Platform for Microgrid Design and Operation

Johan Windahl[1] Håkan Runvik[1] Stéphane Velut[1]

[1]Modelon, Sweden, {johan.windahl,hakan.runvik,stephane.velut}@modelon.com

Abstract

This paper describes the development and requirement specification of a platform for design and operation of microgrids.

The goal is to have a flexible platform based on open standards that can be used to efficiently solve current and future engineering problems for distributed energy sources and storage systems. By basing it on a unified architecture, collaboration and efficient work flows are enabled.

In this work we investigate the requirements on the model and on the tool side. We also demonstrate how an energy storage system can be designed to reduce the maximum peak power and how it can be operated in the most economic efficient way, taking into consideration constraints and limitations of the system.

This work is based on Modelon's web-based modeling and simulation platform and its Modelica library Microgrid.

Keywords: simulation, optimization, peak shaving, battery storage, energy management, economic dispatch

1 Introduction

Environmental considerations and increasing awareness of infrastructure sensitivity have led to reconsiderations of how energy systems should be best configured. The current highly centralized systems which were developed for large production units such as nuclear and fossil power plants are not suitable for renewable, intermittent and distributed energy sources such as wind and solar (Fathima and Palanisamy, 2015). This motivates the use of microgrids, which are specifically developed for this kind of heterogenous energy production.

A microgrid is a group of interconnected energy sources, loads and storage devices that can operate both connected with the surrounding electricity grid and disconnected in islanding mode. It has the potential to offer increased self-sufficiency and reliability at low cost and reduced environmental impact (Eto *et al*, 2018). Microgrids typically include smaller production units such as photovoltaic arrays, wind turbines, microturbines and generators (combustion engines) as well as storage devices such as flywheels and batteries. Their comparably smaller investment cost makes them attractive to install in remote areas and their capacity for reducing transmission congestion makes them interesting for energy suppliers (Venkatraman and Khaitan, 2015).

2 Background

The challenges of microgrid design and operation are attracting considerable research interest. A survey of these, including islanding transitions, power quality improvements and economic optimization can be found in (Venkatraman and Khaitan, 2015). Optimization is an important tool in this regard. (Fathima and Palanisamy, 2015) contains an overview of the different optimization problems considered in the current research, and the methods and tools used to solve them.

A review of previous work reveals that although there are plenty of methods and tools developed to solve specific problems in this domain, there is no unified tool-chain capable of handling all the relevant problems in both design and operation. Common tools used for microgrid optimization and simulation today include HOMER (HOMER Energy, 2018) and PSCAD (Manitoba Hydro International 2018). HOMER (Hybrid Optimization of Multiple Energy Resources) is a commercial tool originally developed by the National Renewable Energy Laboratory (NREL), that can be used for optimization of the system configuration, component sizing and grid operation. PSCAD is a time-domain based power system simulation tool, which can be applied for microgrid configurations. Tools such as these are quite specialized, lacking the needed flexibility of a complete solution, such as

- changing fidelity levels in the system representation
- introducing highly customized component models in specialized systems
- customizing optimization formulations for specific needs
- combining the physical domains such as thermal, electrical, mechanical to describe future heterogeneous cyber-physical systems

Alternative, highly customized implementations based on more generic tools such as GAMS (General Algebraic Modeling System) (GAMS, 2018) also exist, but the work required to setup such systems makes them impractical for the common user.

DOI
10.3384/ecp19157405

A Modelica-based solution is a good candidate for filling this gap, enabling a flexible and user-friendly framework for the design and operation task. Previous microgrid solutions based on Modelica include (Du *et al*, 2014), where two optimization problems for a Modelica microgrid model are solved using Matlab and FMI-toolbox. In (Kehl *et al*, 2017), a heterogeneous modeling process is proposed, where Modelica models are coupled with Simulink for control design. Additional work focusing only on the Modelica modeling also exist, such as (Roy *et al*, 2014) and (Enerbäck and Nalin Nilsson, 2013). None of these present a platform for modeling, simulation and optimization, enabling the user to solve many of the problems of microgrid design and operation with one tool. Presenting the development and requirements of such a tool is the goal of this paper.

