HW 1: Battery Modeling, Analysis, and Simulation

Franklin Zhao
SID: 3033030808

2 Feb. 2018

Problem 1: Review Submission Procedure from HW0

Problem 2: Reading

The first issue is decarbonization, which is the main driver for the smart grid development. With the growing concerns over energy scurity and global climate change, interest in renewables is increasing significantly. More significant progress will be made in understanding of environmental issues and using electricity innovations to reslove the energy issue. The second issue is reliability in the face of growing demand. With the increasing stress in electric grid, a question whether carrying capacity or safety margin will exist to satisify the demand is raised. The third issue is the electrification of transportation. Next-generation transportation systems will depend more electricity supplied by the grid. The fourth issue is that consumers have more power. The size of system peak influences the capacity needs to be built, and short timescale variability determines the flexibility required to follow the demand. The last issues are market designs and regulatory paradigms. New market designs have pushed the system upgrade and investment by allowing competition in several economy strategic sectors.[1]

In terms of the growing demand of consumers, for example, we can minimize a cost function from the view of the utility operator or the consumer, and review fundamental problems in energy storage management. To achieve the optimization objective, we can focus on deciding when and how much to charge and discharge the battery, either in terms of a mismatch between energy supply and demand or the cost.[2]

[1] Annaswamy, A. M. "Vision for smart grid control: 2030 and beyond." *IEEE Standards Publication* (2013).

[2] Koutsopoulos, Iordanis, Thanasis G. Papaioannou, and Vasiliki Hatzi. "Modeling and optimization of the smart grid ecosystem." *Foundations and Trends® in Networking* 10.2-3 (2016): 115-316.

Problem 3: Black-box vs. White-box Modeling

The answer[3] is shown in Table 1.

Table 1: Black-box vs. White-box

	Black-Box Models	**White-Box Models**
Advantages	• Useful for training data rigime • No domain knowledge needed • Minimal required computing power	• Insight into internal phenomena • Extrapolation can be done • Very close to actual behavior
Disadvantages	• Lack of flexibility • Lack of physical meaning • Not good for sensitivity analysis	• May be very complex • Restricted to assumed structure • Large computing overheads

Problem 4: Mathematical Modeling Use

The five potential uses are listed as follows[4]:

1 **Analysis.** Given a future trajectory of $u(t)$, $x(0)$ at the present, and the system model Σ, predict the future of $y(t)$.

2 **State Estimation.** Given a system Σ with time histories $u(t)$ and $y(t)$, find x that is consistent with Σ, u, y.

3 **System Design or Planning.** Given $u(t)$ and some desired $y(t)$, find Σ such that $u(t)$ acting on Σ will produce $y(t)$.

4 **Model Identification.** Given time histories $u(t)$ and $y(t)$, usually obtained from experimental data, determine a model Σ and its parameter values that are consistent with u and y.

5 **Control Synthesis.** Given a system Σ with current state $x(0)$ and some desired $y(t)$, find $u(t)$ such that Σ will produce $y(t)$.

Problem 5: Methematical Modeling

(a)

“Reservoirs”(States): State-of-charge $\mathbf{z(t)}$ and capacitor voltage $\mathbf{V_c(t)}$.

[3] http://www.idc-online.com/technical_references/pdfs/electronic_engineering/Modelling_Choosing_a_Model.pdf

[4] Moura, S. J. “Ch1: Modeling Systems” *CE 295 Lecture Notes* (2018).

(b)

According to Kirchoff's voltage and current laws, the following systems equations and integrator dynamics can be derived:

$$V(t) - OCV(z) - V_C(t) - R_1 I(t) = 0 \tag{1}$$

$$I(t) = C\dot{V}_C(t) + \frac{V_C(t)}{R_2} \tag{2}$$

$$\dot{z}(t) = \frac{1}{Q} I(t) \tag{3}$$

(c)

Parameters: Voltage source $\mathbf{OCV(z)}$, Resistance $\mathbf{R_1}$ and $\mathbf{R_2}$, Capacity C, and charge capacity Q.

