SAND2015-9170R

Database Performance Monitoring for the Photovoltaic Systems

Contents

Katherine A. Klise
Sandia National Laboratories
PO Box 5800 MS 0751
Albuquerque, NM 87185-0751

ph: 505-284-4456

email: kaklise@sandia.gov

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Overview

The Database Performance Monitoring (DPM) software (**copyright in processes**) is being developed at Sandia National Laboratories to perform quality control analysis on time series data. The software loads time indexed databases (currently csv format), performs a series of quality control tests defined by the user, and creates reports which include summary statistics, tables, and graphics. DPM can be setup to run on an automated schedule defined by the user. For example, the software can be run once per day to analyze data collected on the previous day. HTML formatted reports can be sent via email or hosted on a website. To compare performance of several databases, summary statistics and graphics can be gathered in a dashboard view which links to detailed reporting information for each database. The software can be customized for specific applications.

Installation

DPM is a python package which requires Python 2.7 along with several python package dependencies, including:

- Pandas: analysis and storage of time series data, https://pandas.pydata.org
- Numpy: support large, multi-dimensional arrays and matrices, https://numpy.org
- Matplotlib: produce figures, https://matplotlib.org
- pyyaml: configuration file format, https://pyyaml.org

The software has a simple API and can be easily customized. The following guide outlines the use of DPM for PV applications. The data objects are compatible with pvlib, which can be used to model PV systems (https://github.com/pvlib/pvlib-python).

Configuration File

The DPM configuration file includes information about the database and defines quality control tests. The configuration file includes the following information.

General information

General information includes the database name, file format, directory, timestamp index column, and expected timestamp frequency. Multiple databases can be combined in a single DPM analysis. For example, DPM analysis can combine data from PV and weather databases.

```
General:
  Name: System1
  Database: [PV_database, Weather_database] # Root name for database(s)
  Date Format: _%Y_%m_%d
  File Format: TOA5
  File Directory: C:\DatabaseFiles
  Index Column: TIMESTAMP
  Frequency: 60                  # Expected timestamp frequency, seconds
```

Specifications

Specification includes application specific constants. For PV systems, specifications can include latitude, longitude, elevation, UTC offset, Isc, Imp, Voc, Vmp, Ns, Np. These values can be used to define upper and lower bounds in quality control tests, or as constants in composite signals (explained below).

```
Specifications:
  Latitude: '35.1107'            # Latitude in decimal degrees (+N), string
  Longitude: '-106.610'          # Longitude in decimal degrees (+E), string
  ...
```

Translation

The translation dictionary maps database column names into common names. The translation dictionary can also be used to group columns with similar properties into a single variable. There is one translation dictionary per database listed in [General][Database]. Each entry is a key:value pair where 'value' is a list of column names in the database. For example, {temp: [temp1,temp2]} means that columns named 'temp1' and 'temp2' in the database file are assigned to the key 'temp' in DPM. Keys defined in the translation dictionary can be used in composite signals and quality control tests. Database columns not included in the translation dictionary are not used in the DPM analysis. For PV systems, grouping variables is useful when systems contain multiple strings and modules.

```
Translation:
  PV_database:
    Module Temperature 1: [Unit1Temperature1_Avg, Unit1Temperature2_Avg,...]
    DC Current 1: [U1S1_DCAmps_Avg, U1S2_DCAmps_Avg]
    DC Current 2: [U2S1_DCAmps_Avg, U2S2_DCAmps_Avg]
    DC Current 3: [U3S1_DCAmps_Avg, U3S2_DCAmps_Avg]
    DC Voltage: [U1_DCVolt_Avg, U2_DCVolt_Avg, U3_DCVolt_Avg]
    AC Power: [ACWattsTotal1_Avg, ACWattsTotal2_Avg, ACWattsTotal3_Avg]
    ...
```

