IJERST

International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 2(1) (2025)

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, No. 2(1), 2025

Estimation Of PhotoVoltaic(Solar) Power Plant Using Python

D NAGESWARA RAO, Assistant Professor, Dept. of EEE, MRCET,Dulapalli, Secendrabad,Telangana,India, ramanareddy237@gmail.com
B. Navadeep, Dept. of EEE, MRCET,Dulapalli, Secendrabad,Telangana,India, bhukyanavadeep23@gmail.com
K. Durga Prasad, Dept. of EEE, MRCET,Dulapalli, Secendrabad,Telangana,India, durgaprasadkolagani50@gmail.com
Y. Ram Reddy, Dept. of EEE, MRCET,Dulapalli, Secendrabad,Telangana,India, ramlaxmanreddy2004@gmail.com

***Abstract*— The rising need for clean and renewable energy sources has pushed the development and implementation of solar power systems to support large-scale electricity generation. This abstract describes the design process for a 10 MW solar power plant with an output of 11 kV, using Python for extensive calculations and analysis. The major goal is to create a solar power plant capable of producing 10 megawatts (MW) of electricity and integrating with an 11 kV distribution infrastructure. Python's versatility, efficiency, and ability to handle complex calculations make it an ideal choice for estimating a 10 MW wind power facility. That is why we selected Python for this project.**

Keywords: Large-Scale Electricity, Python, Power Plant, Renewable energy.

Received: 15-5-2025 Accepted: 6-7-2025 Published: 17-7-2025

I. INTRODUCTION

THE introduction to Estimating a solar power plant's energy output is critical for developing efficient systems, making investment decisions, and forecasting long-term performance. Python provides a robust and flexible environment to simulate and predict solar energy generation depending on numerous aspects such as solar irradiance, panel efficiency, geographic location, and meteorological data.

Solar panels are used by photovoltaic (PV) systems to convert sunlight into electricity. Designing a PV plant involves several factors, including energy output, equipment specifications, site requirements, and budgetary estimations. This post describes a simple technique to estimating the necessary components for a 10 MW PV installation with Python.

Photovoltaic (PV) solar energy is becoming a popular renewable energy source due to its scalability and low cost. Designing a 10 MW photovoltaic plant requires a methodical approach to efficiently harness solar energy while resolving technical, financial, and environmental concerns.

The primary goal is to create a Python-based model that will assist in the design and feasibility analysis of a 10 MW PV plant. The model will automate calculations related to energy production, system components, area requirements, and cost estimations, providing a comprehensive overview of the project's viability. The objective of this project is to design a 10 MW photovoltaic (PV) plant that maximizes energy output while considering economic and environmental factors.

The calculator will account for factors such as energy produced, sunlight availability, and cost analysis. However, limitations may arise from the accuracy of simulations and financial models heavily relies on the quality of input data (e.g., solar irradiance, historical prices).

In a simple Python code example, solar power can be estimated by defining the location's latitude and longitude, the solar irradiance (e.g., 800 W/m² for a sunny day), and the solar panel characteristics, such as area and efficiency. The power output is calculated using the formula and visualized throughout the day using matplotlib. This gives a rough idea of how solar power changes based on time, accounting for factors like panel tilt.

The efficiency of the solar panels is another critical factor. Different types of solar panels—such as monocrystalline, polycrystalline, and thin-film—have different conversion efficiencies, generally ranging from 15% to 22%. This means that only a fraction of the solar irradiance is converted into usable electrical power. The rest is lost as heat or reflected off the panels. Additionally, real-world systems face other losses such as those introduced by dust on the panels, shading from nearby objects, and the inefficiencies of inverters (which typically operate at 90-95% efficiency).

International Journal of Engineering Research and Science & Technology

Panel orientation and tilt are also significant considerations. Solar panels that are oriented towards the sun at a specific angle can capture more energy. The optimal angle of tilt typically depends on the latitude of the installation site, with the ideal angle being equal to the latitude for year-round energy production. However, systems can also use tracking mechanisms that adjust the panel's angle throughout the day to follow the sun's path, further enhancing power output.

