Power-Flow-Constrained Asset Optimization for Off-Grid Power Systems Using Selected Open-Source Frameworks

Sabine Auer, Jan Liße, Supriya R. Mandha and Christina Horn
elena international
D-10117 Berlin, Germany
Email: sabine.auer@elena-international.com

***Abstract*—In this paper we aim at creating a review of Open-Source tools that can be used for asset optimization of off-grid power systems while considering power-flow constraints.**

In the past a variety of tools for optimal power flow solutions were published and made available. In particular within the Open-Source community, new tools are getting available by the day. The advantage of Open-Source software is that it allows unlimited sharing of developments and unlimited verification of models. Hence, it is important to keep the overview and find the correct tool for the task at hand.

Off-grid power systems are stand-alone islanded power systems that have no connection to a higher-level continental electricity grid. These systems are predestined/ most suitable to be first-users of hybrid power system solutions.

Since the quality of these tools strongly depends on their use case, within this paper we focus on off-grid power systems and analyze the available tools with respect to several functionalities such as nonlinear vs linear power-flow, AC/DC infrastructure, unit commitment, storage and storage types, multi-time step optimization.

I. Introduction

A. Short Motivation

A recent review on Open-Source tools for energy system optimization [1] illustrated the broad range of Open-Source tools in this context [2] whose modelers are often connected via the Open Modeling Initiative [3]. In this paper we focus on a selection of Open-Source frameworks, that is not meant to be exhaustive. We have chosen the Open-Source frameworks shown in Table I, which are helpful for designing hybrid microgrids or off-grid power systems. This selection is influenced by the ability of Open-Source frameworks to explicitly take grid constraints into account in the optimization problem. Also, a special emphasis is put on tools available in Python or Julia. Where the latter are often still in a strong development phase but at promising intermediate stages and are for this reason included in this review nevertheless. All these frameworks employ functionalities necessary in the process of microgrid planning, hence, they do not represent ready-for-use tools with user-friendly interfaces but demand a certain level of expertise in programming.

B. Requirements from stakeholder interviews

To find out about the requirements for designing off-grid power systems (or hybrid microgrids) the authors of [4] undertook a stakeholder workshop with microgrid planners and operators. For the review of Open-Source frameworks in this paper we partially rely on the comparison criteria derived from the stakeholders' needs. They demand a comprehensive software tool that should be capable of:

- multi-objective optimization (cost of energy, emissions, capacity shortage, RE share)
- system sizing and identification of optimal operational strategies
- adjustable time resolution (15 minutes to hourly time increments)
- grid stability functions (e.g. line overloading and transient stability with spinning reserve and rotating mass requirements)
- modeling different HMG components, for different priorities:
  - high: PV, battery storage (lead acid, Li-ion), multiple diesel generators, controller strategies, inverters, AC- and DC coupling
  - medium: wind, biomass, hydro-turbines, thermal storage, fuel cell, grid
  - low: geothermal components, tidal sources, high-temp. battery storage, pumped hydro storage
- secondary functionality (couple with resource data and load profiles, update uncertain parameters over the course of optimization horizon).

As an outcome of this survey in the following we explain the different modeling techniques and functionalities necessary to cover the stakeholders needs for microgrid or off-grid planning.

II. Off-Grid Power Systems Optimization

The adequate planning of off-grid power system from scratch is a multi-step process. First of all, in the process of system sizing the necessary generation, storage and line capacities need to be determined. This optimization problem

Open-Source Frameworks	Github Link
PyPSA	github.com/PyPSA/PyPSA
PowerSimulations.jl	github.com/NREL/PowerSimulations.jl
PowerModels.jl	github.com/lanl-ansi/PowerModels.jl
Joulia	github.com/JuliaEnergy/Joulia.jl
OEMOF	github.com/oemof
GridCal	github.com/SanPen/GridCal
pandapower	github.com/e2nIEE/pandapower

TABLE I
Link of github repository for each Open-Source framework described in this paper.

may have different objectives and constraints as shown in II-A. Different bus types such as PV, diesel generators or wind have different operational constraints and parameters which need to be included.

With the capacities fixed, to determine optimal operational strategies a unit commitment needs to be run. This also functions as a sanity check of the results to see what generation technology is used to what extend (see II-B).

