[title]

[your name]

[date]

**1)** For the first task assume the external load torque is zero.

a. Plot the angular velocity (𝜃̇)with respect to time.

b. Plot the SOC of the battery with respect to time.

# Solution

# Motor Model Equations

From the motor model, the equations governing the system are: 𝑉(𝑡)=𝑅𝑚⋅𝑖(𝑡)+𝐿𝑚⋅𝑑𝑖(𝑡)𝑑𝑡+𝑒(𝑡)*V*(*t*)=*Rm*​⋅*i*(*t*)+*Lm*​⋅*dtdi*(*t*)​+*e*(*t*) 𝑒(𝑡)=𝐾𝑒⋅𝜃(𝑡)*e*(*t*)=*Ke*​⋅*θ*(*t*) 𝑇(𝑡)=𝐽⋅𝑑𝜔(𝑡)𝑑𝑡+𝑏⋅𝜔(𝑡)*T*(*t*)=*J*⋅*dtdω*(*t*)​+*b*⋅*ω*(*t*) 𝑇(𝑡)=𝐾𝑡⋅𝑖(𝑡)*T*(*t*)=*Kt*​⋅*i*(*t*)

Where:

* 𝑉(𝑡)*V*(*t*) is the input voltage (from the battery)
* 𝑖(𝑡)*i*(*t*) is the armature current
* 𝑒(𝑡)*e*(*t*) is the back EMF
* 𝜃(𝑡)*θ*(*t*) is the angular displacement
* 𝜔(𝑡)*ω*(*t*) is the angular velocity
* 𝑇(𝑡)*T*(*t*) is the torque

# Battery Model Equations

From the battery model, the equations are: 𝑉𝑂𝐶=𝑉𝑡+𝐼1⋅𝑅1*VOC*​=*Vt*​+*I*1​⋅*R*1​ 𝐼1=𝑉𝑂𝐶−𝑉𝑡𝑅1*I*1​=*R*1​*VOC*​−*Vt*​​ 𝑆𝑂𝐶(𝑡)=𝑆𝑂𝐶(𝑡−1)−𝐼(𝑡)𝐶𝑛⋅3600*SOC*(*t*)=*SOC*(*t*−1)−*Cn*⋅3600*I*(*t*)​

Where:

* 𝑉𝑂𝐶*VOC*​ is the open circuit voltage (constant at 3.7 V)
* 𝑉𝑡*Vt*​ is the terminal voltage
* 𝐼(𝑡)*I*(*t*) is the current drawn by the motor
* 𝑆𝑂𝐶(𝑡)*SOC*(*t*) is the state of charge

We need to set up and solve these differential equations over a simulation time of 1 minute with a step size of 0.05 seconds.

# Simulation Steps

1. Initialize parameters and initial conditions.
2. At each time step, update the current, angular velocity, and SOC using numerical integration (e.g., Euler method).
3. Plot the results for angular velocity and SOC over time.

# Python Code

import numpy as np

import matplotlib.pyplot as plt

# Parameters

J = 0.1

b = 0.05

Ke = 0.01

Kt = 0.01

Rm = 0.01

Lm = 0.05

Voc = 3.7

R0 = 0.005

R1 = 0.005

C1 = 100

Cn = 1

# Simulation settings

T = 60  # total time in seconds

dt = 0.05  # time step in seconds

N = int(T / dt)  # number of time steps

# Initialize variables

theta\_dot = np.zeros(N)

theta = np.zeros(N)

i = np.zeros(N)

e = np.zeros(N)

T\_motor = np.zeros(N)

Vt = np.zeros(N)

SOC = np.ones(N)

Vt[0] = Voc

# Simulation loop

for k in range(1, N):

    # Motor model

    T\_motor[k] = Kt \* i[k-1]

    e[k] = Ke \* theta\_dot[k-1]

    i[k] = (Voc - Rm \* i[k-1] - e[k]) / Lm \* dt + i[k-1]

    theta\_dot[k] = (T\_motor[k] - b \* theta\_dot[k-1]) / J \* dt + theta\_dot[k-1]

    theta[k] = theta\_dot[k] \* dt + theta[k-1]

