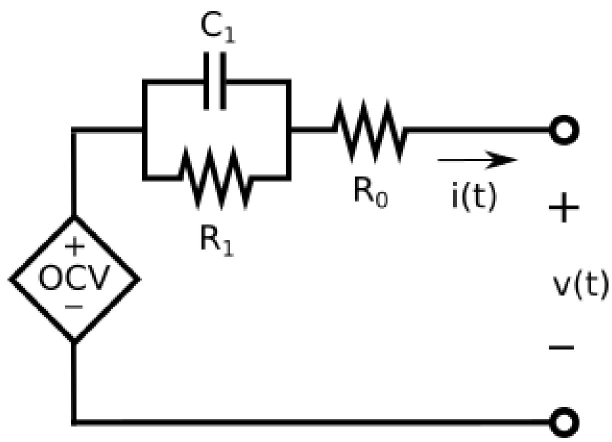
AEM 591 Project 2

Optimal Battery Parameter Estimation

This project is intended to introduce the concept of least-squares in Python for use in optimal parameter estimation.

To that end, consider the following equivalent circuit diagram of a battery.



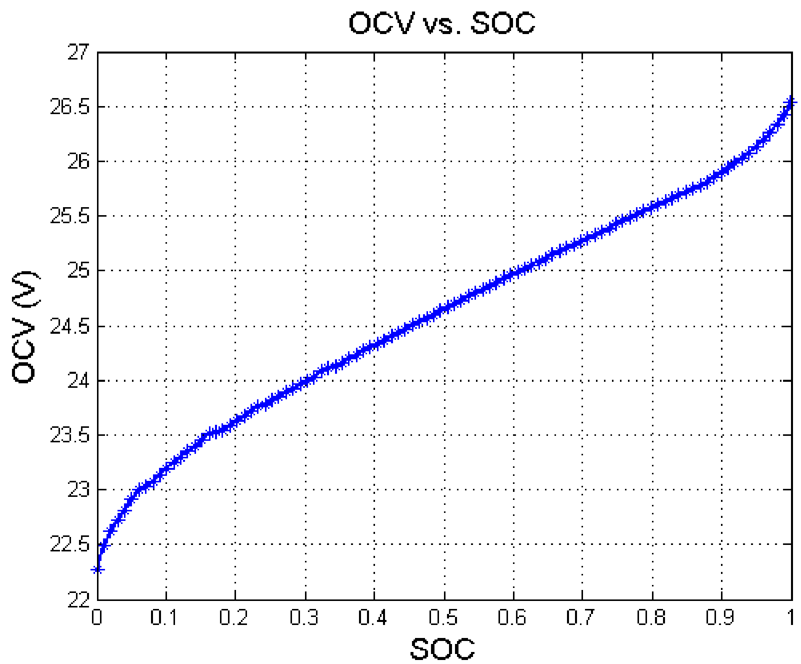
where R_0 is the internal resistance, $\tau_1 = R_1 C_1$ is one dynamic element of the battery (there may be multiple of these), $v(t)$ is the voltage observation, $i(t)$ is the current observation, and OCV is the **open circuit voltage** (OCV) of the battery. In addition, consider the **State-Of-Charge (SOC)** of the battery, SOC, i.e. the percentage of current remaining in the battery relative to the total capacity, Q. which can be written mathematically as

$$SOC(t) = SOC(0) + \frac{\int_0^t i(t)dt}{Q} \tag{1}$$

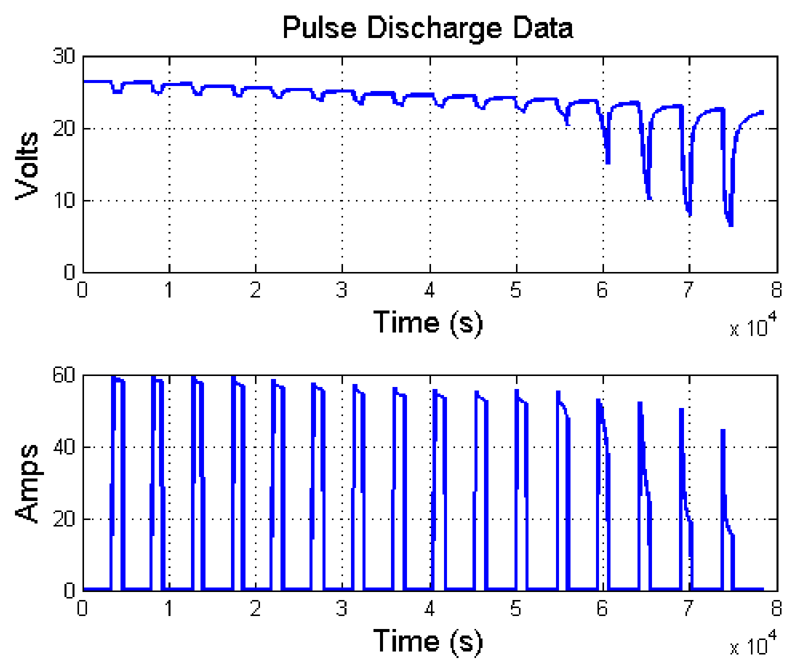
where one can use numerical integration, e.g. a trapezoidal method, to approximate this integral. Note that $i(t)$ should be negative for a discharging battery. Note that the total capacity can be estimated by extracting all of the current from the battery, i.e.