At the end an additional grid analysis can be useful to undertake a stress test of the grid infrastructure (Section II-C.

A. Step 1: System sizing

There are different levels of complexity and accuracy in optimizing the assets of an off-grid power system. For an economic system sizing the total system costs, which for example include the variable and fixed costs of generation, storage and transmission costs are minimized being subject to technical constraints. In a multi-objective optimization also the cost of emissions etc. may be included. Important note: to include storage in this optimization it is necessary to model a multi-period optimal power flow since the state of charge (SOC) of the batteries make time steps interdependent.

In the following we briefly describe the optimization problem where the different levels of complexity come from different grid constraints.

1) Optimization with Power Grid as a Copper Plate: Lowest complexity starting point with respect to the technical constraints is to undertake a simple power balance approach where the power grid is only treated as a copper plate but its overall capacity may be part of the optimization problem:

$$\sum_n S_n = \sum_n P_n + iQ_n = 0 \tag{1}$$

where S_n is the apparent power with the active power P_n as its real part and the reactive power Q_n as its imaginary part.

2) Linear Optimal Power Flow Problem: In a next step, a linear optimal power flow (LOPF) allows to integrate grid constraints if they can be represented by linear equations. Since both the DC and AC power flow are nonlinear (see Section VI-A3 and VI-A1) only the linearized version of both can be taken into account which according to Sections VI-A4, VI-A2 only holds true for certain assumptions.

Nevertheless, it is then possible to take into account the flows on the lines which in the LOPF problem are then constrained by the line capacities. This ensures that in the planning process already line overloading can be avoided.

3) Optimal Power Flow Problem: For optimizing both AC and DC power systems the linearized power flow equations of the LOPF represents a simplification. For an Optimal Power Flow (OPF) analysis nonlinear constraints, such as the DC and AC power flow equations from eqs. (2) and (6) and also objectives, e.g. quadratic cost functions, can be taken into account. Since the linearisation of power flow (see eqs. (5) and (7)) only holds for transmission lines (with low resistance), ideally an OPF should be used for planning microgrids which show distribution grid characteristics.

B. Step 2: Unit Commitment

After the system sizing is completed, the unit commitment allows to investigate the dispatch of generator and storage for a given load time series under optimal operation. This introduces a time series of new binary status variables, which indicate whether the generator is running (1) or not (0) at a certain point in time. The different options for technical constraints are the same as for system sizing described in Section II-A.

C. Step 3: Grid Analysis

If grid constraints were not part of the optimization problem, the grid analysis should at least follow as an additional step to investigate the stress on the power grid infrastructure.

Additionally to the power flow analysis the investigation of further aspects of grid stability would be useful. This includes a transient stability, a short circuit as well as an harmonic analysis. Especially in small power systems their transient stability (due to low system inertia and a lack of rotating masses) becomes a crucial bottleneck for high shares of renewable energies [5], [6]. Also, islanded power systems have a low short-circuit level and frequency variation, which leads to higher sensitivity to contingencies [7]. Hence, it is important to check the short-circuit current and power of the power grid. And additionally, inverter switching and load fluctuations cause frequency harmonics [8] that may dramatically reduces the lifetime of power system infrastructure [9], [10].

III. Selection of Open-Source Frameworks

A. PyPSA / EnergyModels.jl

PyPSA stands for “Python for Power System Analysis” and is maintained by the Energy System Modelling group at the Karlsruhe Institute of Technology.

PyPSA does not only provide an OPF solver but also many predefined problem descriptions. The scope of PyPSA includes the simulation and optimization of modern power systems with features such as unit commitment, coupling to other energy sectors, and mixed alternating and direct current networks. Possible components are conventional generators, variable wind and solar generation and storage units. PyPSA is designed to scale well with large networks and long time series.

EnergyModels.jl is the Julia implementation of PyPSA and as such work in progress [11].

Planned features of PyPSA are for example short-circuit analysis (following the implementation in pandapower [12]) and an OPF with the full non-linear network equations, following the implementations in PYPOWER and MATPOWER.

B. PowerSystems.jl & PowerSimulation.jl

The PowerSystems.jl package is a data model using Julia structures and is used as the foundational data container for the PowerSimulations.jl package, a Julia package for power system simulations. Both have been developed at the National Renewable Energy Laboratory (NREL).