2.1 Design and operation

Design and operation are different types of engineering tasks that typically occur at different places in time using different types of tools of various fidelity levels and time scales. But the design and operation are tightly coupled and would benefit from a unified tool-chain. As an example, a design that requires an aggressive use of the energy storage system may shorten the life-time and lead to a higher total cost compared to a different design. Another example is peak reduction, a design that cannot reduce the peak power may lead to higher costs. In (Fathima and Palanisamy, 2015), the challenges encountered in this regard are divided into generation, control and distribution side optimization problems. Generation side problems consider system design and component sizing, control side problems describe phenomena such as voltage and frequency control and the distribution side considers scheduling and dispatching. The performance indicators and optimization objectives are typically based on economic, environmental or reliability considerations (Luna-Rubio *et al*, 2012). Typical problem formulations are:

1. Which system configuration has the lowest capex and opex cost, for a given load profile and ambient conditions.
2. What is the optimal component sizing under certain reliability requirements.
3. What is the optimal economic dispatch, taking electricity, fuel, maintenance and aging costs into account.

Only a subset of the possible questions can be answered by one single tool, but by providing a flexible framework that can be used by multiple user types, the potential impact is maximized.

3 Requirements

A challenge with designing a model-based framework that should empower different user personas performing various tasks and analyses is the variety of aspects and requirements to consider. Requirements cover the range from model fidelity level, interface design and numerical robustness to user-friendly work-flows, application interface and integration of input data such as solar irradiation and economy parameters.

Here we will look closer into the technical aspects of analysis that are required and the kind of model fidelity that needs to be supported.

3.1 Analyses

A framework that should support both design and operation needs to cover various types of analyses.

System design focuses on the overall system behavior. It requires support of:

- Sizing of components
- Configuration evaluation
- System evaluation

Operation covers a wide area from energy management where energy producers and storage should be used in an optimal way, to grid robustness assuring a stable grid that meets grid code requirements.

For operation we identity following basic analysis types that a tool should support:

- Economic dispatch
- Control design

“Economic dispatch is the short-term determination of the optimal output of a number of electricity generation facilities, to meet the system load, at the lowest possible cost, subject to transmission and operational constraints” (Wikipedia – Economic Dispatch, 2018).

3.2 Technology

The technology needs to be able to support the analyses described in chapter 3.1. In the center there is a model with the right fidelity level to accurately describe the system.

The different analysis types correspond to different computational execution tasks which also set requirements on the tool. The following tasks needs to be supported:

- Simulation
- Multi-simulation
- Optimization

Besides simulation, which is a basic requirement for analyzing the system behavior and to support control design, multi-simulation is required to quickly evaluate configurations. Parallelization of simulations is not a hard requirement but will improve the performance and user experience. Multi-simulation is also needed to perform sensitivity analysis in the presence of uncertainty related to weather or load forecast.

Optimization is required for component sizing and to solve the economic dispatch problem.

3.2.1 Model

Component sizing and control design have different requirements on complexity and time scale of interest. See Figure 1 for an overview of dynamic phenomena and their corresponding time range in an electrical system.

A disadvantage of using a higher complexity level than required is that it results in larger simulation times and potentially also numerical robustness issues. Another issue is that a higher model complexity often has a more detailed parameterization that increases the barrier to get started.

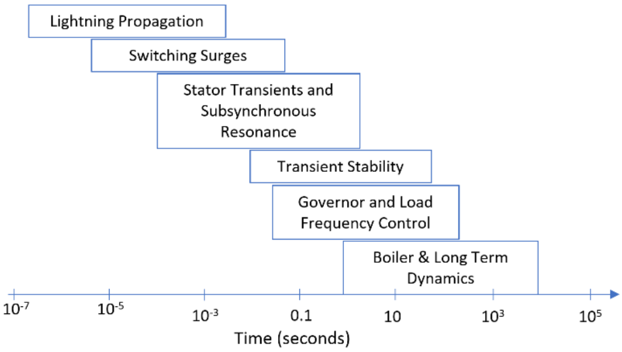


Figure 1 Time ranges of dynamic phenomena. (Sauer et. al, 1997)

One strength of Modelica is the support of generic data types that makes it possible to define a replaceable architecture, where a user can switch the complexity level. Examples of work that take advantage of this in the electrical domain are a multi-level library for electrical machines (Giangrande, *et al*, 2014) and an electric power library that covers arbitrary phases and transforms in one generic framework (Franke and Wiesmann, 2014).

