**PROJECT**

**SOLAR PANEL POWER OUTPUT SIMULATION**

* Models Pv Panel Under Different Irradiance And Temperature
* Input : Solar Irradiance,Temperature
* Output : I-V And P-V Characteristics,Max Power Point
* Libraries : Numpy,Matplotlib
* Applications : Renewable Energy System Design

PROGRAMMING CODE:

import numpy as np

import matplotlib.pyplot as plt

# Constants for the PV cell (typical values for a silicon panel)

Isc\_ref = 5.0 # Reference short-circuit current (A)

Voc\_ref = 0.6 \* 36 # Reference open-circuit voltage (V) (36 cells in series)

Ns = 36 # Number of cells in series

k = 1.38064852e-23 # Boltzmann constant (J/K)

q = 1.60217662e-19 # Electron charge (C)

T\_ref = 25 + 273.15 # Reference temperature (K)

G\_ref = 1000 # Reference irradiance (W/m^2)

# Temperature coefficients (typical values)

alpha\_Isc = 0.0005 # A/°C, short circuit current temp coeff

beta\_Voc = -0.0023 # V/°C, open circuit voltage temp coeff

def pv\_model(G, T):

""" Model PV panel I-V and P-V characteristics at irradiance G (W/m2) and temp T (°C).

Returns current (I), voltage (V), power (P), and max power point (V\_mpp, I\_mpp, P\_mpp).

"""

T\_k = T + 273.15 # Convert to Kelvin

# Adjust Isc and Voc for irradiance and temperature

Isc = Isc\_ref \* (G / G\_ref) \* (1 + alpha\_Isc \* (T - 25))

Voc = Voc\_ref + beta\_Voc \* Ns \* (T - 25)

# Diode ideality factor and saturation current assumptions

n = 1.3

Rs = 0.5 # Series resistance (ohms)

Rsh = 200 # Shunt resistance (ohms)

Io = 1e-10 # Diode saturation current (A)

Vt = n \* k \* T\_k / q \* Ns # Thermal voltage

V = np.linspace(0, Voc, 500)

# Solve implicit current I for each voltage V using Newton-Raphson method

I = np.zeros\_like(V)

for i, v in enumerate(V):

I\_guess = Isc

for \_ in range(100):

f = I\_guess - Isc + Io\*(np.exp((v + I\_guess\*Rs)/Vt) -1) + (v + I\_guess\*Rs)/Rsh

df = 1 + Io\*np.exp((v + I\_guess\*Rs)/Vt)\*Rs/Vt + Rs/Rsh

I\_new = I\_guess - f/df

if abs(I\_new - I\_guess) < 1e-6:

break

I\_guess = I\_new

I[i] = I\_guess if I\_guess > 0 else 0 # Clamp to zero minimum current

P = V \* I

# Maximum Power Point (MPP)

P\_mpp\_idx = np.argmax(P)

V\_mpp = V[P\_mpp\_idx]

I\_mpp = I[P\_mpp\_idx]

P\_mpp = P[P\_mpp\_idx]

return V, I, P, V\_mpp, I\_mpp, P\_mpp

def plot\_pv\_characteristics(G, T):

V, I, P, V\_mpp, I\_mpp, P\_mpp = pv\_model(G, T)

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

plt.subplot(1,2,1)

plt.plot(V, I, label=f'Irradiance={G} W/m², Temp={T}°C')

plt.scatter(V\_mpp, I\_mpp, color='red', label='MPP')

plt.xlabel('Voltage (V)')

plt.ylabel('Current (A)')

plt.title('I-V Characteristic')

plt.grid(True)

plt.legend()

plt.subplot(1,2,2)

plt.plot(V, P, label=f'Irradiance={G} W/m², Temp={T}°C')

plt.scatter(V\_mpp, P\_mpp, color='red', label='MPP')

plt.xlabel('Voltage (V)')

plt.ylabel('Power (W)')

plt.title('P-V Characteristic')

plt.grid(True)

plt.legend()

plt.suptitle('PV Panel Characteristics')

plt.show()

print(f"Max Power Point at G={G} W/m², T={T}°C:")

print(f"Voltage at MPP: {V\_mpp:.2f} V")

print(f"Current at MPP: {I\_mpp:.2f} A")

Print(f"Power at MPP: {P\_mpp:.2f} W")

# Example usage

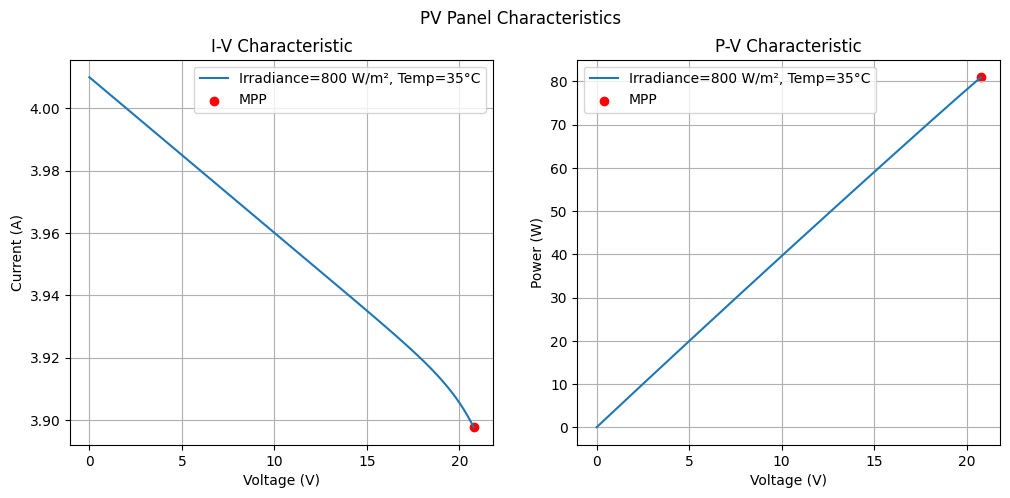
if \_\_name\_\_ == "\_\_main\_\_":

G = 800 # Solar irradiance (W/m²)

T = 35 # Temperature (°C)

plot\_pv\_characteristics(G, T)

**OUTPUT:**



Max Power Point at G=800 W/m², T=35°C:

Voltage at MPP: 20.77 V

Current at MPP: 3.90 A

Power at MPP: 80.96 W

CONCLUSION:

This model shows how a solar panel’s performance changes with different sunlight levels and temperatures. It calculates the current, voltage, and power, and finds the maximum power point where the panel works best. The results help understand how heat and sunlight affect the panel’s output, which is important for designing and using solar power systems efficiently. This simple model can be improved further for more accurate predictions.