(d)

State space form:

$$\dot{z}(t) = \frac{1}{Q} I(t) \tag{4}$$

$$\dot{V}_C(t) = -\frac{1}{R_2 C} V_C(t) + \frac{1}{C} I(t) \tag{5}$$

$$V(t) = OCV(z) + V_C(t) + R_1 I(t) \tag{6}$$

(e)

Denote $[z(t),\ V_C(t)]^T$ by X; Denote I by u. The state equations can be written as:

$$\dot{X} = AX + Bu \tag{7}$$

It is a linear state equation, where

$$A = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{R_2 C} \end{bmatrix} \text{ and } B = \begin{bmatrix} \frac{1}{Q} \\ \frac{1}{C} \end{bmatrix}$$

However, the model does **not** have a linear output equation, since the $\mathbf{OCV(z)}$ term produces nonlinearity.

Problem 6: Stability and Linearization

(a)

From the last problem we know that the matrix A is:

$$\begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{R_2C} \end{bmatrix}$$

where we could easily identify that the eigenvalues are $\lambda_1 = 0$ and $\lambda_2 = -\frac{1}{R_2C}$. Since $\lambda_1 = 0$ and $\lambda_2 < 0$, we can conclude that the model is **marginally stable**.

(b)

Now let's perturbate a little around equilibrium:

$$V = V^{eq} + \tilde{V} \tag{8}$$

$$V_C = V_C^{eq} + \tilde{V_C} \tag{9}$$

$$I = I^{eq} + \tilde{I} \tag{10}$$

$$z = z^{eq} + \tilde{z} \tag{11}$$

From Equation 6, the output equation can be derived:

$$\begin{aligned} V &\approx V^{eq} + \tilde{V}_c \frac{\partial V}{\partial V_C}\Big|_{V_C^{eq}} + (I - I^{eq})\frac{\partial V}{\partial I}\Big|_{I^{eq}} + (z - z^{eq})\frac{\partial V}{\partial z}\Big|_{z^{eq}} \\ &= V^{eq} + (V_C - V_C^{eq})\frac{\partial V}{\partial V_C}\Big|_{V_C^{eq}} + (I - I^{eq})\frac{\partial V}{\partial I}\Big|_{I^{eq}} + (z - z^{eq})\frac{\partial V}{\partial z}\Big|_{z^{eq}} \\ &= V_C^{eq} + R_1 \times I^{eq} + (p_0 + p_1 z^{eq} + p_2 z^{eq2} + p_3 z^{eq3}) + (V_C - 0) \times 1 + (I - 0) \times R_1 + \\ &\quad (z - 0.5)(p_1 + 2p_2 \times 0.5 + 3p_3 \times 0.5^2) \\ &= V_C + 0.05I + (z - 0.5)(p_1 + p_2 + 0.75p_3) + (p_0 + 0.5p_1 + 0.25p_2 + 0.125p_3) \\ &= V_C + (p_1 + p_2 + 0.75p_3)z + 0.05I + (p_0 - 0.25p_2 - 0.25p_3) \end{aligned} \tag{12}$$

which becomes a linear equation.

Problem 7: Simulation and Analysis

(a)

```
## Part(a): Model Parameters
```

```

# ECM Model Parameters
Q = 3600 # [Coulombs]
R1 = 0.05 # [Ohms]
R2 = 0.005 # [Ohms]
C = 500 # [Farads]

# OCV polynomial coefficients
p_0 = 3.4707
p_1 = 1.6112
p_2 = -2.6287
p_3 = 1.7175

# Plot nonlinear OCV function
z_vec = np.linspace(0,1,25)
OCV = p_0 + p_1*z_vec + p_2 * (z_vec ** 2) + p_3 * (z_vec ** 3)
print(z_vec)
print(OCV)

plt.plot(z_vec, OCV)
plt.xlabel('SOC, $z$ [-]',fontsize=fs)
plt.ylabel('OCV [volts]',fontsize=fs)
plt.tick_params(axis='both', which='major', labelsize=fs)
plt.show()
```

The plot of nonlinear OCV functions vs. state-of-charge is shown in Figure 1.

