

nasa-battery-data-analysis

December 6, 2016

```
In [102]: from scipy.io import loadmat, whosmat
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline
```

```
In [65]: struct = loadmat('./nasa-battery-data/BatteryAgingARC-FY08Q4/B0005.mat')
data = struct['B0005'][0][0][0][0]
```

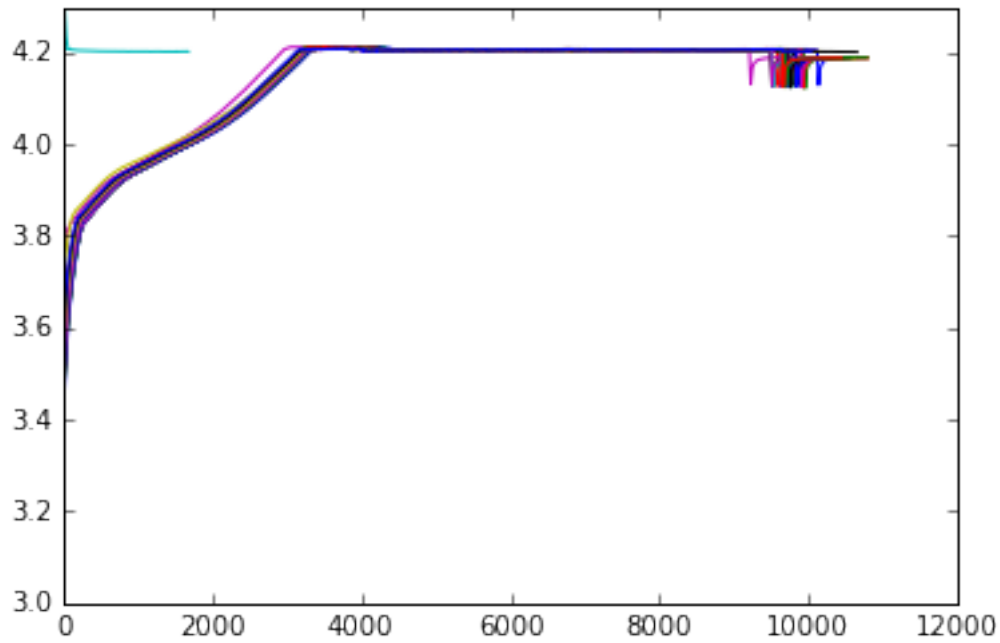
0.1 charge/discharge data structure

for each element in data...

- element[0] = charge/discharge/impedance
- if element[0] = charge/discharge
 - element[1] = ambient temperature
 - element[2] = date/time
 - element[3] = data
 - * data fields:
 - Voltage_measured
 - Current_measured
 - Temperature_measured
 - Current_charge
 - Voltage_charge
 - Time

```
In [129]: for cycle in data[1:100]:
    if (cycle[0] == 'charge'):
        time = np.hstack(cycle[3]['Time'])[0][0]
        voltage = np.hstack(cycle[3]['Voltage_measured'])[0][0]
        current = np.hstack(cycle[3]['Current_measured'])[0][0]
        plt.plot(time, voltage)

plt.ylim(3, 4.3)
plt.show()
```

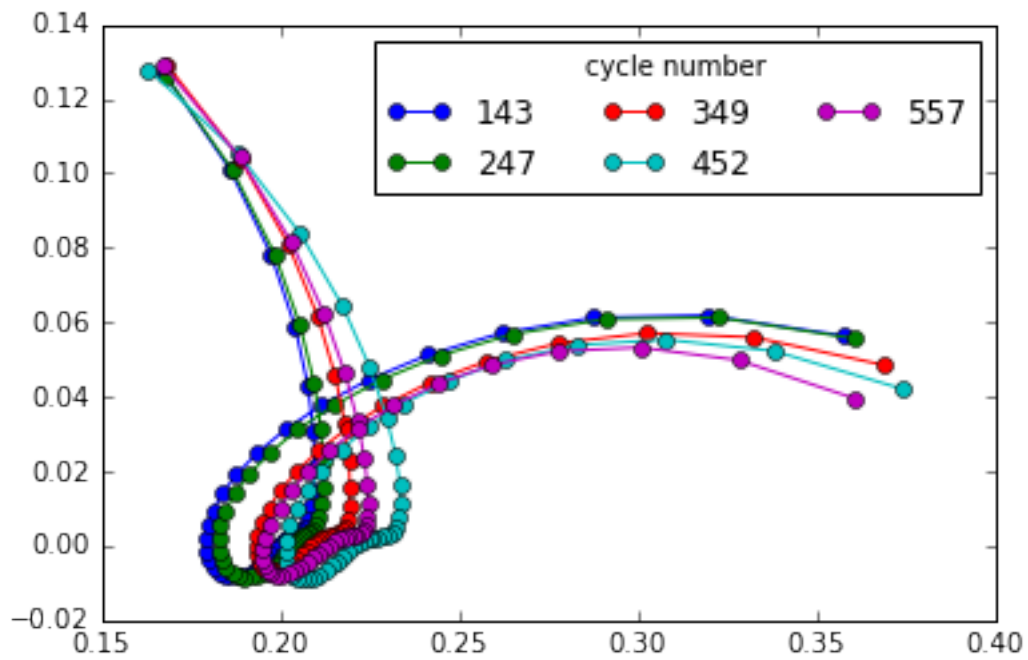


0.2 impedance data structure

- if element[0] = impedance
 - element[1] = ambient temperature
 - element[2] = date/time
 - element[3] = data
 - * data fields:
 - Sense_current
 - Battery_current
 - Current_ratio
 - Battery_impedance
 - Rectified_Impedance
 - Re
 - Rct

```
In [173]: count = 0
          cycle_num = 0
          for cycle in data:
              cycle_num+=1
              if (cycle[0] == 'impedance'):
                  count += 1
                  impedance = np.hstack(cycle[3])[0][3] #Battery_impedance
                  if count % 50 == 0:
                      plt.plot(np.real(impedance), -np.imag(impedance), 'o-', label=
```

```
plt.legend(title='cycle number', ncol=3)
plt.show()
```



```
In [ ]: count = 0
        cycle_num = 0
        for cycle in data:
            cycle_num+=1
            if (cycle[0] == 'impedance'):
                count += 1
                impedance = np.hstack(cycle[3])[0][3] #
                if count % 50 == 0:
                    plt.plot(np.real(impedance), -np.imag(impedance), 'o-', label=cycle_num)
        plt.legend(title='cycle number', ncol=3)
        plt.show()
```