pandapower - an Open Source Framework for Automated Evaluations of Future Power Systems

L. Thurner, A. Scheidler, M. Braun
Fraunhofer IWES & University of Kassel

1st International Conference on Large-Scale Grid Integration of Renewable Energy in India
New Delhi, India
September 6th – 8th, 2017
Introduction

- Increasing penetration of power systems with renewable energy resources (RES)
  - Germany: 10 GW (1999) $\rightarrow$ >100 GW (2016)
  - India: 43 GW (now) $\rightarrow$ 175 GW (plan for 2022)

- Majority of RES are installed in distribution networks
  - Large amount of different networks
  - High diversity of networks

- Studying the impact of RES installation is vital to guarantee cost-efficient planning and operation of future power systems
  - General conclusions are difficult to draw
  - Reliable analysis has to be based on large amount of network data

$\rightarrow$ New tools for the automated analysis of networks are necessary

$\rightarrow$ Introduction of the new open source power systems analysis tool pandapower
# pandapower vs. PyPSA

<table>
<thead>
<tr>
<th>pandapower</th>
<th>PyPSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python based open source Tool for Power System Analysis</td>
<td></td>
</tr>
<tr>
<td>Able to solve power flow and optimal power flow problems</td>
<td></td>
</tr>
<tr>
<td>Based on a tabular data structure using the Python library pandas</td>
<td></td>
</tr>
<tr>
<td>Focus on Detailed Power System Modelling</td>
<td>Focus on Energy Modelling</td>
</tr>
<tr>
<td>Static Power Flow, Short Circuit and Topological Analysis</td>
<td>Multi-Period Optimal Power Flow</td>
</tr>
<tr>
<td>Distribution System Analysis and Planning</td>
<td>Transmission System Analysis and Planning</td>
</tr>
<tr>
<td><a href="http://www.uni-kassel.de/go/pandapower">http://www.uni-kassel.de/go/pandapower</a></td>
<td><a href="https://github.com/FRESNA/PyPSA">https://github.com/FRESNA/PyPSA</a></td>
</tr>
</tbody>
</table>
pandapower Pro

**GRID PLANNING**
- Metaheuristic Grid Extension Planning
- Automated Sectioning Point Optimization
- Topological Network Optimization

**GRID ANALYSIS**
- Grid Loss Analysis
- Determination of Hosting Capacity
- Reliability and n-1 Outage Analysis

**GRID OPERATION**
- Time Series Simulations
- Local, Decentral and Central Controllers
- Co-Simulation Framework

**APPLICATION**
- Grid Loss Analysis
- Determination of Hosting Capacity
- Reliability and n-1 Outage Analysis

**FRAMEWORK**
- pandapower
  - State Estimation
  - Network Building API
  - Topological Graph Search Analysis
  - Optimal Power Flow
  - Different Power Flow Solvers
- Large Library of Grid Element Models
- IEC 60909 Short Circuit Calculation
- Data structure based on pandas
- Plotting

**INPUT**
- Conversion from other Tools / Formats
  - PowerFactory
  - PSS Sincal
  - Neplan
  - CIM
- Forecasts for
  - Probabilistic DER installation scenario
  - Generic load profiles
- Boundary Conditions
  - Operational Limits
  - Planning Principles
  - Topology Constraints
  - Contingency Constraints

**OPEN SOURCE**

---

Martin Braun
Integration Conference
New Delhi, India | 6-9 Sep 2017
http://www.uni-kassel.de/go/pandapower
Overview

- pandapower
  - Element Models
  - Analysis Functionality and Validation
  - Minimal Example

- Hosting Capacity
  - Boxplot Distribution
  - Example Implementation in pandapower
  - Real World Case Study Results

- Conclusion
Overview

- pandapower
  - Element Models
  - Analysis Functionality and Validation
  - Minimal Example

- Hosting Capacity
  - Boxplot Distribution
  - Example Implementation in pandapower
  - Real World Case Study Results

- Conclusion
# pandapower Electric Element Models

<table>
<thead>
<tr>
<th></th>
<th>MATPOWER</th>
<th>PSAT</th>
<th>OpenDSS</th>
<th>FyPSA</th>
<th>GridCal</th>
<th>pandapower</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZIP-load</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Line</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2W-Transformer ($\pi$)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2W-Transformer (T)</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3W-Transformer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>DC Line</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ideal Switches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Synchronous Generator</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Static Load / Generator</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Shunt</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Unsym. Impedance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ward Equivalents</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
pandapower Switch Model