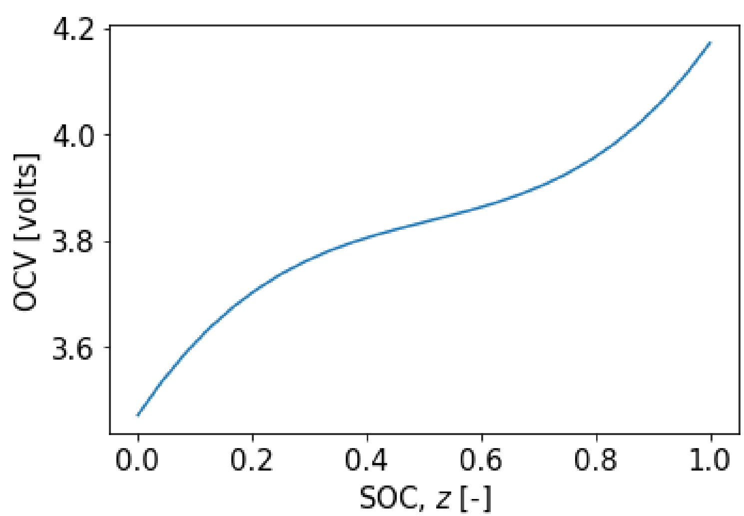

Fig. 1: Nonlinear OCV functions vs. state-of-charge

(b)

```
## Part(b): Simulate

# Assemble (A,B) state-space matrices
A = [[0, 0], [0, -1/(R2 * C)]]
B = [[1/Q], [1/C]]
print(A)
print(B)
C_dummy = [1, 1]
D_dummy = R1
print(C_dummy)
print(D_dummy)

# Create state-space model
sys = signal.lti(A, B, C_dummy, D_dummy)

# Create time vector
DeltaT = 1 # Time step size [sec]
t = np.arange(0,10*60,DeltaT)   # Total Simulation time (min*sec/min)
#print('time')
#print(t)

# Input current signals
#print(np.shape(t))
Current = np.zeros_like(t)*0
for k in range(0, len(Current)):
    if (t[k] % 40) < 20:
        Current[k] = -5

# Initial Conditions
z0 = 0.5 # state-of-charge
V_c0 = 0 # capacitor voltage
x0 = [z0,V_c0] # np.array([[z0],[V_c0]])   # Vectorize initial conditions
print('x0')
print(x0)

print(np.shape(x0))
print(np.shape(Current))
print(np.shape(t))

# Simulate linear dynamics (Read documentation on scipy.signal.lsim)
tsim, y, x = signal.lsim2(sys, Current, t, x0)

```

```python
# Parse out states
z = x[:,0]
V_c = x[:,1]

# Compute nonlinear output function
V_nl = p_0 + p_1 * z + p_2 * (z ** 2) + p_3 * (z ** 3) + V_c + R1 * Current

### Compute linearized output function
# Linearization Points
zeq = 0.5    # state-of-charge
V_ceq = 0 # capacitor voltage
Ieq = 0 # Current

V_lin = V_c + (p_1 + p_2 + 0.75 * p_3) * z + R1 * Current + p_0 - 0.25 * p_2 -
    0.25 * p_3

## Part(b): Plot results

# Current
plt.figure(num=2, figsize=(8, 8), dpi=80, facecolor='w', edgecolor='k')
plt.subplot(3, 1, 1)
plt.plot(t, Current)
plt.ylabel('Current (A)')

# State-of-charge
plt.subplot(3, 1, 2)
plt.plot(t, z)
plt.ylabel('SOC')

# Nonlinear and linearized voltage
plt.subplot(3, 1, 3)
plt.plot(t, V_nl,label="Nonlinear")
plt.plot(t, V_lin, 'r--', label="Linearized")
plt.ylabel('Voltage (V)')
plt.xlabel('Time (s)')
plt.legend()

plt.show()
```

The subplots are shown in Figure 2.

