SAND2016-5177C

# Automated Performance Monitoring for PV Systems using Pecos

Katherine A. Klise and Joshua S. Stein

Sandia National Laboratories, Albuquerque, NM 87185, USA

*ABSTRACT* — **Photovoltaic system monitoring can generate vast amounts of data. Analytical methods are required to post-process this data into useful information. Pecos is open source software designed to address this need. The software was developed at Sandia National Laboratories and is compatible with performance models in PVLIB-Python. The software can be used to automatically run a series of quality control tests and generate reports which include performance metrics, test results, and graphics.**

## I. Introduction

Photovoltaic (PV) system performance depends on a range of complex conditions, including seasonal and daily weather variation, component degradation, intermittent component failure, and operating conditions. Monitoring this complex system is critical to ensure the system meets long term projections. Meteorological stations, pyranometers, and thermocouples are used to collect valuable data on PV system inputs and outputs. The challenge is to turn that data into meaningful information that can help support operations and maintenance over the lifetime of the system. When collecting large amounts of data, automated data quality control and performance monitoring can allow system operators to quickly detect performance issues.

Pecos is open source software designed to address this need. Pecos includes built-in functionality to monitor performance of time series data. The software can be used to automatically run a series of quality control tests and generate customized reports which include performance metrics, test results, and graphics. The software was developed specifically for PV performance monitoring, but it can be customized for other applications. The initial software release includes a core framework for performance monitoring. The functionality in Pecos can be used to establish standard procedures to track long term system health and analyze energy production, such as the standards outlined in IEC 61724 [1]. Future software development will work to incorporate these data collection and analysis standards.

Pecos is a Python package that utilizes powerful scientific computing functionality from several Python packages, including Pandas [2], NumPy [3], and Matplotlib [4]. For PV applications, PVLIB-Python [5][6] can be used to include performance models. Compatibility with PVLIB-Python allows the user to compare system measurements with model results and flag data points that fall outside expected bounds.

Some high level features of Pecos include:

- Flexible quality control tests can be used to identify a wide range of anomalous conditions in time series data.
- Data can be grouped into common variable names that represent system architecture, facilitating module-level, string-level, and system-level monitoring.
- Time filters can be used to eliminate data at specific times from quality control tests.
- Composite signals can be added to represent relationships between measured data and to compare measured data to modeled values.
- Compatibility with PVLIB-Python allows the user to integrate performance models in the analysis and create time filters that are a function of sun position.
- Compatibility with Pandas facilitates a comprehensive set of time series analysis options and date-time functionality.
- General and custom performance metrics can be stored to keep a running history of system health.
- Performance monitoring can be run on an automated schedule.
- HTML formatted reports can be sent to system operators via email or hosted on a website.

## II. Performance Monitoring with Pecos

Pecos monitors system performance over a specified time frame defined by the user. For example, that time frame could be the previous hour, day, week, or year. The time series data is read from single or multiple files and loaded into Pecos as a Pandas DataFrame. Dependence on Pandas allows the user to easily read data from several file formats, including CSV, Excel, and SQL, and recognize a wide range of date-time stings used to index time series data. Pecos builds on these features to include a built-in function to read data from Campbell Scientific file format. Additional data logger formats could be added. Within Pecos, several data files can be easily merged into a single analysis. For example, electrical measurements and weather data can be combined, or files containing data for different time frames can be combined.

The column names used in the database can be mapped into common names. This translation allows the user to access data using common names when running the analysis. For example, the user can specify that the name 'AC Power' will be used to run analysis on the column of data called 'Sys1Wac_Avg.' The translation can also be used to group similar variables into module-level, string-level, and system-level parameters. For example, DC current can be grouped according to strings. This grouping is useful when the user wants to run the same quality control test on multiple columns of data.