The objective of PowerSimulations.jl is to provide a flexible modeling framework that can accommodate problems of different complexity and at different time-scales. It incorporates generators (Thermal, Renewable, Synchronous Condensers, and Hydro), transmission (Lines, and Transformers), active flow control devices (DC Lines and phase-shifters), topological elements (Buses, Areas), battery storage, loads (static, and curtailable), services (reserves, inter-regional transfers) and forecasts (deterministic, scenario, stochastic).

Model formulations, that are contained so far, are unit commitment, economic dispatch and DC power flow. Capacity expansion modeling (necessary for system sizing) as well as load flow and contingency analysis are planned features.

PowerSystems.jl can parse the MATPOWER CaseFormat, the PSS/E - PTI Format and the RTS-GMLC data format

C. PowerModels.jl

PowerModels.jl is a Julia/JuMP package for Steady-State Power Network Optimization which has been developed as part of the Advanced Network Science Initiative at Los Alamos National Laboratory [13].

It aims at decoupling problem specifications from the power network formulations in order to compare a wide variety of power network formulations for common problem specifications. Core problem specifications in PowerModels.jl are Power Flow (PF), Optimal Power Flow, Optimal Transmission Switching (OTS) and Transmission Network Expansion Planning (TNEP). Generally, users which are interested in generation asset investment planning and as such focus more on the problem specifications rather than power flow formulations, are referred to other software frameworks. Since the user shall use PowerModels.jl to build packages for problem specifications. This means packages for investment planning and unit-commitment can be developed with PowerModels.jl but will not be part of the core package. Such extensions could then be done in PowerSimulations.jl, for example.

Core network formulations are AC (polar and rectangular coordinates), DC Approximation (polar coordinates) LPAC Approximation (polar coordinates), SDP Relaxation (W-space), SOC Relaxation (W-space) and QC Relaxation (W+L-space).

PowerModels.jl is able to parse Matpower ".m" files and PTI ".raw" files (PSS(R)E v33 specfication).

D. Joulia.jl

Joulia.jl is a recently released bottom-up electricity sector model with high spatial resolution using the Julia programming environment [14], developed by the Workgroup for Infrastructure Policy (WIP) at the Technical University Berlin. It solves an economic dispatch problem by minimizing the total system generation costs constrained by DC power flows on high-voltage transmission lines.

Its demonstration/ application example is the German energy system. Here, it calculates the economic dispatch on an hourly basis for a full year, taking into account demand, infeed from renewables, storage, and exchanges with neighboring countries.

E. Oemof/ micrOgridS

Oemof (short for "Open Energy System Modelling Framework") is an Open Source Python toolbox which provides base packages for energy system modelling and optimisation. [15], maintained and developed within the Reiner Lemoine Institut and the Center for Sustainable Energy Systems. It represents an energy system as a network consisting of nodes and flows connecting them and as such can be mathematically described using concepts from graph theory.

One of the base packages of oemof is the the oemof-solph library which is designed to create and solve linear or mixed-integer linear optimization problems. Currently, there is no integration of grid constraints in oemof. However, including this is a planned feature. Instead it is unique and powerful at supporting the user in the process of assembling the necessary input data for the optimization problem. Here, the demandlib can be used to create load profiles for electricity and heat knowing the annual demand. In addition, the feedinlib library provides an interface between Open Data weather data and libraries (pvlib and windpowerlib) to calculate feedin time series for fluctuating renewable energy sources.

The Open-Source model micrOgridS was developed by applying the Open Energy Modelling Framework (Oemof) [4] to microgrid planning comparable to Homer. This means there is no integration of grid analysis functions. However, stability criteria as rotating mass and spinning reserve are incorporated with empiric values, giving as fractions of the overall system load.

F. GridCal

GridCal is a research oriented power systems software with GUI developed by Santiago Peate Vera. It does not focus on power system optimization and thus system sizing but rather on the grid analysis. For this, it has different power flow formulations (Robust Newton Raphson, Newton Raphson Iwamoto, Fast Decoupled Power Flow, Levenberg-Marquardt, Holomorphic Embedding Power Flow, DC approximation, Linear AC approximation). It is able to undertake Monte Carlo / Latin Hypercube stochastic power flow based on the input profiles, blackout cascading simulations, three-phase short circuit etc. Further GridCal can do overhead line construction from wire scheme, grid reduction based on branch type and it has device templates (lines and transformers).

