2 Project Plan

2.1 Project Management/Tracking Procedures

We have chosen to use the Agile methodology, as we are working alongside our client rather than for our client. Most of our project goals are appearing as we meet with our client and come up with some information for him.

Our group has decided to use Discord, Google Docs, and GitLab to track our progress throughout the course.

2.2 Task Decomposition

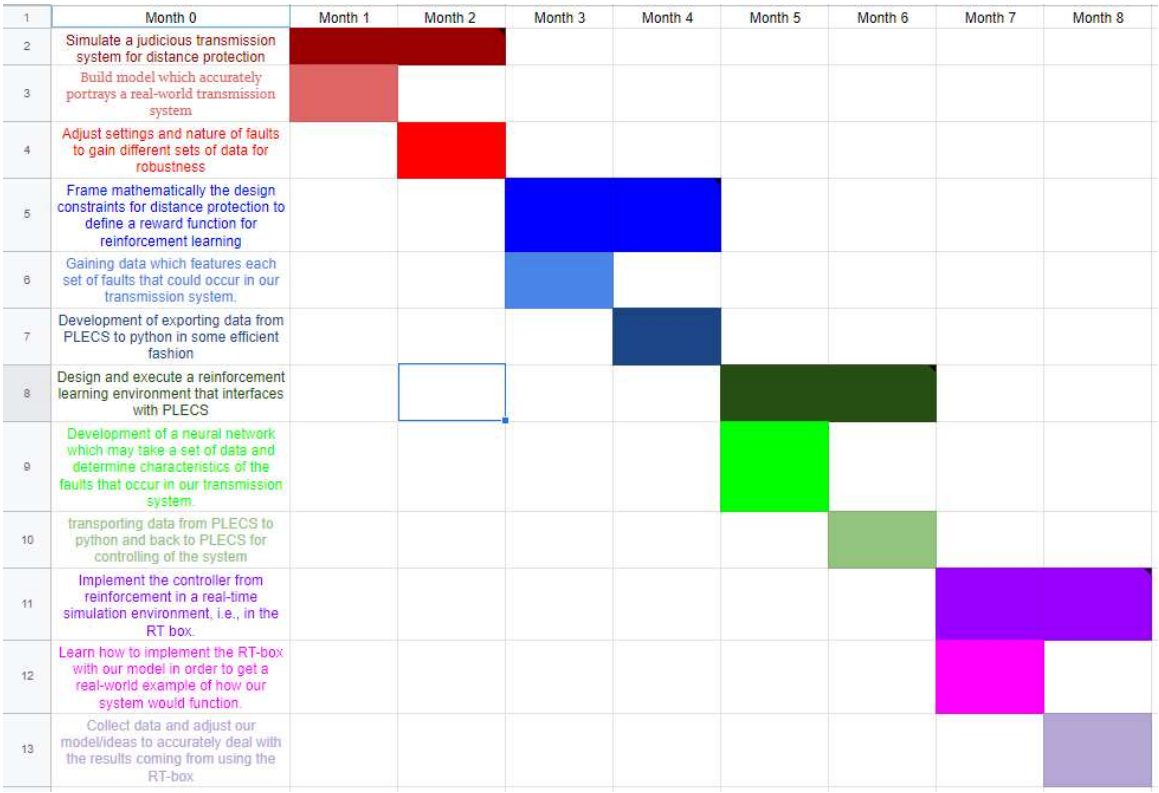
1. Simulate a judicious transmission system for distance protection studies in PLECS
    a. Build model which accurately portrays a real-world transmission system
    b. Adjust settings and nature of faults to gain different sets of data for robustness
2. Frame mathematically the design constraints for distance protection to define a reward function for reinforcement learning
    a. Gaining data which features each set of faults that could occur in our transmission system.
    b. Development of exporting data from PLECS to python in some efficient fashion
    c. Open circuit breakers based on numerical analysis of faults that occur.
3. Design and execute a reinforcement learning environment that interfaces with PLECS
    a. Development of a neural network which may take a set of data and determine characteristics of the faults that occur in our transmission system.
    b. transporting data from PLECS to python and back to PLECS for controlling of the system
4. Implement the controller from reinforcement in a real-time simulation environment, i.e., in the RT box.
    a. Learn how to implement the RT-box with our model in order to get a real-world example of how our system would function.
    b. Collect data and adjust our model/ideas to accurately deal with the results coming from using the RT-box

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

1. Simulation model - Measurement of progress would be if this is complete with all aspects taken into account.
    a. Our model should be able to simulate a fault occurrence at some point in the line with a decent amount of accuracy (>75%).
    b. The model should also be able to detect that a fault has occurred with near perfect accuracy (>90%).
2. Mathematical model of a reward function - Measurement of progress would be if this is complete and able to detect a fault.

    a. Our mathematical model should be able to detect a fault with 75% accuracy.
3. Script for reinforcement learning - Measurement of progress would be if this is complete and is able to detect a fault and open the circuit breaker.
    a. The script should be accurately determining if a fault is present and decide what to do in the instance of a fault (>50%).
4. Final proof of concept while using a real-time simulator - Measurement of progress would be if this is complete and works around 80% of the time correctly.
    a. The model and final proof should be able to accurately find that a fault is occurring and determine what type. This should also be able to de-energize the line and give a rough estimate of the distance at which the fault occurred. This should be around >80% accurate.

2.4 Project Timeline/Schedule



2.5 Risks And Risk Management/Mitigation

PLECS components don't run or work as intended which then skews our data and give us incorrect information

Model developed in PLECS does not interface sufficiently with RT-box and gives erroneous data/results.

There is some internal issue, or malfunction, that occurs with the RT box that isn't obvious and interferes with our real-time simulation.

- For any issue with the RT-box, we shall feed in control cases to get a baseline for how the RT box is functioning as well as any adjustments that need to be made.

2.6 Personnel Effort Requirements

Task	Reference/explanation	Estimate (person-hours)
Simulation model	Reference: We have spent around 3 hours running each of our faults. Explanation: This will include each of the faults, but will need much more data for our program.	10-20 hours
Mathematical model of a reward function	Reference: This is also based on the time we have spent in PLECS up to this point. Explanation: This will include the execution of the fault detection, not the data we collected to find it.	5-10 hours
Script for reinforcement learning	Reference: This is based on our overall knowledge of Python and what it will take to complete. Explanation: This will include the initial simulation model, and our AI system that will execute the location of the fault.	20-30 hours
Final proof of concept while using a real-time simulator	Reference: We do not have a current reference because of where we are in the project and do not have any prior knowledge using this. Explanation: This will include the final simulation and testing our fault location system.	10-20 hours

2.7 Other Resource Requirements

Other resources that we will need to use for our project will include a RT box that will run our simulations in real time. We will also need to use Python to run and examine our testing from our PLECS outputs, which is another resource that we are using.