Time filters indicate if specific time index should be used in quality control tests. The time filter can be defined using elapsed time, clock time, or other custom algorithms. Using the system location and data time index, sun position can be used to define a custom time filter using methods in PVLIB-Python. A time filter can also be used to add conditional statements to any test. For example, a time filter can be defined based on apparent sun elevation greater than 10 degrees and plane of array irradiance greater than 250 W/m$^2$. Isolating data that meets specific criteria helps eliminate false warnings.

Composite signals can be included to generate new data based on existing data. For example, composite signals can be used to compute the relative error in power output as compared to a model, or compute inverter efficiency from measured AC and DC power output. Data created from composite signals can be used in quality control tests to ensure that the new data is within expected bounds.

Pecos includes several built-in quality control tests. When a test fails, information about the failure is saved in memory and printed in the final report. Quality control tests fall into five categories:

1) Timestamp tests are used to check the time index for non-monotonic, duplicate, and missing indexes. If the index is not monotonically increasing, the timestamp is reordered. If a duplicate time index is identified, the first occurrence is kept. If a time index is missing, then that index is added with NaN (not a number) for each column.
2) Corrupt data tests are used to flag corrupt values in the data. Some data loggers will record a value (i.e. -999) instead of recording a blank cell or NaN when the data is corrupt.
3) Missing data tests are used to flag missing values (blank or NaN) in the data. Missing data can occur across all columns or some columns in the data.
4) Range tests are used to check if data is within an expected range. The range test can be used to test if specific data columns fall outside expected bounds, or if a composite signal falls outside expected bounds. This can be useful when comparing the relative or absolute error between measured data and a model. An upper bound, lower bound, or both can be specified.
5) Increment tests are used to check that the difference between consecutive data values (or other specified increment) is within an expected range. Like the range test, the increment test can be performed on raw data or composite signals. This method can be used to test if data is not changing over a specified time interval or if the data has an abrupt change (either positive or negative). An upper bound, lower bound, or both can be specified.

For each test, the minimum number of consecutive failures can be specified. For example, the user can specify that a test failure is recorded only after the test has failed for 5 consecutive timestamps. This threshold can help ensure that reported failures are significant. Data can also be smoothed with a moving window before using range and increment tests.

Performance metrics are stored to help track long term system health. Pecos includes a general quality control index which computes the percent of the data points that passed quality control tests. PV specific performance metrics include performance ratio, performance index, energy yield, and clearness index. Pecos also includes methods to compute insolation from irradiance and energy from power. Custom performance metrics can easily be included. Pecos is commonly used to perform quality control test on the previous day's data. Performance metrics from each day are appended in a file so that the system operator can review trends in system performance over the year. These daily performance metrics can also be analyzed in Pecos to report on long term performance issues.

Pecos can be used to generate automated reports, which include custom graphics, performance metrics, a table that includes test results, graphics associated with the test results, and notes on any runtime errors and warnings. Custom graphics can be a simple plot of the time series data, or a more specialized graphic showing a composite signal or a history of performance health. The graphics associated with the test results highlight data points that failed a quality control test. Fig. 1 includes an example of inverter efficiency that is below a lower bound of 0.7, and wind speed data that is not changing for several hours and then is missing. These graphics include the time filter. In this example, the filter excludes data from the quality control test when the apparent sun elevation is less than 10 degrees. For each test failure, the system name and variable name is stored in memory along with the start and end time of the failure, and an error flag. Error flags indicate the type of test that failed, and if applicable, the lower or

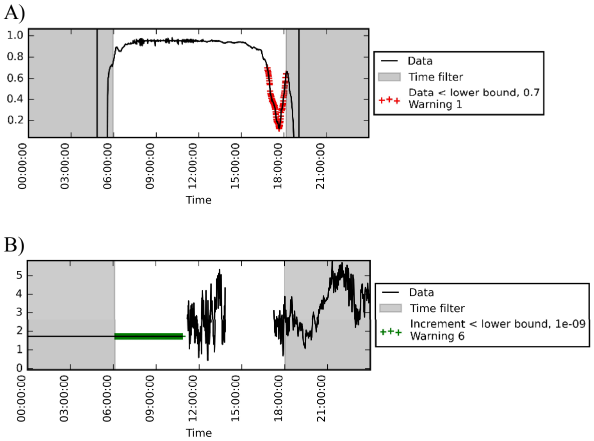


Figure 1. Quality control test graphics showing test failures in A) inverter efficiency (unitless) and B) wind speed (m/s).