GridCal is able to parse CIM (Common Information Model v16), PSS/e RAW (versions 30, 32 and 33), Matpower, DigSilent files .DGS (only for positive sequences and devices like loads, generators, etc.). It can export Excel, custom JSON and CIM (Common Information Model v16).

G. pandapower

pandapower combines the data analysis library pandas and the power flow solver PYPOWER to create a network calculation program aimed at automation of analysis and optimization in power systems [12]. It includes validated equivalent circuit models for lines, transformers, switches etc. It is a jointly developed by the research group Energy Management and Power System Operation, University of Kassel and the Department for Distribution System Operation at the Fraunhofer IEE, Kassel. Since pandapower has a strong power grid focus in the optimization of the predefined OPF objectives it includes e.g. maximization of generation, maximization of load, loss minimization. The implementation of asset optimization is not included and can be incorporated by using the PowerModels.jl interface of pandapower.

Pandapower enables an exchange of network data with PYPOWER and MATPOWER.

IV. Comparison Criteria

From the requirements for off-grid planning tools identified through stakeholder interviews (as described in Section I-B) this paper uses the following comparison criteria that are each explained briefly.

A **multi-objective optimization** allows for more than one objective during optimization, e.g. the objective is to reach a certain renewable energy share with minimal cost.

System sizing refers to the ability of the framework to optimize the power system allowing for capacity extensions of generation and storage.

An identification of optimal **operational strategies** for a microgrid of given conventional generation and storage capacity as well as renewable generation and load time series requires a framework to be able of unit-commitment (UC) simulations. It allows to find optimal dispatch scenarios for generation and storage sites.

Sector coupling enables to not only look at the electricity sector but also allows to include the heat and eventually gas sector by means of power-to-heat and power-to-gas as well as means of gas and heat storage.

Stakeholders favor an **adjustable time resolution** which means the time steps in both the investment planning and unit-commitment should not be hard-coded but flexible.

Technical constraints of the optimization problem can be of different complexity as described in Section II-A:

- **Power balance** with the so-called copper plate approach (see Section II-A1)
- **Linear Optimal Power Flow** (see Section II-A2),
- **Nonlinear Optimal Power Flow** (see Section II-A3),
- **Multi-period Optimization** that allows to adequately describe storage systems and their load shifting capability and required inter-time dependent constraints e.g. for the batteries' state-of-charge.

Bus components that were identified by stakeholders to be included in a software tool:

- **variable renewable energy sources (VRES)** such as photovoltaics (PV) and wind that have fixed but fluctuating feed-in time series,
- **inverters** that need to be represented extra to the VRES so that the capacity of battery and solar inverters can be optimized independent of the generators' capacity,
- **battery storage** with pre-implemented features like state-of-charge, costs per MWh etc.,
- **diesel generators** with their ramping limits and minimum up and down times,
- **biomass** that is similar to the representation of diesel generators but with different carriers,
- **hydro turbines** that are for example influenced by water stream velocities.
- **thermal storage** that at least requires the implementation of power-to-heat and some sector coupling feature.
- **fuel cells** require the implementation of electrolyzers.

The renewable energy generator requires **renewable feed-in** data as time series to represent their fluctuating power generation characteristics.

For many purposes a stand-alone grid analysis may be useful. Here, we check for the ability of the framework to undertake an

- **AC Power Flow** (see App. VI-A1)
- **DC Power Flow** (see App. VI-A3)
- **a short-circuit analysis** allows determining the short-circuit power in case of
- **an harmonic analysis** allows to investigate the stress of different frequency harmonics on power grid infrastructure.
- **a transient stability analysis** shows whether after sudden power drops, short-circuit events, line dripping or similar frequency and voltage may return back to their stable point of operation.

V. Evaluation

With the help of the above comparison criteria, that are especially relevant for microgrid planners, we have evaluated the different Open-Source frameworks in Table II.

In Table III we have provided general information on each software framework that allows to evaluate how mature it is, how big the community is, whether it is still maintained and under what type of license it was published.

