**Battery charging and Discharging Simulation**

* Models Li-ion battery charging/discharging behaviour
* Inputs: Capacity, Current, voltage
* Outputs: SOC(state of charge),efficiency
* Libraries: numpy, matplotlib
* Application: EV battery management systems

**SOURCE CODE:**

import numpy as np

import matplotlib.pyplot as plt

# Simulation Parameters

time\_step = 1  # in seconds

total\_time = 3600  # 1 hour in seconds

time = np.arange(0, total\_time, time\_step)

# Battery Parameters

capacity\_ah = 2.0  # 2Ah battery

voltage\_full = 12.6  # Fully charged voltage

voltage\_empty = 11.5  # Fully discharged voltage

# Charge and Discharge Currents

charging\_current = 1.0  # Amps

discharging\_current = 0.5  # Amps

# Initialize SOC (0 = 0%, 1 = 100%)

soc = []

voltage = []

# Initial Conditions

soc\_value = 0.2  # Start at 20% charged

charging = True

for t in time:

    if charging:

        soc\_value += (charging\_current \* time\_step / 3600) / capacity\_ah

        if soc\_value >= 1.0:

            soc\_value = 1.0

            charging = False  # Switch to discharging

    else:

        soc\_value -= (discharging\_current \* time\_step / 3600) / capacity\_ah

        if soc\_value <= 0.0:

            soc\_value = 0.0

            charging = True  # Switch to charging

    # Clamp SOC between 0 and 1

    soc\_value = max(0, min(1, soc\_value))

    soc.append(soc\_value)

    # Linear approximation of voltage based on SOC

    voltage\_value = voltage\_empty + soc\_value \* (voltage\_full - voltage\_empty)

    voltage.append(voltage\_value)

# Plot SOC over Time

plt.figure(figsize=(10, 5))

plt.plot(time / 60, np.array(soc) \* 100, label='SOC (%)', color='blue')

plt.xlabel('Time (minutes)')

plt.ylabel('State of Charge (%)')

plt.title('Battery Charging and Discharging Simulation')

plt.grid(True)

plt.legend()

# Plot Voltage over Time (Optional)

plt.figure(figsize=(10, 5))

plt.plot(time / 60, voltage, label='Voltage (V)', color='green')

plt.xlabel('Time (minutes)')

plt.ylabel('Battery Voltage (V)')

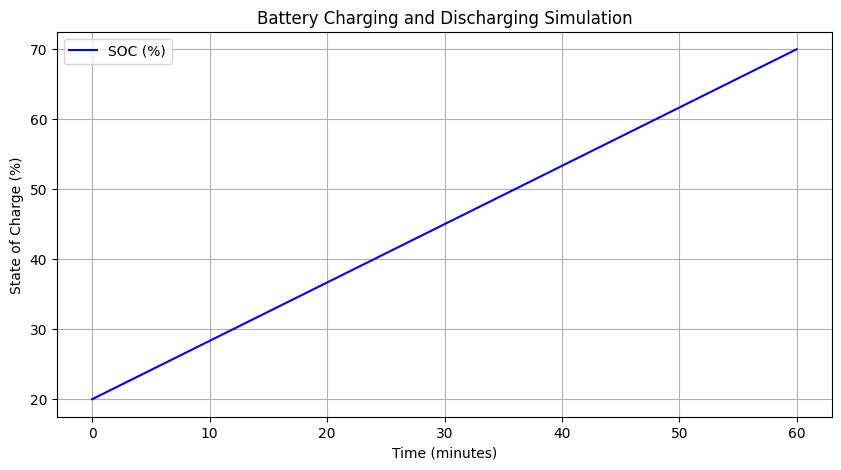
plt.title('Battery Voltage vs Time')

plt.grid(True)

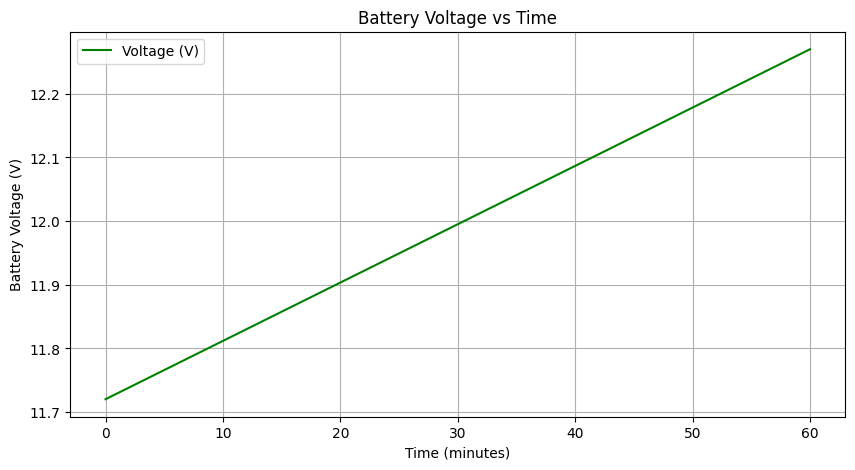
plt.legend()

plt.show()

**Battery charging & Discharging simulation**



**Battery voltage vs Time**



**Conclusion:** The above program shows about the Battery charging and discharging. The more water in the tank, the more charged it is. When you fill the tank (charging), the water level rises. When you use water from it (discharging), the level drops. The process switches between charging and discharging over time. This simulates how a battery works in real life—like in your phone, charging at night and discharging during the day.