Composite Signals

Composite signals are used to generate new data columns based on existing data. Composite signals can be used to add modeled data values or relationships between data columns. Data created from composite signals can be used in the quality control tests. Composite signals are defined using strings of python code, to be evaluated in the order specified in the configuration file. Numpy and pvlib (and other python modules if needed) can be used for computation. Examples are shown below**.** Using a PV database, DC current can be summed over each string, DC power can be computed based on DC current and voltage, and inverter efficiency can be computed from AC power and DC power. Using a weather database, GHI can be computed from DNI and DHI and the relative error can be computed. Code strings should be tested. If the composite signal fails, the error message is printed in the final report.

```
Composite Signals:
- DC Current: "np.sum([DC Current 1],axis=1), np.sum([DC Current 2],axis=1),
np.sum([DC Current 3],axis=1)"
- DC Power: "np.multiply([DC Current],[DC Voltage])"
- Inverter Efficiency: "np.divide([AC Power],[DC Power])"
```

```
Composite Signals:
```

```
- GHI from DHI DNI: np.add([DHI],np.multiply([DNI],np.cos(np.deg2rad(90-[SunEl]))))
- GHI Relative Error: "np.abs(np.divide(np.subtract([GHI], [GHI from DHI DNI]),
[GHI]))"
```

Time Filter

A time filter isolates timestamps to be used in the quality control analysis. The time filter is a special type of composite signal applied to all data columns. For PV systems, the time filter can be defined based on the sun position. Sun position is computed for the specific location and day using pvlib. Complex time filters can be used to specify different sun positions in the morning and evening. The time filter can be relaxed to avoid quality control test failures in the early morning and late evening.

```
Time Filter: "[SunEl] > 5"
```

Quality Control Tests

Quality controls tests fall into five categories: timestamp, corrupt data, missing data, zero deviation, and expected range.

1. **Timestamp test.** When DPM reads a database, it recognizes and converts a wide range of timestamp strings to datetime indexes. The time of missing, duplicate, and non-monotonic indexes are included in the summary table. To activate the timestamp test, set the “Check Timestamp” flag to true in the configuration file.

```
Check Timestamp: True
```

2. **Corrupt data test**. Databases often have a default value for corrupt values. Multiple values can be specified in the configuration file. Corrupt data points are included in the summary table. To activate the corrupt data test, include values in the “Check Corrupt” list.

```
Check Corrupt: [-999]
```

3. **Missing data test.** Missing data points are included in the summary table. To activate the missing data test, set the “Check Missing” flag to true in the configuration file.

```
Check Missing: True
```

4. **Zero deviation test.** Data that has zero deviation over the analysis time period is included in the summary table and graphics. Specific variables can be listed for the zero deviation check as shown below. Data is that is expected to have the same value throughout the day should not be included in this test.

```
Check Deviation: [Temperature, DC Voltage, DC Current, DC Power, ...]
```

5. **Range test.** Range tests are very flexible. They can be used to test if data falls within the expected range, but they can also be combined with composite signals to test if the absolute error or relative error falls within an expected range. An upper bound, lower bound or both can be specified. Upper

and lower bounds can be defined using values in the specifications section of the configuration file. Data points that fall outside the expected range are included in the summary table and graphics.

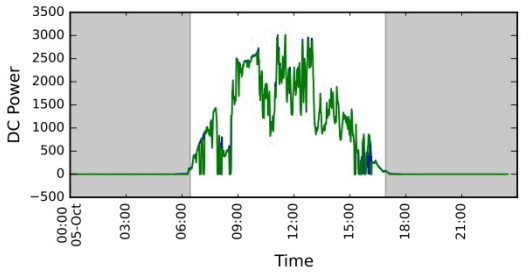
```
Check Range:
  Ambient Temperature: [-20,45]
  Module Temperature: [-20,90]
  AC Voltage: [0, None]
  DC Voltage: [0, "[Vmp]*[Ns]*1.2"]
  Inverter Efficiency: [0.75, 1.01]
  ...
```