upper bound that caused the failure (i.e. 'Missing timestamp, or 'Data < lower bound, 0.7). This information, along with associated graphics, is included in the report.

The results from quality control tests can be used to filter out or modify data points before including the data in subsequent analysis. Data points that fail a quality control test can be removed or replaced. In some cases, interpolation methods or modeling results could be used to replace detected errors. The "scrubbed" data can be saved to a file for later use.

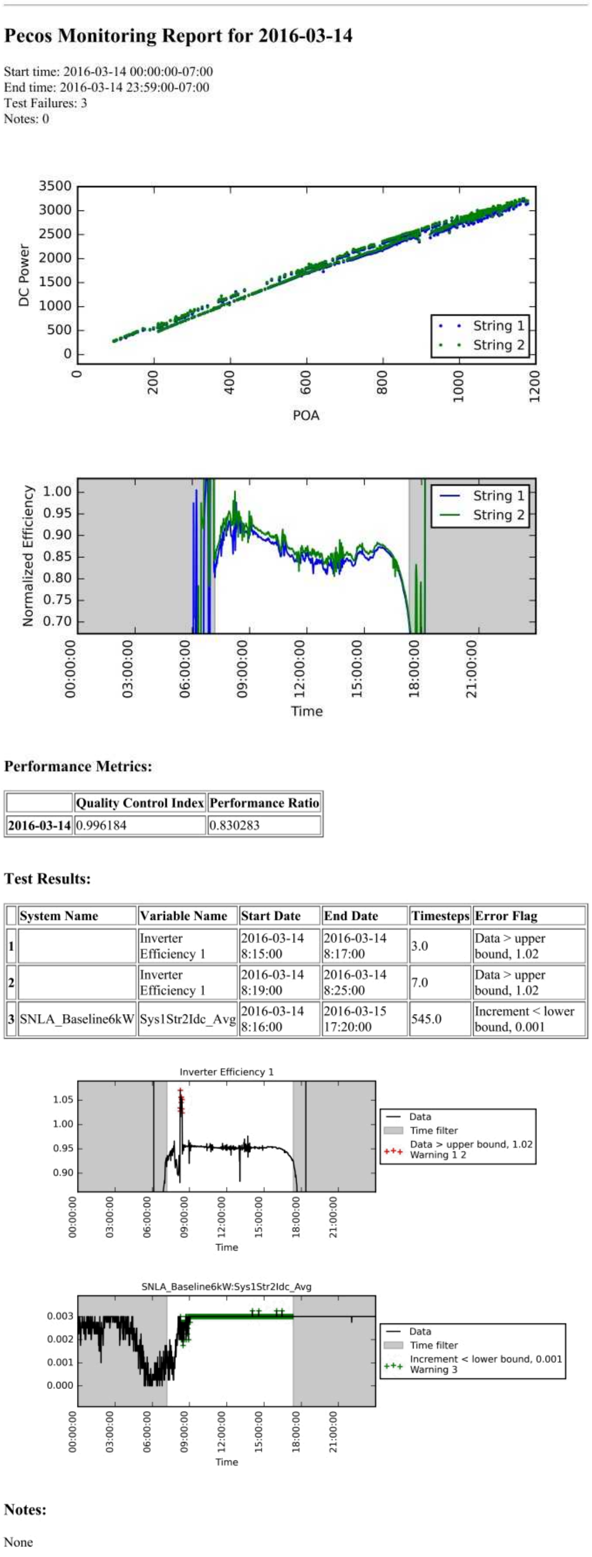
When analyzing multiple systems, it is sometimes useful to summarize analysis results in a "dashboard." Pecos includes functionality to generate dashboards that include custom graphics, performance metrics, and links to detailed reports for each system.