VI. Conclusion and Outlook

In this brief and non-exhaustive comparison of Open-Source frameworks for microgrid or off-grid power system planning we especially focused on tools in Python and Julia. Each of the tools evaluated has its strengths with respect to some criteria according to its modeling focus.

PyPSA has many pre-implemented problem definitions that easily allow system-sizing and unit commitment modeling with many different bus components including even sector-coupling. Since the developers mainly use PyPSA for German or European power system analyses, the nonlinear optimal power flow formulation is not implemented yet, but is a planned feature for the future. With PyPSA being translated to Julia with the development EnergyModels.jl the computational performance will be significantly increased.

In this review we also included new and promising frameworks from large Open-Source collaborations. This includes PowerSimulations.jl. Despite being only pre-released so far and the documentation being still limited, it already shows a great range of applications for economic dispatch and power-flow analysis. The feature for system-sizing is supposed to be comping up soon. With PowerSystems.jl incorporating the data structure for power system modeling, it is possible to decouple data and simulations. Also, PowerSystems.jl puts great effort in parsing different open and commercial file formats making it easier to incorporate grid models from different sources.

PowerModels.jl is especially strong on solving different power-flow formulations. As such it covers nonlinear power-flow modeling but also optimal power flow problems. However, PowerModels.jl does not provide much pre-implemented models for system-sizing and unit-commitment

Comparison Criteria	Open-Source Frameworks						
	PyPSA	PowerSimulations.jl	PowerModels.jl	Joulia	oemof	GridCal	pandapower
multi-objective optimization	yes and manual	no?	manual	no	yes?	no	no
system sizing	yes	planned	manual	no	yes	no	manual
operational strategies/ UC	yes	yes	manual	yes	yes	no	no
Sector coupling	yes	yes	manual	yes	yes	no	yes
Time resolution	flexible	flexible	flexible	hourly	flexible	flexible	flexible
Optimization Constraints:							
Power Balance	yes	yes	yes	yes	yes	no	yes
Linear Optimal Power Flow	yes	yes	yes	yes	no	no	yes
Nonlinear Optimal Power Flow	no (planned)	yes	yes	no	no	no	yes
Multi-period Optimization	yes	yes	yes	yes	yes	no	yes
Bus components:							
Variable Renewable Energy Sources	yes	yes	manual	yes	yes	yes	yes
Inverters	yes	yes	manual	no	yes	yes	yes
Battery Storage	yes	yes	manual	yes	yes	yes	yes
Diesel generators	yes	yes	manual	yes	yes	yes	yes
Biomass	yes	no?	manual	yes	yes	no	no
Hydro turbines	yes	yes	manual	yes	yes	no	no
Thermal storage	yes	yes	manual	yes	yes	no	no
Fuel cell	yes	no	manual	no	manual?	no	no
Renewable feed-in data	partially	no	no	no	yes	no	no
Grid Analysis:							
DC power flow	yes	yes	yes	yes	yes	yes	yes
AC power flow	yes	no	yes	no	no	yes	yes
Short-circuit analysis	no (planned)	no	no	no	no	yes	yes
Transient Stability	no	no	no	no	no	no	no
Harmonic Analysis	no	no	no	no	no	no	no

TABLE II
OPEN-SOURCE SOFTWARE FRAMEWORKS ARE EVALUATED ACCORDING TO THE COMPARISON CRITERIA DESCRIBED IN SECTION IV. "MANUAL" MEANS SUCH FEATURES ARE POSSIBLE TO BE IMPLEMENTED WITHIN THIS FRAMEWORK.

General Information	Open-Source Frameworks						
	PyPSA	PowerSimulations.jl	PowerModels.jl	Joulia	OEMOF	GridCal	pandapower
license	GPLv3	BSD	BSD	MIT	GPLv3	GPLv3	BSD
programming language	Python	Julia	Julia	Julia	Python	Python	Python
first published	01/2016	upcoming	08/2017	03/2019	11/2015	03/2016	01/2017
last commit	04/2019	04/2019	04/2019	03/2019	02/2019	03/2019	04/2019
contributors	12	8	14	1	24	4	31

TABLE III
THE GENERAL INFORMATION FOR THE DIFFERENT OPEN-SOURCE FRAMEWORK WAS ACCESSED AT GITHUB AT APRIL 15, 2019.

and serves as a basis for further development or may be integrated into tools that lack these features.

