4 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, power system, or software.

The testing plan should connect the requirements and the design to the adopting test strategy and instruments. In this overarching introduction, given an overview of the testing strategy. Emphasize any unique challenges to testing for your system/design.

4.1 Unit Testing

Using PLECS and the PLECS Model our team advisor gave us we are to collect current and voltage waveforms. The given PLECS Model allows us to export these waveforms as excel files.

4.2 Interface Testing

Our current PLECS Model is a block diagram of a mini power grind with an emphasis on having multiple transmission lines set up.PLECS allows us to add in a “fault block” that can simulate the different types of faults along the transmission line. When done properly the user can see the effect on the voltage and current waveforms through the output within their respective graphs.

4.3 Integration Testing

Our integration paths are going to be our RT Box sending a signal to our PLECS model which then feeds data to a csv file that our python code will be able to use to determine if there is a fault in the signal. To test that the RT Box is properly sending a signal to PLECS we can make a basic model that shows us the signal being used in PLECS and if it matches what the RT Box is sending. Then, to make sure the PLECS model is sending the correct information to the csv file we will just have to use analysis tools in PLECS to make sure the information in PLECS matches what is on the csv file. Finally, to test the python code, we would run the whole project together and if it can accurately detect faults then we know that our python code is running properly and our project can detect faults.

4.4 System Testing

Describe system level testing strategy. What set of unit tests, interface tests, and integration tests suffice for system level testing? This should be closely tied to the requirements. Tools?

As for the unit tests, we will be testing the specific faults, their detection, and the circuit breaker switch. As for interface testing, we will need to test between PLECS, python, C, excel, and our real-time simulator (RT box). Our integration tests will include the final runs of our testing. This will include all of our softwares coordination of the C code into the real-time simulator with the data that we previously collected from PLECS. Other than our previously stated platforms of softwares, we will be using an RT box as a tool.

4.5 Regression Testing

How are you ensuring that any new additions do not break the old functionality? What implemented critical features do you need to ensure they do not break? Is it driven by requirements? Tools?

To ensure new additions will not break the system, the neural network will be hosted on the GitLab repository. As we add new changes and features, we will commit and push them to the version control software. Each of these commits represent a version of the program that we can return to and work from at a later point if we feel the current direction is not working. To make sure our main product is working properly, we will use branches to work on larger features before merging them into the main branch once they are finished.

4.6 Acceptance Testing

We will ensure that our design is meeting the specified time requirements as well as perform as it is expected to by repetitive testing and ensuring that time constraints and results are not being missed. Accurate distance detection should also occur and be calculated correctly with multiple sets of data being fed to the neural network. We will test all different scenarios and ensure that the testing done for each scenario is performing as satisfactory. We will involve our client by running them through the data and double checking that the system is performing as expected as set by the client.

4.7 Results

Our result is going to be an accurate function that allows us to analyze a group of faults that will then determine if a circuit breaker needs to operate. This will ensure compliance with the requirements by being able to accurately detect and categorize a fault that is occurring. If the circuit breaker operates, it helps us to minimize damages to both the environment and the system as a whole.

If everything were to work as planned, we would be able to run many different simulations to collect a bunch of data. From here, Python would analyze the data and be able to craft a function that would accurately determine if a fault was occurring during a certain amount of time. If a fault is detected, our system should send a signal to the circuit breaker. When it receives this command, the circuit breaker should operate and isolate the fault. Our function should then determine the type of fault and the location at which the fault occurred. This would simulate a correct operation of the system and confirm that our design works as expected.