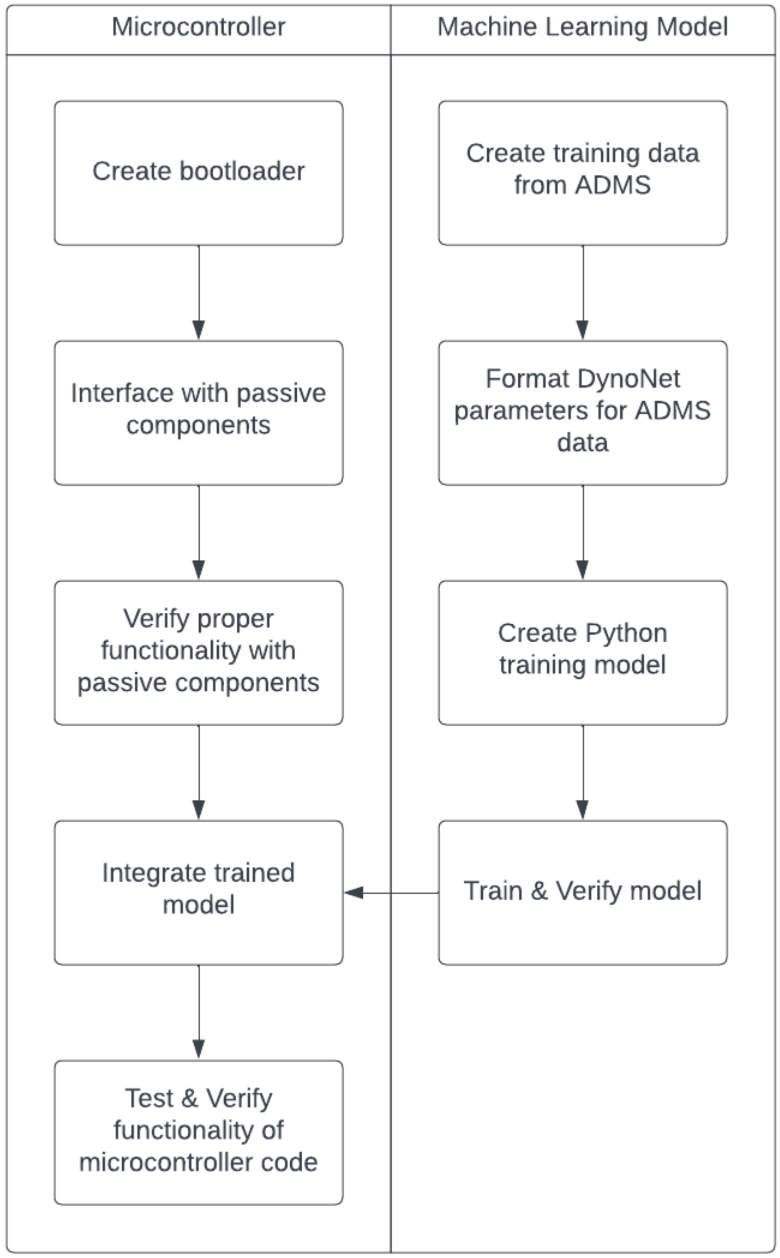


Figure 53: High level flowchart of software design

6.7.1 Bootloader design

The ATMega328p comes 'raw' from the factory, that is there is no way of communicating with it, at least not for a usable purpose. To begin communicating with the ATMega328p, we will need to burn a bootloader onto it. Thankfully, the Arduino documentation provides us a convenient way of burning said bootloader onto it. You may now be asking, why is burning the bootloader a necessity? Well, if we burn the bootloader onto the ATMega328p, we can then upload our sketches (Arduino's naming of programs) directly to the ATMega328p through its Tx/Rx pins instead of having to use an In-circuit Serial Programmer (ISP) every time. This will save us a considerable amount of time as there will be a lot of debugging and chances in our prototyping phase. As a result of this, however, we will need a USB-to-TTL converter. Thankfully, we have two options for this, we can use a normal USB-to-TTL cable, which will allow us to connect directly to a host PC. Or we can use the Arduino as a USB-to-TTL converter by connecting the TXD and RXD pinouts on the board to our ATMega328p's Tx and Rx pinout, as well as grounding the RST pin and providing Vcc to the ATMega328p.

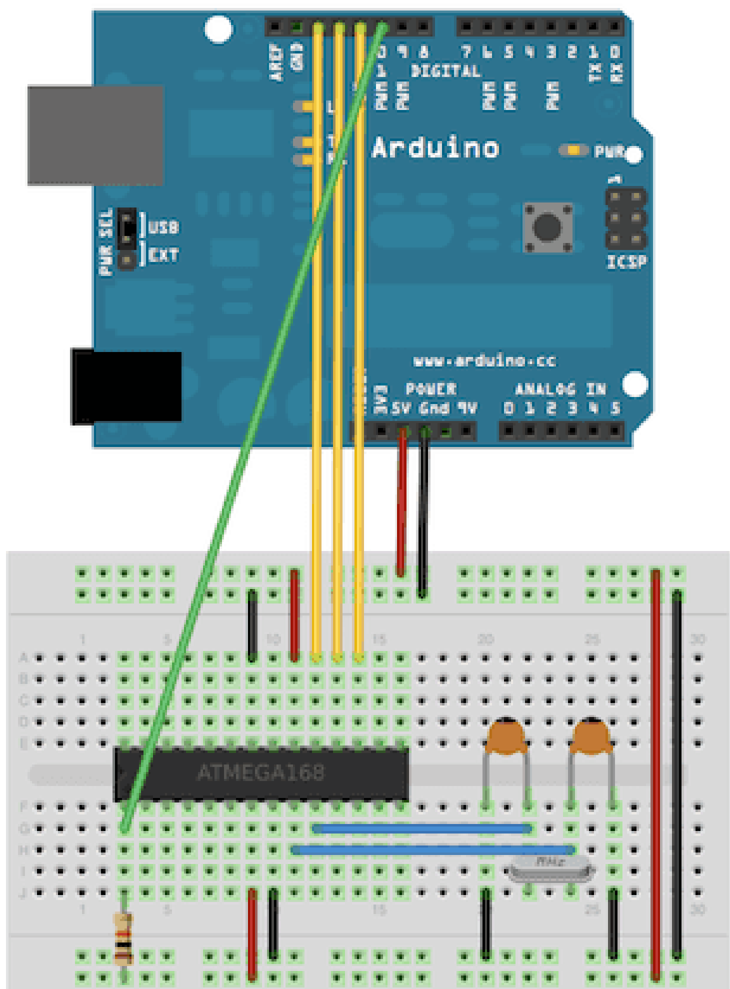


Figure 54: Breadboard diagram of Arduino ISP [8]

As we stated in a previous section, we will be using an Arduino board as our ISP, this is "the device that connects to a specific set of pins of the microcontroller to perform the programming of the whole flash memory of the microcontroller, bootloader included. The ISP programming procedure also includes the writing of fuses: a special set of bits that define how the microcontroller works under specific circumstances" [7]. To do this, we will begin by uploading the "ArduinoISP" sketch provided by Arduino to our ISP Arduino board. This basically configures our Arduino board as an ISP capable of flashing the raw ATMega328p.

From here, we will begin wiring up our Arduino ISP to our raw ATMega328p as shown in the image above. As can be seen, we will have to use a 16MHz clock, this is because the Arduino board we are using as our ISP runs at 16MHz, but the internal clock of the ATMega328p runs at 8MHz, therefore, to successfully have them communicate we must be using the same clock speeds. We will also be pulling up the RESET pin using a 10kΩ resistor since it is an active low RESET pin and connecting it to pin 10 on our Arduino ISP. Finally, we will connect pin 11 to pin 17 (MOSI), pin 12 to pin 18 (MISO), and finally pin 13 to pin 19 (SCK).

Next, we will then select the Arduino board model we are using as our ISP which is the Arduino Uno. Afterwards, We will configure it as ISP in the same tools menu. We can then finally burn our bootloader on the ATMega328p. The menu examples on the Arduino IDE are shown below.

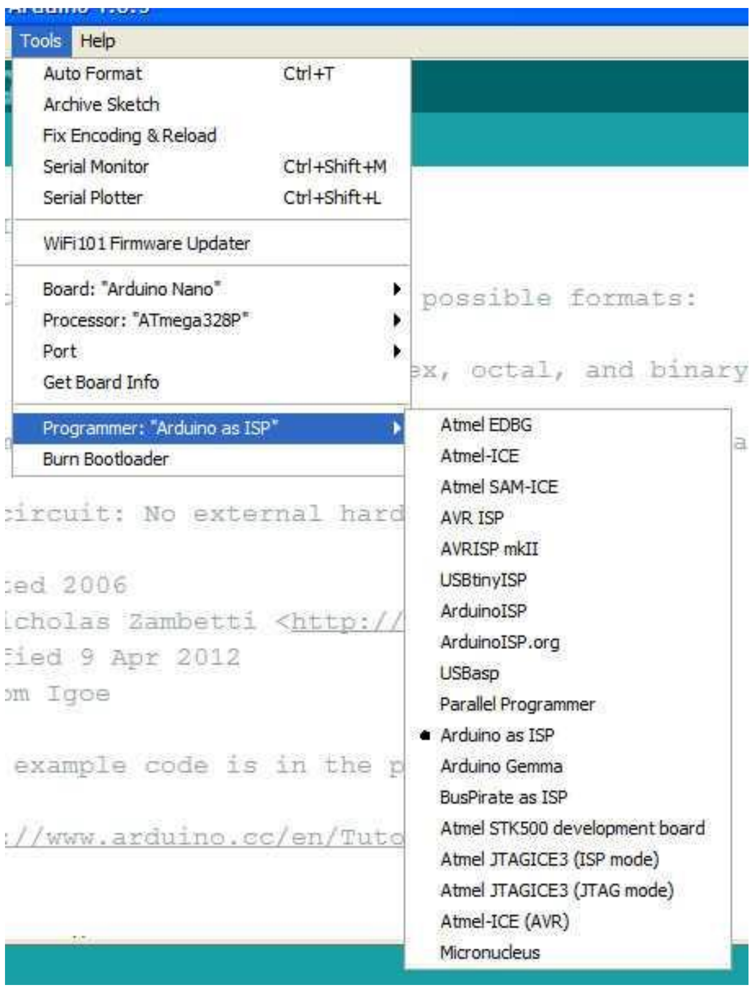


Figure 55: Menu options of interest [7]

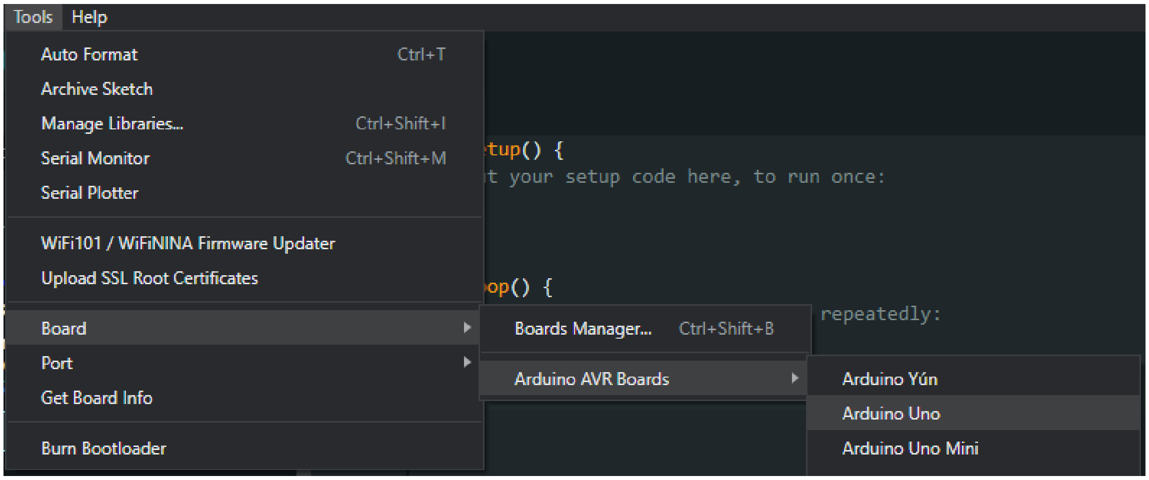


Figure 56: Menu options of interest [7]

6.7.2 Microcontroller Software Design

The software design that will be flashed into our microcontroller will play a critical role in ensuring that our project is a success. Special care must be taken in regard to the available amount of flash storage on our microcontroller, as 32kB is not enough memory to implement luxurious, space hungry algorithms or data structures. However, if this project were to be applied on a bigger scale, the ATMega328p would not be capable of handling this computational load, thankfully, swapping to a stronger microcontroller is not a big task. We only chose the ATMega328p due to its low price and since our project is proof of concept on a small scale.