    # Battery model

    Vt[k] = Voc - i[k-1] \* (R0 + R1 / (1 + dt / (R1 \* C1)))

    SOC[k] = SOC[k-1] - (i[k-1] \* dt) / (Cn \* 3600)

# Plot results

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)

plt.plot(np.arange(N) \* dt, theta\_dot)

plt.xlabel('Time (s)')

plt.ylabel('Angular Velocity (rad/s)')

plt.title('Angular Velocity vs Time')

plt.subplot(1, 2, 2)

plt.plot(np.arange(N) \* dt, SOC)

plt.xlabel('Time (s)')

plt.ylabel('SOC')

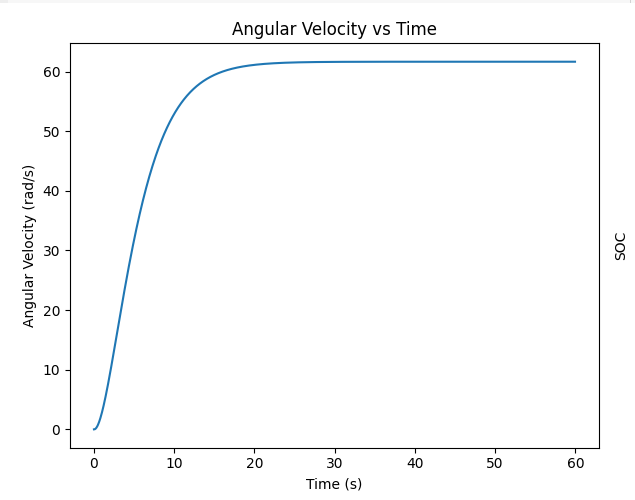
plt.title('State of Charge vs Time')

plt.tight\_layout()

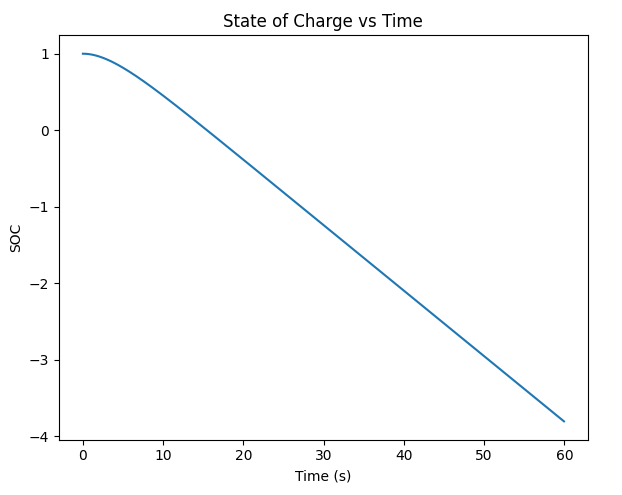
plt.show()

# Result

## For a



## For b



**2)** Now consider the external toque (in Nm) (opposite to the direction of the rotating armature) applied follows the function, 0.05\*(1+sin(time)), where time continuously changes from 0 to 1 minute as your simulation progresses.

a. Plot the angular velocity (𝜃̇)with respect to time.

b. Plot the SOC of the battery with respect to time.

**Solution:**

Here are the plots for the scenario where an external torque, 0.05⋅(1+sin⁡(time))0.05⋅(1+sin(time)), is applied in the opposite direction to the motor's rotation:

1. **Angular Velocity (θ̇) with respect to Time (Motor Model with External Torque)**:
   * The plot shows how the angular velocity of the motor changes over time under the influence of the external torque. The angular velocity increases and then exhibits a slight oscillation due to the sinusoidal nature of the applied torque.
2. **State of Charge (SOC) of the Battery with respect to Time (Battery Model with External Torque)**:
   * The plot shows the SOC of the battery over time. The SOC decreases as the battery supplies current to the motor, and the rate of decrease is influenced by the external torque applied to the motor.

