Performance Monitoring using Pecos

2016 PV System Symposium
Santa Clara, CA, May 10, 2016

Katherine A. Klise
Sandia National Laboratories, Albuquerque, NM
Overview

- What is Pecos?
  - Performance monitoring for time series data

- Why use Pecos?
  - Collect large amounts of data on multiple systems and locations
  - Run automatic quality control tests on that data
  - Alert system operators when the system has changed
  - Generate reports
  - Collect performance statistics to track long term system health
  - Compare system performance across sites

- Pecos was developed specifically for solar photovoltaic system monitoring, but it can be customized for other applications
Software Framework

- Open-source python package
  - Revised BSD License
- Software repository
  - https://github.com/sandialabs/pecos
- Documentation
  - http://pecos.readthedocs.io
  - Includes API documentation
- Software testing results
  - https://travis-ci.org/sandialabs/pecos
  - https://coveralls.io/github/sandialabs/pecos
- ‘Getting started’ examples included with the software
  - simple
  - pv
  - metrics
  - dashboard
Installation

- Required dependencies
  - Python 2.7
  - pandas
  - numpy
  - Matplotlib

- Optional dependencies
  - pvlib
  - pyyaml
  - win32com
  - nose

- Building pecos from source
  ```
git clone https://github.com/sandialabs/pecos
cd pecos
python setup.py install
  ```

- Installing the latest release
  ```
pip install pecos
  ```
Basic Workflow

- Load time series data
  - From excel, csv, etc

- Define system variables
  - Translate data column names to common names

- Define time filter
  - Conditional statement

- Add composite signals
  - Relationship between data columns

- Run quality control tests
  - Timestamp errors
  - Missing data or corrupt data
  - Data out of range
  - Increment out of range

- Compute metrics
  - Quality control index

- Generate reports
  - HTML reports, csv results, metrics history

- Monitoring reports (html) include
  - Custom graphics
  - Table and graphics of test results
  - Performance metrics

- Test results file (csv) includes
  - System/variable name
  - Start and end time of failure
  - Number of time steps involved
  - Error flag

- Performance metrics file (csv) includes
  - Running history of stats

- Dashboards (html) includes
  - System summary
  - Link to monitoring report

pecos.io.write_monitoring_report(...)  
pecos.io.write_test_results(...)  
pecos.io.write_metrics(...)  
pecos.io.write_dashboard(...)
Basic Example

- **simple_example.py**
  - The data includes missing timestamps, duplicate timestamps, non-monotonic timestamps, corrupt data, data out of expected range, data that doesn’t change, and data that changes abruptly.
  - A = elapsed time in days
  - B = uniform random number between 0 and 1
  - C = sin(10*A)
  - D = C+(B-0.5)/2
PV Workflow

Load time series data
Campbell scientific file format, merge electrical and weather data

Define system variables
i.e. GHI = Global_Wm2_Avg
String current = [Str1dc_Avg, Str2ldc_Avg]

Define time filter
Related to solar position and/or irradiance conditions

Add composite signals
i.e. Total string current, inverter efficiency, normalized efficiency

Run quality control tests
Timestamp errors
Missing or corrupt data
Data out of range
Increment out of range

Generate a model
Predicted performance using a pvlib model

Add additional composite signals to compare the model to the data
i.e. Relative error in power output

Run additional quality control tests
Data out of range
Increment out of range

Compute metrics
Quality control index
Performance index
Energy yield
etc

Generate reports
Detailed monitoring reports
Performance metrics history
Dashboards
PV Examples

- `pv_example.py`
  - YAML configuration file
  - Electrical and weather data
  - Time filter based on sun position
  - pvlbl performance model and metric

- `metrics_example.py`
  - Track long term system health
  - Performance metrics from daily analysis
Dashboard Example

- `dashboard_example.py`
  - Compare performance of several systems
  - Generic dashboard
  - Includes text, graphics, table and link
Dashboard Example

Pecos Dashboard for 2016-03-28

<table>
<thead>
<tr>
<th>Location 1</th>
<th>Location 2</th>
<th>Location 3</th>
<th>Location 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Weather" /></td>
<td><img src="image2" alt="Weather" /></td>
<td><img src="image3" alt="Weather" /></td>
<td><img src="image4" alt="Weather" /></td>
</tr>
<tr>
<td>Quality Control Index: 0.392238</td>
<td>Quality Control Index: 0.997323</td>
<td>Quality Control Index: 0.971821</td>
<td>Quality Control Index: 0.984164</td>
</tr>
<tr>
<td>Clearness Index: 0.499177</td>
<td>Clearness Index: 0.029179</td>
<td>Clearness Index: 0.000057</td>
<td>Clearness Index: 0.128900</td>
</tr>
<tr>
<td><img src="image5" alt="Baseline" /></td>
<td><img src="image6" alt="Baseline" /></td>
<td><img src="image7" alt="Baseline" /></td>
<td><img src="image8" alt="Baseline" /></td>
</tr>
<tr>
<td>Quality Control Index: 0.992929</td>
<td>Quality Control Index: 0.942320</td>
<td>Quality Control Index: 0.962247</td>
<td>Quality Control Index: 0.988573</td>
</tr>
<tr>
<td>Performance Ratio: 0.882932</td>
<td>Performance Ratio: 0.705677</td>
<td>Performance Ratio: 0.754065</td>
<td>Performance Ratio: 0.893605</td>
</tr>
</tbody>
</table>

Link to Detailed Report | Link to Detailed Report | Link to Detailed Report | Link to Detailed Report
Future Development

- Tighter integration with IEC 61724
  - Photovoltaic system performance monitoring - Guidelines for measurement, data exchange and analysis
  - Four types of filters
    - Range, missing data, dead value, abrupt change
  - Example quality control tests
    - Power sensor is out of range if the value < -0.01*rating or > 1.02*rating
    - Irradiance sensor is dead if the derivative < 0.0001 W/m² while value > 5 W/m² (15 minute data)
    - Temperature sensor is erratic if the derivative > 4 C (15 minute data)
  - Bounds depend on precision requirements (Class A,B,C) and calibration accuracy
  - Data binned into times when inverters are on and off line
  - Integrated performance model and performance metrics

- Community input