

# Flexible modeling of electrical power systems – the Modelica PowerSystems library

Rüdiger Franke, ABB, Germany – Ruediger.Franke@de.abb.com,  
Hansjürg Wiesmann, Switzerland – HJ.Wiesmann@bluewin.ch

## Abstract

New trends such as renewable power, virtual power plants, electric mobility and smart grids raise the importance of electrical power systems. The systems are manifold, including e.g. DC, single- and multi-phase AC with fixed and variable frequency. Often times such systems cover other physical domains as well, such as rotational mechanics and thermo-fluid. Required system models range from simple flow calculations of active power to detailed transient and asymmetric studies of three-phase systems. Transformed modal coordinates play an important role for the treatment of three-phase AC systems.

The paper introduces the new Modelica PowerSystems library. It covers arbitrary phase systems in one modeling framework. Besides simple generic models that are valid with all phase systems, also large sets of detailed component models for DC and three-phase AC are included. The detailed component models have been ported from the former Spot library to PowerSystems.

The usefulness of the library is shown on behalf of manifold examples and applications carried out so far for power/frequency control, intraday optimization and virtual power plants as well as electrical drive trains of wind turbines.

**Keywords:** *Modelica, electrical systems, power systems, dq0, quasi-static, transient, three-phase AC, DC, wind turbines, virtual power plants, electric mobility.*

## 1 Introduction

The Modelica Standard Library, version 3.2.1, contains the Electrical.Digital and Electrical.Analog sub-libraries, besides Electrical.Spice3 for digital and analog electronics. The Electrical.Multiphase and Electrical.Machines sub-libraries cover detailed

dynamic models of power electronics in the time domain [1]. The Electrical.QuasiStationary sub-library treats AC systems with complex phasors [2].

Some more electrical libraries have been developed during the last years by different authors. One of them, the Spot library, introduced a couple of important features, such as the uniform treatment of natural (abc) and modal (dq0) coordinates in one framework, making it well suited for the detailed modeling of transient effects [3]. Moreover the Spot library supported the per-unit system for parameterization and it contained an extensive set of component models.

Being well suited for detailed transient modeling of three-phase power systems, the Spot library has sometimes been considered overkill if simpler quasi-static models were sufficient. A further drawback of the Spot library was that the same models appeared multiple times in distinct sub-libraries for particular phase systems. The Spot library hasn't been available since Modelica version 3 anymore.

This paper introduces the new PowerSystems library that integrates the Spot library with a new concept of replaceable phase systems. The aim is to support different single and poly phase systems and different mathematical formulations in one framework. In particular this shall cover systems like:

- AC systems, including steady-state, transient, and asymmetric,
- Variable frequency systems, e.g. in wind turbines or for drive control, and
- DC power systems, like HVDC

The replaceable phase systems also enable the application of one and the same library with one universal connector to different levels of detail. This starts from simple power balances and ranges through quasi-static models up to the treatment of detailed transient effects.

## 2 Library Structure and Interfaces

### 2.1 Library Structure

Figure 1 shows the package structure of the PowerSystems library.

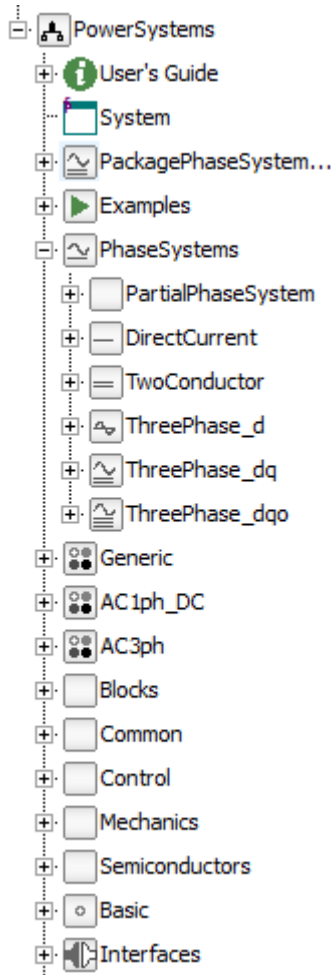


Figure 1: Library structure of PowerSystems

The main packages are:

- **Examples** cover textbook power flow calculations, a demo for power/frequency control and introductory tutorials to the detailed component models;
- **PhaseSystems** defines different mathematical representations of electrical systems;
- **Generic** contains simple component models that can be used with any phase system for basic investigations;
- **AC1ph\_DC** contains detailed component models for DC and one phase AC;
- **AC3ph** contains detailed component models for three phase AC;

- **Blocks** contains signal oriented models, such as Multiplex and Transforms;
- **Control** contains special control blocks, such as Exciters or Governors;
- **Mechanics** provides TurboGroups, complementing a generator with rotor and gears;
- **Semiconductors** define required components such as Diodes and Thyristors.

