Manually Tuning ESC Model code

August 4, 2024

0.1 Capstone project: Manually tuning an ESC cell model

Welcome to the capstone project! To complete this assignment, follow the instructions below. When you have completed your modifications to the sample code that I have provided as a starting point, click on the "Submit Assignment" button to submit your code to the grader.

0.1.1 Concept

This Jupyter notebook operates in the same way as all of the other Jupyter notebooks that you have used so far in this specialization. You can type Octave code into notebook cells and execute that code to see how it functions. This allows you to test your code fully before submitting it for grading.

Only one notebook cell is actually graded – the one marked with "% GRADED FUNCTION" in its first line. Do not modify that line – otherwise the grader will not be able to find the correct function for grading.

0.1.2 Deliverables

The goal of the capstone project is to hand-tune the resistor and capacitor value(s) for a simplified ESC model of a battery cell. To reduce project complexity, this model will not consider hysteresis terms. You will do this tuning by trial-and-error to get the best result you are able to find. Some of the guidelines that you learned in the course will be helpful to you.

When optimizing a model for a real application, the model's parameter values are tuned so that the model gives good and robust predictions over a wide variety of operating conditions. However, for this project you will tune the model to make good predictions for only a single set of input/output (current/voltage) data (otherwise, the project would take too long to complete).

The data that you will be working with were collected from an actual cell in the laboratory. At the beginning of the test, the cell was resting at a high state of charge. The test rested for 5 minutes and then a constant-current discharge pulse was applied for 15 minutes. The cell was then allowed to rest for one hour. Voltage and current were measured every second.

Your deliverable will be a set of resistor and capacitor value(s) within the tuneModel function, below.

0.2 Implementation of the cell simulator

This remainder of this notebook implements a cell simulator using the resistor and capacitor value(s) that you provide. It is similar to simCell.m, which you have seen before, except that it deletes hysteresis states.