Pecos, Open Source Software for PV Performance Monitoring

8th PV Performance Modeling and Monitoring Workshop
Albuquerque, NM, May 9-10, 2017

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

- **What is Pecos? (PAY-cose)**
  - Software for automated quality control and performance monitoring of time series data

- **Why use Pecos?**
  - Collect large amounts of data on multiple systems and locations
  - Run automated 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
Getting started

- Retrieve data
  - From sensor, database, files, or from the web
  - 3Vs (volume, velocity, and variety)
  - Single or repeat (automated)

- Define analysis
  - Analysis/reporting time interval
  - Filters
  - Integrate models
  - Quality control tests
  - Metrics

- Final product
  - Simple to complex
  - Red/yellow/green approach
  - Time series or interactive graphics
  - Performance history
  - Dashboards hosted on the web
  - Email alerts
Time series data

- Time series data loaded into Pecos as a Pandas DataFrame
  - Powerful time series analysis options
  - Datetime and timezone recognition
  - Merge multiple DataFrames in a single analysis (i.e. electrical and weather)
  - Data can be easily loaded from database, file, or web

- New Data acquisition methods recently added to Pecos
  - Transfer data from sensors to an SQL database

- User defines the analysis timeframe (minute, hour, day, month, …)
- Data can be grouped and renamed according to type
- Repeat analysis automated using OS task scheduler (cron, tasks)

```python
From database
sql_con = MySQLdb.connect(host=ip_address, port=...)
sql_query = "SELECT * FROM table..."
df = pandas.read_sql(sql_query, con=sql_con)

From file
df = pandas.read_csv(filename)

From the web
response = requests.get(url=http://developer.nrel.gov/pvdaq/api/...)
data = json.loads(response.text)
df = pandas.DataFrame(data=data['outputs']['data'])
```
Pre-processing filters

- **Filter data**
  - Smoothing
  - Upscale/downscale

- **Fill missing data**
  - Interpolation (linear, polynomial, etc.)
  - Duplicate sensors
  - Historic/regional data
  - Data generated from models

- **Time filter**
  - Conditional statement that exclude specific timestamps from quality control tests
  - Time filter can be based on:
    - Time of day (i.e. before 8 am and after 5 pm)
    - Sun position (i.e. sun elevation < 10 degrees)
    - Data properties (i.e. irradiance < 200 W/m²)
Composite signals

- Composite signals are used to create new data from existing data or from a model
  - Compute relationships between data columns
  - Compare measured data to a model
    - PVLIB performance model
    - Machine learning

- Examples
  - DC Power from current and voltage
  - Inverter efficiency from DC and AC power
  - Normalized efficiency from power and irradiance
  - Module temperature deviation
  - Relative error between model and data

- Composite signals can be used in the quality control tests
Quality Control tests

- Quality controls tests fall into five categories
  - Timestamp test
  - Missing data test
  - Corrupt data test
  - Range test
  - Dead sensor/abrupt change tests

- When a test fails, information is stored in a summary table which can be included in automated reports and saved to file/database. Graphics can be produced that pin point the data points that caused the test failure.

<table>
<thead>
<tr>
<th>System Name</th>
<th>Variable Name</th>
<th>Start Date</th>
<th>End Date</th>
<th>Timesteps</th>
<th>Error Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV System 1</td>
<td>Direct_Wm2</td>
<td>2017-04-19 12:36:00</td>
<td>2017-04-19 17:40:00</td>
<td>305</td>
<td>Increment &lt; lower bound, 0.0001</td>
</tr>
</tbody>
</table>
Quality Control tests

- **Timestamp test** identifies duplicate, non-monotonic, and missing timestamps. *New* Irregular timestamps can be preserved.
- **Missing data test** identifies column-time pairs that are missing.
- **Corrupt data test** screens for datalogger values that indicate corrupt data.

![Quality Control Tests Diagram]

### Original data

<table>
<thead>
<tr>
<th>TIMESTAMP</th>
<th>Column A</th>
<th>Column B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/2017 0:00</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1/1/2017 1:00</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1/1/2017 2:00</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>1/1/2017 3:00</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>1/1/2017 4:00</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>1/1/2017 5:00</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>1/1/2017 6:00</td>
<td>8</td>
<td>-999</td>
</tr>
<tr>
<td>1/1/2017 7:00</td>
<td>7</td>
<td>-999</td>
</tr>
<tr>
<td>1/1/2017 8:00</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>1/1/2017 9:00</td>
<td>9.5</td>
<td>2</td>
</tr>
</tbody>
</table>