Originally, we were planning on implementing a multi linked list data structure but decided otherwise due to spatial constraints and inefficiencies in traversal of O(n). Spatial complexity is similar for both; however, an adjacency list can be more efficient. It is also easier to visualize and is commonly used to model circuits. Our adjacency list will consist of a unique singly linked list initialization for each node. Each vertex node will contain its name, how many edges are connected to it, and an array of pointers to structures containing the information for each edge. The edge array of pointers will contain the impedance value of that edge, along with a pointer to the next vertex node. Basically, we access a vertex node structure to view the start of that node, then access its edge structure which contains the impedance as well as a pointer to the vertex node at the end of it. This system allows us to construct a bidirectional circuit representation of our grid while having a desirable time complexity of O(1) for add operations as well as O(V) for querying operations, where V is the number of vertices on the graph. Once our data structure has been initialized in our program, the program will then move on to the multiplexing and querying stage of operation. Below is pseudocode of our proposed data structure.

```
// Define the Vertices structure
typedef struct Vertex {
    char name;
    struct Edge* neighbors;
    int neighbor_count;
} Vertex;

// Define the Edge structure
typedef struct Edge {
    Vertex* vertex;
    double impedance;
    struct Edge* next;
} Edge;

// Create Data Structure
Vertex* createVertex(char name) {
    Create a new Vertex with the given name;
    Initialize the neighbors list to NULL;
    Set the neighbor_count to 0;
    return the new Vertex;
}
Edge* createEdge(Vertex* Vertex, int impedance) {
    Create a new Edge with the given Vertex and impedance;
    Initialize the next pointer to NULL;
    return the new Edge;
}
void addEdgeToNeighbors(Vertex* Vertex, Edge* edge) {
    Set edge->next to Vertex->neighbors;
    Set Vertex->neighbors to edge;
    Increment Vertex->neighbor_count;
}
```

Figure 57: Pseudocode of proposed data structure [9]

Our program will then multiplex through all the voltage and current meters present in our system. Each multiplex step will correspond with a change of the current vertex and edge of interest in our data structure. For each multiplexing step, it will receive the values of the voltage and current meters and then using Ohm's law $Z = \frac{V}{I}$, it will calculate the impedance seen at that injection site. This calculated impedance value is then stores in the edge structure; this edge structure has two pointers that each correspond to its respective connected vertex. This will be repeated at every injection site until all injection site vertices and edges have been visited.

```
void multiplexThroughInjections(){
    for(each injection site){
        double impedance = (voltage read from meter) / (current read from meter);
        addEdgeToNeighbors(Current Vertex, createEdge(Vertices, impedance));
    }
}
```

Figure 58: Pseudocode of proposed multiplexing

Once all the impedance data has been added to our data structure, we can then begin to process our data to determine where the fault is most likely to be located. For this, we believe the simplest way is to just iterate through each node of our data structure and find the edge with the highest impedance. One can visualize this as a sort of path finding algorithm, where the algorithm will "walk" itself to the edge with the highest impedance point. We believe this to be a naturally robust system given that the edge with the highest measured impedance will eventually lead to the fault as all other edges will either read at the circuits steady-state impedance or less. As it is traversing, if the impedance value of all succeeding edges is less than that of the current edge, then the algorithm will be considered finished, and the fault will be determined found. Of course, our algorithm includes a verification scheme that will be covered later. We believe the time complexity of this algorithm is of O(V) where V is the number of vertices. Below is some pseudocode of the proposed function that will walk the data structure trying to find the highest point.

```
void findHighestImpedance(adjacencyList){
    Edge* highestImpedanceEdge = NULL;
    double maxImpedance = -1;

    for(each vertex in adjacencyList){
        for(each edge in vertex){
            if(edge.impedance > maxImpedance){
                maxImpedance = edge.impedance;
                highestImpedanceEdge = edge;
            }
        }
    }
}
```

Figure 59: Pseudocode of proposed traversal algorithm [10]

Once we have reached the edge with what the algorithm believes to be the highest impedance, we will need a way to verify that it truly a fault or if it is just faulty metering equipment. This is discussed in heavy detail on section 6.7.6 however, we will be introducing our approach in this section. Basically, we will be utilizing a hill vs. cliff concept to confirm if it is a true fault or faulty metering equipment. Below is a visualization of this concept.

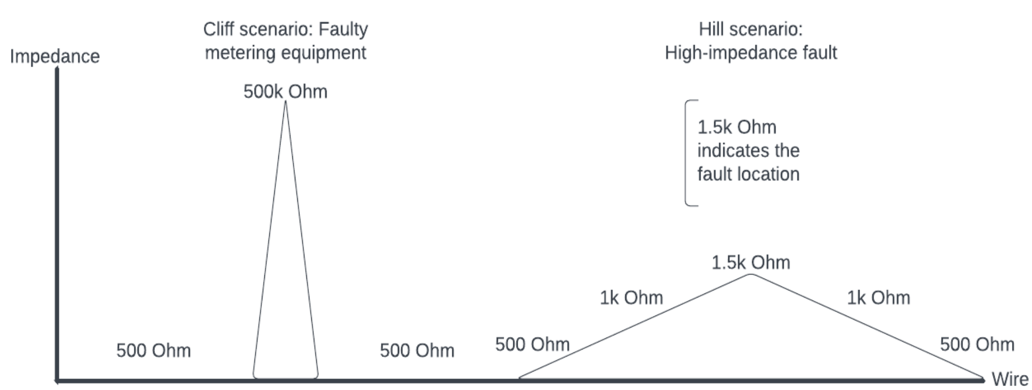


Figure 60: Visualization of scenarios

Our verification technique involves both statistical verification as well as using a pattern matching machine learning model. In this section, we will talk about the statistical solution. In 6.7.3 we will go over the pattern matching part of our solution. Our statistical solution begins from the edge with the highest impedance. From this edge, we will traverse to its connected vertices, from these vertices, we will read each the impedance of each of its connected edges and store them as part of our standard deviation sample. We will be storing all but the edge from where we came from. Once we assemble our sample set from both connected vertices, we will then compute both the mean $\overline{x} = \frac{\sum xi}{n}$ and the standard deviation $s = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{x})^2}{N-1}}$ where N is the amount of samples taken, x_i is the values of the impedances, and $\overline{x}$ is the mean value of the sample.

Using these calculated values, we can establish a mathematical definition for our hill vs. cliff scenarios. Our scenario classification will be as follows, if the standard deviation s is greater than the mean $\overline{x}$, then that will be classified as having a high probability of being faulty metering equipment. Likewise, a standard deviation s that is less than the mean $\overline{x}$ will be classified as having a high probability of being an actual high-impedance fault on our grid. Below is some pseudocode of how our proposed verification algorithm will work.

```
bool verifyEdge(Edge* highestImpedanceEdge){
    Edge* neighbor = highestImpedanceEdge->vertex->neighbors;
    while(neighbor != NULL){
        Add neighboring edge impedance values to tempDataStruct
        neighbor = neighbor->next;
    }
    double mean = calculateMean(tempDataStruct);
    double stdDev = calculateStdDev(tempDataStruct, mean);

    if(stdDev > mean){
        // High probability of being faulty metering equipment
        return false;
    }
    else if(stdDev < mean){
        // High probability of being an actual high-impedance fault
        return true;
    }
}
```

Figure 61: Pseudocode of proposed statistical verification algorithm [10]

6.7.3 Machine Learning Software Design

Another verification method we will be using involves a pattern recognition machine learning model that will be capable of discerning between an actual fault or faulty metering equipment and assigning a confidence value to its judgement. This method will work together with our statistical verification method to ensure that our final decision is of the highest confidence possible.

We will implement this using a supervised learning model. Basically, a supervised learning model is a machine learning model that is trained "using data that is well-labelled. Which means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that the supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labeled data." [11] This well labeled data will be coming from the results of our statistical verification method. There are 6 main steps in employing a supervised learning model [12]. In this section, we will go over the 6 steps of implementation and what our approach will be for these 6 steps.

Step 1 involves "determining the type of training examples". [11] For this project, our training example will consist of a sample set of impedance values. These sets will contain actual faults, this sample set will look much like our hill example. It will also contain faulty metering equipment read-out values, which will look like the cliff example. Specifically, since we are training it on identifying these hills vs cliffs, we will be highlighting to the model what specifically constitutes an actual high-impedance fault vs just faulty metering equipment.

Step 2 involved gathering our training set. Specifically, our training set will be coming from two sources. Source one is from the ADMS system our sponsors have allowed us to use. This data will be representative of the local power grid of a section of New York City. Of course, we will have to insert faults into the data because the data we will be receiving will be of a functional grid with no faults to our knowledge. Source two will come from our own circuit, we will gather the data into a format like that of the ADMS system. However, since we have total control of our grid representation, instead of altering the data, we will be creating a fault and then recording the impedances seen by the meters. This will give us a more "natural" impedance reading since we will be creating these faults ourselves. We will also need to gather out output set. The output set will consist of the input set, but with our desired outputs highlighted so that the model knows what the look for and what its result should be. For our purposes, this consists of highlighting the true fault and emphasizing the hill-like structure of an actual fault. If trained correctly, it should automatically disregard a cliff-like structure of impedance values.

Step 3 is about determining the input feature representation of the learned function. "The accuracy of the learned function depends strongly on how the input object is represented. Typically, the input object is transformed into a feature vector, which contains a number of features that are descriptive of the object. The number of features should not be too large, because of the curse of dimensionality; but should contain enough information to accurately predict the output." [11] Basically, for our designed input training set, we will have to describe features that are important to identifying the object, object in our case being the set of impedance values and the hill-like pattern. For binary classifications such as ours, a linear predictor function would suffice. Linear predictors involve creating a linear combination of a set of coefficients and independent "explanatory" "variables whose value is used to predict the outcome of a dependent (coefficients) variable." [13] For our purposes, we will feed this linear predictor function with our assembled feature vector. Quite literally, a feature vector for our purposes can best be described as assigning a numeric weight to each impedance, in our case, higher impedances carry with them a higher weight. Below is a good high-level interpretation of our approach.

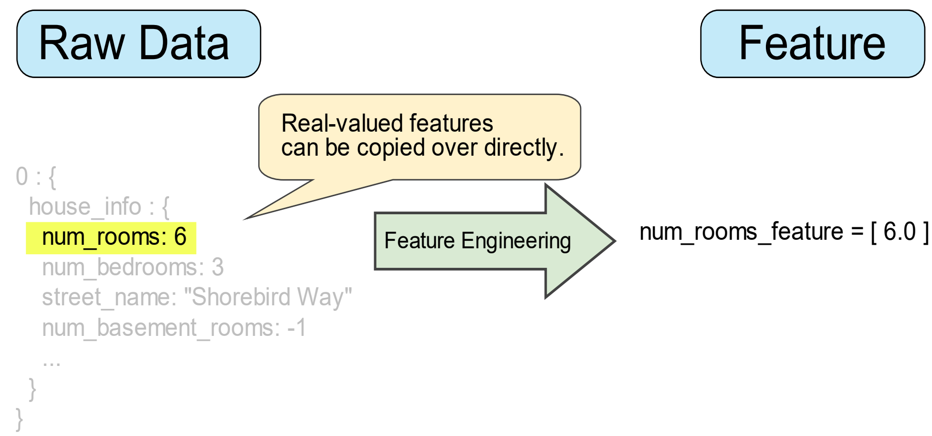


Figure 62: Feature vector [14]

Step 4 of this process involves determining the structure of the learned function and its corresponding learning algorithm. We believe for our model, which is responsible for doing binary classification, a logistic regression structure would be best suited. This model will take our assembled feature vector and using the data provided, will find the best parameters by finding the minimum of the loss function θ using the gradient descent method, which is a "method that finds a minimum of a function by figuring out in which direction (in the space of the parameters θ) the function's slope is rising the most steeply, and moving in the opposite direction. The intuition is that if you are hiking in a canyon and trying to descend most quickly down to the river at the bottom, you might look around yourself 360 degrees, find the direction where the ground is sloping the steepest, and walk downhill in that direction". [15] Once this minimum is found, it will then be classified using the decision boundary, which is 0.5 for binary classification.