In the different analyses, we categorize model requirements in following two groups:

1. System design and energy management. Time scale of interest from minutes to hours where a simulation case can cover a whole year. Here the main interest is energy flows including production, consumption and storage. Detailed electrical behavior is not of interest with an assumption of a balanced grid that neglects fast electrical transients.
2. Grid robustness. Time scale of interest from milli-second to minutes where a typical simulation may be a few seconds to a few minutes. The main interest is transient electrical behavior with a focus on grid frequency and voltage stability.

For use case 1, it should be possible to efficiently apply optimization. This requires that the model equations are at least twice continuously differentiable (Nocedel and Wright, 2006)

3.2.2 Interface

Modelica models are typically implemented to describe physical properties such as geometry and its corresponding fundamental physical equations, see. e.g. Modelica Standard Library (Modelica Libraries, 2018).

But for models to be used for system design decisions which often include economic aspects, economic information also needs to be included. Examples of economic data are nominal lifetime, capital cost and maintenance cost and interval. This information could be provided in e.g. a Modelica record.

Boundary conditions and input data of importance are:

- Weather data for renewable producers
- Electric load defining the consumption
- Fuel and electricity prices

It would be beneficial to have an interface that handles this type of location-dependent data in a user-friendly way, e.g. by integration with a web-map such as Google maps or Bing.

3.3 Collaboration

A general requirement for model-based development that is also true for micro-grid application is the ability to easily share models and results. Different tasks are typically done by people with different roles. Examples include managers that want to supervise an ongoing new design, and sales personnel that want to find a suitable configuration for a customer by simulating various configurations and present the most appropriate one.

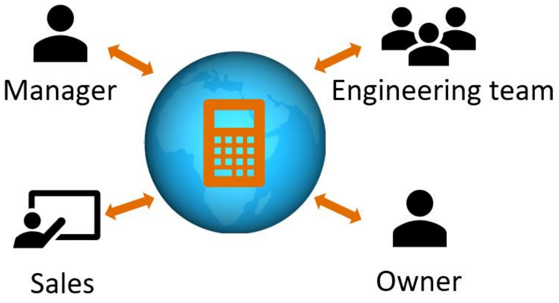


Figure 2 Illustration of a model centric collaboration approach.

Various tasks require different levels of model abstraction and exposure of model information. A sale person may not need to know about the details of Modelica or numerical integration algorithms. Instead he or she may only require an accessible and easy to use interface where a sub-set of all parameters and outputs is shown.

4 Computational Platform

In this chapter we present our framework for the design and operation of micro-grids.

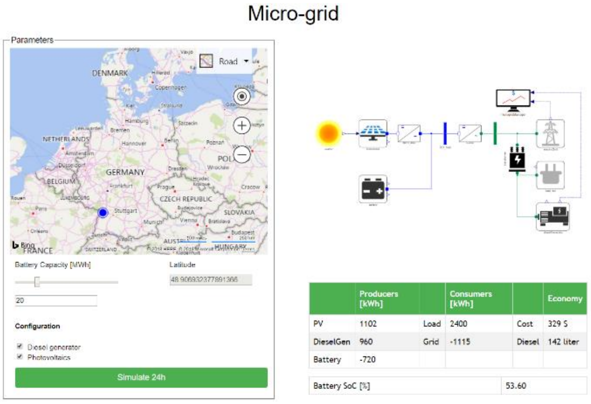
The solution consists of several parts to meet the wide variety of needs and requirements. In the center of the framework is a Microgrid Modelica library and a web modeling and simulation platform from Modelon, including the Optimica Compiler Toolkit for model simulation and optimization.

The following artefacts define the micro-grid framework:

- Modelica models are used to describe the behavior of the components and include data structures with economic data
- Optimica, an extension of Modelica (Åkesson, 2008), is used to formulate optimization problems
- Python scripts are used for optimization workflows and post processing
- HTML and JavaScript are used to create a customizable end-user interface
- FMI is used for deployment.