### 2.2 Interfaces and PhaseSystems

The interfaces define a general power terminal.

```

connector Terminal
  replaceable package PhaseSystem;

  PhaseSystem.Voltage
    v[PhaseSystem.n];

  flow PhaseSystem.Current
    i[PhaseSystem.n];

  PhaseSystem.ReferenceAngle
    theta[PhaseSystem.m]
    if PhaseSystem.m > 0;
end Terminal;
  
```

The connector contains a replaceable PhaseSystem package making it applicable to different mathematical representations. The PhaseSystem defines:

- Number  $n$  of independent current and voltage components
- Number  $m$  of reference angles
- Types used in the connector (Voltage, Current, ReferenceAngle) so that terminals of different phase systems cannot be directly connected
- Functions
  - $j$  “operator”
  - angle and phase
  - phase quantities for voltage, current and power
  - system quantities for voltage, current and power

The vector of reference angles  $\theta[m]$  allows the definition of a rotating reference system for the description of AC systems with modal components. It is known from the Spot library that this enables the treatment of modal quantities in the time domain, covering transient and asymmetric systems as well.

The power Terminal is overdetermined with the reference angles though. This is treated with the operators `Connections.root`, `Connections.potentialRoot`, `Connections.isRoot` and `Connections.branch`. A Modelica tool needs to analyze connection graphs and eliminate redundant equations.

The following table summarizes the `PhaseSystems` that are predefined in the `PowerSystems` library:

PhaseSystem	n	m	Description
DirectCurrent	1	0	One voltage and one current component in natural coordinates
TwoConductor	2	0	Two voltage and two current components for Spot AC1ph_DC components
ThreePhase_d	1	0	One modal component for active power — like DirectCurrent, but converting voltage values to three phase
ThreePhase_dq	2	1	Two modal components for active and reactive power; one reference angle for frequency — cf. complex phasors with variable frequency
ThreePhase_dqo	3	2	Three modal components for active, reactive and dc power; two reference angles for Spot dqo components

A generic steady-state impedance model can be defined as:

```

model GenericInductiveImpedance
  replaceable package PhaseSystem =
    PackagePhaseSystem;
  function j = PhaseSystem.j;
  Terminal terminal_p(
    redeclare package PhaseSystem =
      PhaseSystem);
  Terminal terminal_n(
    redeclare package PhaseSystem =
      PhaseSystem);
  PhaseSystem.Voltage v[:] =
    terminal_p.v - terminal_n.v;
  PhaseSystem.Current i[:] =
    terminal_p.i;
  PhaseSystem.Frequency w =

```

```

    der(PhaseSystem.angle(
      terminal_p.theta));
  parameter
    Modelica.SIunits.Resistance
    R = 1 "active component";
  parameter
    Modelica.SIunits.Inductance
    L = 1/50
    "reactive component";
  equation
    v = R*i + w*L*j(i);
    zeros(PhaseSystem.n) =
      terminal_p.i + terminal_n.i;
    if PhaseSystem.m > 0 then
      terminal_p.theta =
        terminal_n.theta;
      Connections.branch(
        terminal_p.theta,
        terminal_n.theta);
    end if;
end GenericInductiveImpedance;

```

Note the use of the function `j` that generalizes complex calculations known from quasi-static AC models to arbitrary phase systems. `ThreePhase_dq` with two model components for active and reactive power defines a multiplication with the complex `j`:

```

function j
  input Real x[n];
  output Real y[n];
algorithm
  y := {-x[2], x[1]};
end j;

```

The simpler `ThreePhase_d` neglecting reactive power defines:

```

function j
  input Real x[n];
  output Real y[n];
algorithm
  y := zeros(n);
end j;

```

The more detailed `PhaseSystem_dqo` also considers a component for dc power in asymmetric systems, besides active and reactive power. It defines:

```

function j
  input Real x[n];
  output Real y[n];
algorithm
  y := cat(1, {-x[2], x[1]},
    zeros(size(x,1)-2));
end j;

```

The GenericInductiveImpedance model adapts to the selected PhaseSystem.

