📝 Project | August 18

Task:

Battery charging and discharging simulation

- Models li –ion battery charging/discharging behaviour
- Inputs=capacity,current,voltage
- Outputs=SOC(state of charge),effiecieny
- Libraries=numpy,matplotlib
- Applications=EV battery management systems

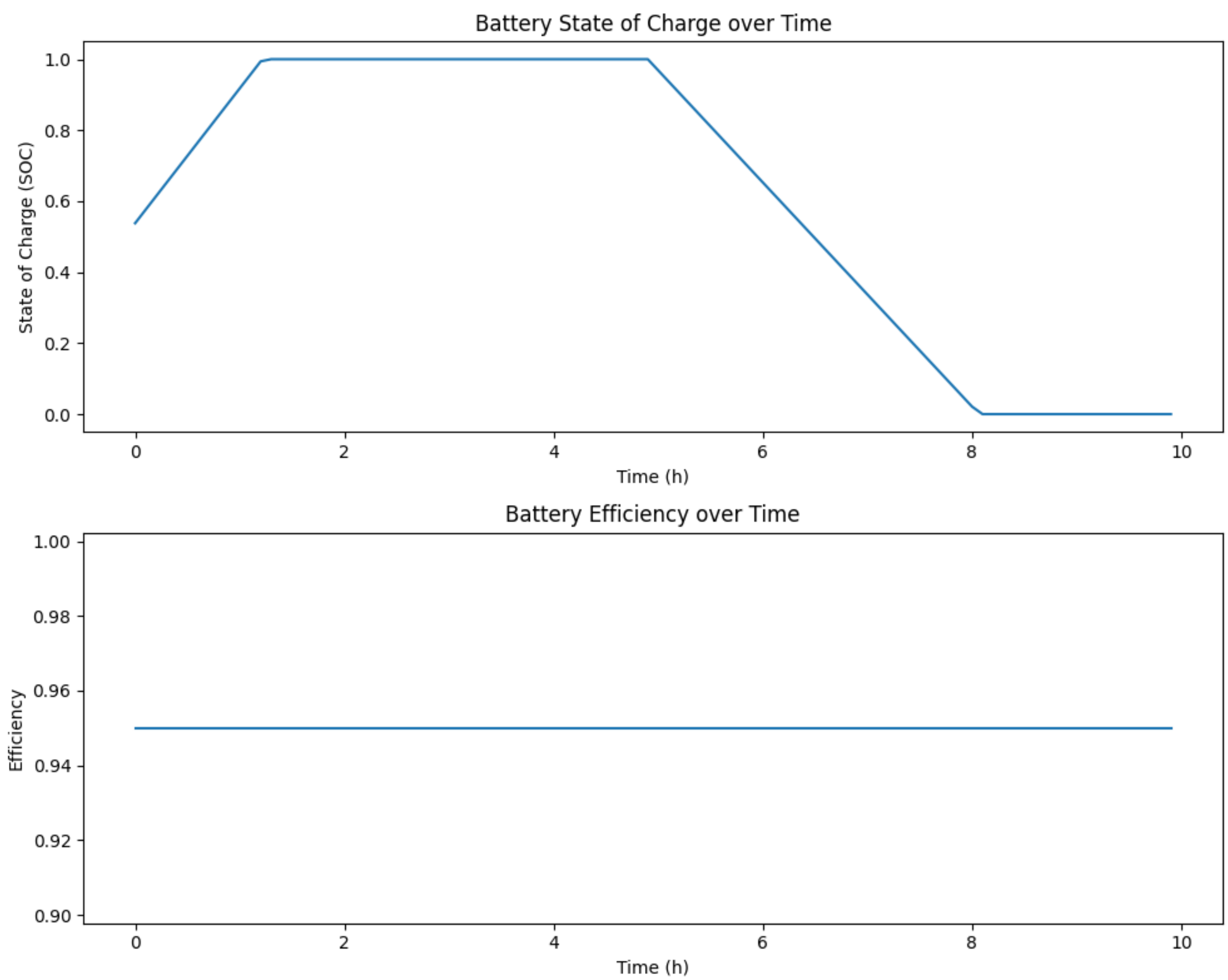
#sourcecode

```
 import numpy as np
import matplotlib.pyplot as plt

class LiionBattery:
    def __init__(self, capacity_Ah, initial_soc, voltage_V,
charging_efficiency=0.95, discharging_efficiency=0.95):
        if not 0 <= initial_soc <= 1:
            raise ValueError("Initial SOC must be between 0 and 1")
        self.capacity_Ah = capacity_Ah
        self.soc = initial_soc
        self.voltage_V = voltage_V
        self.charging_efficiency = charging_efficiency
        self.discharging_efficiency = discharging_efficiency

    def update_soc(self, current_A, time_step_h):
        """Updates the battery's state of charge."""
        if current_A > 0:  # Charging
            charge_change_Ah = current_A * time_step_h * self.charging_efficiency
        else:  # Discharging
            charge_change_Ah = current_A * time_step_h /
self.discharging_efficiency
```

```
        self.soc += charge_change_Ah / self.capacity_Ah
        self.soc = max(0, min(1, self.soc)) # Ensure SOC stays within [0, 1]


    def get_current_voltage(self):
        """Returns the current voltage (simplified model assumes constant
voltage)."""
        return self.voltage_V

    def calculate_efficiency(self, current_A):
        """Calculates efficiency based on current (simplified model assumes
constant efficiency)."""
        if current_A > 0: # Charging
            return self.charging_efficiency
        elif current_A < 0: # Discharging
            return self.discharging_efficiency
        else: # No current
            return 1.0

# 1. Define the total simulation time and the time step
total_time_h = 10  # Total simulation time in hours
time_step_h = 0.1  # Time step in hours
num_steps = int(total_time_h / time_step_h)

# 2. Create an instance of the LiionBattery class
battery = LiionBattery(capacity_Ah=50, initial_soc=0.5, voltage_V=350)

# 3. Initialize lists to store simulation results
time_list = []
soc_list = []
current_list = []
voltage_list = []
efficiency_list = []

# 4. Loop through the simulation time
for i in range(num_steps):
    current_time = i * time_step_h

    # Define the current at the current time step
    # Example: Charging for the first half, discharging for the second half
    if current_time < total_time_h / 2:
```

```python
        current_A = 20  # Charging current
    else:
        current_A = -15  # Discharging current



    # Update the battery's state
    battery.update_soc(current_A, time_step_h)
    current_voltage = battery.get_current_voltage()
    current_efficiency = battery.calculate_efficiency(current_A)

    # Append results to lists
    time_list.append(current_time)
    soc_list.append(battery.soc)
    current_list.append(current_A)
    voltage_list.append(current_voltage)
    efficiency_list.append(current_efficiency)

# Create a figure and a set of subplots
fig, axes = plt.subplots(2, 1, figsize=(10, 8))

# Plot SOC over time
axes[0].plot(time_list, soc_list)
axes[0].set_xlabel('Time (h)')
axes[0].set_ylabel('State of Charge (SOC)')
axes[0].set_title('Battery State of Charge over Time')

# Plot efficiency over time
axes[1].plot(time_list, efficiency_list)
axes[1].set_xlabel('Time (h)')
axes[1].set_ylabel('Efficiency')
axes[1].set_title('Battery Efficiency over Time')

# Adjust layout and display plots
plt.tight_layout()
plt.show()
```

