
G-PST Consortium Pillar 5
May 27, 2021
Housekeeping & WebEx Tips

During the presentation(s):
- If you have a question:
  - Type it in the Q&A box
  - They will be addressed during our Q&A session at the end.
- This webinar is being recorded and will be available on the G-PST website.

Change layout:
- Grid – All videos shown
- Stage – Speaker highlighted, other videos below
- Focus – Only speaker video shown

Audio Issues?
- Check audio outputs and volumes
- If listening by computer, try dialing in by phone
- If listening by phone, try using computer audio
Contributors

Roman Bolgarny, Fraunhofer IEE
Researcher (primary speaker)

Friederike Wenderoth, Fraunhofer IEE
Researcher

Hannele Holttinen, Recognis Consulting
Pillar 5 Lead

Juha Kiviluoma, VTT
Pillar 5 Lead

Clayton Barrows, NREL
Senior Engineer

Dheepak Krishnamurthy, NREL
Distribution Systems Researcher
Global Power System Transformation (G-PST) Consortium

What?
A new global Consortium focused on support to power system operators with advanced high RE & other low-emission solutions

Who?
Founding System Operators

How?
5 Pillars
1. System Operator Research and Peer Learning
2. System Operator Technical Assistance
3. Workforce Development
4. Localized Technology Adoption Support
5. Open Tools and Data

G-PST Core Team Technical Institutes

Developing Country System Operators - Confirmed partners - Indonesia, Vietnam, India, South Africa, and Peru

Interim Secretariat - NREL

Global Power System Transformation Consortium
How You Can Engage

• Join our network to receive webinar invitations, our newsletter and other important updates - [https://globalpst.org/get-involved/](https://globalpst.org/get-involved/).

• Engage in our regional peer learning networks and/or pillar groups – submit interest in particular pillars through the website - [https://globalpst.org/get-involved/](https://globalpst.org/get-involved/)

• Reach out to explore remote light touch technical assistance or with any questions at: [globalpst@nrel.gov](mailto:globalpst@nrel.gov)
Pillar 5 Introduction
How to find a useful tool for a specific problem

- High VRE shares call for new capabilities in analysis tools
- Tools are needed both for learning and for analysis
- New tools need to demonstrate capability and they require validation, benchmarking, as well as support

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Flow</td>
<td>Inertia</td>
<td>Data analytics</td>
</tr>
<tr>
<td>Contingency analysis</td>
<td>Open source data to model input</td>
<td>State estimation</td>
</tr>
</tbody>
</table>
Pillar 5 Open Source Tools & Data - Vision

**Portal**
- Access to useful tools, for each modelling issue: starting with power flow, inertia, and stability
- Description on the strengths and weaknesses/applicability

**Interface**
- Model/tool integration (data and workflow management)
- Easy-to-use simple tools (with learning focus) and High fidelity tools (for advanced uses) with same datasets and enabling model linkage

**Improvement**
- Leverage partner engagements to make tools more accessible and applicable
- From offline, analytical use towards real-time tools
Why open-source?

Advantages:
- Transparency
- Free
- Available source code for customization/extension
- Minimizes duplicate efforts
- Community oriented
- Focus on research

Disadvantages:
- Maintenance burden
- Ambiguous revenue/funding
- IP protection
- Focus on research

Common concerns:
- Support
- Security
- Validation
pandapower
Official website:
http://www.pandapower.org/
GitHub repository:
https://github.com/e2nIEE/pandapower
Documentation:
https://pandapower.readthedocs.io
Background: Python ecosystem

- Python is an open-source general purpose interpreted programming language
- Most popular programming language with a large community that provides thousands of packages
- Python supports data analysis and visualization, extensibility, automation and parallelization of workflows
Benefits of Python for pandapower:

- Power system analysis can be part of a larger workflow in the same environment
- Parallelization possibilities
- Access to a wide range of libraries to extend standard pandapower capabilities
- Popularity of Python makes it easier to find and hire talent to work with pandapower
Background: pandas

- pandas is an open-source data analysis and manipulation library
- Provides a DataFrame object to store and manipulate tabular data
- Supports different data formats e.g. CSV, Excel, JSON, SQL
- Provides functionality to read, manipulate, visualize and save data
Benefits of pandas for pandapower:

• Organizing data in a label-based tabular form
• Pre-processing of input data, as well as interpretation and visualization of calculation results
• Familiar to most Python users, which simplifies getting started with pandapower
Background: MATPOWER

- MATPOWER is an open-source package for MATLAB or GNU Octave
- Grid data can be exchanged using the .mat format
- Analysis capabilities:
  - power flow
  - continuation power flow
  - optimal power flow
  - unit commitment
Implementation gap addressed by pandapower:

- Practical parameters and standard types for elements
- Implementation of switches, 3-winding transformers
- Realistic grid topology
- Control framework
- Open file format
- 3-phase load flow
Why use pandapower?