### 3 Component models and Examples

The PowerSystems library contains simple examples, including textbook power flow calculations, demonstration of power/frequency control, and tutorials for the modeling of power system transients.

#### 3.1 Network flow calculations

`PowerSystems.Examples.Network` contains examples that treat power flow with the quasi-static ThreePhase\_dq by using simple generic component models.

Figure 2 shows the Generic component models used by these examples.

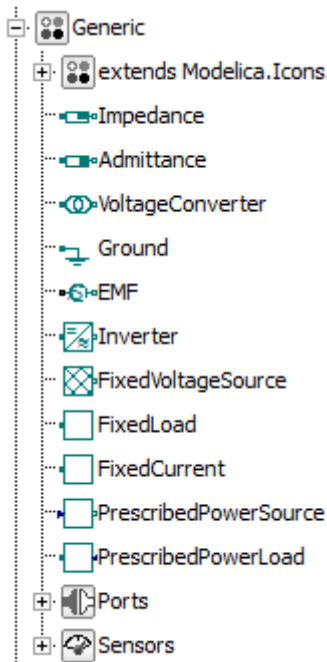


Figure 2: Generic component models

The NetworkLoop example implements a simple power flow model; see Figure 3. The textbook example is taken from [4].

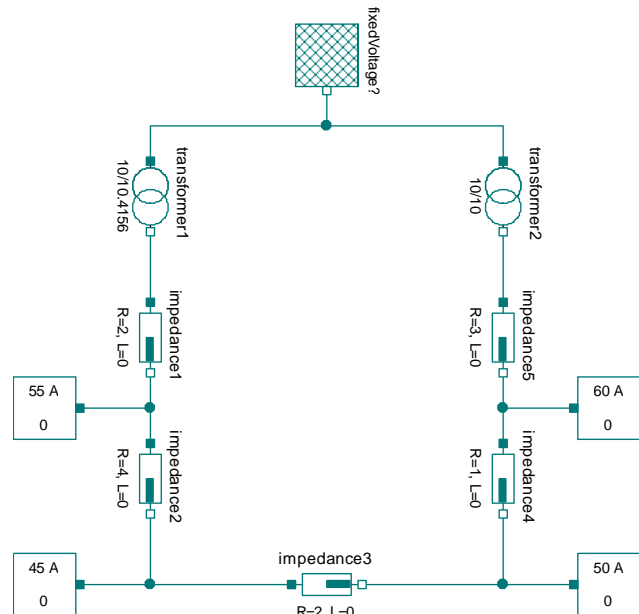


Figure 3: NetworkLoop example

The example demonstrates that the PowerSystems library enables the direct implementation of power flow models using established graphical representations. The graphical representation differs from electrical circuits as treated e.g. with the Electrical.Analog library. The FixedVoltageSource has only one terminal, as opposed to two pins in the Electrical.Analog library. Moreover there is no Ground component needed to treat loops.

#### 3.2 PowerWorld

`PowerSystems.Examples.PowerWorld` demonstrates the principles of power/frequency control; see Figure 4.