- Ideal Bus-Bus and Bus-Branch switch models

<table>
<thead>
<tr>
<th>Switch Configuration</th>
<th>Bus-Bus Switches</th>
<th>Bus-Line Switches</th>
<th>Bus-Transformer Switches</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>![Bus-Bus Switches Diagram]</td>
<td>![Bus-Line Switches Diagram]</td>
<td>![Bus-Transformer Switches Diagram]</td>
</tr>
<tr>
<td>Common Approximation</td>
<td>![Common Approximation Diagram]</td>
<td>![Common Approximation Diagram]</td>
<td>![Common Approximation Diagram]</td>
</tr>
<tr>
<td>pandapower Switch Model</td>
<td>![pandapower Switch Model Diagram]</td>
<td>![pandapower Switch Model Diagram]</td>
<td>![pandapower Switch Model Diagram]</td>
</tr>
</tbody>
</table>

→ Allows detailed modelling of substations and tie-line switches
pandapower Functionality

**Electric Analysis**
- Power Flow
- Optimal Power Flow
- State Estimation
- Short-Circuit Calculation

**Further Functionality**
- Plotting
- Converter
- Generic Networks

**Topological Analysis**
- Graph Searches on Electric Network
Model Tests and Validation

- Power Flow Results are compared with commercial software to validate models
- Validation tests exist for every pandapower element
- About 250 unit tests for overall pandapower functionality
Creating Minimal Example

```python
import pandapower as pp

# create empty net
net = pp.create_empty_network()

# create buses
bus1 = pp.create_bus(net, vn_kv=20., name="Bus 1")
bus2 = pp.create_bus(net, vn_kv=0.4, name="Bus 2")
bus3 = pp.create_bus(net, vn_kv=0.4, name="Bus 3")

# create bus elements
pp.create_ext_grid(net, bus=bus1, vm_pu=1.02)
pp.create_load(net, bus=bus3, p_kw=100, q_kvar=50)

# create branch elements
trafo = pp.create_transformer(net, hv_bus=bus1, lv_bus=bus2,
                               std_type="0.4 MVA 20/0.4 kV")
line = pp.create_line(net, from_bus=bus2, to_bus=bus3,
                      length_km=0.1, std_type="NAYY 4x50 SE")
```

Grid Connection:
- Voltage: 1.02 pu

Transformer:
- Transformer Ratio: 20 / 0.4 kV
- Rated Power: 400 kVA
- Short Circuit Voltage: 6 %
- Short Circuit Voltage (real part): 1.425 %
- Open Loop Losses: 0.3375 %
- Iron Losses: 1.35 kW

Line:
- Length: 100 m
- Cable Type: NAYY 4x50 SE
- Resistance: 0.642 Ω/km
- Reactance: 0.083 Ω/km
- Capacity: 210 mF/km
- Max. thermal current: 142 A

Load:
- Active Power: 100 kW
- Reactive Power: 50 kVar
Running A Power Flow

- Running a power flow and inspecting the results:

```python
pp.runpp(net)
print("Trafo loading: %.2f %%\"\net.res_trafo.loading_percent.at[0])
print("Line loading: %.3f %%\"\net.res_line.loading_percent.at[0])
print("Voltage vector: %s\"\net.res_bus.vm pu.values)
```
Overview

- pandapower
  - Element Models
  - Analysis Functionality and Validation
  - Minimal Example

- Hosting Capacity
  - Boxplot Distribution
  - Example Implementation in pandapower
  - Real World Case Study Results

- Conclusion
Hosting Capacity Distribution

- Concentrated distribution allows installation of only 8 PV plants
- Even distribution allows installation of 15 PV plants

Hosting Capacity is not one value but distribution of values
Hosting Capacity in pandapower

```python
import pandas as pd

results = pd.DataFrame(columns=["installed", "violation"])
for i in range(iterations):
    # Initialize new run
    net = load_network()
    installed_kw = 0
    while 1:
        # Check for violation of any constraint
        violated, violation_type = violations(net)
        if violated:
            # Save result and end iteration
            results.loc[i] = [installed_kw, violation_type]
            break
        else:
            # Add additional PV plant
            plant_size = get_plant_size_kw()
            bus = choose_bus(net)
            pp.create_einj(net, bus, p_kw=-plant_size, q_kvar=0)
            installed_kw += plant_size
```

```python
import pandapower as pp

def violations(net):
    pp.runpp(net)
    if net.res_line.loading_percent_max() > 50:
        return (True, "Line \n Overloading")
    elif net.res_trafo.loading_percent_max() > 50:
        return (True, "Transformer \n Overloading")
    elif net.res_bus.vm_pu_max() > 1.04:
        return (True, "Voltage \n Violation")
    else:
        return (False, None)
```