Step 5 involves validating our design. Basically, we run our finished algorithm on our training set and then evaluate the results. If the results are not to our liking, we can introduce control parameters to fine tune our algorithm. One method is using cross-validation, basically, this means that once we finish training the model on our training set, we introduce another set unknown to the model and judge its performance. This is useful in preventing problems like overfitting, which is a situation that arises when a model is too finely tuned to the point that it can only operate on the original training set. There is also the problem of selection bias, whereby it will only choose events (i.e., impedance values/gradients) that are similar to the original training set. Another method involved taking a subset of the original training set and optimizing it by adjusting its parameters, and then using that optimized set to retrain the model.

Step 6 is much like step 5. Once our model passes the validation tests on step 5, we will evaluate its accuracy using the actual set of impedances measured on our grid representation. In this evaluation test, we will be feeding our trained model with data sets from our actual grid, meaning it will be receiving a data set of real impedance values calculated from the meters in our grid. Using this real-world data, we can then quantitatively judge the accuracy of the model. Given that this is a binary classification model, we must provide a numerical cutoff between the two outcomes. That begs the question of what this cutoff must be. There is no quantifiable way of deciding this, and it falls on to the user to decide what percentage of accuracy is good enough. For our purposes, we will go with a greater than 97.457% fault identification rate. This percentage was chosen from [16] and was chosen due to the similarity in approaches. We believe such a high rate will be the key to proving that our solution is one that has merit and can be applied on a much larger scale.

Should the model not perform to our liking, we will have to return to step 5 and continue optimizations. We expect this to happen a fair number of times and is a natural step in developing a competent model. Special care however must be taken by us in not overfitting our model to our training set, this can cause a positive feedback loop which may look good to us as our accuracy would be improving, but due to the model becoming hyperspecialized on the data sets we fed it, its efficiency given a vast number of data sets will suffer. Overfitting can also create a black box problem whereby the model we've finely tuned becomes so complicated that it cannot be explained, modified, or enhanced.

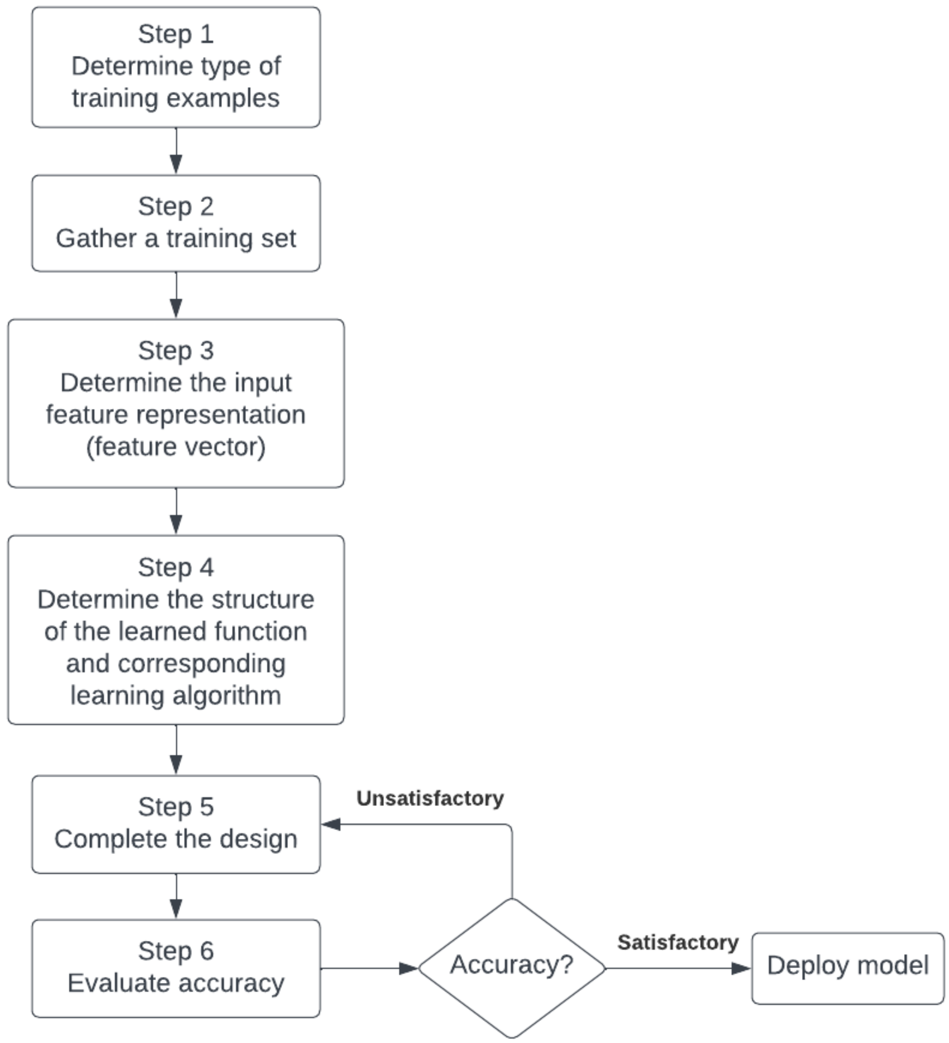


Figure 63: Flowchart of the machine learning training

6.7.4 Integration of Trained Model to the Microcontroller

After we've trained our model and it is performing at a level we deem accurate enough to roll out. Now, we will be discussing how we will integrate our trained model into our microcontroller. Given the limited computing resources available on the ATMega328p, special care must be taken to integrate our trained model as efficiently as possible. Given the simplicity and relative scope of our project, an ATMega328p proves to be capable of handling our relatively simple, but scalable machine learning model.

Integration of our model involved taking special care not to waste any of the ATMega328p's 2kB RAM. Instead, we will try to offload as many variables as possible to the flash memory, which contains a much roomier 32kB of memory. While at first this can seem daunting, these types of models do exist and are possible. TinyML, a popular foundation aimed at bringing deep learning models to microcontrollers states that on average, a TinyML model comes to be around 18kB in size [17]. This comes well short of our maximum flash memory of 32kB, which gives us headroom to run the other peripherals connected to the ATMega328p such as the current and voltage meters.

As stated, our implementation involves using as much of the flash memory as possible and saving only the onboard RAM to whenever truly needed. Given the possible size of our data set, we will be loading that into the flash instead of the RAM. Special care must also be taken when we initialize variables. As a reminder, a “byte” datatype uses 8-bits of data, while an “int” utilizes 16-bits. This can be applied to variables that we expect will not ever be more than 255. Otherwise, implementation of our machine learning model involves it being programmed using the same source files, or otherwise linked through a header file and called as a library function. However, we are specifically aiming to reduce the number of header files and libraries we need in order to have modular code with minimal memory overhead.

6.7.5 Flashing the ATMega328p

Flashing the ATMega328p is the final step in the design of our software. Flashing allows us to upload our written code to the ATMega328p and perform the computations we had programmed in our chosen IDE. AS discussed in section 6.7.1, we will be using a spare Arduino Uno as an ISP programmer for the ATMega328p. To flash, we will connect our ATMega328p to both Vcc and GND to activate the chip. Afterwards, we will connect our 16MHz clock source to the ATMega328p so that the Arduino Uno and the ATMega328p have synchronized clock cycles.

Since we have already flashed our bootloader as discussed in 6.7.1, the proceeding operations are fairly simple. We will then navigate to the Arduino IDE and select “Arduino as ISP” and then go to the sketch menu and clicking on “Upload Using Programmer”. After this procedure, we should have an ATMega328p that flashes with our own code. We will then verify that the flashed circuit is behaving as expected.

6.7.6 Fault Detection Algorithm

The fault detection algorithm for our project will be familiar to the famous Dijkstra’s algorithm from a high-level view. However, there are a couple of differences in how our approach will work out. We will be using the meters we have chosen to build out nodes on our grid, these nodes will serve to break up the overall grid into sections that can then be spatially identified on whether they contain a fault or not.

6.7.6.1 Data Structure

Our dataset will consist of the impedance values at these nodes. We will be calculating the impedances using $Z = R + jX$. To simulate a fault, we will be increasing the resistance of our fault site (which consists of a potentiometer) to its maximum of 1MΩ. This will increase the impedance seen by our meters. We will then be capturing a snapshot of our grid with its measured impedances at the various nodes. With this, we will have a completed dataset to execute our algorithm on.

Once the dataset has been created, we will create a data structure consisting of all the nodes with their associated impedances. However, the data structured as is serves us no purpose. Therefore, we will also be linking all our nodes with their physical neighbor notes. We believe the most suitable data structure for us to use would be an adjacency list. The adjacency list data structure is commonly used to represent circuits and lends itself well to our uses.

The nodes in our adjacency list will be a programmatic representation of the junction of our main feeder line and any lateral offshoots. This gives us an effective resolution of the fault status of all lateral offshoots on the feeder line. There will also be a node V_s which represent the AC voltage source and a node V_{Tf} which represents the Thevenin equivalent of the rest of the feeder line. Using V_{Tf} to represent the rest of the grid allows us to limit the scope of our project to a manageable, proof-of-concept level. Furthermore, each lateral offshoot will have its own, V_{TLn} node, which is also a representation of the Thevenin equivalent of the rest of the lateral line. n corresponds to each lateral offshoot on our feeder line and is used to provide a unique node to each lateral for the purposes of our algorithm.

Our adjacency list will consist of a unique singly linked list initialization for each node. Each vertex node will contain its name, how many edges are connected to it, and an array of pointers to structures containing the information for each edge. The edge array of pointers will contain the impedance value of that edge, along with a pointer to the next vertex node. Basically, we access a vertex node structure to view the start of that node, then access its edge structure which contains the impedance as well as a pointer to the vertex node at the end of it. This system allows us to construct a bidirectional circuit representation of our grid while having a desirable time complexity of O(1) for add operations as well as O(V) for querying operations, where V is the number of vertices on the graph.

Below is a high-level visualization of what our data structure will consist of, the illustrated circuit is used to demonstrate the concept and may not be our actual experimental circuit.

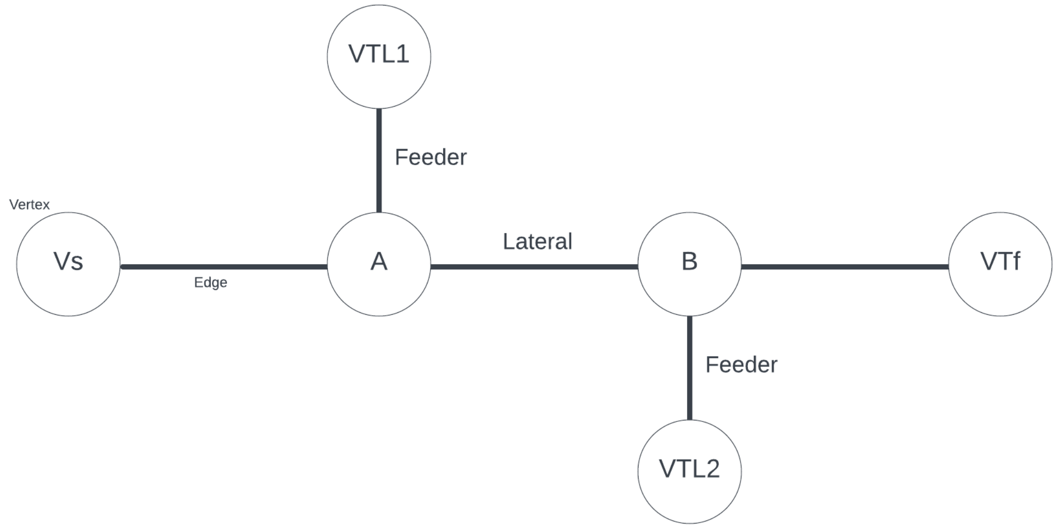


Figure 64: High level physical representation of data structure

Using this data structure design allows us to build a picture of our grid using code, and we believe that this method will allow us to tell the use where in the physical space is the fault occurring. We mainly chose this data structure too because it allows us to assign weights to nodes, with the weight being the impedance seen by each node. These associated weights will be the critical piece into our algorithm will work, and we believe it to be a scalable solution. It also allows us to apply pattern recognition, which can then be used to detect whether a fault is a true fault, or faulty equipment that is producing incorrect data due to damage. So as can be seen, it serves a secondary purpose in being able to detect faulty equipment.