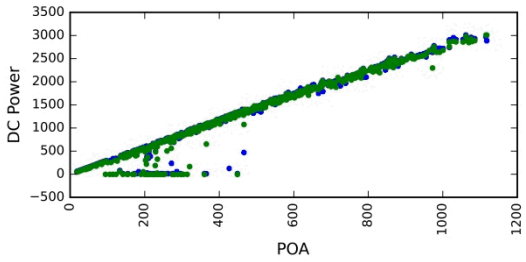
Reporting

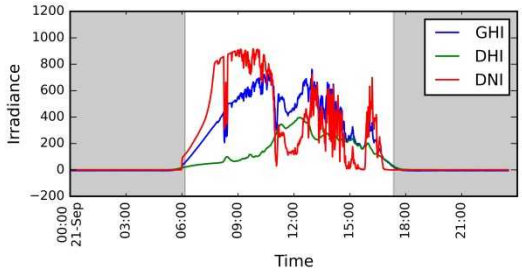
DPM reports include the start and end time for analysis, summary graphics and statistics, a table that includes missing/corrupt data points, a table and graphics for warnings (data points that failed a quality control tests), and notes on runtime errors and warnings.

Summary Graphics

Summary graphics can be customized for a particular application. For example, PV databases generate a plot of DC Power vs. time and DC Power vs. POA. Weather databases generate a plot of GHI, DHI, and DNI vs. Time.







Summary Statistics

A Quality Control Index (QCI) is computed for each system. QCI is the percent of data points that pass quality control tests. Duplicate and non-monotonic indexes are not counted as failed tests. Duplicates are removed and non-monotonic indexes are reordered. The occurrence of duplicate and non-monotonic indexes are listed in the final report.

QCI is defined as: $QCI = \frac{\sum_{d \in D} \sum_{t \in T} X_{dt}}{|DT|}$

where D is the set of data columns and T is the set of timestamps in the analysis. X_{dt} is a data point for column d time t that passed all quality control test. |DT| is the number of data points in the analysis. A value of 1 indicates that all data passed all tests. For example, if the database consists of 10 columns and 720 times that are used in the analysis, then |DT| = 7200. If 7000 data points pass all quality control tests, then the QCI is 0.972.

The Performance Ratio (PR) is computed for PV databases and the Clearness Index (CI) is computed for Weather databases.

PR is defined as: $PR = \frac{Y_{fAC}}{Y_r}$

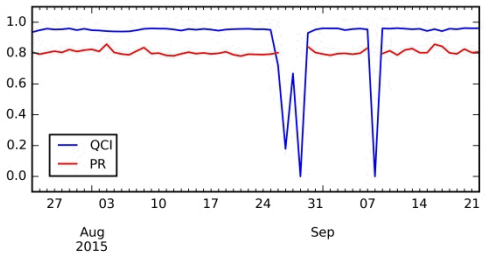
where Y_{fAC} is the AC system yield defined as the measured AC energy produced by the PV system in the day (kWh/d) divided by the rated power of the PV system (kW) . Y_r is is the plane-of-array insolation (kWh/m^2) divided by the reference irradiance (1000 W/m^2).

CI is defined as: $CI = \frac{\sum_{t \in T} DNI_t}{\sum_{t \in T} Ea_t}$

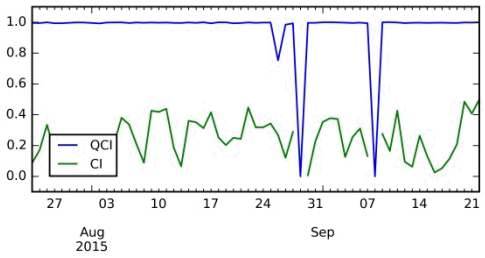
where DNI_t is the direct-normal irradiance at time t (kWh/m^2) and Ea_t is the extraterrestrial irradiance at time t (kWh/m^2) modeled using pvlib.