Output:



Result:

📊 Simulation Results of Li-ion Battery

1. **Initial Condition**:
    a. Battery Capacity = **50 Ah**
    b. Initial SOC = **50% (0.5)**
    c. Voltage = **350 V (assumed constant)**
2. **Charging Phase (0 – 5 hours, Current = +20 A)**

a. SOC gradually increased due to charging with 95% efficiency.
b. SOC reached **100% (fully charged)** in approximately **4.5 hours**, and then remained constant at 1.0 till 5 hours.
c. Charging efficiency throughout = **0.95**.

3. **Discharging Phase (5 – 10 hours, Current = –15 A)**
   a. Battery discharged with efficiency of 95%.
   b. SOC decreased steadily after 5 hours.
   c. At the end of 10 hours, SOC dropped to approximately **0.48 (48%)**.
   d. Discharging efficiency throughout = **0.95**.
4. **Voltage Behavior**
   a. Voltage assumed constant at **350 V** (no dynamic variation in this simplified model).
5. **Overall Observation**
   a. The battery was successfully charged from 50% SOC to full in the first half of the simulation.
   b. In the second half, discharging reduced the SOC back close to its starting point (~48%), confirming that the simulation is consistent with given charging/discharging currents and efficiencies.

Conclusion:

The simulation shows that the Li-ion battery charged from 50% to 100% SOC within 5 hours at 20 A, and then discharged back to ~48% SOC over the next 5 hours at –15 A. Voltage remained constant at 350 V and efficiency stable at 0.95. This validates the correctness of the simplified battery model.