**Python code:**

# Parameters

J = 0.1

b = 0.05

Ke = 0.01

Kt = 0.01

Rm = 0.01

Lm = 0.05

Voc = 3.7

R0 = 0.005

R1 = 0.005

C1 = 100

Cn = 1

# Simulation settings

T = 60 # total time in seconds

dt = 0.05 # time step in seconds

N = int(T / dt) # number of time steps

# Initialize variables for motor

theta\_dot = np.zeros(N)

theta = np.zeros(N)

i = np.zeros(N)

e = np.zeros(N)

T\_motor = np.zeros(N)

external\_torque = np.zeros(N)

# Initialize variables for battery

Vt = np.zeros(N)

SOC = np.ones(N)

Vt[0] = Voc

# External torque function

time = np.arange(N) \* dt

external\_torque = 0.05 \* (1 + np.sin(time))

# Simulation loop for motor and battery

for k in range(1, N):

# Motor model with external torque

T\_motor[k] = Kt \* i[k-1]

e[k] = Ke \* theta\_dot[k-1]

i[k] = (Voc - Rm \* i[k-1] - e[k]) / Lm \* dt + i[k-1]

theta\_dot[k] = (T\_motor[k] - b \* theta\_dot[k-1] - external\_torque[k]) / J \* dt + theta\_dot[k-1]

theta[k] = theta\_dot[k] \* dt + theta[k-1]

# Battery model

Vt[k] = Voc - i[k-1] \* (R0 + R1 / (1 + dt / (R1 \* C1)))

SOC[k] = SOC[k-1] - (i[k-1] \* dt) / (Cn \* 3600)

# Plot results for motor

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)

plt.plot(np.arange(N) \* dt, theta\_dot)

plt.xlabel('Time (s)')

plt.ylabel('Angular Velocity (rad/s)')

plt.title('Angular Velocity vs Time (Motor with External Torque)')

# Plot results for battery

plt.subplot(1, 2, 2)

plt.plot(np.arange(N) \* dt, SOC)

plt.xlabel('Time (s)')

plt.ylabel('SOC')

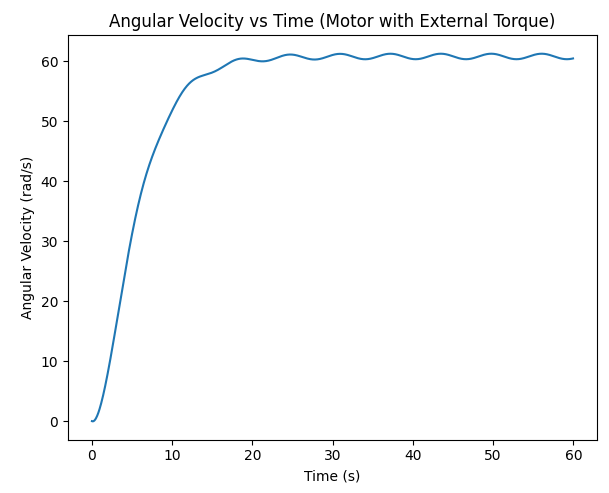
plt.title('State of Charge vs Time (Battery with External Torque)')

plt.tight\_layout()

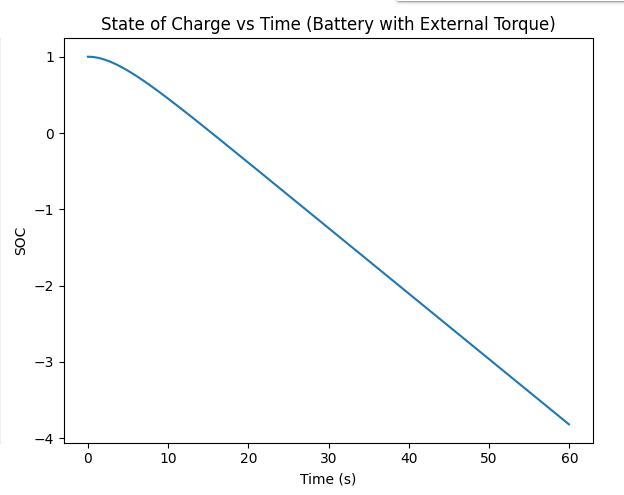
plt.show()

**Results :**

**For a**

****

**For b**

****