6.7.6.2 Analyzing the Dataset

As one can imagine, given a big enough grid size, the data that is being created would get massive over time. Considering Dijkstra's algorithm has a spatial runtime of O(NlogX), where N is the number of edges and X is the number of nodes, our runtime would be scaling at a log-linear rate, which is not terrible but also not desirable. So, how do we reduce our dataset to only include nodes of interest is the main question. Our solution is a multi-criteria solution that we believe will work well.

Given our assembled data structure, our algorithm will traverse the data structure with the intent of finding the edge with the highest impedance. For a given node, it will traverse through all the edges connected to that node and check which edge contains the highest impedance value. Accuracy is important here as we will be using this impedance differential to "see" our way towards the fault. Whichever edge for the given node has the highest impedance will then be accessed and its corresponding node pointer (which points to the other connecting node) will be made the new current node.

Eventually, the algorithm will make its way to the edge that has the highest impedance. The algorithm is considered to have finished when, for a given node all of its connected edges have an impedance lower than that of the previous edge. This physically signifies that the fault must be located on the edge between the current node and the previous node. As a reminder, a node signifies the physical junction between the feeder and the lateral line.

6.7.6.3 Verifying the Fault with Pattern Matching

Once our algorithm arrives at what is believed to be the potential fault, we must verify that it is indeed the actual fault and not an anomaly caused by faulty equipment returning wrong values. For this, we believe the best application is a pattern matching machine learning model. Our approach involves establishing a “gradient” of sorts, starting from the identified fault and expanding out through its neighboring edges.

Our method of discerning between an actual fault or faulty equipment uses this gradient method. Conceptually, an actual fault on the line will cause a gradient of gradually increasing impedance values up until the location of the fault, any measurements taken past this point will result in gradually decreasing impedance values. Likewise, faulty metering equipment will create an unnaturally high impedance reading between two nodes with the subsequent node returning the expected impedance measurement. One can visualize this as follows, an actual impedance measurement results in a hill-like shape, with slowly increasing impedance values, likewise, faulty equipment will create a cliff-like shape, with the anomalous value sticking out.

Now of course, a human can easily spot these patterns, but the goal is this project is to provide a scalable, autonomous way to detect these faults. This is where pattern matching comes in. We will be utilizing a machine learning model and training the model to discern if the edge the algorithm determined to be a fault is one or just faulty metering equipment. The basis of its operation will revolve around the hill vs. cliff concept described in the last paragraph. Essentially, the data set provided to train the model will contain actual fault “hills”, we will then apply this trained model to our program, where it will be able to discern if the chosen fault has a hill shape, or a cliff shape. It will then assign a confidence level to its decision; this confidence level can be relayed to the end user where they can make the final decision on whether it is an actual fault or just faulty metering equipment. Figure 41 shows a high-level visualization of what a hill vs. cliff scenario would look like on our physical grid.

Now that we have defined the hill vs. cliff scenario, we need to find a way to quantify what is a hill or what is a cliff. After all, we as humans can spot this but a computer needs a mathematical method of determining whether it is a hill or whether it is a cliff. Thankfully, there is one statistical method well suited for this. We will be using standard deviations to determine whether something is a hill or a cliff. To determine this, we will begin by reading the measurement of the edge that the algorithm believes to be the fault. We will then fan out to its connected vertices, and then read the value of the edges of those vertices. This will give us a small dataset of impedance values which we can then use to calculate the standard deviation of these values.

A problem with this approach however is, how do we determine what is too high of a standard deviation. There is no mathematical approach to this, and it all depends on the type of data you are looking at. We believe a good approach to determining what is a hill or a cliff is by comparing the standard deviation with the mean. If the calculated standard deviation s is greater than the calculated mean $\bar{x}$ then we will consider that an anomalous reading. Likewise, a standard deviation s less than the calculated mean $\bar{x}$ signifies that there is a potential fault at the chosen edge. Below is both a visual representation along with a mathematical explanation of our calculations.

$$Given\ edge\ values: 1M\ Ohm, 1k\ Ohm, 3k\ Ohm$$
$$Where\ n = 3$$
$$where\ mean\ \overline{x} = \frac{1E^6 + 1E^3 + 3E^3}{3} = 334{,}666.67$$
$$Standard\ deviation,\ s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}}$$
$$s = \sqrt{\frac{SS}{n-1}}$$
$$Calculating\ sum\ of\ squares,\ s = \sqrt{\frac{664004670000}{3-1}}$$
$$s = \sqrt{\frac{664004670000}{2}}$$
$$s = \sqrt{332002340000}$$
$$Solving\ for\ s,\quad s = 576196.44$$
$$s > \overline{x},$$
$$therefore\ the\ chosen\ fault\ has\ a\ high\ probability\ of\ being\ faulty\ metering\ equipment.$$

$$Likewise, Given\ edge\ values: 5k\ Ohm, 3k\ Ohm, 4k\ Ohm, 3k\ Ohm$$
$$Where\ n = 4$$
$$Solving\ for\ s,\ s = 957.42711$$
$$Solving\ for\ \overline{x}, \overline{x} = \frac{3E^3+3E^3+4E^3+5E^3}{4} = 3{,}750$$
$$s < \overline{x},$$
$$therefore\ the\ chosen\ fault\ has\ a\ high\ probability\ of\ being\ an\ actual\ fault\ on\ the\ line.$$

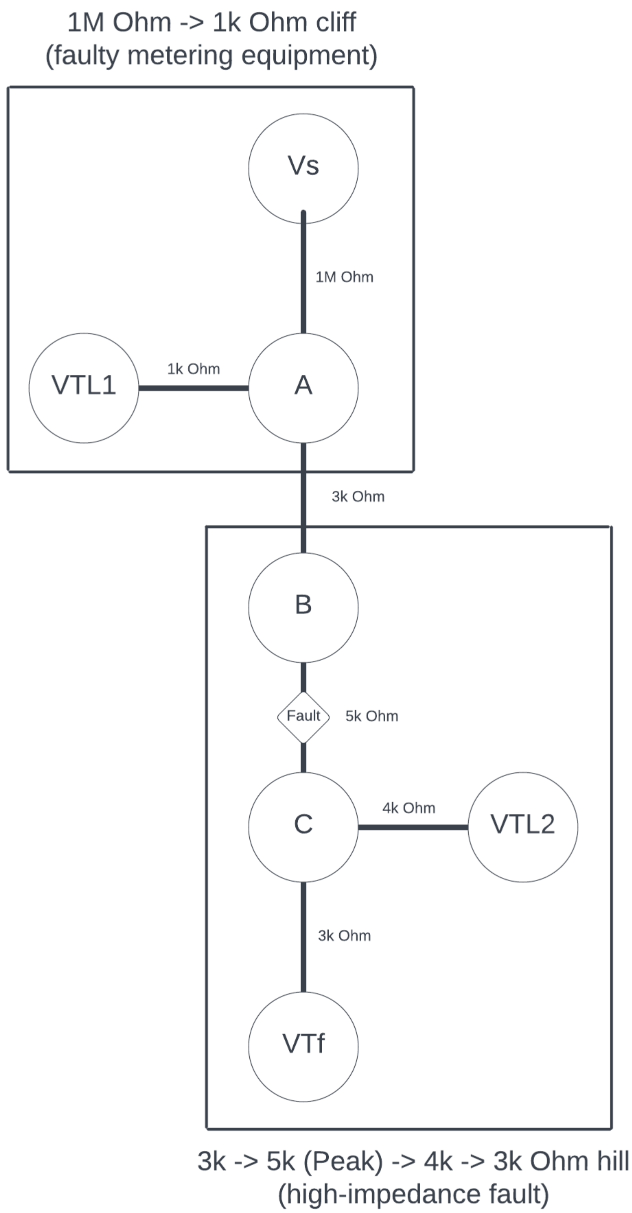


Figure 65: Visualization of pattern matching events.

We believe a combination of a pattern matching machine learning algorithm along with verification through statistical methods will allow us to make a decision with high confidence on whether a fault on the line is legitimate or just a faulty metering equipment.

6.7.7 Triangulation

For fault location triangulation, we plan to use the method "A fault locator for radial sub-transmission and distribution lines" published in the 2000 Power Engineering Society Summer Meeting. This method presents a fault locator that estimates the location of shunt faults on radial sub-transmission and distribution lines. These techniques rely on the fundamental frequency voltage and the measured current, which we will mimic in our physical system by using multimeters and current clips. These measurements are taken for calculating the impedance values at line terminals before and during the fault.

This method works for both ground fault and phase-to-phase fault, measuring and analyzing the high impedance values. We will be collecting data in the fault-loop impedance and feeder impedance. Although we will only be focusing on ground fault for the scope of this project, we are keeping scalability in mind while we choose to implement this algorithm.

The before-fault measured impedance should be consistent with the calculated impedance at the feeder, if there is a difference, that should be a good indicator of the location of the fault along the line. This technique works for the radical line only because if it is in a loop, the direction of the power flow could change, which will impact the measurements the mathematical model is based on.

The "Development of a New Type Fault Locator Using the One-Terminal Voltage and Current Data" method was referenced which is a more direct version of the method we are implementing. This one applies to single-phase ground faults with no lateral system only. It utilizes one-terminal voltage and current data of the transmission line to calculate the distance of the fault from the bus with a microprocessor. The distance of the fault can be calculated by taking a fraction of the voltage and current with the imaginary and conjugate components, with voltage being represented by impedance times current. This method has a simple mathematical model.

$$x = \frac{Im(Vs \cdot (Is\, pre\, fault - Is\, post\, fault)^{*})}{Im(ZIs \cdot (Is\, pre\, fault - Is\, post\, fault)^{*})}$$

This reference method has proven to be successful, and the method we will be implementing is a more advanced version that can handle laterals. Although we are expecting a more difficult model mathematical involving a matrix due to the higher amount of measurement points neighboring each other. In our implementation, we will section the simulation and the physical system off with a certain conversion ratio, to help keep track of the location of the fault in a distance value and compute the accuracy of our model.

