Project Instructions

Operational research for urban solar development

“PV failure detection based on operational time series”

Alexandre MATHIEU, PhD student

Project summary:

The project aims at modeling a PV system and detecting underperformances to, eventually, correct them.

The first objective relies on the estimation of the operating variables such as the POA irradiance, module temperature, DC and AC power from satellite data while the second objective is to quantify the losses from five underperformances categorized as inverter failure, inverter clipping, shading, string open-circuit and module short-circuit.

Everyone is going to be associated to a student number “n” for the project.

Deadline: 13/01/2025 at 11h15

If late (less than 15mins late): - 10%
If late (less than 1h late): - 50%
If more than 1h late: -100%

Deliverables:

- One jupyter notebook as html file.
- 7 csv files with DateTime index starting from the 01/07/2020 up to the 01/01/2022 (excluded)
    - 1 for the part A
    - 6 for the part B

All those files are to be sent by email through a zip file on the 13/01/2025 at 11h15 at the latest to alexandre.mathieu@heliocity.io (make sure it is received !)

You can export a pd.DataFrame into csv files with the “to_csv()” command.

Input datasets of the project:

Some datasets are imported in the notebook according to your student number.

- "input_data": Data to use as inputs for models with irradiance data, air temperature, wind_speed and sun_path.
- "pv_data": Data to compare with the outputs of the models. It contains the irradiance in the plane of array, module temperature, DC power and AC power.
- "pv_data_invf": Data which includes inverter failure effect.
- "pv_data_clp": Data which includes the inverter clipping effect.
- "pv_data_sd": Data which includes the shading effect.
- "pv_data_oc": Data which includes the string open-circuit.
- "pv_data_sc": Data which includes the module short-circuit effect.
- "pv_data_all": Data with all failures combined.

The six last datasets contain the operating variables with the underperformance effects included on idc, vdc, pdc and pac. Idc, Vdc and Pdc are the current, voltage and power at the input of the inverter. Pac refers to the AC power.

PV Installation data:

- Location: Sophia Antipolis, France (Lat 43.61 N , Long 7.05 W , Altitude 150m)

- Installation orientation: 5° tilt, 180° azimuth (South)

- Albedo can be assumed to be equal to 0.2

- 120 modules: 6 strings in parallel of 20 modules connected in series.

- Module nominal power: 300 W

datasheet: https://www.aeet-energy.com/pdf/q-cells/Hanwha_Q_CELLS_Datenblatt_QPEAK_DUO_BLK-G5_300-320_2018-10_Rev03_DE.pdf

- Inverter datasheet: SUN2000-30KTL-M3 (30 kW capacity)

https://solar.huawei.com/en-GB/download?p=%2F-%2Fmedia%2FSolar%2Fattachment%2Fpdf%2Ffr%2Fdatasheet%2FSUN2000-30-40KTL-M3.pdf

Tasks

A. AC Power modelling

1. Estimate the irradiance in the plane of the array.
2. Estimate the module temperature.
3. Estimate the DC power.
4. Estimate AC power.
5. Calculate the performance of the installation at the AC level through two indicators for ***the whole period***:
    a. Performance Ratio
    b. Energy Performance Index

If you develop some ***ML*** models, only use the data from 2020 to train your models !

Expected outputs:

- 1 CSV file with 4 columns [“gpoa_estimated”, “t_mod_estimated”, “dc_power_estimated”, “ac_power_estimated”] and Datetime index with the estimated POA irradiance [W], module temperature [°C], DC power [W] and AC power [W].
  The csv file should be named “student_n_acestimation.csv”, with “n” your student number.
- The notebook with the explained approach and the performance metrics (PR, EPI) calculated and printed.

B. Underperformance detection

Detect when the following underperformances occur and quantify the losses:

1. Inverter shutdown.
2. Inverter clipping.
3. Shading.
4. String open-circuit.
5. Short-circuited module failure.
6. ***Bonus***: All failures combined – to dissociate.

Expected outputs

- 1 csv file for each underperformance which corresponds to a pandas DataFrame (one column) with its values which are equal to the energy losses (Wh) from the underperformance. The column name should be “losses”.
  Each file should be named with the following convention with "n" your student number:
    - student_n_invshutdown.csv
    - student_n_invclip.csv
    - student_n_shading.csv
    - student_n_opencirc.csv
    - student_n_modulesc.csv
    - student_n_allfailures.csv

  For the 6., the csv file should contain 5 columns, 1 with the energy losses in Wh for each underperformance.

Grading scale

Score distribution:

- 45% A. AC power modeling
  - 10% for each task for the first 4 tasks.
  - 5% for the last 5th task.
  - For the first 4 tasks, the following points will be assessed:
    - Relevance of the estimation method
    - RMSE accuracy of the estimation
      (3/5 for the accuracy is the same as the most basic method
      5/5 if the accuracy is the same as the teacher's method)
  - For the 5th task: The correctness of the calculation is assessed.

- 50% B. Failure detection
  - 10% for each failure, the following points will be assessed:
    - Relevance and originality of the approach
    - Accuracy of the energy loss estimation (Wh)
      (3/5 if the accuracy is the same as the most basic method
      5/5 if the accuracy is the same as the teacher's method)
- 5% for the overall structure and the ease of reviewing the notebook.
- ***Bonus*** 10%: All failure signals are combined together.
  Dissociate them and estimate their energy losses separately, the following points will be assessed:
  - Relevance and originality of the approach
  - Accuracy of the estimation
    (3/5 if the accuracy is the same as the most basic method
    5/5 if the accuracy is the same as the teacher's method)

Potential bonus/penalty points:

- Original approaches.
- Better accuracy than the teacher's benchmark.
- Illustrations to explain the approach/the results.

Advice:

- Refer to the course formulas.
- Make concise explanations.
- Illustrate your approach with some relevant plots.
- Comments to explain your code with "#"
  For example,

```
# Check when the inverter efficiency is under 60%
filter = (ac_power / dc_power) < 0.6
ac_power.loc[filter]

# Consider that a failure occurs when the inverter efficiency < 60% and quantify the losses
loss = ac_power_estimated.loc[filter] - ac_power.loc[filter]
print(loss.sum())
```