- Automate workflows for power system analysis
  - Access to thousands of useful packages in Python
- No license limitations on parallelization for extensive calculations
  - Can be scaled on a high-performance computing (HPC) cluster
- Broad user base (over 100k downloads and growing) and productive use at Fraunhofer IEE and University of Kassel
  - Linus's law: "given enough eyeballs, all bugs are shallow"
  - Continuous quality improvement and addition of features by the community, Fraunhofer IEE, and the University of Kassel
- pandapower format allows automated screening of the grid data for irregularities, e.g., errors of data input
Types of devices and circuits that can be modeled with pandapower

Node elements:
- Shunt
- Load (incl. ZIP-load)
- Asymmetric load
- Motor
- Static generator
- Asymmetric static generator
- Synchronous generator
- External grid
- Shunt
- Ward
- Extended ward

Branch elements:
- Switch
- Line
- DC Line
- Impedance
- 2-W transformer
- 3-W transformer
Types of analysis possible using pandapower

- Power Flow
- Optimal Power Flow
- Controller simulation
- Time-series calculation (quasi-static analysis)
- State Estimation
- Short-Circuit Calculation
- Topological Graph Searches
- A customized analysis for a particular purpose
Interfaces with pandapower

- **PowerModels.jl**
  - Optimal power flow
  - Expansion planning
  - Optimal transmission switching
- **pandapipes**
  - fluid/gas pipe networks
- **MATPOWER** converter
- Further interfaces/converters possible by users
Visualization capabilities

Create schematic coordinates with igraph

Identify and highlight topological zones

Visualize results with plotly

Visualize results with plotly mapbox
Visualization capabilities

Results of a reinforcement study for HV grids

Visualize results with matplotlib

A grid with multiple voltage levels

- EHV grid
- Zone Flensburg
- Zone Göhl
- Zone Heide-West
- Zone Kummerfeld
- EHV substation
- Transit node
Visualization capabilities

Results of a reinforcement study for HV grids
Python – pandapower
Demo
Workflow using pandapower

Load input data
- pandapower net (JSON, MATPOWER, SQL etc)
- Time-series data (CSV, SQL, Excel etc)
- GIS-Data (Shapefile, etc.)

Pre-processing of data
- Validate the data
- Clean-up errors
- Combine grid model and input data
- Grid-preprocessing: topological searches, etc.
- Configure the grid model: add controllers, etc.

Run power system analysis
- Load flow
- OPF
- Controller simulation
- Time-series simulation
- Short-circuit
- State estimation
- Customized analysis

Post-processing of results
- Visualization with matplotlib, pandas, seaborn
- Data analysis with pandas
- Identify issues in the grid
- (Automated) modification of the grid

Save the results
- Visualizations (pdf, etc)
- Modified grid (JSON, SQL, etc)
- Calculation results (CSV, Excel, JSON, SQL etc)
• We run examples with Jupyter Notebook: jupyter.org
• First, install pandapower and all required packages.
• Start command prompt.
• Navigate to the directory pandapower/tutorials.
• Run the command "jupyter notebook":

```
C:\>
C:\>cd pandapower\tutorials
C:\pandapower\tutorials>jupyter notebook
```
Next Steps
Please suggest future events!

Possible future webinars to showcase other tools:

- **Julia:**
  - PowerModels.jl
  - PowerSystems.jl & PowerSimulations.jl

- **Python:**
  - PyPsa
  - Andes
Q&A

Contacts
pandapower:
Roman.Bolgaryn@iee.fraunhofer.de
Friederike.Wenderoth@iee.fraunhofer.de
GPS-T Pillar 5:
Hannele.Holtthinen@recognis.fi
Juha.Kiviluoma@vtt.fi
Clayton.Barrows@nrel.gov
Dheepak.Krishnamurthy@nrel.gov