### Corrected data

<table>
<thead>
<tr>
<th>TIMESTAMP</th>
<th>Column A</th>
<th>Column B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/2017 0:00</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1/1/2017 1:00</td>
<td>NaN</td>
<td>2</td>
</tr>
<tr>
<td>1/1/2017 2:00</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1/1/2017 3:00</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>1/1/2017 4:00</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>1/1/2017 5:00</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>1/1/2017 6:00</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>1/1/2017 7:00</td>
<td>7</td>
<td>NaN</td>
</tr>
<tr>
<td>1/1/2017 8:00</td>
<td>8</td>
<td>NaN</td>
</tr>
<tr>
<td>1/1/2017 9:00</td>
<td>9</td>
<td>1</td>
</tr>
</tbody>
</table>
Quality Control tests

- **Range tests** checks if data is within expected bounds
  - Ambient temperature should be between -30 and 50 degrees C
  - Normalized efficiency (composite signal) should be between 0.5 and 1
- **New Dead sensor/abrupt change test** checks if the difference between min and max is within expected bounds over a given time span
  - Voltage should not change by more than 80% rating within 15 minutes
  - The rain gauge should not increase by more than 2 inches in an hour
  - If the irradiance sensor changes by less than 0.0001 in 5 hours, it’s probably dead
Evaluating quality control tests

- **New** Evaluate how well a quality control test (or set of quality control tests) distinguishes normal from anomalous conditions.
  - Probability of detection
  - False alarm rate

- Strategies to reduce false positives and false negatives
  - Adjust thresholds
  - Specify the minimum number of consecutive failures needed to signal a warning
  - Smooth data before running quality control tests

<table>
<thead>
<tr>
<th>Estimated normal condition</th>
<th>Actual normal condition</th>
<th>Actual anomalous condition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>True negative (TN)</td>
<td>False negative (FN)</td>
</tr>
<tr>
<td>Estimated anomalous condition</td>
<td>False positive (FP)</td>
<td>True positive (TP)</td>
</tr>
</tbody>
</table>

\[
\text{FAR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \\
\text{FAR} = 1 - \text{Specificity} \\
\text{PD} = \frac{\text{TP}}{\text{TP} + \text{FN}} \\
\text{PD} = \text{Sensitivity}
\]
RTC quality control analysis

- Regional Test Center Baseline and Weather systems
  - New Mexico, Florida, Vermont, Nevada
  - 2 strings of 12 Suniva Optimus 270 Black modules

- Quality control tests and performance metrics based on IEC 61724
  - Check for data outside expected range, dead sensors, and abrupt changes
  - Compute in-service and all-in energy performance index

- Analysis run daily (near real-time), results emailed to stakeholders.

- End of year report

<table>
<thead>
<tr>
<th>Weather data</th>
<th>Baseline PV data</th>
</tr>
</thead>
<tbody>
<tr>
<td>GHI, DNI, DHI, air pressure, wind speed, wind direction, relative humidity</td>
<td>For each string: DC voltage, DC current, AC voltage, AC current, AC power, power factor, frequency, reference cell irradiance, and reference cell temperature</td>
</tr>
</tbody>
</table>

Module specs: $P_{\text{max}} = 270$ W, $V_{\text{mp}} = 31.2$ V, $V_{\text{oc}} = 38.5$ V, $I_{\text{mp}} = 8.68$ A, $I_{\text{sc}} = 9.15$ A
RTC quality control analysis

Load time series data
- Daily or entire year

Define system variables
- Group according to type

Define filters
- Apply a linear filter if data is missing for < 2 hours
- Screen out data when POA < 200 W/m² or sun elevation < 20 degrees

Add composite signals
- DC power
- Inverter efficiency
- Normalized efficiency
- Module temperature deviation

Run quality control tests
- Duplicate, non-monotonic, missing timestamps
- Missing if > 2 hours
- Corrupt if -999
- Expected range
- Dead/abrupt change

Performance model
- PVWatts model run using PVLIB
- Filter out data that failed quality control test before running the model

Compute metrics
- Data availability
- Quality control index
- System availability
- Energy performance index

Generate reports
- Daily dashboard with link to interactive graphics and detailed report
- Performance history over the year

Compute metrics
- Data availability
- Quality control index
- System availability
- Energy performance index

Generate reports
- Daily dashboard with link to interactive graphics and detailed report
- Performance history over the year
## RTC quality control analysis