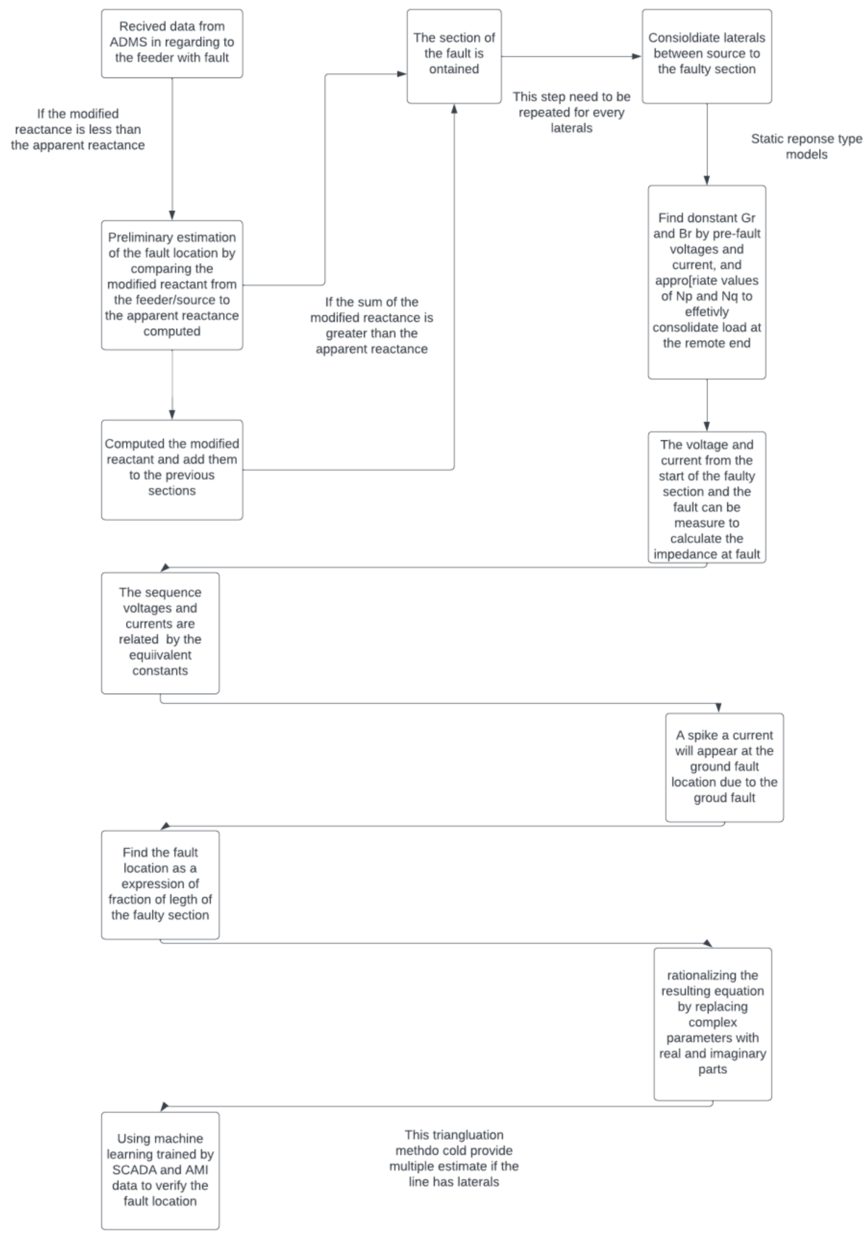


Figure 66: Triangulation Flowchart

Because we will work with a system with 2 laterals, we need to expand on this method to compensate for its shortcoming by providing multiple estimates to help narrow down the fault location. This method utilizes pre-fault and during-fault fundamental frequency voltage and current measurements at the feeder/sources only. We will have at least three measurement points to collect voltage and current data to ensure the neighboring data points are also supporting the fault location estimation. This is a reasonable assumption because there are a lot of data acquisition devices in our current grid. We will also be using a machine learning module trained by SCADA and AMI data to verify the fault location. It should give us a confidence rating.

In the purposed paper, they had done an extensive sensitivity test. Which perform tests by systematically varying parameters such as fault impedance and, fault type, fault location, and fault resistance. Our algorithm is designed to be able to perform fault location detection with varying fault impedance and location.

7 Project Prototype Construction and Coding

For our project's prototype construction, we will be using a breadboard to test each component that will be on our PCB. We will be designing it to be as close to our printed PCB product. On the software side, we will be using our spare Arduino as an ISP and as a USB-to-TTL converter. Since we also have burned a bootloader onto the ATMega328p, we will only be using the TxD and RxD ports on our Arduino ISP to program our ATMega328p.

Since our code relies on external physical inputs, we will be prototyping our code by programming it onto the ATMega328p and then running through our fault detection routine as normal. We will consider our prototype ready for testing and debugging once the outputs are as expected.

7.1 Breadboard testing

Our Breadboard testing will be divided into two parts. First, we need to make sure that all our measurement circuits work. We will do these by using an Arduino UNO and connecting everything on a breadboard. For this test, we will only need to order the multiplexer and the current sensors since all the other passive components are going to be available in the Senior Design Lab. Once we test that we are effectively receiving data from our sensors, we will start the second part of the breadboard testing.

For the second stage, we will have our power grid model built, and we are going to start doing trials to observe if we can detect the fault and check if we can find its location.

Dividing our testing into two stages is better because we are going to be able to break our overall testing into small more manageable phases. By doing this, the number of potential errors is reduced, so the process of debugging the problem becomes easier.

7.2 PCB Vendor and Assembly

After testing our system on a breadboard, we will check if everything worked the way we wanted. We will have a great idea if our system is functioning correctly. If it doesn't work, we still have the chance to find the problem and fix it. If it is working fine, then we will start ordering our PCB. For that purpose, we must choose a vendor that satisfies our three requirements. First, they must do fast shipping and delivery. We have a limited time available to finish this project and to be able to present it, therefore we can't afford to wait too long for products to arrive. Also, it must have a good quality-price ratio since we have a restricted budget. Although it can't be too cheap either due to the importance of this project and the fact that we want to maintain a high-quality standard of product.

After reviewing some of the most renowned companies in the World, we decided to do a closer analysis of PCBWay, Camptech, and JLCPCB. They all have similar benefits, but we decided to go with JLCPCB since it is the option that most closely fulfills our criteria.

JLCPCB is the largest PCB prototype manufacturer in China. It is very well-known due to its enormous influence around the World. They tend to keep their prices down, but that doesn't harm the quality since it is one of the most reliable. A disadvantage of JLCPCB is its shipping cost. As it is not a United States company, there are usually high fees to ship the products.

Camptech is a company based on Canada. They offer excellent quality products and fast delivery with minimal shipping fees. Out of all three, Camptech has the quickest delivery. Their products are usually delivered in around two to ten days. The main downside is its prices.

The last option we considered was PCBWay. It is a widely popular company worldwide. They tend to have great quality standards for their products, and additionally, they offer a free engineering file review to minimize errors on the final PCB design. In terms of price and delivery time it is very similar to JLCPCB because both are based outside the United States.

In summary, these three companies offer very similar product qualities but each one has its own advantages and disadvantages. In our case, we will be working with JCLPCB because we think that it is the one that satisfies our price-quality ratio the most. And although it is not the fastest one, it usually delivers its products within a week, which is acceptable for us.

7.3 Final Coding Plan

Section 6.7 contained a lot of our more technical data, where we broke down each of the steps our program is going to take in detecting a fault. In this section, we would like to summarize what our final coding plan is going to be. It is a multi-faceted approach implementing various topics from computer science such as statistical analysis, machine learning, algorithms, embedded systems, and electrical networking. As such, while our implementation is on a very small scale, there are various complexities that are being resolved in order to create an efficient, scalable solution to the problem posed to us by our sponsors.

We begin by assembling the low-level programming of our system. That is creating and flashing the bootloader onto the ATMega328p, which comes ‘raw’ from the factory, meaning there is no way of communicating with it, at least not for any usable purpose. To begin communicating with the ATMega328p, we will need to burn a bootloader onto it. Once our bootloader is installed and the ATMega328p boots as expected, we can run a simple program such as a blinking LED to confirm if it is indeed functional. Next, we will begin setting up our communication protocols to the peripherals attached on the ATMega382p, peripherals such as the multiplexers and the voltage and current meters. Our communication protocol of choice for the multiplexers will be a serial UART interface and a SPI interface between the meters and the microcontroller.

After, we will move on to higher-level programming languages. On this higher-level language, we will be programming all our data structures, algorithms, and machine learning model. We believe our first step should be in creating the framework of our data structure, that is the graph adjacency list we discussed in 6.7.6.1. We will then verify the data structure by creating an example graph of our circuit and seeing if it matches to how our physical circuit looks like. Once our data structure is in place, we can then begin analyzing the dataset using the methods presented in 6.7.6.2, once again verifying that these methods are predicting the right impedance values.

Our final step involves verifying that these chosen faults are indeed the correct faults. And again, as stated on 6.7.6.3, we will be applying two verification methods, with one being statistical in nature and the other one being a machine learning model for binary classification. We believe we will be spending most of our development efforts in this part of the process. Once, our verification methods are at a point where we are content with their quality. We will begin the final verification and polishing of our project. Afterwards, we will release our system as a finished product to our sponsor for critique.

8 Project Prototype Testing Plan

Our testing environment for the project will involve creating a small-scale grid. Using this small-scale grid, a single-phase ground fault will be simulated in the system using a potentiometer. This fault will short the potentiometer, creating an open circuit for the rest of the grid. This will draw current to the site of the fault, at which point our pulse injections will be triggered by the MCU. These pulse injections will return an impedance reading, at which point our software will preprocess the data, assign the injection sites into a cluster based on which sites returned the highest impedance values, and from there our software will use these highest impedance values to triangulate the location of the fault. At this point, the user will be alerted with the determined location so that repairs to the system can be made.

8.1 Hardware Test Environment

Several components are required for the hardware test environment to be properly implemented. These components are the PCB, the small-scale grid, a potentiometer, the site injections, and current sensors.

8.1.1 Small-Scale Grid

Since our design focuses on fault detection and location identification in a distribution system, it is important for the testing environment to resemble a distribution grid as much as possible. The grid will feature a main feeder system as well as laterals in a single-phase configuration. The goal of the test will be to add voltage to junctions within the system and use the resulting current to calculate the impedance of the line. Using these impedance values the software part of our test will be able to determine the location of the fault along our grid. The inclusion of the lateral system makes the grid a more realistic representation of a real-life distribution system, as lateral offshoots are the main way power is supplied to the consumer. Although fault detection and location identification in a feeder system is important, having the ability to also detect the location of faults in a lateral vastly increases the versatility of the design, while also helping consumers. Since laterals exist mainly for the purpose of bringing power to customers who don't live directly next to the feeder system, whenever a lateral goes out of power a large number of customers can be immediately affected. Although a fuse or Tripsaver will protect the main feeder system from the permanent fault, having the ability to precisely locate the fault before linemen arrive for repairs will vastly reduce the time needed to restore power.

One key concept related to the grid test environment will be the concept of scaling distance. Since feeder and lateral systems can span several miles, it is important to have distance factor brought into the testing environment so that we can verify the accuracy of the determined location. Each inch of cable on the grid will be assigned a scaled distance, something like each inch of cable representing 10 feet in a distribution grid. From here, the precision of our location estimate can be directly tested. If our estimate is off by three inches, then that would be representative of being off by 30 feet in a real system. It is important to assign some sort of representative distance so that we can truly understand the margin of error that the design has.

Using Thevenin's Theorem, we'll be able to determine the impedance differences between different sections of the feeder and lateral systems. These impedance differences will be vital in helping to determine the location of the fault. Comparing the current and impedance values of different sections of the feeder and laterals after a fault has occurred will be an important part in how the fault location is determined. Seeing that there is a huge impedance and current increase in a certain section will let us know that there may be a fault in that section of distribution line. The current will flow towards the ground fault as though it is an open circuit, which means that our main goal is narrowing down certain sections of distribution line using mathematical models and impedance values until we have a precise idea of where the fault lies. From here, the user will then be alerted of the determined fault location, and they can send operators to the location to begin repairs.

8.1.2 Creating a Ground Fault

The goal of this design is to come up with a way to determine the location of a fault in the distribution system, thus, for our testing we have to create a fault. The fault that will be created will be a single-phase ground fault in a small-scale grid. This fault must be created in a way that the other sections of the grid will see this fault as an open circuit, meaning current will want to move towards the fault location.

To create a fault in our system, a certain section of our grid will have a potentiometer attached. By lowering the resistance in the potentiometer, the potentiometer will be shorted, creating an open circuit for the rest of the grid. The current will then want to flow to the path of least resistance and will rush towards the open part of the circuit. Using this current flow, we can gather an idea on where the fault will be located.

8.2 Hardware Specific Testing

Now that the general hardware test environment has been discussed, it is important to discuss the roles of specific components in the hardware phase and how their designs have been catered to a positive testing environment.

8.2.1 PCB in the Test Environment

The PCB will be the backbone of the physical testing structure for our design. There are currently several ideas for the exact testing process, one idea is using site injections to measure impedance values, and the other is using the PCB to compute impedance values using Thevenin's theorem.

8.2.1.1 PCB with Site Injections

Our PCB is a vital piece in the testing process. It will be responsible for the site injections as well as reading the data that the injections return. To power the PCB, we will have a 120V/9V AC to DC converter plugged into a wall outlet and the microcontroller. This converter will allow the DC microcontroller to use the AC power from the wall outlet as a power source.