4.1 Software and Tools

A web-platform matches well with the requirements in chapter 3. It offers an accessible solution that supports a model centric collaboration approach. The web interface makes it possible to create various abstraction levels for a model. This is used to fulfill the different simulation needs of e.g. engineering teams and sales organization.



Micro-grid

Parameters

Battery Capacity [MWh]: 20

Latitude: 48.906932377891366

Configuration

☑ Diesel generator
☑ Photovoltaics

Simulate 24h

	Producers [kWh]		Consumers [kWh]		Economy
PV	1102	Load	2400	Cost	329 $
DieselGen	960	Grid	-1115	Diesel	142 liter
Battery	-720				

Battery SoC [%]	53.60

Figure 3 Simulation web-interface with integration of Bing map into Modelon's web-based modeling and simulation platform web-API.

Another aspect of a web simulation platform is scalability, where simulations can be distributed and executed on the cloud.

Modelon's new web-based modeling and simulation platform builds on open standards. An overview of the architecture is seen in Figure 4. The platform has support for various open formats and programming languages that have been used to create a framework that empowers the user. The following concepts have been used:

1. **View -** with views it is possible to add a visualization layer above the model. The separation makes it possible to create custom views targeting different users using a single model.
2. **Custom function** – using a custom function a python-based workflow can be associated with a model, see Figure 5. In this work it has been used to integrate dynamic optimization in a user-friendly way into the tool.
3. **Web-API**. A web simulation interface has been used to create a customizable end-user interface targeting sales personnel needing an easy-to use simulation tool, see Figure 3.

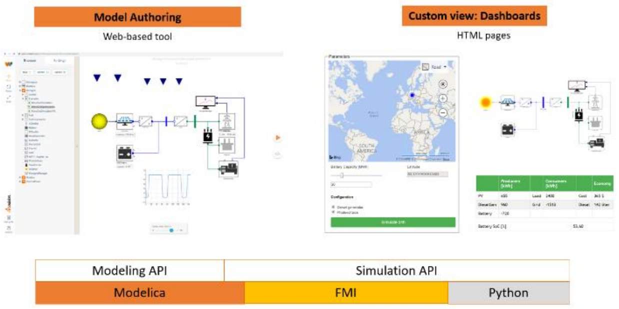
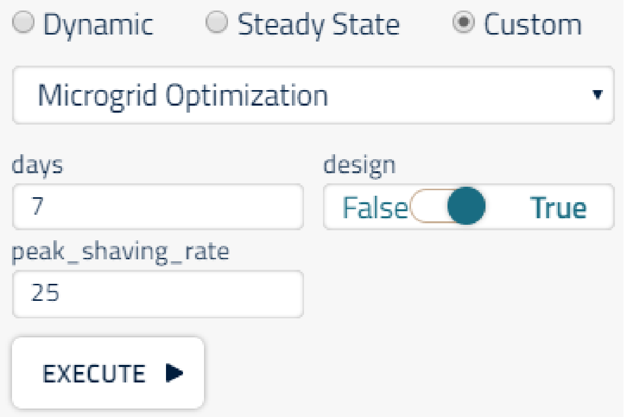


Figure 4 Simplified architecture overview of Modelon's web-based modeling and simulation platform.

4.1.1 Optimization framework

Optimization capabilities are important for many design and operation challenges, as highlighted in the previous analysis. Our solution tightly integrates the optimization formulation with the model formulation using the Modelica extension Optimica. Optimica enables the user to define the objective function, constraints and the optimization time horizon in an *optimization* class, which incorporates the Modelica models via extension or instantiation. The dynamic optimization capabilities of the Optimica Compiler Toolkit are used to solve the dynamic optimization problems defined in this way. The problems are symbolically transformed and transcribed into nonlinear programs (NLP) through direct collocation (Magnusson & Åkesson 2015). These are solved using IPOPT (Wächter and Biegler 2006).