![Histogram and graph](image-url)
Hosting Capacity in pandapower

Hosting capacity analysis yields

- Distribution of installable capacity
- Distribution of limiting causes
Comparing Hosting Capacity to Expected PV Expansion

- Analysis of 111 Low Voltage Networks
- Comparing hosting capacity to expected installations

→ Study shows that most networks can host expected additional installation

Figure courtesy of Romande Energie
Assessment of Smart Grid Technologies

- Advanced PV Inverters functions
  - Reactive Power Control
    - Constant CosPhi
    - Q(U)-control (volt-var function)
    - CosPhi(P)-control (watt-var function)
  - Active power curtailment (e.g. peak shaving)

Figure courtesy of Romande Energie
Assessment of Smart Grid Technologies

- Advanced OLTC transformer control (AOLTC)
- Installation of additional voltage regulators (e.g. MV/LV transformers with OLTC-contoller)
- Combination of Smart Grid Technologies

Figure courtesy of Romande Energie
Technical and Economical Assessment of MV/LV OLTC transformer (rONT)

- Analysis of 329 Low Voltage Networks
  - PV Hosting Capacity
  - Full rooftop potential (maximum installable PV capacity)

→ Voltage problems in 85 LV networks

→ can potentially be mitigated with controllable MV/LV transformer (rONT)

Figure courtesy of Bayernwerk
Assessment of Smart Grid Technologies

- Increase of Hosting Capacity (median value) by MV/LV OLTC 85 real LV grids
Technical and Economical Assessment of MV/LV OLTC transformer

- Analysis of 85 Low Voltage Networks with voltage violations
- rONT mitigates all problems in 29 networks

⇒ rONT only effective in 29 / 329 overall networks

Figure courtesy of Bayernwerk
Overview

- pandapower
  - Element Models
  - Analysis Functionality and Validation
  - Minimal Example
- Hosting Capacity
  - Boxplot Distribution
  - Example Implementation in pandapower
  - Real World Case Study Results
- Conclusion
Conclusion

- pandapower was published in November 2016
  - Very positive feedback and participation from the community
  - Already deployed in multiple projects worldwide
  - Presented examples (and many more) are available as interactive notebooks on https://github.com/lthurner/pandapower/tree/develop/tutorials

- Applications of pandapower
  - Hosting capacity analysis was presented as example application with real world examples
  - pandapower has been successfully deployed in other applications, such as power system planning, operation, network studies, loss studies etc.

⇒ continuous development on github: https://github.com/lthurner/pandapower
⇒ Subscribe for pandapower updates on https://www.uni-kassel.de/go/pandapower
Annex – Minimal Example
Creating Minimal Example

```python
import pandapower as pp

# create empty net
net = pp.create_empty_network()

# create buses
bus1 = pp.create_bus(net, vn_kv=20., name="Bus 1")
bus2 = pp.create_bus(net, vn_kv=0.4, name="Bus 2")
bus3 = pp.create_bus(net, vn_kv=0.4, name="Bus 3")

# create bus elements
pp.create_ext_grid(net, bus=bus1, vm_pu=1.02)
pp.create_load(net, bus=bus3, p_kw=100, q_kvar=50)

# create branch elements
trafo = pp.create_transformer(net, hv_bus=bus1, lv_bus=bus2,
                              std_type="0.4 MVA 20/0.4 kV")
line = pp.create_line(net, from_bus=bus2, to_bus=bus3,
                      length_km=0.1, std_type="NAYY 4x50 SE")
```

Grid Connection:
- Voltage: 1.02 pu

Transformer:
- Transformer Ratio: 20 / 0.4 kV
- Rated Power: 400 kVA
- Short Circuit Voltage: 6 %
- Short Circuit Voltage (real part): 1.425 %
- Open Loop Losses: 0.3375 %
- Iron Losses: 1.35 kW

Line:
- Length: 100 m
- Cable Type: NAYY 4x50 SE
- Resistance: 0.642 Ω/km
- Reactance: 0.083 Ω/km
- Capacity: 210 nF/km
- Max. thermal current: 142 A