Pecos can be run on an automated schedule. For example, the software can be run each day to analyze data collected on the previous day and generate a daily report. Alternatively, the software can be used to generate weekly, monthly, or annual reports.

## III. Applications

Pecos is currently used by Sandia National Laboratories to monitor performance of multiple PV systems at different locations across the U.S. For each system, data collected the previous day is loaded into Pecos, a series of quality control tests are run, performance metrics are computed, and reports are automatically sent to system operators.

Information in the automated reports has helped diagnose missing data, corrupt or malfunctioning sensors, file transfer errors, system degradation, inverter performance issues, and mismatched string current due to module failures. An example monitoring report generated by Pecos is shown in Fig. 2. This report includes data from a 6kW PV system. Performance metrics in this example include the quality control index and performance ratio. These metrics are included in the report and are appended to a CSV file and kept to maintain a record of long term performance health. Custom graphics for this report include a plot of DC Power vs. POA irradiance and a time series of normalized efficiency. Normalized efficiency is a composite signal computed from DC power, DC power rating, POA irradiance, and reference irradiance. Normalized efficiency can help diagnose differences between strings. The report includes test failures associated with inverter efficiency that is higher than expected for several timestamps and DC current that did not change for more than an hour. The test results include the start and end time for each failure. Figures at the end of the report show the data points involved in each failure. Each figure includes the time filter in gray, which is defined based on the apparent sun elevation at the system location for the day of analysis. Apparent sun elevation is computed using PVLIB-Python. The information in the 'Test Results' section of Fig. 2 can be saved as a CSV file and used



**Pecos Monitoring Report for 2016-03-14**

Start time: 2016-03-14 00:00:00-07:00
End time: 2016-03-14 23:59:00-07:00
Test Failures: 3
Notes: 0

**Performance Metrics:**

| | Quality Control Index | Performance Ratio |
|---|---|---|
| **2016-03-14** | 0.996184 | 0.830283 |

**Test Results:**

| | System Name | Variable Name | Start Date | End Date | Timesteps | Error Flag |
|---|---|---|---|---|---|---|
| **1** | | Inverter Efficiency 1 | 2016-03-14 8:15:00 | 2016-03-14 8:17:00 | 3.0 | Data > upper bound, 1.02 |
| **2** | | Inverter Efficiency 1 | 2016-03-14 8:19:00 | 2016-03-14 8:25:00 | 7.0 | Data > upper bound, 1.02 |
| **3** | SNLA_Baseline6kW | Sys1Str2Idc_Avg | 2016-03-14 8:16:00 | 2016-03-15 17:20:00 | 545.0 | Increment < lower bound, 0.001 |

**Notes:**

None

Figure 2. Example Pecos performance monitoring report.