○ Dynamic ○ Steady State ● Custom

Microgrid Optimization ▾

days: 7

design: False ● True

peak_shaving_rate: 25

EXECUTE ▶

Figure 5 Custom parameter interface in Modelon's web-based modeling and simulation platform. The user interface is automatically created based on input definitions in a Python file.

4.2 Model

Microgrid is a Modelica library developed by Modelon. It contains the basic components needed for building models aimed at evaluation and optimization of micro-grid configurations. The different components will be described in the following section.

4.2.1 Microgrid components

- **Weather:** Periodic solar insolation model.
- **Photovoltaics:** Calculates the generated power from solar insolation, based on rated capacity and efficiency.
- **Battery:** The battery model is defined by parameters for capacity, minimal and maximal state of charge and a maximal charge and discharge rate. A DC-connector is used for the charging and discharging.
- **Grid:** Ideal grid model, providing the electricity needed to balance the micro-grid through an AC-connector, at user-defined voltage.
- **Diesel generator:** Provides power to the micro-grid through an AC-connector, based on an input signal. The corresponding fuel consumption is calculated from a fuel curve defined by an intercept coefficient and a slope.
- **Load:** AC electric power consumption defined by input signal.
- **Transformer and inverters:** Efficiency based models changing voltage or current type.
- **Micro-grid manager:** Component containing replaceable control models determining the control strategy for the battery and the diesel generator. By connecting it to external sources, it supports using the micro-grid in optimization.

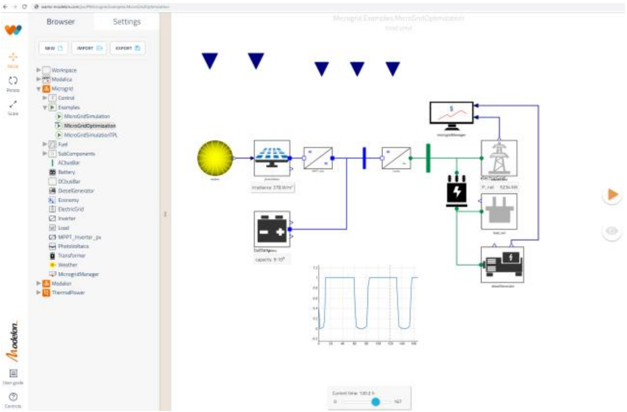


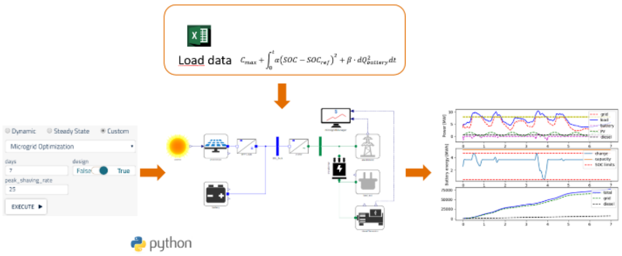
Figure 6 Microgrid Modelica library in Modelon's web-based modeling and simulation platform.

5 Use cases

Two different use cases are considered; peak shaving and economic dispatch. The same Modelica model of the micro-grid is used in both cases, a setup with all components is listed in Section 4.2.1.

The Modelica model is extended to Optimica models, which define optimization problems corresponding to the two use cases. They are determined by choosing different degrees of freedom and objectives in the optimization formulations. The optimization horizon of 7 days is divided into hourly samples in the economic dispatch and 30-minute samples in the peak shaving case.

The same electric load profile is used in both cases. It contains one week of electricity consumption data with a sampling time of 15 minutes. Simplified trajectories are used to represent the electricity price and insolation, but real data could be used for these too, by importing from a file in the same way.



○ Dynamic ○ Steady State ● Custom
Microgrid Optimization
days 7
design False True
peak_shaving_rate 25

Figure 7 Illustration of optimization workflow. Running a custom function starts the execution of a Python script that compiles and optimizes a Modelica model. The result is automatically plotted and returned to the tool.