Other summary statistics can be added to the analysis. Summary statistics are stored over the entire time history the system has been analyzed by DPM. Summary statistic history can be plotted for any time window. Example graphics using a 60 day time window are shown below

PV Database statistics



Weather Database statistics



Missing/Corrupt Data

Missing and corrupt data is summarized in a table that includes the system name, the start and end time for the missing/corrupt data, the number of timesteps the data is missing/corrupt, and the number of variables associated with the missing/corrupt data. An example table is shown below. In this example, Direct_Wm2_Avg (DNI) is missing data between 8:00 and 8:05, the database is missing a timestamp at 9:10

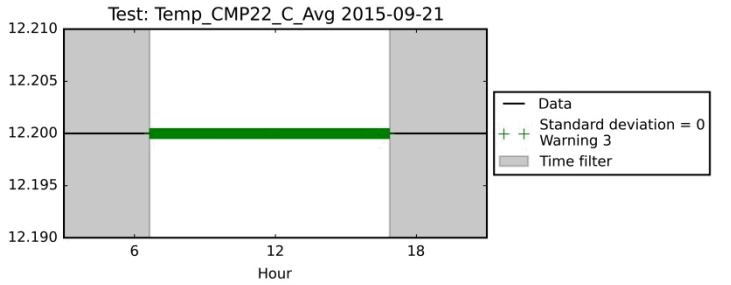
(all 16 variables are missing), and WS_ms_Mean (wind speed) has corrupt values (-999) between 14:50 and 14:51.

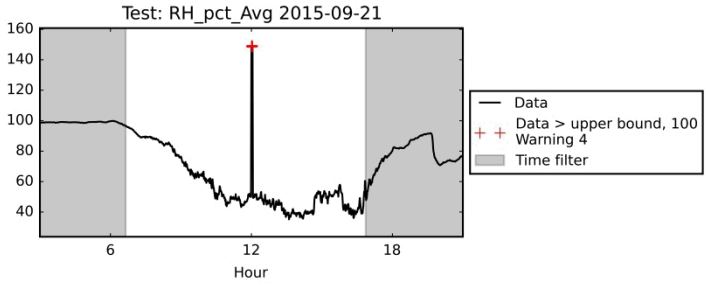
	System Name	Start Date	End Date	Timesteps	Number of Variables	Variables	Error Flag
1	Test	2015-09-21 08:00:00	2015-09-21 08:05:00	6	1	[Direct_Wm2_Avg]	Missing data
2	Test	2015-09-21 09:10:00	2015-09-21 09:10:00	1	16	[Battery_V_Avg, Diffuse_Wm2_Avg, Direct_Wm2_Avg, Global_Wm2_Avg, …	Missing timestamp
3	Test	2015-09-21 14:50:00	2015-09-21 14:51:00	2	1	[WS_ms_Mean]	Corrupt data

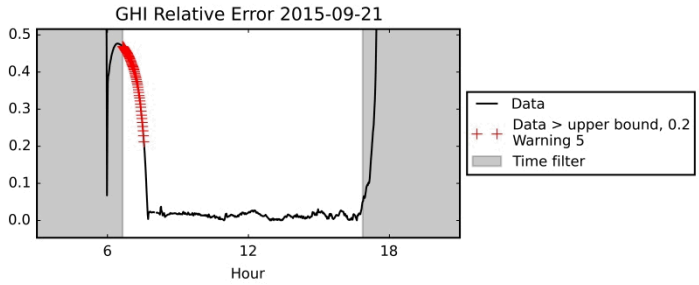
Warnings

Warnings are associated with data points that failed a specific quality control test. If the database includes warnings, a table and graphics are generated. The table includes the system name, the variable name, the start and end date, the number of continuous timestamps associated with the test failure, and an error flag that indicates what type of test failed. The graphics display the data point(s) that caused the failure. An example table is below. In this example, the database includes timestamp 9:41 that is out of order (non-monotonic), there is a duplicate timestamp at 11:46, Temp_CMP22_C_Avg (temperature) is always recoding 12.2, the RH_pct_Avg (relative humidity) has 4 consecutive values that are over 100, and GHI relative error is greater than 20% in the early morning.