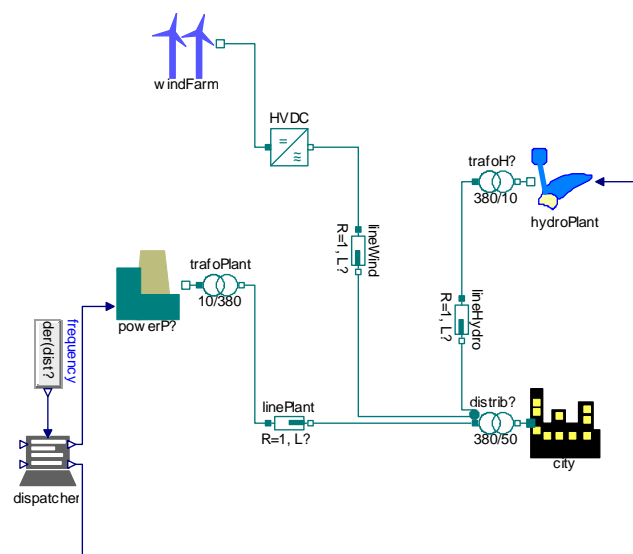
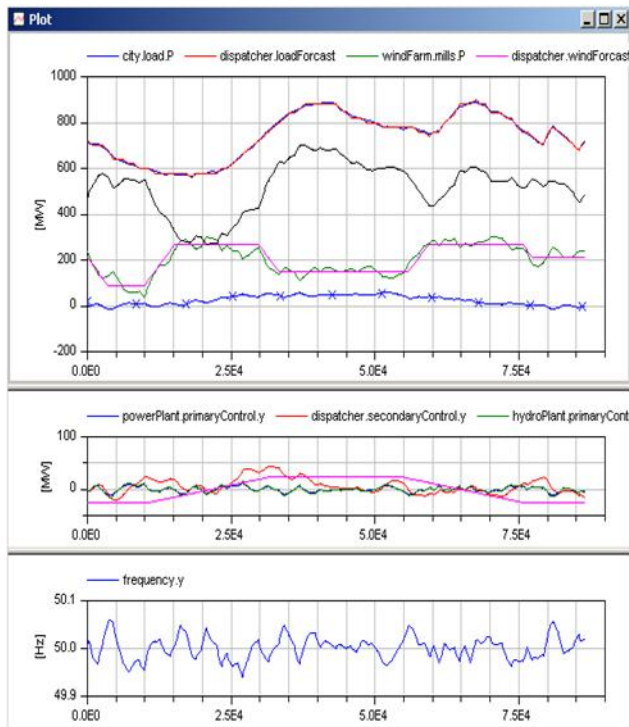


Figure 4: PowerWorld example

The example uses Generic components with the quasi-static ThreePhase\_dq because fast electrical transients and asymmetries are neglected. Rotating masses in the power plants and active power/frequency control determine the system dynamics.

Figure 5 shows simulation results covering one day of operation.



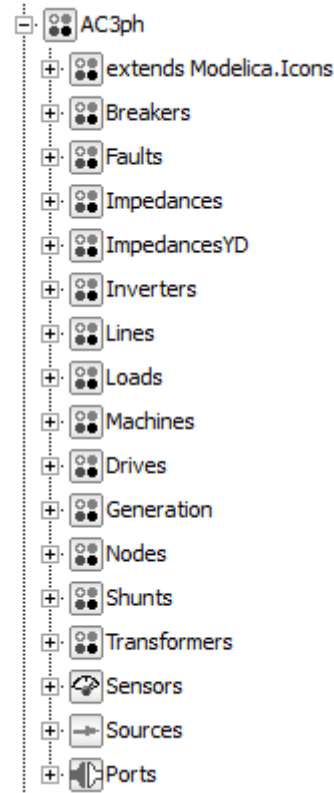
**Figure 5: Simulation results for the PowerWorld example**

The wind farm introduces disturbances into the system, caused by deviations of the actual weather from the forecast – see forecast in magenta vs. actual feed-in in green in the upper plot. A misbalance between production and consumption leads to deviations of the grid frequency from the nominal value of 50 Hz. The conventional power plant and the hydro plant accommodate for deviations with primary frequency control by means of proportional control. The remaining control error is exploited by the load dispatcher to determine the misbalance and to send out set points for secondary frequency control. This way the fast primary frequency control is relieved with the slower secondary frequency control, so that subsequent disturbances can be treated as well. The overall balance between production and consumption is maintained.

The example does not cover tertiary frequency control and intraday optimization that form further control loops on top of secondary and primary frequency control – see also the application in section 4.2 below.

### 3.3 Spot examples

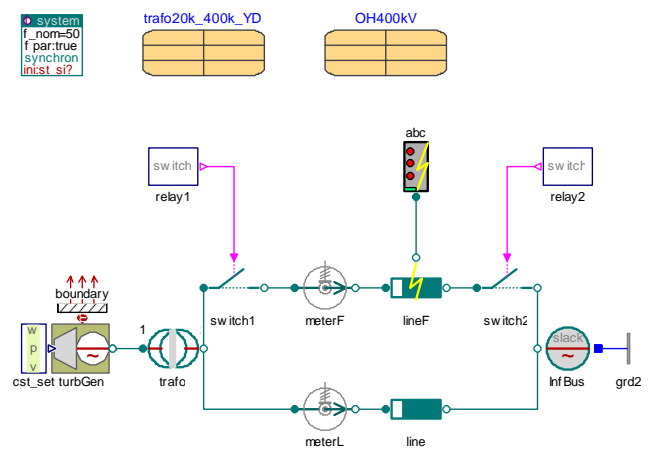
PowerSystems.Examples.Spot has been ported from the Spot library [3]. They use the components from AC1ph\_DC and AC3ph; see Figure 6.