5.1 Peak shaving

Peak shaving is a method to reduce economic cost by limiting power consumption during high loads. In the modelled micro-grid, the battery is used to decrease the electricity demand from the grid. For this reason, a user-defined maximum constraint on the grid power is imposed on the system, reducing the maximum power peaks. The optimal operation of the micro-grid system is determined under this constraint, assuming constant diesel generator load and penalizing deviations from nominal state of charge for the battery. The battery size is in particular a degree of freedom in the optimization formulation and the derived size is the main result of the use case.

5.1.1 Optimization formulation

The optimization formulation is defined as the minimization of the following cost function:

$$C_{max} + \int_0^t \alpha\left(SOC - SOC_{ref}\right)^2 + \beta \cdot dQ_{battery}^2 dt$$

Where C_{max} is the battery capacity, SOC is the state of charge, SOC_{ref} the nominal state of charge and $dQ_{battery}$ is the battery charge control signal. α and β are constants determining the relative cost of the terms. The control signal is penalized to avoid unreasonably fast control action. The degrees of freedom are the battery size and the battery charge/discharge profile.

5.1.2 Results

Optimization results are displayed in Figure 8. The peak shaving of the grid load is displayed in the upper plot, for each period of high production the grid load is limited. The battery size can be seen in the middle plot, it has been chosen as small as possible without violating the prescribed state of charge limits.

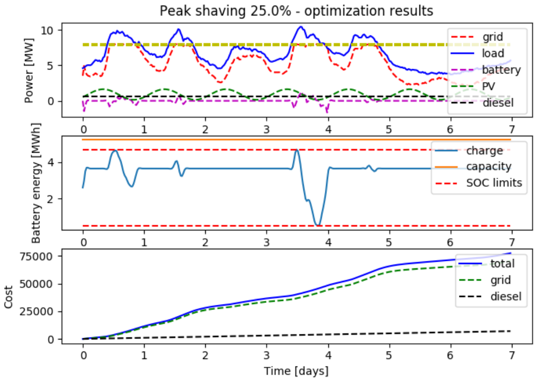


Figure 8 Results of peak shaving optimization.

5.2 Economic dispatch

The economic dispatch use case solves the problem of finding the optimal operation of the units in the micro-grid system, taking variations in load, electricity cost and solar irradiance into account. The main result is the operation mode of the battery and diesel generator and the resulting economic cost.

5.2.1 Optimization formulation

The following integral is minimized, taking economic costs and control signal changes into account:

$$\int_0^t p_{fuel} \cdot F_{fuel} + p_{el} \cdot P_{el} + \alpha \cdot dQ_{generator}^2 + \beta \cdot dQ_{battery}^2 dt$$

Where p_{fuel} and p_{el} are the diesel fuel and grid electricity prices, respectively, F_{fuel} is the diesel consumption, P_{el} is the grid electricity usage, and α and β determine the penalties for the generator load and battery charge control signals.

5.2.2 Results

The results from the economic dispatch experiment can be seen in Figure 9. The battery usage and the diesel generator load follow the variations in the electricity price, so that grid electricity consumption is minimized during the price peaks.

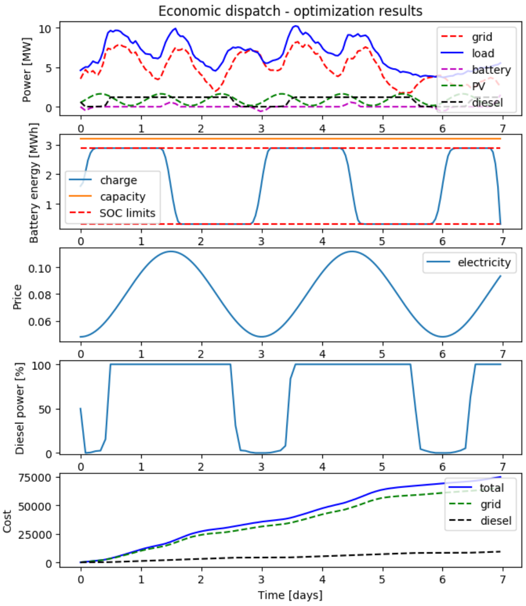


Figure 9 Result of economic dispatch.

