Battery Charging &amp; Discharging Simulation

Information:

 Models Li-ion battery charging/discharging behavior.

 Inputs : Capacity , current , voltage

 Outputs : SOC(State of charge),efficiency

 Libraries : numpy , matplotlib

 Application : EV battery management systems

Source code:

#Battery charging&amp;Discharging Simulation

import numpy as np

import matplotlib.pyplot as plt

capacity\_Ah = 60.0

voltage\_nominal = 3.7

internal\_resistance = 0.05

total\_time = 3600

dt = 1

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

current = np.zeros\_like(time)

voltage = np.zeros\_like(time)

current[:1800] = 12

voltage[:1800] = voltage\_nominal + 0.1

current[1800:] = -20

voltage[1800:] = voltage\_nominal - 0.15

soc = np.zeros\_like(time, dtype=float)

efficiency = np.zeros\_like(time, dtype=float)

soc[0] = 40.0

for t in range(1, len(time)):

    I = current[t]

    V = voltage[t]

    delta\_soc = (I \* dt / 3600) / capacity\_Ah \* 100

    soc[t] = soc[t - 1] + delta\_soc

    soc[t] = np.clip(soc[t], 0, 100)

    if I &gt; 0:

        power\_in = I \* V

        power\_stored = I \* voltage\_nominal

        eff = power\_stored / power\_in if power\_in &gt; 0 else 1

    elif I &lt; 0:

        power\_out = -I \* voltage\_nominal

        power\_drawn = -I \* V

        eff = power\_out / power\_drawn if power\_drawn &gt; 0 else 1

    else:

        eff = 1.0

    efficiency[t] = np.clip(eff, 0, 1)

print(&quot;------ Battery Simulation Results ------&quot;)

print(f&quot;Initial SOC: {soc[0]:.2f}%&quot;)

print(f&quot;SOC after 30 minutes (charging): {soc[1800]:.2f}%&quot;)

print(f&quot;SOC after 60 minutes (discharging): {soc[-1]:.2f}%\n&quot;)

print(f&quot;Efficiency at 30 minutes: {efficiency[1800]:.3f}&quot;)

print(f&quot;Efficiency at 60 minutes: {efficiency[-1]:.3f}&quot;)

print(&quot;----------------------------------------&quot;)

Output:

------ Battery Simulation Results ------

Initial SOC: 40.00%

SOC after 30 minutes (charging): 49.99%

SOC after 60 minutes (discharging): 33.33%

Efficiency at 30 minutes: 1.000

Efficiency at 60 minutes: 1.000

Conclusion:

This program shows how a Li-ion battery charges and discharges using information

like capacity, current, and voltage. It calculates the battery charge level (SOC) by

keeping track of the current flow over time. The efficiency is estimated by comparing

the power going into or coming out of the battery. The results show that the charge

level goes up when charging and goes down when discharging, while efficiency

changes depending on energy loss. This easy-to-understand model can help manage

electric vehicle batteries better by monitoring their condition and improving how they

are used.