	System Name	Variable Name	Start Date	End Date	Timesteps	Error Flag
1	Test		2015-09-21 09:41:00	2015-09-21 09:41:00	1	Nonmonotonic timestamp
2	Test		2015-09-21 11:46:00	2015-09-21 11:46:00	1	Duplicate timestamp
3	Test	Temp_CMP22_C_Avg	2015-09-21 06:39:00	2015-09-21 16:51:00	613	Standard deviation = 0
4	Test	RH_pct_Avg	2015-09-21 12:00:00	2015-09-21 12:03:00	4	Data > upper bound, 100
5		GHI Relative Error	2015-09-21 06:39:00	2015-09-21 07:33:00	55	Data > upper bound, 0.2







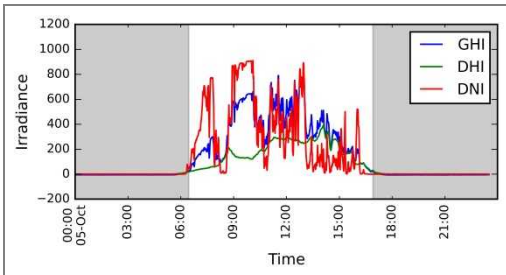
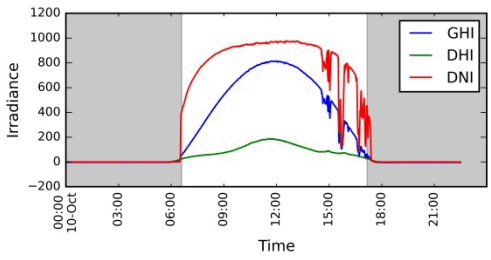
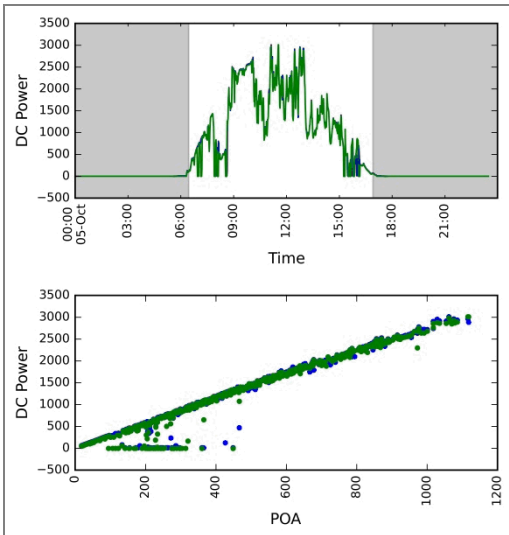
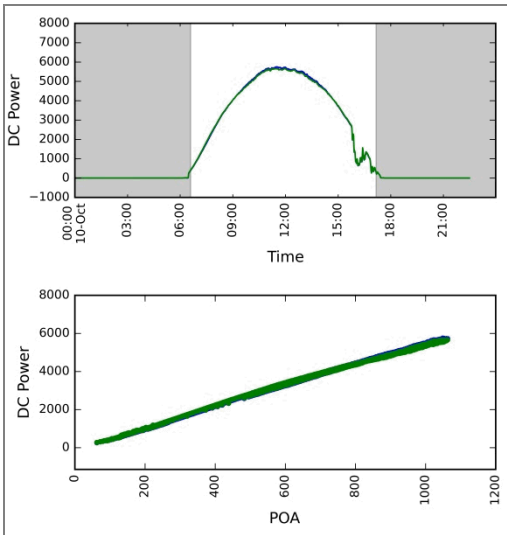
Notes

Notes include DPM runtime errors and warnings. Notes include:

- Empty/missing database
- Formatting error in the translation dictionary
- Insufficient data for a specific quality control test
- Insufficient data or error when computing a composite signals

Dashboard View

Custom dashboards can be created to help compare performance across multiple locations and databases. The dashboard includes summary statistics and graphics which link to detailed reporting information.

	Location 1	Location 2
Weather Database	 QCI: 0.95 CI: 0.16	 QCI: 1.0, CI: 0.62
PV Database	 QCI: 0.90, PR: 0.87	 QCI: 0.96, PR: 0.89