Lightning Talk: Project Plan

Group 9

Chye Stecher, Josh Vrenick, Keegan Kraft , Matthew Dobrzynski, Taylor Semple, Anthony Ruffalo

Project Management/Tracking Procedures

We plan on using the agile development methodology. To assist with tracking our progress and building our project, we want to use GitLab for code sharing, collaborative work, and issue tracking, Google Drive for file sharing and collaborative documents, and Discord for communication, announcements, and updates.

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
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

Project Proposed Milestones, Metrics, and Evaluation Criteria

- Simulation model - Measurement of progress would be if this is complete with all aspects taken into account.
- Mathematical model of a reward function - Measurement of progress would be if this is complete and able to detect a fault.
- 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.
- 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.

Project timeline/Schedule

	Month 0	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Month 7	Month 8
2	Simulate a judicious transmission system for distance protection								
3	Build model which accurately portrays a real-world transmission system								
4	Adjust settings and nature of faults to gain different sets of data for robustness								
5	Frame mathematically the design constraints for distance protection to define a reward function for reinforcement learning								
6	Gaining data which features each set of faults that could occur in our transmission system.								
7	Development of exporting data from PLECS to python in some efficient fashion								
8	Design and execute a reinforcement learning environment that interfaces with PLECS								
9	Development of a neural network which may take a set of data and determine characteristics of the faults that occur in our transmission system.								
10	transporting data from PLECS to python and back to PLECS for controlling of the system								
11	Implement the controller from reinforcement in a real-time simulation environment, i.e., in the RT box.								
12	Learn how to implement the RT-box with our model in order to get a real-world example of how our system would function.								
13	Collect data and adjust our model/ideas to accurately deal with the results coming from using the RT-box								

Risks and Risk Management/Mitigation

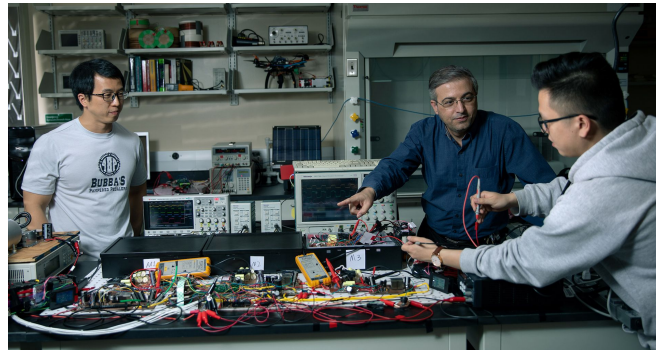Model developed in PLECS does not interface sufficiently with RT-box and gives erroneous data/results.

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.
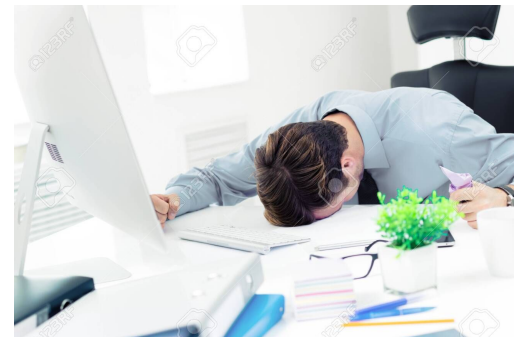
Personal Effort Requirements

- Simulation model: 10-20 hours
- Mathematical model of a reward function: 5-10 hours
- Script for reinforcement learning: 20-30 hours
- Final proof of concept while using a real-time simulator: 10-20 hours

Other Resource Requirements

Other resources that we will need to use for our project will include...

- PLECS
  - Runs our models for each fault type and outputs our results.
- Excel
  - Will hold the format of our outputted data from PLECS models.
- RT Box
  - Runs our simulations in real time.
- Python
  - Examine and compile PLECS results.
  - Read excel information into our neural network.