The role of the PCB in testing will be to send pulses into the system at certain site injection points. The pulses sent into the grid will be 5V square waves sent in from a signal generator. The goal of these pulses will be to receive an impedance value which will then be read by an MCU attached to the PCB. At this point the testing transitions to a software level. Though, to receive these impedance readings, more equipment is required than just the injection sites. Connected to the PCB will also be current meters that will be used to read the current caused by these pulses. These current meters will be connected to the PCB using a multiplexer that will allow for multiple current meters to attach to one pin on the PCB. This way, the device is able to read from multiple locations in an efficient way. Without the multiplexer, only one current meter could be used at a time and would have to be moved from place to place to get readings from the pulses. This would create an incredibly inefficient and time-consuming system. The MCU attached to the PCB will then take the readings from the current sensors, calculate the impedance based on those values, and continue to work at the software level to triangulate the location of the fault. In addition to the current meters being attached with a multiplexer, the injection points will also use a multiplexer to create a multi-point switch. This way, the user can determine which node they want to create the pulse at. Having the ability to send a pulse to multiple nodes in quick succession gives the ability to implement this design on a higher scale. With a larger system than our current PCB or with more injection sites, having the ability to target multiple sites quickly becomes paramount to the success and efficiency of the design.

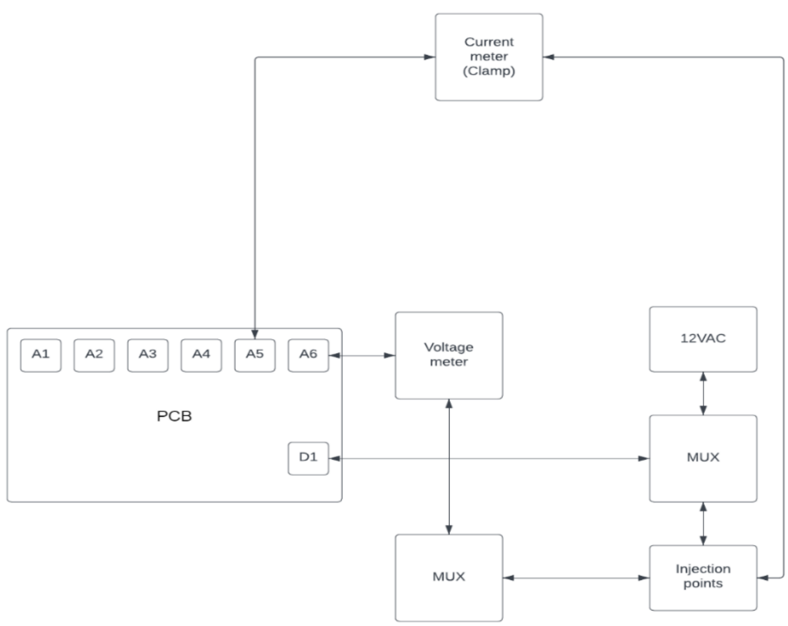


Figure 67: 67 A Conceptual Schematic of the PCB in the Site Injection Testing Process

8.2.1.2 PCB without Site Injections

If the site injection idea is not implemented, the PCB has less work to do on the hardware side. Attached to the PCB will be a multiplexer and a voltmeter. The multiplexer will be connected to multiple current sensors. This configuration allows us to switch which current sensor is read at will. Being able to use multiple current sensors is vital to the design because it allows us to take multiple readings in quick succession. The ability to check the current at a location and get the impedance value from that measurement and repeat that across our grid quickly will mean that our design is able to work at an efficient rate. Using a multiplexer would mean that our PCB could only read one current sensor measurement at once, but it would be able to swap quickly between locations to take multiple readings. The time it takes to switch input pins to a new current sensor is quick enough that it does not have a noticeable impact on the performance of the PCB. The other benefit of using a multiplexer is that although the multiplexer can take multiple inputs from multiple current sensors, it only takes up one pin on the PCB.

8.3 Software Test Environment

The software testing environment has several important components that will be discussed in this section. The software portion of this lab begins its main implementation following the measurement of impedances from the current sensors attached to the small-scale grid.

The main environment that the software testing will take place in will be using the C++ coding language. The reason for using C++ is that it is compatible with the hardware components of the testing, meaning that it will be easier to code our hardware interactions using a C++ system. Another benefit of using C++ is the ability to code in C within a C++ program. Having the ability to code in both languages gives us a bit more versatility and flexibility with how we approach the software testing for our design. The main usage of our coding portion of the test will be to process the data received from the hardware, sorting the data into useful clusters, and then calculating the location of the fault.

8.4 Software Specific Testing

The goal of our software is to efficiently process the readings obtained from the physical components within our test environment and calculate the fault location by using the readings from the physical components in combination with mathematical models and algorithms. This intricate task involves several steps that are crucial for the successful operation of our system, and in this section of the senior design report, we will outline the entire process.

The software section of our testing procedure initiates once the impedances have been accurately calculated. The impedance calculation can only take place after we receive the current values from the current sensing hardware, which will be securely attached to both the grid and PCB. These current values act as a foundation for further analysis and are integral to the accuracy of our results. There are several testing procedures we have in place, with some being visual testing while others have a more quantitative approach.

For our data structure, our testing approach will involve first populating it with predefined impedance values and then visualizing said data structure. If it matches our physical structure in terms of vertices, connected edges along with their weights (impedance) then we deem the data structure as properly implemented. For our dataset analyzation, this also involves a visual approach. We will have our analyzation algorithm run and return to us through the console the highest impedance it has found. We will then compare this given impedance to what we know is the highest impedance. If the impedance values match, we will deem the algorithm as properly functioning. We may also test against edge cases, for example, negative impedances due to overflow, or equal impedances. However due to the double-precision floating point nature of our variables, we do not foresee this ever occurring.

Our verification algorithms require a more quantitative approach, as one involves statistical analysis and the other is an implementation of a machine learning model. To verify our statistical model, we believe the simplest and most elegant solution is to just implement the same dataset into a calculator capable of calculating the standard deviation of a sample. If both the output from the verification algorithm and the calculator match, then we will deem the algorithm as functional. For the machine learning model, we will be employing a commonly used method of splitting our sets into three sets, a training set, a testing set, and a validation set. For example, we will only be using our training set during the training of our machine learning model. Afterwards, to test and optimize our machine learning model, we will utilize a separate testing set. Finally, to validate that our model is performing as expected, we will have a separate validation set that will be fed to the model.

Separating sets like this prevents common problems like over fixing when developing a machine learning model. Exposing it to a new set at each step allows it to not be hyper-tuned to one set and then fail catastrophically when exposed to a different set.

8.4.1 Data Preprocessing

Once the impedance values have been calculated, the data enters a phase of preprocessing so that it can be organized into a more useful format. During this preprocessing, the raw data values will be read and converted into a graph data format, this will allow for easier sorting later in the test. This preprocessing will be done with the use of algorithms to create a data structure that links the physical location of all of our nodes with their physical neighbor notes. From here, the data will be converted into an adjacency list.

Adjacency lists are already commonplace as a representation of circuits and will be a useful representation of our small-scale grid. Arranging the system into a set of nodes that represent a main feeder line with lateral offshoots, the adjacency list will have a linked list initialization for each node. The adjacency list will keep track of the name of the node, how many edges it is connected to, and the information for each edge. This is key to recreating the grid structure in a code format. Recreating the grid structure in code format makes understanding which part of the grid the fault lies within a much easier process, because the code will have drawn an image that looks similar to the grid that is existing in the field. This makes the information more digestible for the operators that receive the fault location, since it will already be based on the existing grid configuration.

Another important reason for using the adjacency list technique is that it has a desirable time complexity. Time complexity is the time required for an algorithm to run based on the length of the input. Having a bad time complexity means that the more inputs that are added, the time to operate the algorithm increases in a way that may make operating this program on a higher scale a complete impossibility. If the time to execute is increasing exponentially with the number of inputs, then our design will not work fast enough to be beneficial to the crew compared to existing methods of finding fault locations. Thus, it is important that we have a favorable time complexity. Using the adjacency list gives us a favorable time complexity. Having a favorable time complexity means that the time to execute is fast but also means that this project will be scalable at a high level. This is because as the number of inputs increases, it doesn't change the time to execute very much. This concept of time complexity is shown in the graph below. Having a good Big-O complexity will be vital for any algorithms used in the project so that we can keep a scalable design.

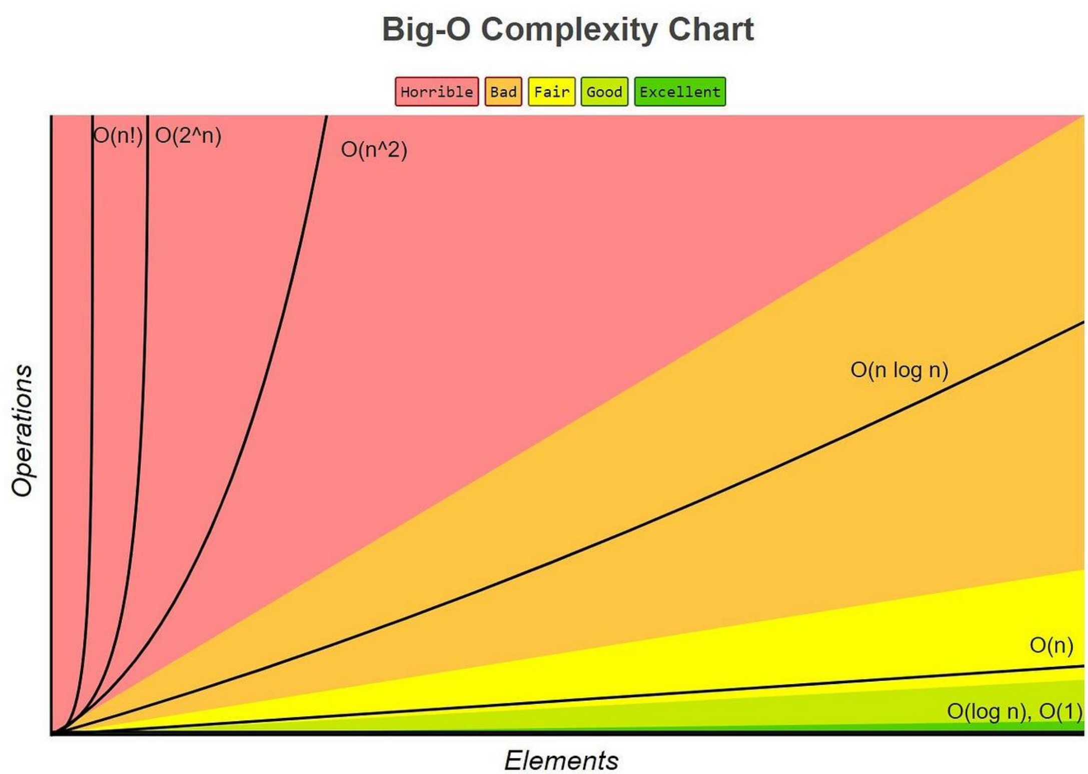


Figure 68:68 Big-O Complexity Chart.

Without the preprocessing stage, the data would remain significantly disordered and cumbersome to work with, hindering the efficiency of our software. By converting it into a graph data format via an adjacency list, we ensure that the subsequent stages of the software testing will be more seamless and productive.

8.4.2 Cluster Identification & Pattern Recognition

Following the preprocessing sequence, our data has now been put into a more usable format via the adjacency lists and from there the more technical side of our software test begins. A key part of this technical software test will be creating a data cluster using a grouping of impedance values obtained from different sections of the small-scale grid. These clusters will tend to be from junctions that neighbor each other, since the impedance values will be higher the closer the junction is to the original fault location. It is important that our program is accurate in the grouping of nodes to not confuse the grid locations. The goal of this cluster identification will be to find the node with the highest impedance and based on the impedance of nearby nodes determine the location of the fault. The two nodes with the highest impedances, if adjacent, will tend to be on either side of the fault depending on the grid layout. Though the reason that we are using a clustering technique is to reduce the number of locations that experience Dijkstra's algorithm. This is because it has a runtime of O(NlogX), which is not the most ideal time complexity for a large input. Therefore, we want to minimize the amount of data that is used by Dijkstra's algorithm, proving the importance of the data clustering process.

By combining the data preprocessing phase and data clustering techniques, we have optimized the overall software testing process. The generation of data clusters will both help in our ability to determine the location of a fault and make sure that those calculations are carried out as efficiently as possible. This approach will also aid in the ability to work with large-scale inputs and environments. As a result, our software will be equipped to meet the demands of real-world scenarios and systems, where quick and efficient fault identification is of utmost importance.