Joulia is a well investigated tool for the German power system. It focuses especially on identifying optimal operational strategies. As such it for sure has a greater relevance for microgrids that can be operated both in islanded and grid-connected mode.

Oemof as a framework was developed from a user's perspective and as such does not only allow to model system-sizing and unit-commitment (so far with a copper-plate approach without LOPF or OPF) but also helps the user to aggregate the necessary input data ranging from demand data to weather data to generate renewable infeed time series. Grid analysis functions are implemented with a DC power flow so far. More grid features are planned for the future.

GridCal is a framework that can be used for additional grid analyses but not for the actual planning phase. It can calculate the nonlinear formulations of both DC and AC power flow and is also able to undertake a short-circuit analysis. It can also be of help to parse grid models from other sources.

pandapower's focus is on distribution grid analysis and as such also interesting for microgrids. It has a great data base for power grid components. It is able to undertake both LOPF and OPF simulations. For a grid analysis it can undertake DC, AC and short-circuit analyses. To enable power system sizing, it has to interface with PowerModels.jl though. Also, for unit-commitment modeling other tools need to be included.

So far, none of the tools above is able to undertake a harmonic analysis or a transient stability analysis. E.g. for the latter additional Open-Source frameworks such as PowerDynamics.jl [16], [17] specialized on dynamic modeling need to be used.

All in all, depending on the focus of microgrid planning different Open-Source frameworks or a combination of them may be a good modeler's choice. Since many promising frameworks are still under strong development we expect a further improvement of these tools according to the needs of stakeholders for planning off-grid power systems or microgrids.

ACKNOWLEDGMENT

We would like to thank the Federal Ministry of Economics (BMWi), the European Union and the European Social Fund (ESF) for supporting Sabine Auer, Jan Lie and Christina Horn with an EXIST scholar ship that allowed us to undertake this paper work and attend the Hybrid Power Systems Workshop. Also, we want to acknowledge the support of the Leibniz Association which grants us a Leibniz Research Transfer Project which funds Supriya Reddy Mandha. Also, we want to thank our collaborators from the Potsdam Institute for Climate Impact Research (the Coen group with Prof. Kurths, Frank Hellmann, Paul Schultz and Anton Plietzsch), from Reiner Lemoine Institute (Philipp Blechinger, Paul Berthau and Sarah Berendes), from Karlsruhe Institute of Technology (Dr. Tom Brown, Dr. Jonas Hörsch) and from TU Berlin (Prof. Hirschhausen, Jens Weibezahn, Prof. Raisch and Lia Strenge). Thank you also to Santiage Peate Vera for fruitful discussions on GridCal.

References

[1] M. Groissböck, "Are open source energy system optimization tools mature enough for serious use?" *Renewable and Sustainable Energy Reviews*, vol. 102, pp. 234–248, 2019.

[2] Wikipedia contributors, "Open energy system models," 2019, [accessed 5-April-2019]. [Online]. Available: https://en.wikipedia.org/wiki/Open_energy_system_models

[3] Open Modeling Initiative contributors, "Open models," 2019, [accessed 5-April-2019]. [Online]. Available: https://wiki.openmod-initiative.org/wiki/Open_Models

[4] S. Berendes, P. Bertheau, and P. Blechinger, "Sizing and optimization of hybrid mini-grids with microgrids-an open-source modelling tool."

[5] S. Auer, "The stability and control of power grids with high renewable energy share," Ph.D. dissertation, 2018.

[6] ENTSO-E, "High Penetration of Power Electronic Interfaced Power Sources (HPoPEIPS) – ENTSO-E Guidance document for national implementation for network codes on grid connection," 2017.

[7] A. C. Z. de Souza and M. Castilla, *Microgrids Design and Implementation*. Springer Nature Switzerland AG, 2019.

[8] J. Arrillaga and N. R. Watson, *Power system harmonics*. John Wiley & Sons, 2004.

[9] D. Said, K. Nor, and M. Majid, "Analysis of distribution transformer losses and life expectancy using measured harmonic data," 10 2010, pp. 1 – 6.