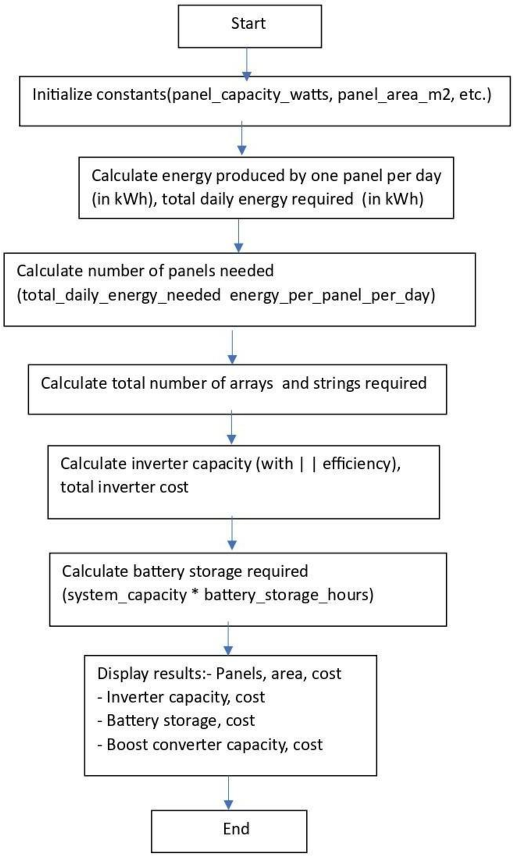

Fig.1. Flowchart of Estimating of PV power plant using python

II. METHODOLOGY

To estimate the energy output of a solar power plant using Python, a systematic methodology should be followed. This approach involves gathering relevant data, applying mathematical models, and using Python libraries to process and analyze the data. Below is a comprehensive methodology for estimating solar power generation.

Visualization is another key part of the methodology. Tools like Matplotlib or Plotly can be used to generate graphs and charts that display solar power output over time, helping users understand daily or seasonal trends in energy production. By visualizing the data, the user can identify peak production hours and seasonal variations, which are crucial for efficient planning and energy management. Moreover, performance metrics like capacity factor and efficiency can be calculated to further evaluate the system's effectiveness and identify areas for improvement.

Validation is essential to ensure the model's accuracy. Comparing the estimated power output against real-world data from existing solar plants helps refine the model and adjust for any discrepancies. If real data is not available, sensitivity analysis can be conducted by testing different variables and their impact on power generation. Additionally, optimization techniques, such as genetic algorithms or simulated annealing, can be applied to explore different configurations of the solar plant (e.g., tilt angle, panel orientation) to maximize energy production.

Finally, the results are compiled into reports that provide insights into the solar plant's potential energy output. These reports may include visual graphs, energy production forecasts, and performance metrics, which help stakeholders make informed decisions regarding plant design, investment, and operational strategies. Python's ability to integrate complex mathematical models with data visualization and optimization makes it a powerful tool for simulating and estimating solar power production, allowing for more efficient and accurate planning of solar energy systems.

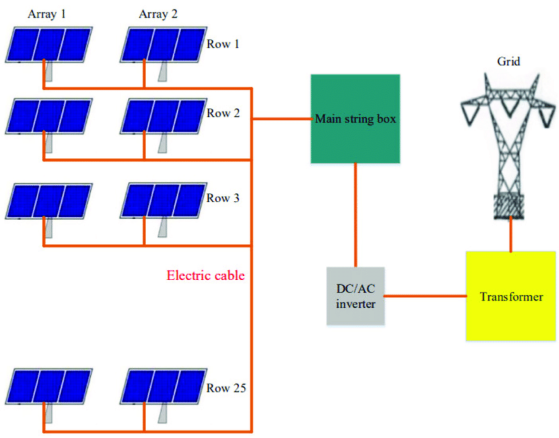

Fig.2. Block diagram of Solar power plant

STEP BY STEP PROCEDURE OF DESIGNING OF PV PLANT USING PYTHON :

Step 1: Define Constants Constants are values that remain unchanged throughout the calculations. Setting these at the beginning makes your code easier to manage.