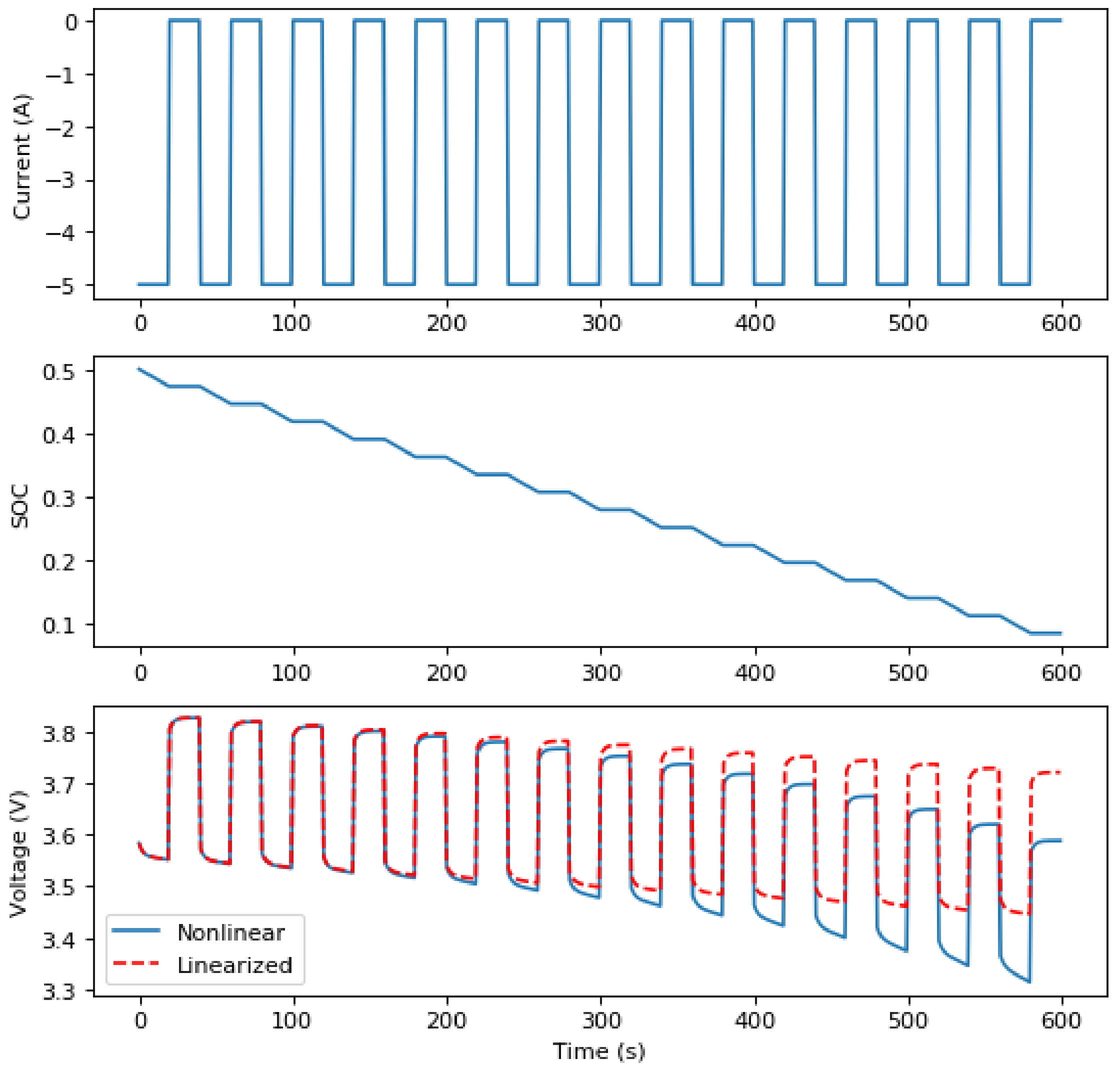

Fig. 2: Current, SOC, and voltage vs. time

(c)

From the OCV and SOC plots in Figure 1 and Figure 2, we could see that when SOC drops below 25%, the nonlinearity of OCV starts increasing dramatically (i.e., the inflection point is around 0.25), which means the system will continue moving away from the linearization point over time. Such results indicate that the linearized model performs well only within a small range, which means the perturbation cannot be too large. If we would like to study a wide range, new linearized models should be made in every small intervals.