[10] "Chapter 6 - lifetime reduction of transformers and induction machines," in *Power Quality in Power Systems and Electrical Machines*, E. F. Fuchs and M. A. Masoum, Eds. Burlington: Academic Press, 2008, pp. 227 – 260. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780123695369500073

[11] Jonas Hrsch, "Energymodels.jl," 2019, [accessed 6-April-2019]. [Online]. Available: https://github.com/PyPSA/EnergyModels.jl

[12] L. Thurner, A. Scheidler, F. Schäfer, J. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "pandapower an open-source python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6510–6521, Nov 2018.

[13] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, "Powermodels.jl: An open-source framework for exploring power flow formulations," in *2018 Power Systems Computation Conference (PSCC)*, June 2018, pp. 1–8.

[14] J. Weibezahn and M. Kendziorski, "Illustrating the benefits of openness: A large-scale spatial economic dispatch model using the julia language," *Energies*, vol. 12, no. 6, p. 1153, Mar 2019. [Online]. Available: http://dx.doi.org/10.3390/en12061153

[15] S. Hilpert, C. Kaldemeyer, U. Krien, S. Günther, C. Wingenbach, and G. Plessmann, "The open energy modelling framework (oemof)a novel approach in energy system modelling," *DoiOrg*, vol. 49, pp. 1–24, 2017.

[16] T. Kittel, S. Auer, and C. Horn, "Sneak preview: Powerdynamics.jl – an open-source library for analyzing dynamic stability in power grids with high shares of renewable energy," *Wind Integration Workshop 2018*, 2018.

[17] S. Auer and T. Kittel, "Modeling the dynamics and control of power systems with high share of renewable energies," *Wind Integration Workshop 2018*, 2018.

Appendix

A. Power Flow Analysis

1) AC Power Flow: The nonlinear AC power-flow equation reads as

$$S_n = P_n + iQ_n = U_n I_n^* = \sum_m U_n Y_{nm}^* U_m^* \tag{2}$$

where S_n is the apparent power at node n with the active power P_n as its real part and the reactive power Q_n as its imaginary part. U_n is the complex voltage at node n and I_n the complex current flowing through n. The value in the admittance matrix Y_{mn} defines whether node m is connected to node n via a line of the given admittance.

The AC power flow equation can be written for its real, $P = \Re(UI^*)$, and imaginary part $Q = \Im(UI^*)$ of the apparent power:

$$P_n = \sum_m V_n V_m (G_{nm} \cos(\phi_n - \phi_m) + B_{nm} \sin(\phi_n - \phi_m)) \tag{3}$$

$$Q_n = \sum_m , V_n V_m (G_{nm} \sin(\phi_n - \phi_m) - B_{nm} \cos(\phi_n - \phi_m)) \tag{4}$$

where V and ϕ are the voltage magnitude and phase of the complex voltage $U = Ve^{i\phi}$ and G and B are the real and imaginary part of the complex admittance, respectively: $Y = G+iB$. This non-linear equation can then be solved e.g. with the Newton-Raphson algorithm.

2) Linearized AC Power flow: Under the following assumption the linearized AC power flow represents a good approximation of the nonlinear AC power flow: reactive power flow decouples from active power flow, that there are no voltage magnitude variations, voltage angles differences across branches are small enough and branch resistances are negligible compared to branch reactances.

With the assumptions (that approximately hold for AC transmission systems) that $\phi_n - \phi_m$ is very small and $G_{nm} \approx 0$ the nonlinear power flow equations can be simplified:

$$\begin{aligned} P_n &= \sum_m V_n V_m B_{nm} (\phi_n - \phi_m) \\ Q_n &= \sum_m V_n V_m B_{nm} \end{aligned} \tag{5}$$

3) DC Power Flow: The nonlinear DC power flow equations reads as:

$$P_n = V_n I_n = \sum_m V_n G_{nm} V_m \tag{6}$$

This non-linear equation is also solved with the Newton-Raphson algorithm.

4) Linearized DC Power flow: The linearized DC power flow equations can be derived from the linearized AC power flow equations with the additional assumption that only active power is considered and the voltage magnitude is $1pu$ for all busses. The equations then correspond to a simple linear relationship:

$$P_n = \sum_m B_{nm} (\phi_n - \phi_m), \tag{7}$$

which can be solved with simple matrix inversion:

$$\phi = B^{-1} P \tag{8}$$