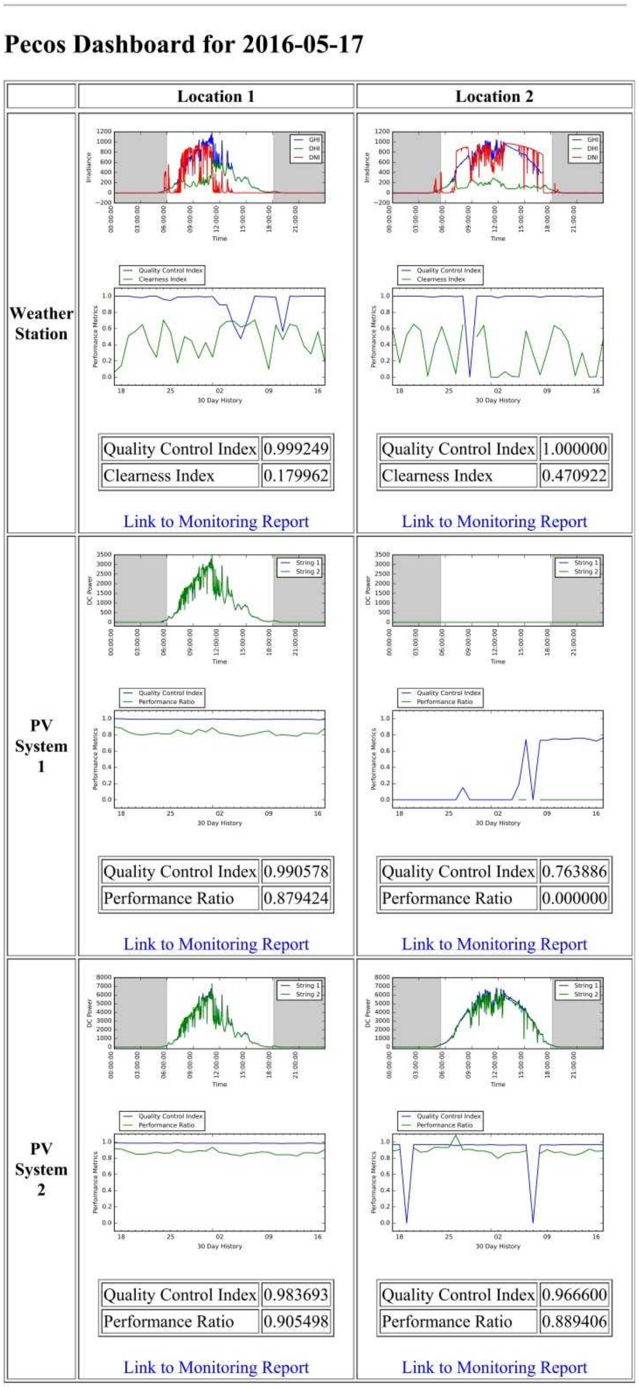
to later screen data points that failed quality control tests from the original data.

When analyzing multiple systems, possibly across multiple sites, it is convenient to display results in one place to compare systems side by side. Fig. 3 shows a dashboard generated by Pecos which includes three systems (one weather station and two PV systems) monitored at two different locations. Custom graphics include a thirty day history of performance metrics, a time series plot of GHI, DHI, and DNI for the weather station, and a time series plot of DC power for the PV systems. Performance metrics include the quality control index, clearness index for the weather station, and performance ratio for the PV systems. These graphics and metrics can be customized by the user for specific applications.

The dashboard in Fig. 3 illustrates several trends for the systems being monitored. The quality control index for both weather stations occasionally dips below 1, but quickly recovers. These failures are associated with intermittent missing data at Location 1 and a file transfer error at Location 2. The performance ratio is consistently higher for PV System 1 as compared to PV System 2 at Location 1. At the time when the monitoring report was run, PV System 1 at Location 2 was being installed. The quality control index indicates that 76% of the sensor data is being recorded and passes quality control tests at that time, and the performance ratio indicates that the system is not yet generating power. PV System 2 at Location 2 has had two file transfer issues in the past month, indicated by a quality control index of 0. Each system includes a link to a performance monitoring report, similar to the report shown in Fig. 2. This allows the user to quickly get an overview of system performance, and then see more detailed information on the quality control test results.