8.4.3 SCADA and AMI Fault Verification

An important final component of the project timeline will be verifying the potential fault locations that are determined by the cluster identification and pattern recognition software. It is incredibly important to verify the information after it has been provided because without proper verification, we cannot truly be sure that the locations generated are precise to the fault location. The way that we will be verifying these locations is by using AMI and SCADA data to train a machine learning system. AMI and SCADA are two distribution resources used for many different purposes, but in this case raw data from both systems will be used to train our machine learning. To first understand how these two programs can train our machine learning system it is important to understand what these two programs are.

8.4.3.1 SCADA

SCADA is an acronym that stands for Supervisory Control and Data Acquisition, and it is a type of control system that collects real-time data and analyzes it for the purpose of equipment monitoring. The type of data that SCADA could collect would be information given by devices in the field. This type of information could be current and voltage values, temperature data, wind and vibrational data, and many other types of data. This data is then sent to a high-level central facility, generally an operating center for the owner of the field devices. An important thing to note with SCADA is that SCADA is capable of creating alarms when systems reach a certain threshold. An example of this that would be relevant to our project is SCADA creating an alarm whenever a distribution line is experiencing overcurrent issues. Some other potential examples of useful SCADA information and alarms would be monitoring the power factor of the distribution line and monitoring voltage levels and creating an alarm whenever going above or below a certain threshold.

These SCADA data and their alarms could prove to be very beneficial when training a machine learning system. Taking a large amount of SCADA data and plugging it into a machine learning software could result in a computer system that understands when a fault may occur by understanding the conditions that may lead up to a fault. Having the real-time system data as well as having the location where the fault that caused an alarm may create a system where knowing both of those pieces of information and giving them to a machine learning system may create a system that is able to better understand fault distances and location estimates using a plethora of previous incidents as a learning experience.

To conclude, SCADA is a powerful and useful tool in real-time data collection and analysis, providing important insight and alarms that will ensure the functioning of complex systems like a distribution system or power generation system. Combining machine learning systems, the plethora of SCADA data, as well as alarm information and fault locations creates an opportunity for a high-potential fault prediction and localization system. Using machine learning and SCADA data, we will use that machine learning system to test our generated fault locations to verify their precision and accuracy.

8.4.3.2 AMI

AMI is an acronym that stands for Advanced Metering Infrastructure. This is a communication system that collects information from meter points in the field. This could be a meter connected to a house, business, or even a piece of equipment in the field that is powered by the distribution grid. The type of information collected by AMI includes power consumption, load profile, voltage sag and swell events, outage logs, outage counts, and power factor at the point of the meter. Some important pieces of information that will be most relevant to our project will be outage logs and voltage information. Using a combination of the location of the AMI instrument and the data it collects, we can train a machine learning system to better identify fault information and location determination. Although AMI systems are mainly used as a verification that the outage exists and when the outage is restored, keeping track of previous location data and collected outage information could help as an input in a machine learning system. Comparing the information gathered from our machine learning software with the data tracked by AMI may allow us to check our mathematical models to ensure that we are properly determining our fault location estimates.

In conclusion, AMI is an incredibly useful existing infrastructure and communication system and we believe that its plethora of past data and utility information will provide heaps of value in training a machine learning system to identify and verify fault locations.

9 Administrative Content

During the creation of our project, it is important to understand the milestones for good progress as well as having a firm understanding of the cost of materials to make the design into a reality. In this section there will be a discussion of the milestone timeline that our group estimates to happen during Senior Design I and II as well as a drafted budget which will estimate the cost of parts to create the hardware section of our testing.

9.1 Milestone Discussion

To be able to accomplish our goals, we set a few milestones so we can keep good track of the project development. This way we can also ensure that we will meet the final deadlines for the report and presentation. Also, the chart shows the responsibilities of each member of the group so everyone can get a fair share of work and contribution.

Table 12: Project Milestones

Week	Date	Task	Status	Responsible
Planning				
1	5/29	Brainstorm	Completed	Group
		Choose Project	Completed	Group
	6/02	Divide & Conquer paper	Completed	Group
2	6/05	Meeting with advisor	Complete	Group
Research				
3	6/12	Research Dyno Net ML algorithm	Completed	Michael
		Determine I/O model for algorithm	Completed	Group
		Understand ADMS and AMI	Completed	Group
4	6/19	Impedance injection/detection	Completed	Group
		PCB System Requirements	Completed	Michael/Julio

		MCU requirements for ML model	Completed	Michael
5	6/26	Import real world data from ADMS and AMI	Completed	Yuejun
	6/30	60-page draft documentation	Completed	Group
Design & Order				
6	7/03	Build schematic	Completed	Michael / Julio
		BoM	Completed	David
7	7/10	Order parts	On Track	Group
8	7/17	Write ML model	On Track	Michael
		Connect I/O model to our ML model	On Track	Group
9	7/24	Document review	Completed	Group
	7/25	Final 120-page document	Completed	Group
Prototyping & testing				
10	7/31	Test components & code	On Track	Group
11	8/07			
12	8/14			
13	8/21	Breadboard testing	On Track	Group
14	8/28			
15	9/04			
16	9/11	Design & Order PCB	On Track	Group
17	9/18	Flash & Train ML model	On Track	Group
18	9/25	Debug/Refine model	On Track	Group
19	10/02			
20	10/09	Build grid representation	On Track	Group

21	10/16	Testing & Redesign	On Track	Group
22	10/23			
23	10/30			
Final Details				
24	11/06	Finalize prototype	On Track	Group
25	11/13			
26	11/20	Peer Presentation	TBA	Group
27	11/27	Final Report	TBA	Group
28	12/04	Final Presentation	TBA	Group

9.2 Budget and Finance Discussion

The budget for this project will be divided into parts for assembling the printed circuit board (PCB) and any other components needed to get the physical simulation running. Some items needed for the PCB will be passive components like resistors and capacitors, a timer and clock system, a power module, OP-AMPs for the power source schematic, and a stepdown transformer so that the system can be powered through a wall outlet. To reduce costs, a timer that can also function as a clock is likely to be used in the PCB design.

This project will be funded through assistance from UCF and the design team. Also, some of the equipment that will be used in the hardware simulation will be provided by the UCF lab and will not need to be purchased beforehand. Using equipment from the lab reduces the overall cost of the project as long as the equipment is properly taken care of. Examples of this provided equipment will be passive components like resistors and capacitors as well as the cable that will be used to simulate our feeder and lateral systems.

Table 13: Prototype Budget

Item Description	Quantity	Total Estimated Cost
ATMega328P	1	$3-5
Printed Circuit Board (PCB)	1	$50-300
Passive Components	20-30	Provided by Lab
Cable for Simulated Grid	50	Provided by Lab

Item Description	Quantity	Total Estimated Cost
16MHz Crystal Clock	1	$5-10
Power module for Printed Circuit Board	1	$10-15
Potentiometers for Short Circuit Simulation	3-5	$36-60
120V/12V Step Down Transformer	1	$10-35
4-pin Header	1	$2
RDT1034 Switch	1	$5-7
Jack Barrel Plug	1	$1-3
5.1V Zener Diode	1	$1-3
Terminal Connectors	24	$15-20
CD4051BM	2	$2-4
LMZM23600S LR	1	$5-6
	Total:	~$100-470

9.3 Other resources

In this section we want to point out the resources we planned or used in this project but were graciously acquired at no cost, therefore not a part of our budget. These resources compress three main categories, hardware, software, and data. Please note, we did not include any resources that's already open sources and free for the public on this list. And we will only be using these resources for this project unless it is accessible for all UCF students. We would like to express our gratitude as we understand that the realization of this project would have been unattainable without their provision.

9.3.1 Hardware

Below we will list all hardware equipment we have used or will use. We are also naming the person that facilitated us the access to this equipment.

Table 14 Hardware equipment provided

Item	Sponsor
SEL-551 Overcurrent/Reclosing Relay from Schweitzer Engineering Laboratory	Dr. Aleksandar Dimitrovski Kwasi Opoku
Chroma 61704 Programmable AC Power Source	Dr. Qu Zhihua
California Instruments MX22.5 amplifier	Dr. Qu Zhihua
Tektronix AFG3022C Arbitrary/Function Generator	UCF senior design lab
Fuse adapter for Tektronix AFG3022C Arbitrary/Function Generator	David Douglas

9.3.2 Software

We were also able to obtain different software that were necessary in the completion of this project, it is important to notice that a big part of a project consists of simulation and analysis through software. Below we will list the software that was provided to us to ensure a successful project.

Table 15: Software equipment provided

Item	Sponsor
Matlab	UCF
ADMS	Dr. Qu Zhihua GE FPL

9.3.3 Data

Finally, since we want our project to be implemented in real-life situations, we used multiple past data to do trials and create algorithms that help us achieve our goal. We will also be using this data in the future to feed a machine learning algorithm that will help us identify any faults in our system. Below we will list all data that was provided to us, and the sponsor that facilitate us with it. A major aspect of our project falls into understanding patterns to identify faults, and the use of this data was fundamental in the success of this project.

Table 16 Data provided

Item	Sponsor
ADMS data	FPL
AMI data	FPL

9.4 LLM Declaration

Declaration: We hereby declare that we have not copied more than 7 pages from the Large Language Model (LLM). We have utilized LLM for drafting, outlining, summarizing, and proofreading purposes.

10 Conclusion

In conclusion, our senior design project has been an incredible journey that has allowed us to showcase our skills, creativity, and dedication to quality in engineering. Throughout this project, we have faced numerous challenges and setbacks and have collaborated tirelessly to deliver a project that exceeds expectations.

Working as a team, we have not only honed our technical abilities but also learned the importance of proper communication, collaboration, adaptability, and time management. Without these things this project could never have reached the level of quality that it is at today and we are all so proud of what we have accomplished as a team and collaborative unit.

We would like to thank once again our faculty advisors, mentors, and all of those who have helped and supported us throughout this experience. Without their guidance and support, this project would not be possible.

Although the completion of this senior design project marks the end of our academic journey, it also marks the beginning of our transitions into the professional world and our future careers as engineers. The invaluable lessons we have learned in this project and in our time as students at UCF will undoubtably serve us well as we move forward in our careers.

In closing, we are immensely proud of our senior design project and the positive impact it has had on us as both individuals and as a team. We are incredibly eager to continue the drive and passion that has led to the completion of this project and use it to create positive change in the world.

Citation

[1] Murari Mohan Saha, Ratan Das, Pekka Verho, & Damir Novosel. (2002). REVIEW OF FAULT LOCATION TECHNIQUES FOR DISTRIBUTION SYSTEMS. Beijing; Power Systems and Communications Infrastructures for the future.

[2] M. Forgione and D. Piga. DynoNet: A neural network architecture for learning dynamical systems. International Journal of Adaptive Control and Signal Processing, 35(4):612–626, 2021.

[4] DroneBot Workshop. (2023, April 12). Controlling AC devices with Arduino - safely. DroneBot Workshop. https://dronebotworkshop.com/ac-arduino/

[5] SEL-551/SEL-551C overcurrent and reclosing relay. Available at: https://selinc.com/api/download/2846/ (Accessed: 25 July 2023).

[7] Team, T.A. (no date) Arduino as ISP and Arduino bootloaders, Arduino Documentation. Available at: https://docs.arduino.cc/built-in-examples/arduino-isp/ArduinoISP (Accessed: 25 July 2023).

[8] Team, T.A. (no date b) From Arduino to a microcontroller on a Breadboard, Arduino Documentation. Available at: https://docs.arduino.cc/built-in-examples/arduino-isp/ArduinoToBreadboard (Accessed: 25 July 2023).

[9] ChatGPT

[10] Partial assistance from ChatGPT

[11] Supervised and unsupervised learning (2023) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/supervised-unsupervised-learning/# (Accessed: 25 July 2023).

[12] Supervised learning (2023) Wikipedia. Available at: https://en.wikipedia.org/wiki/Supervised_learning (Accessed: 25 July 2023).

[13] Linear predictor function (2023) Wikipedia. Available at: https://en.wikipedia.org/wiki/Linear_predictor_function (Accessed: 25 July 2023).

[14] Representation: Feature engineering | machine learning | google for developers (no date) Google. Available at: https://developers.google.com/machine-learning/crash-course/representation/feature-engineering (Accessed: 25 July 2023).