6 Conclusions

In this paper we have presented a unified platform for microgrid design and operation with the scope to efficiently solve todays and future engineering problems for distributed energy sources and storage systems. We also investigated the requirements from a model and tool perspective with a focus on both technology and user interface. The framework is based on the open standards Modelica, Python, HTML and JavaScript and is built around Modelon's new web-based modeling and simulation platform. The use case demonstrated successfully that the framework can be used to solve peak shaving and economic dispatch optimization problems.

A major benefit with a Modelica based platform is the openness and flexibility, where it's possible to define an architecture with different fidelity levels that can be used for various types of execution and analysis types.

Future work will focus on adding additional components such as wind power and defining a flexibility architecture for 3-phase AC systems. Other focus areas are improving the fidelity level of already existing components and adding aging effects of batteries in the optimization problem.

References

Wenbo Du, Humberto E. Garcia and Christiaan J. J. Paredis. An Optimization Framework for Dynamic Hybrid Energy Systems. *10th International Modelica 2014 Conference*. Lund, Sweden.

Jonas Enerbäck and Oscar Nalin Nilsson. Modelling and Simulation of Smart grids using Dymola/Modelica. Division of Industrial Electrical Engineering and Automation Faculty of Engineering, Lund University, 2013

Joseph H. Eto, Robert Lasseter, David Klapp, Amrit Khalsa, Ben Schenkman, Mahesh Illindala and Surya Baktiono. The CERTS Microgrid Concept, as Demonstrated at the CERTS/AEP Microgrid Test Bed, Energy Analysis and Environmental Impacts Division Lawrence Berkeley National Laboratory, 2018

A. Hina Fathima and K. Palanisamy. Optimization in microgrids with hybrid energy systems – A review. *Renewable and Sustainable Energy Reviews*, 45, 431-446, 2015

Rüdiger Franke and Hansjürg Wiesmann. Flexible modeling of electrical power systems – the Modelica PowerSystems library. *10th International Modelica 2014 Conference*. Lund, Sweden.

GAMS (2018). URL https://www.gams.com/

Paulo Giangrande, Christopher Hill, Chris Gerada and Serhiy Bozhko. Multi-Level Library of Electrical Machines for Aerospace Applications. *10th International Modelica 2014 Conference*. Lund, Sweden.

Homer Energy (2018). URL https://www.homerenergy.com/

Pierre Emanuel Kehl, Raja Rehan Khalid and Georg Frey. Heterogeneous Modeling: A Need to Model Future Energy Systems. *Power and Energy Student Summit 2017*

Ricardo Luna-Rubio, Mario Trejo-Perea, Damián Vargas Vázq and José Ríos-Morena. Optimal sizing of renewable hybrid energy systems. A review of methodologies. *Solar Energy* 86, Issue 4:1077-1088, 2012

Fredrik Magnusson and Johan Åkesson. Dynamic optimization in JModelica.org. *Processes*, 3(2):471–496, 2015

Manitoba Hydro International (2018). URL https://hvdc.ca/pscad/

Modelica Association Libraries. Available at https://www.modelica.org/libraries, accessed 2018-11-19.

Jorge Nocedel and Stephen J. Wright (2006). Numerical Optimization. New York, NY: Springer New York.

Juan Van Roy, Robbe Salenbien and Johan Driesen. Modelica Library for Building and Low-Voltage Electrical AC and DC Grid Modeling. *10th International Modelica 2014 Conference*. Lund, Sweden.

Peter W. Sauer, M. A. Pai. and Joe H. Chow. (1997). Power System Dynamics and Stability, John Wiley & Sons Inc.

Ramakrishnan Venkatraman and Siddharta Kumar Khaitan. A Survey of Techniques for Designing and Managing Microgrids. *IEEE Power & Energy Society General Meeting*, 2015

Wikipedia contributors. (2018, August 28). Economic dispatch. In Wikipedia, The Free Encyclopedia. Retrieved 13:51, November 12, 2018, from https://en.wikipedia.org/w/index.php?title=Economic_dispatch&oldid=856915225

Andreas Wächter and Lorenz T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

Johan Åkesson. (2008). Optimica - An Extension of Modelica Supporting Dynamic Optimization. *Proceedings of the 8th International Modelica 2008 Conference*. Bielefeld, Germany.