$$Q = \int_0^\infty i(t)dt \tag{2}$$

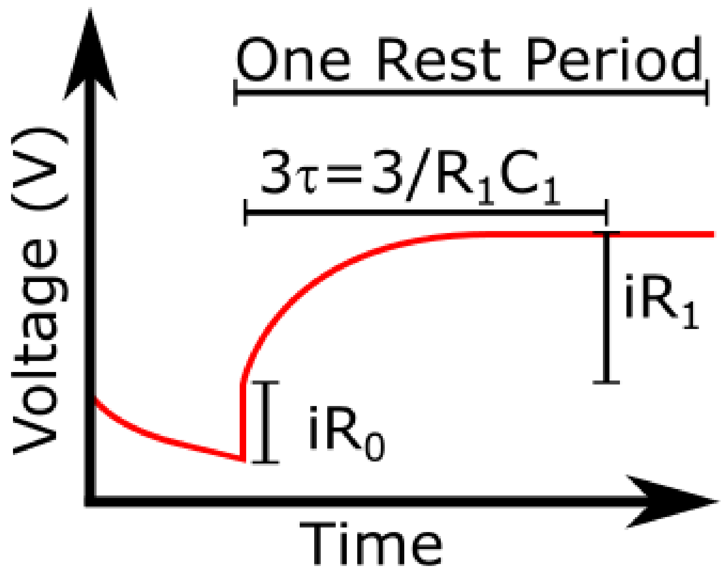
Empirically, it is known that the OCV is a nonlinear function of the SOC as shown in the following figure.



which remains relatively constant over the lifetime of the battery. This OCV-SOC curve can be obtained using a **pulse discharge test** where one can measure the OCV after a long rest period and smooth the samples over multiple tests.



The purpose of this project is to estimate the circuit parameters, including the OCV-SOC curve, from the pulse discharge tests using Nonlinear Least-Squares where each rest period allows one to model the parameters by



However, the number of RC elements is **unknown** for batteries in general thereby the total increase in voltage may occur due to multiple exponential terms, i.e. iR_1, iR_2, ... Thus, one could attempt to fit the time series data of one rest period to an exponential decaying function with any number of terms, i.e.

$$v(t_r) = OCV - i(t_s)R_0 + i(t_s)R_1 \exp\left(\frac{-t_r}{\tau_1}\right) + i(t_s)R_2 \exp\left(\frac{-t_r}{\tau_2}\right) + \cdots \tag{3}$$

where $\tau_i = R_iC_i$ is the time constant of the i^{th} exponential, t_r is the time across one rest period, and t_s is the single time step immediately before the rest period. Note that the OCV is the value of $v(t_r)$ as $t \to \infty$ when no current is being discharged.

Project Assignment and Deliverables

For this project, perform the following tasks in Python:

1. load the battery data for both sets of pulse discharge tests,
   - i.e. columns 2-4 and columns 5-7 of the accompanying CSV file are different sets;
2. partition the data into each rest period using the current measurement as the trigger,
   - where one can see where the current is set back close to zero instead of the discharge amperage;
3. for each rest period, in each pulse discharge data set,
   - compute four sets of optimal battery parameters for the exponential decaying function using nonlinear least-squares with 1, 2, 3, and 4 exponential terms, e.g. `scipy.optimize.least_squa`
   - compare the models through an analysis of the observed residuals;

4. plot the estimated parameters as a function of SOC over each pulse discharge test and comment on their variety with SOC;
    - comment on the improvement from using additional RC elements which make the model more complicated, both in terms of residuals and
5. numerically integrate the current to obtain the SOC at each rest period; and
6. estimate the OCV-SOC curve for each battery and compare for both batteries,
    - use the OCV from each of the exponential function fits.