[15] (No date a) Chapter logistic regression - Stanford University. Available at: https://www.web.stanford.edu/~jurafsky/slp3/5.pdf (Accessed: 25 July 2023).

[16] H. Zhai, X. Wang, M. Ge, Z. Yang, S. Feng and Y. Liu, "Research on Fault Identification of the Transmission Line Based on Improved Convolutional Neural Network," *2021 China International Conference on Electricity Distribution (CICED)*, Shanghai, China, 2021, pp. 606-610, doi: 10.1109/CICED50259.2021.9556730.

[17] Gavrilova, Y. (2023) Introduction to tiny ML, Serokell Software Development Company. Available at: https://serokell.io/blog/introduction-to-tinyml (Accessed: 25 July 2023).

[18] AFG2021 arbitrary-function generator Datasheet - Tektronix. Available at: https://download.tek.com/datasheet/AFG2021-Arbitrary-Function-Generator-Datasheet-75W280895.pdf (Accessed: 25 July 2023).

[19] Arbitrary waveform generators (no date) Tektronix. Available at: https://www.tek.com/en/products/arbitrary-waveform-generators?&utm_source=google&utm_medium=ppc&utm_campaign=tek-awg&utm_term=wave+form+generator&_bt=588919138079&_bk=wave+form+generator&_bm=b&_bn=g&_bg=136364431162&gclid=CjwKCAjwzJmlBhBBEiwAEJyLu1PEOC7CD27pNtfOPsb5SK_Cefsc_mWYM0cc8bY4d6aaRwWqYBxd-BoCCwUQAvD_BwE (Accessed: 25 July 2023).

[20] (No date a) California Instruments MX22.5 delivers three-phase power. Available at: https://www.electronicdesign.com/technologies/test-measurement/article/21207363/california-instruments-mx225-delivers-threephase-power (Accessed: 25 July 2023).

[21] (No date a) High Power AC & DC source | MX series - 15kva to 135kva. Available at: https://www.programmablepower.com/products/ac-power-sources/mx-series (Accessed: 25 July 2023).

[22] (No date a) BPS / MX / RS Series SCPI Programming Manual. Available at: https://ametekprogrammablepowercn.live.ametekweb.com/-/media/ametekprogrammablepower/files/ac-power-sources/mx-series/manuals/scpi_programming_manual_7003-961-rab.pdf?la=zh-cn&revision=6013614d-1d90-44ee-bf38-2148e8db6158 (Accessed: 25 July 2023).

[23] Ametek MX Series User Manual PDF Download (2022) ManualsLib. Available at: https://www.manualslib.com/manual/2552906/Ametek-Mx-Series.html (Accessed: 25 July 2023).

[24] 013034500 - test accessory, Fuse Adapter (no date) Newark. Available at: https://www.newark.com/tektronix/013034500/fuse-adapter/dp/58K0057?source=Datasheets360&CMP=AFC-ECIA (Accessed: 25 July 2023).

[25] AUTOMATICA.IT 2020: WEA2 - Learning & Control (2020) YouTube. Available at: https://www.youtube.com/live/SrrlhGPLBrA?feature=share (Accessed: 25 July 2023).

[26] Getting started with LTspice (no date) Getting Started with LTspice - SparkFun Learn. Available at: https://learn.sparkfun.com/tutorials/getting-started-with-ltspice/all (Accessed: 25 July 2023).

[27] (No date a) New Jersey Institute of Technology. Available at: http://ecelabs.njit.edu/ece449/intro_to_powerworld_simulator%20.pdf (Accessed: 25 July 2023).

[28] Li, S. (2017) Solving a simple classification problem with Python - Fruits Lovers' edition, Medium. Available at: https://towardsdatascience.com/solving-a-simple-classification-problem-with-python-fruits-lovers-edition-d20ab6b071d2 (Accessed: 25 July 2023).

[29] 5 types of classification algorithms in machine learning (2020) MonkeyLearn Blog. Available at: https://monkeylearn.com/blog/classification-algorithms/ (Accessed: 25 July 2023).

[30] Forgione, M. and Piga, D. (2021) DynoNet: A neural network architecture for learning dynamical systems, arXiv.org. Available at: https://arxiv.org/abs/2006.02250 (Accessed: 25 July 2023).

[31] forgi86 (no date) Forgi86/dynonet: 'dynonet: A neural network architecture for learning dynamical systems' by Marco Forgione and Dario Piga, GitHub. Available at: https://github.com/forgi86/dynonet (Accessed: 25 July 2023).

[32] (No date) Topology Processing and real-time applications - powerworld. Available at: https://www.powerworld.com/files/GrijalvaRealTimeApps.pdf (Accessed: 25 July 2023).

[33] EPRI Home. Available at: https://www.epri.com/pages/sa/opendss#:~:text=OpenDSS%20is%20an%20electric%20power,grid%20integration%20and%20grid%20modernization. (Accessed: 25 July 2023).

[34] Simulink - simulation and model-based design (no date) Simulation and Model-Based Design - MATLAB. Available at: https://www.mathworks.com/products/simulink.html (Accessed: 25 July 2023).

[35] Multi-agent opendss, an open source and scalable distribution grid platform (no date) IEEE Smart Grid. Available at: https://smartgrid.ieee.org/bulletins/may-2021/multi-agent-opendss-an-open-source-and-scalable-distribution-grid-

platform#:~:text=The%20open%20source%20Distribution%20System,quasi%20steady%2Dstate)%20simulation. (Accessed: 25 July 2023).

[36] Chroma 61704 programmable AC power source 6 KVA, 1 kW (no date) Advanced Test Equipment Corp. - Rentals, Sales, Calibration, Service. Available at: https://www.atecorp.com/products/chroma/61704#:~:text=The%2061704%20supplies%20the%20output,simulate%20unbalance%203%2Dphase%20power. (Accessed: 25 July 2023).

[37] Automated Test Equipment - Chroma Systems Solutions (2023) Chroma Systems Solutions, Inc. Available at: https://www.chromausa.com/ (Accessed: 25 July 2023).

[38] RS-232C - what is RS-232C? (2020) Computer Notes. Available at: https://ecomputernotes.com/computernetworkingnotes/communication-networks/what-is-rs-232c (Accessed: 25 July 2023).

[39] Chroma 61704 3-phase AC power source for MIL-STD-704 and RTCA DO-160 testing (no date) The EMC Shop. Available at: https://theemcshop.com/electronic-test-equipment/power-supplies/chroma-61704-3-phase-ac-power-source-for-mil-std-704-and-rtca-do-160-testing/ (Accessed: 25 July 2023).

[40] Chroma 61704AC source 300V DC, 15 - 1.2KHz, 6KVA, 3 phase (no date) Axiom Test Equipment Rental. Available at: https://www.axiomtest.com/Power-Supplies/AC-Power-Sources,-More-than-3kVA-up-to-10kVA/Chroma/61704/AC-Source-300V-DC,-15-_-1.2kHz,-6kVA,-3-Phase/?gclid=Cj0KCQjw2eilBhCCARIsAG0Pf8tlAqClwEMTPtSRcXRvOQjy6tcY8JKNQcKAQTiZ-iGsdYaG0sNHjzQaAuiPEALw_wcB (Accessed: 25 July 2023).

[41] Prado, K.S. do (2020) Understanding time complexity with python examples, Medium. Available at: https://towardsdatascience.com/understanding-time-complexity-with-python-examples-2bda6e8158a7 (Accessed: 25 July 2023).

[42] R. Das, M. S. Sachdev and T. S. Sidhu, "A fault locator for radial subtransmission and distribution lines," 2000 Power Engineering Society Summer Meeting (Cat. No.00CH37134), Seattle, WA, USA, 2000, pp. 443-448 vol. 1, doi: 10.1109/PESS.2000.867627. https://ieeexplore.ieee.org/document/867627

[43] T. Takagi, Y. Yamakoshi, M. Yamaura, R. Kondow and T. Matsushima, "Development of a New Type Fault Locator Using the One-Terminal Voltage and Current Data," in IEEE Power Engineering Review, vol. PER-2, no. 8, pp. 59-60, Aug. 1982, doi: 10.1109/MPER.1982.5519781. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5519781&isnumber=5519703

[44] 12V wall adapter ACU120100D0531 Jameco Reliapro. (n.d.-a). https://www.jameco.com/z/ACU120100D0531-Jameco-Reliapro-AC-AC-Wall-Transformer-12-VAC-1A-2-5mm-Plug_100167.html

[45] Eagle (V 9.6.2) Computer software. Autodesk.

[46] *About Us*. JLCPCB. (n.d.). https://jlcpcb.com/about

[47] *ACS37002LMCATR-050B3: Digi-key electronics*. Digi. (n.d.). https://www.digikey.com/en/products/detail/ACS37002LMCATR-050B3/620-ACS37002LMCATR-050B3CT-ND/13682936?curr=usd&utm_campaign=buynow&utm_medium=aggregator&utm_source=octopart

[48] *Affordable prototype PCB manufacturer in China*. Custom PCB Prototype Manufacturer. (n.d.). https://www.pcbway.com/

[49] *Altium Designer - PCB design software*. Altium. (n.d.). https://www.altium.com/altium-designer

[50] Cadence PCB Solutions. (2022, October 13). *IPC component spacing guidelines*. IPC Component Spacing Guidelines. https://resources.pcb.cadence.com/blog/2021-ipc-component-spacing-guidelines

[51] *CS60-010 50/60 hz current sensor*. Coilcraft. (n.d.). https://www.coilcraft.com/en-us/products/transformers/power-transformers/current-sensing/cs60-10/

[52] *Eagle key features 2022: Upcoming advanced features 2023*. Autodesk. (2023, May 2). https://www.autodesk.com/products/eagle/features

[53] *IPC-2221: The standard for printed circuit board design*. NextPCB. (n.d.). https://www.nextpcb.com/blog/ipc-2221

[54] KiCad Eda Reviews 2023: Details, pricing, & features | G2. (n.d.-b). https://www.g2.com/products/kicad-eda/reviews

[55] *KiCad Eda*. Schematic Capture & PCB Design Software. (n.d.). https://www.kicad.org/

[56] *Libguides: Engineering Subject Guide: Standards*. Standards - Engineering Subject Guide - LibGuides at Trinity University. (n.d.). https://libguides.trinity.edu/engr/standards#:~:text=A%20standard%20can%20be%20defined,on%20engineering%20components%20or%20equipment.

[57] Matric Group. (n.d.). *IPC class definitions: Class 1, 2, & 3 electronics*. IPC Class Definitions | Class 1, 2, & 3 Electronics. https://blog.matric.com/ipc-class-definitions-class-1-2-3-electronics

[58] *Non-invasive AC current sensor-30a*. Elecrow. (n.d.). https://www.elecrow.com/noninvasive-ac-current-sensor30a-p-842.html

[59] *PCB trace and pad clearance: Low vs. high voltage*. Altium. (2023, June 25). https://resources.altium.com/p/pcb-trace-and-pad-clearance-low-vs-high-voltage

[60] Phoebe. (2023, January 7). *A comprehensive guide to IPC-2221 standards in PCB design*. Swimbi. https://swimbi.com/ipc-2221/

[61] *Printed circuit board (PCB) manufacturing - Camptech II circuits*. Camptech II Circuits Inc. (2021, November 30). https://camptechii.com/services/pcb-fabrication-2/

[62] Yuehuam current Transformer, DROK 100A 100mA AC current sensor ... (n.d.-c). https://www.amazon.com/Current-Transformer-Sensor-PEACEFAIR-PZCT-02/dp/B081CQNXSL

[63] *Coding standards for quality and compliance*. Perforce Software. (n.d.). https://www.perforce.com/resources/qac/coding-standards

[64] *Google C++ Style Guide*. Google C++ style guide. (n.d.). https://google.github.io/styleguide/cppguide.html

[65] Wikimedia Foundation. (2023, July 24). *SCADA*. Wikipedia. https://en.wikipedia.org/wiki/SCADA

[66] *Ami: Fundamentals of advanced metering infrastructure*. Eaton. (n.d.). https://www.eaton.com/in/en-us/products/utility-grid-solutions/advanced-metering-infrastructure/fundamentals-of-ami.html

[67] MediLexicon International. (n.d.). *Electric shock: Effects on body, symptoms, first aid, and more*. Medical News Today. https://www.medicalnewstoday.com/articles/electric-shock#what-is-it