**Figure 6: AC3ph component models**

The extensive AC3ph components range from simple impedances through special devices such as breakers up to transformers, inverters and electrical machines. Moreover they cover composed models for Drives and Generation.

The Spot examples serve as tutorials for detailed transient modeling of electrical power systems. Figure 7 shows the fault clearance by short-time line switched off as example.



**Figure 7: DoubleRXline example**

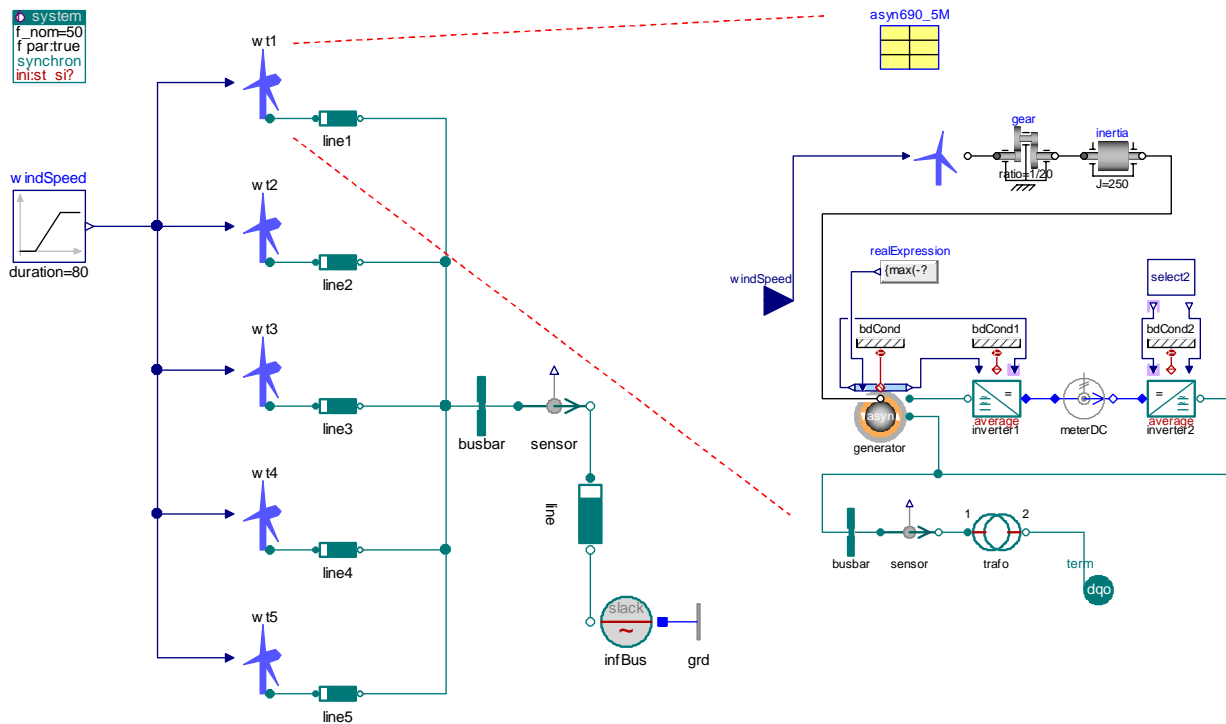


Figure 8: Wind farm model (electrical side)

## 4 Applications

### 4.1 Drive trains of wind turbines

The ITEA research project MODRIO aims for model driven online application. One application area addresses wind turbines. They are challenging because multiple physical domains need to be covered to master the overall system dynamics. The PowerSystems library has been used to implement transient models of electrical drive trains. Figure 8 shows an example with doubly fed induction generator (DFIG) on the right hand side. Multiple such wind turbine models have been assembled to a wind farm model as shown on the left hand side.

The electrical model will be integrated with a model for the mechanical side by SIMPACK.

### 4.2 Intraday optimization of municipal power

Traditional power/frequency control runs into limitations when facing large uncertainties due to increasing use of renewable energy. Intraday optimization reacts on new conditions by re-planning the power production. Such an intraday optimization was developed within the project econnect Germany for the investigation of the integration of emerging electric mobility with power generation.