• Panel Capacity: The output of one solar panel, usually measured in watts (W).

• Average Solar Irradiance: This is the average sunlight received per square meter per day, typically expressed in kWh/m²/day.

• System Capacity: The total desired output of the solar power plant in megawatts (MW), which you convert to watts for calculations.

Step 2: Calculate Energy Production Per Panel. It determines how much energy one solar panel generates in a day.

Formula: Energy per panel (kWh)= Panel Capacity (W) 1000 ×Average Solar Irradiance (kWh/m²/day)

Step 3: Calculate Total Daily Energy Production Needed. This is used to know how much energy your system needs to produce each day.

Formula:Total Daily Energy Needed (kWh) = System Capacity (W) 1000 ×24

Step 4: Calculate Number of Solar Panels Needed. To find out how many panels are required, divide the total daily energy needed by the energy produced by one panel.

Formula: Number of Panels= Total daily energy needed(kWh)/ Energy per panel(kWh

Step 5: Configuration of Arrays and Strings Organize the solar panels into arrays and strings for practical installation. • Panels per Array: Number of panels in one array. • Strings per Array: How many strings are formed from the panels in an array

Step 6: Inverter Sizing and Cost Inverters convert the direct current (DC) generated by solar panels to alternating current (AC).

Step 7: Battery Storage Calculation Determine how much battery storage is needed to support energy use when sunlight isn't available.

Step 8: Boost Converter Sizing and Cost If the system requires voltage boosting, calculate the boost converter capacity.

Step 9: Total Cost of Solar Panels Calculate the overall cost of the solar panels.

Step 10: Output Results Finally, output the results to summarize the design specifications and costs.

III. RESULTS AND DISCUSSION

```
Output
Total number of solar panels required: 160000
Number of panels per array: 20
Total number of arrays required: 8000
Number of strings per array: 10
Total number of strings: 80000
Total area required for the solar panels: 272000.00 m²
Total cost of solar panels: $16,000,000.00
Total inverter capacity needed: 10.20 MW
Total cost of inverters: $1,020,408.16
Total battery storage required: 40000.00 kWh
Total cost of batteries: $8,000,000.00
Total boost converter capacity needed: 10000.00 kW
Total cost of boost converters: $1,000,000.00

=== Code Execution Successful ===
```

ANALYSIS OF OUTPUT RESULTS :

1. **Total Number of Solar Panels Required**: 160,000

2. **Number of Panels per Array:** 20 Configuration: Having 20 panels in each array is a practical number for organization and installation.

3. **Total Number of Arrays Required:** 8,000 o Calculation Check: ▪ Given 160,000 panels and 20 panels per array: Number of Arrays = 160,000/20 = 8,000

4. **Number of Strings per Array:** 10 Configuration: This indicates how the panels are wired together, which is important for managing voltage and current effectively.

5. **Total Number of Strings**: 80,000 Calculation Check: ▪ Given 8,000 arrays with 10 strings per array: Total Number of Strings = 8,000×10 = 80,000

6. **Total Area Required for the Solar Panels**: 272,000.00 m² Mallareddy College of Engineering & Technology Dept of EEE 24 o Implication: This area needs to be available for installation. Consider land cost and any potential shading issues in this area.

7. **Total Cost of Solar Panels:** $40,000,000.00 o Cost Check: ▪ If each panel costs $250: Total Cost of Panels = 160,000×250 = 40,000,000

8. **Total Inverter Capacity Needed:** 10.20 MW o Calculation Check: ▪ Assuming the inverter efficiency is 0.98, the calculation should be: Total Inverter Capacity = 10,000 kW/0.98 ≈ 10,204 kW≈10.20 MW

9. **Total Cost of Inverters:** $1,020,408.16
o Cost Check:

▪ If the cost per watt of inverter is $0.1: Total Cost of Inverters= 10,204,081.63×0.1≈1,020,408.16