## IV. THE PECOS SOFTWARE PACKAGE

Pecos is a Python package which requires several Python package dependencies, including: Pandas, NumPy, and Matplotlib. Pecos is compatible with Python 2.7, 3.4, and 3.5. Python distributions, such as Anaconda, can help users manage the Python environment. Pecos can be installed through the Sandia National Laboratories GitHub organization at https://github.com/sandialabs/pecos. That site includes links to software documentation, hosted at http://pecos.readthedocs.org, and software testing results, hosted at https://coveralls.io/github/sandialabs/pecos. Users are encouraged to download the software, request features, and report issues through the GitHub site. Software developers interested in contributing to Pecos are expected to follow standard practices to document and test new code. Requests to merge code from outside developers will be reviewed by the maintainers of the software. The Pecos package includes simple examples to help get new users started. These examples demonstrate how to load data for analysis, define system variables, add time filters and composite signals, run quality control tests, and generate reports. The examples cover



**Pecos Dashboard for 2016-05-17**

| | Location 1 | Location 2 |
|---|---|---|
| Weather Station | Quality Control Index 0.999249<br>Clearness Index 0.179962<br>Link to Monitoring Report | Quality Control Index 1.000000<br>Clearness Index 0.470922<br>Link to Monitoring Report |
| PV System 1 | Quality Control Index 0.990578<br>Performance Ratio 0.879424<br>Link to Monitoring Report | Quality Control Index 0.763886<br>Performance Ratio 0.000000<br>Link to Monitoring Report |
| PV System 2 | Quality Control Index 0.983693<br>Performance Ratio 0.905498<br>Link to Monitoring Report | Quality Control Index 0.966600<br>Performance Ratio 0.889406<br>Link to Monitoring Report |

Figure 3. Example Pecos dashboard.

general and PV related use cases.

The package includes the following modules:

- *monitoring* contains the base performance monitoring class, PerformanceMonitoring, used to store time series data, run quality control tests, and collect results.
- *io* contains input/output methods used to load data from Campbell Scientific files and write HTML formatted reports. Pandas includes additional methods to read data from other standard formats.
- *metrics* contains general performance metrics used to track system health. This includes the quality control index and a general time integration method.
- *pv* contains custom methods for PV applications. This includes methods to compute energy from power, insolation from irradiance, performance ratio, performance index, energy yield, and clearness index. Additional methods will be added in future releases.
- *graphics* contains methods to plot time series data with superimposed test failures and time filters. These graphics are formatted for reports.
- *logger* contains a function to initialize a Pecos logger which saves all warnings and errors associated with the analysis. This information is printed in the monitoring report in the "Notes" section.
- *utils* contains helper functions. Current utility methods allow the user to round index values to a specified frequency and convert HTML files to image files.

Pecos is copyright through Sandia National Laboratories. The software is licensed under the Revised BSD license. The full license is included with the software. This license is intended to encourage users to integrate the code into other programs while maintaining proper reference to the original copyright.

## IV. Conclusion

This paper introduces the Pecos software package developed by Sandia National Laboratories. Pecos is an open source Python package designed to monitor performance of time series data, subject to a series of quality control tests. The software includes methods to run quality control tests defined by the user and generate reports which include performance metrics, test results, and graphics. The software is intended to facilitate monitoring of complex systems and alert system operators when the system has changed. Pecos integrates powerful Python tools for time series analysis, data storage, and report generation. The software was developed specifically for solar photovoltaic system monitoring, but it can be customized for other applications. Future development will focus on integration with standards such as IEC 61724 to help establish standard procedures to analyze energy production and track long term system health.

## Acknowledgements

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

## References

[1] IEC 61724, Photovoltaic system performance monitoring - Guidelines for measurement, data exchange and analysis, 2015

[2] W. McKinne, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, O'Reilly Media , 2012

[3] Stefan van der Walt, S. Chris Colbert and Gael Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30, 2011

[3] John D. Hunter. Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95, 2007

[4] W.F. Holmgren, R.W. Andrews, A.T. Lorenzo, and J.S. Stein, "PVLIB Python 2015," in *Photovoltaic Specialists Conference*, 2015.

[5] R.W. Andrews, J.S. Stein, C Hansen, and D Riley, Introduction to the Open Source PV LIB for Python Photovoltaic System Modelling Package, Photovoltaic Specialists Conference (PVSC), 2014.