- **Expected range, dead sensor, abrupt change thresholds**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Expected range</th>
<th>Dead sensor threshold</th>
<th>Abrupt change threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC current and AC current (A)</td>
<td>&gt; 0 and &lt; Imp*1.5</td>
<td>&lt; 0.0001 in 5 hours</td>
<td></td>
</tr>
<tr>
<td>DC voltage and AC voltage (V)</td>
<td>&gt; 0 and &lt; Vmp<em>12</em>1.5</td>
<td>&lt; 0.0001 in 5 hours</td>
<td></td>
</tr>
<tr>
<td>DC power* and AC power (W)</td>
<td>&gt; 0 and &lt; Pmp<em>12</em>1.5</td>
<td>&lt; 0.0001 in 5 hours</td>
<td>&gt; Pmp<em>12</em>0.8 in 15 min</td>
</tr>
<tr>
<td>Power factor</td>
<td>&gt; -1 and &lt; 1</td>
<td>&lt; 0.0001 in 5 hours</td>
<td></td>
</tr>
<tr>
<td>Frequency(Hz)</td>
<td>&gt; 57 and &lt; 63</td>
<td>&lt; 0.0001 in 5 hours</td>
<td></td>
</tr>
<tr>
<td>POA, DNI, GHI, and ref cell irradiance (W/m²)</td>
<td>&gt; 0 and &lt; 1500</td>
<td>&lt; 0.0001 in 5 hours</td>
<td></td>
</tr>
<tr>
<td>DHI (W/m²)</td>
<td>&gt; 0 and &lt; 500</td>
<td>&lt; 0.0001 in 5 hours</td>
<td></td>
</tr>
<tr>
<td>Air pressure (mbar)</td>
<td>&gt; 800 and &lt; 1020</td>
<td>&lt; 0.0001 in 5 hours</td>
<td>&gt; 100 in 15 minutes</td>
</tr>
<tr>
<td>Wind speed (m/s)</td>
<td>&gt; 0 and &lt; 32</td>
<td>&lt; 0.0001 in 5 hours</td>
<td></td>
</tr>
<tr>
<td>Wind direction</td>
<td>&gt; 0 and &lt; 360</td>
<td>&lt; 0.0001 in 5 hours</td>
<td></td>
</tr>
<tr>
<td>Relative humidity</td>
<td>&gt; 0 and &lt; 100</td>
<td>&lt; 0.0001 in 5 hours</td>
<td>&gt; 50 in 15 minutes</td>
</tr>
<tr>
<td>Ambient temperature (°C)</td>
<td>&gt; -30 and &lt; 50</td>
<td>&lt; 0.0001 in 5 hours</td>
<td>&gt; 20 in 15 minutes</td>
</tr>
<tr>
<td>Module and ref cell temp (°C)</td>
<td>&gt; -30 and &lt; 90</td>
<td>&lt; 0.0001 in 5 hours</td>
<td>&gt; 20 in 15 minutes</td>
</tr>
<tr>
<td>Module temp deviation (°C)*</td>
<td>&gt; -10 and &lt; 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inverter efficiency*</td>
<td>&gt; 0.5 and &lt; 1</td>
<td></td>
<td>&gt; 0.25 in 15 minutes</td>
</tr>
<tr>
<td>Normalized efficiency*</td>
<td>&gt; 0.5 and &lt; 1</td>
<td></td>
<td>&gt; 0.25 in 15 minutes</td>
</tr>
</tbody>
</table>

* Composite signal

---

---
RTC quality control analysis

Daily report, red/yellow/green dashboard with links to details and interactive graphics

### RTC Dashboard for 2017-03-24

<table>
<thead>
<tr>
<th></th>
<th>New Mexico</th>
<th>Florida</th>
<th>Vermont</th>
<th>Nevada</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Weather</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Irradiance</td>
<td>1.00</td>
<td>1.00</td>
<td>0.57</td>
<td>1.00</td>
</tr>
<tr>
<td>Wind</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Air Pressure</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Humidity</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Rainfall</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Datalogger</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Detailed Report</td>
<td>Interactive Plot</td>
<td>Interactive Plot</td>
<td>Interactive Plot</td>
<td>Interactive Plot</td>
</tr>
<tr>
<td>Interactive Plot</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| **Baseline**        |            |         |         |        |
| Irradiance          | 1.00       | 1.00    | 1.00    | 1.00   |
| Temperature         |            | 0.49    | 0.27    | 1.00   |
| Current             | 1.00       | 1.00    | 1.00    | 1.00   |
| Voltage             | 1.00       | 1.00    | 1.00    | 1.00   |
| Power               | 1.00       | 0.93    | 0.24    | 1.00   |
| Detailed Report     | Interactive Plot | Interactive Plot | Interactive Plot | Interactive Plot |
| Interactive Plot    |            |         |         |        |

![Graphs and Interactive Plots](image)
RTC quality control analysis

Yearly report, daily and monthly metrics

New Mexico

- GHI, DNI, DHI sensors dead
- Daily average
- Monthly average

Vermont

- DC current and DNI sensors dead
- Periodic low normalized efficiency
- String 2 I and V sensors are dead, DC power < 0
- String 2 EPI computed using very little data
Pecos

- Open-source python package
  - Python 2.7, 3.4, or 3.5
  - Revised BSD License
- Software repository
  - https://github.com/sandialabs/pecos
- Documentation
  - http://pecos.readthedocs.io
- Software testing
  - https://travis-ci.org/sandialabs/pecos
- ‘Getting started’ examples included with the software
- Version 0.1.5 (master branch)
  - New features include data acquisition, more flexible dashboards, PD and FAR metrics, compatibility with irregular timestamps, improved efficiency