Load:
- Active Power: 100 kW
- Reactive Power: 50 kVar

http://www.uni-kassel.de/go/pandapower
Running A Power Flow

- Running a power flow and inspecting the results:

```python
pp.runpp(net)
print("Trafo loading: \%.2f \%"%net.res_trafo.loading_percent.at[0])
print("Line loading: \%.3f \%"%net.res_line.loading_percent.at[0])
print("Voltage vector: \%s\%"%net.res_bus.vm_fu.values)
```

Trafo loading: 29.29 \%
Line loading: 117.835 \%
Voltage vector: [ 1.02 1.00884272 0.96443057]

→ Detailed results for each element
Running A Power Flow – Tap Changers

- Change the transformer tap position:

```python
net.trafo.tp_pos.at[trafo] = -1
pp.runpp(net)
print("Trafo loading: %.2f %%%n"%net.res_trafo.loading_percent.at[0])
print("Line loading: %.3f %%%n"%net.res_line.loading_percent.at[0])
print("Voltage vector: %s"%net.res_bus.vm pu.values)
```

→ Transformer ratio changes
→ Voltage at the low voltage side of the transformer rises
Running A Power Flow – Switches

- Introduce an open switch at the end of the line:

```python
pp.create_switch(net, bus=bus3, element=line, et="1", closed=False)
pp.runpp(net)
print("Trafo loading: %.2f %%")%net.res_trafo.loading_percent.at[0])
print("Line loading: %.3f %%")%net.res_line.loading_percent.at[0])
print("Voltage vector: %s")%net.res_bus.vm_pu.values)
```

- Load bus is cut from power supply
- Voltage at isolated bus is returned as *nan (not a number)*
Topological Analysis

- Find all buses without galvanic connection to slack bus:
  ```python
  import pandapower.topology as top
  top.unsupplied_buses(net)
  ```
  
  {2}

- Find all buses on the same voltage level as Bus 2 (after closing switch)
  ```python
  net.switch.closed.at[0] = True
  mg = top.create_nxgraph(net, include_trafos=False)
  list(top.connected_component(mg, 2))
  ```
  
  [2, 1]
Short Circuit Analysis

- Define short circuit parameters of external grid
- Calculate short circuit currents according to IEC 60909
  - Initial short circuit current $I'_{k}$
  - Peak short circuit current $i_p$

```python
import pandapower.shortcircuit as sc
net.ext_grid["s_sc_max_mva"] = 100
net.ext_grid["rx_max"] = 0.1
sc.calc_sc(net, case="max", ip=True, r_fault_ohm=2.)
print(net.res_bus_sc)
```

```
<table>
<thead>
<tr>
<th>ikss_ka</th>
<th>ip_ka</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.534707 4.317318</td>
</tr>
<tr>
<td>1</td>
<td>0.126631 0.182666</td>
</tr>
<tr>
<td>2</td>
<td>0.122698 0.176991</td>
</tr>
</tbody>
</table>
```
## Contact

**Prof. Dr. Martin Braun**  
Head of Business Field  
**Grid Planning and Operation**  
- Mail: martin.braun@iwes.fraunhofer.de  
- Phone: +49 561 7294 118  
- www.iwes.fraunhofer.de

**Dr. Alexander Scheidler**  
Research associate  
- Mail: alexander.scheidler@iwes.fraunhofer.de

## Fraunhofer IWES - Business Field Grid Planning and Operation

- Techno-economic studies for analyzing, planning, operation, control, stability of power systems  
- Automated Planning Tools  
  (e.g. pandapower [http://www.uni-kassel.de/go/pandapower](http://www.uni-kassel.de/go/pandapower))  
- Operational Tools (algorithms for ancillary services, hardware/software platform for pilot systems)  
- (Co-Simulation) Test Platforms for operational solutions ([www.opsim.net](http://www.opsim.net))  
- Multi-Energy System Planning and Operation (Power, Heat, Gas)  
- Microgrid/ Hybrid System Test Bench and PHiL Tests
Contact

Prof. Dr. Martin Braun  
Chair of Energy Management and Power System Operation  
• Mail: martin.braun@uni-kassel.de  
• Phone: +49 561 804 6202  
• http://www.uni-kassel.de/eecs/e2n

Leon Thurner  
Research associate  
• Mail: leon.thurner@uni-kassel.de

Department Energy Management and Power System Operation - e²n

• Development of models, methods, algorithms and tools for analysis, operation and control, and design of the future decentralized power system with high share of renewable energies. e.g. pandapower  
• Multi-Objective/Perspective/Level Optimisation of the power system  
• Simulation of the power system over time scales and system levels.  
• Resilient Control Design incl. power system stability, network restoration, microgrid structures