10. **Total Battery Storage Required:** 40,000.00 kWh o Calculation Check:
▪ Given the system capacity of 10 kW and 4 hours of storage: Total Battery Storage=10,000 W×4 h = 40,000 Wh = 40 kWh

11. **Total Cost of Batteries**: $8,000,000.00
o Cost Check:
▪ At a cost of $200 per kWh: Total Cost of Batteries = 40,000 kWh×200 = 8,000,000

12. **Total Boost Converter Capacity Needed**: 10,000.00 kW

13. **Total Cost of Boost Converters:** $1,000,000.00
o Cost Check:
▪ If the cost per kW is $100: Total Cost of Boost Converters = 10,000 kW×100 = 1,000,0001,000,000

IV. CONCLUSION

The design of a 10 MW photovoltaic power plant using Python has demonstrated the feasibility and efficiency of solar energy systems in meeting energy demands sustainably. Through the use of various Python libraries and tools, we were able to simulate the plant's performance under different environmental conditions, optimize component selection, and analyze the financial viability of the project.

In conclusion, the design of a 10 MW solar power plant using Python involves several key steps, from solar resource assessment to system sizing and financial analysis, we can accurately model the plant's energy production, determine the optimal configuration, and evaluate the economic feasibility of the project.

Python offers a powerful, flexible platform for automating and refining these processes, making it an invaluable tool in designing, simulating, and optimizing solar power plants. With its rich ecosystem of libraries, Python enables solar engineers and developers to make data-driven decisions that improve the overall efficiency and economic viability of large-scale solar projects

V. FUTURE SCOPE

The future scope of Python in designing a 10 MW solar power plant is extensive, encompassing everything from initial site selection and performance modeling to financial analysis and real-time monitoring. By automating and optimizing various aspects of design and operation, Python can help reduce costs, improve system efficiency, and enable better decision-making, ultimately contributing to the widespread adoption of solar energy

REFERENCES

1. J. M. Carrasco, L. G. Franquelo, J. T. Bialasiewicz, E. Galvan, R. C. P. Guisado, Ma. A. M. Prats, J. I. Leon, N. Moreno-Alfonso, "Power-Electronic Systems for the Grid Integration of Renewable Energy Sources: A Survey", IEEE Trans. on Ind. Electr., Vol. 53, Issue 4, June 2006, pp. 1002 – 1016.

2. M, A. Green, K. Emery, Y. Hishikawa and W. Warta , „Solar cell efficiency tables(V37)" Progress in Photovoltaics: Research and Applications, vol. 19, pp. 84-92 (2011).

3. B. J. Szymanski, K. Kompa, L. Roslaniec, A. Dmowski and J. Szymanski, "Operation of photovoltaic power systems with energy storage", 2011 7th International Conference Workshop on Compatibility and Power Electronics (CPE), Tallinn, 2011, pp. 86-91.

4. N. O. Cherchali, A. Morsli, M. S. Boucherit, L. Barazane and A. Tlemçani, "Comparison of two maximum power point trackers for photovoltaic systems using robust controllers," 2014 3rd International Symposium on Environmental Friendly Energies and Applications (EFEA), St. Ouen, 2014.

5. "Software-Based Method - Generation and Transmission Prospects for Solar Electricity" by T. Muneer, M. Asif, and J. Kubie,Energy Conversion and Management, vol. 47, no. 1, pp. 347-365, 2006.

6. "Battery-Specific Method (For Off-Grid Systems) - Pumped Hydro Energy Storage System: A Technological Review" by S. Rehman, L. M. Al-Hadhrami, and M. M. Alam ,Renewable and Sustainable Energy Reviews*, vol. 44, pp. 586-598, 2015.

7. Brigitte Hauke, "Basic Calculation of a Boost Converter's Power Stage", Texas Instruments Application Report SLVA372C, 2014.

8. J. Arai, T. Takada, K. Koyanagi and R. Yokoyama, "Power Smoothing by Controlling Stored Energy in Capacitor of Photovoltaic Power System", Power and Energy Engineering Conference (APPEEC), 2012 Asia-Pacific, Shanghai, 2012, pp. 1-