The PowerSystems library has been used as basis for an optimization library. The optimization library predefines component models that contain, besides physical equations, also optimization constraints and objective terms. This way optimization programs can be implemented graphically like simulation models.

Figure 9 shows an exemplary model. The library is shown as tree on the left hand side. The intraday optimization treats a virtual power plant (VPP) consisting of multiple power generation units, such as wind, solar, hydro, as well as combined heat and power plants. Available storage capacities include heat buffers and a large battery. Moreover the exploitation of electric cars and of heat pumps as controllable consumers is investigated.

The Modelica model is translated to C code and exported as Functional Model Unit (FMU) to the ABB control system. The FMU is used a basis for a large-scale mixed integer optimization program, providing a model predictive control of the VPP. The online optimization approach is further discussed in [5].

Figure 10 shows the result of one optimization run. Dotted lines mark the original day-ahead plan. Caused by a surplus of wind production, the intraday optimization reduces the use of combined heat and power BHKW plants (green areas) until the heat buffers reach their lower limit (red areas). A battery that is installed in a parking house is charged at times when there is too much power in the grid. More details of the intraday optimization are given in [6].



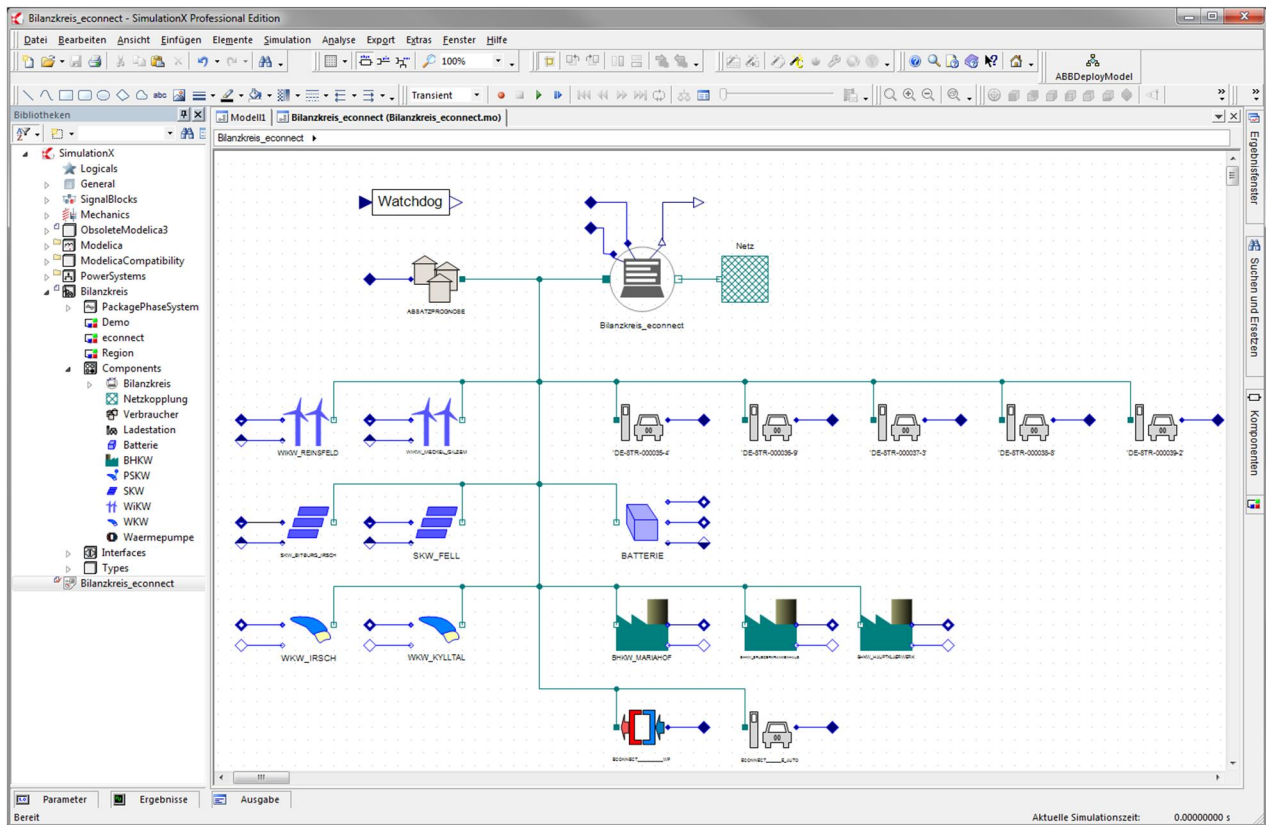


Figure 9: Optimization model for municipal power

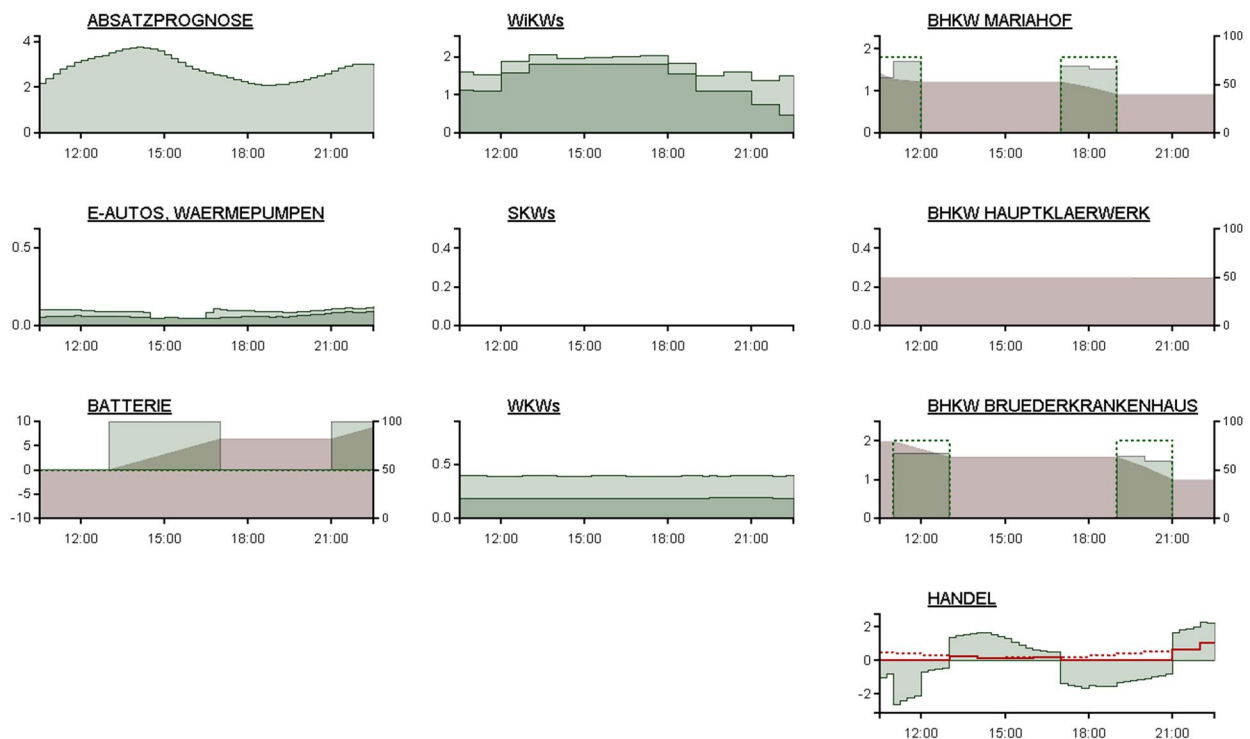


Figure 10: Exemplary intraday optimization result

## 5 Conclusions

The PowerSystems library started as a concept study for defining configurable phase systems, together with simple generic component models and universal power connectors. Later on the Spot library was integrated, so that extensive sets of detailed component models for DC and three-phase AC are available.

This way PowerSystems provides the comprehensiveness of Spot on the one hand side. Moreover, the new replaceable phase systems enable models at different level of detail, including also simple models for active power balances, besides symmetric and asymmetric AC models for active and reactive power, both steady-state and transient. A couple of examples and applications show the usefulness of the library at different levels of detail.

The PowerSystems Library is available under the Modelica license, version 2, at: <https://github.com/Modelica/PowerSystems>

Given sufficient interest by others, future work shall address the integration of the PowerSystems library or its concepts with the Modelica Standard Library.

## Acknowledgements

We thank Martin Otter who has converted the original Spot library to Modelica 3. This significantly simplified its integration with PowerSystems.

This work was supported in parts by BMBF within the ITEA2 project MODRIO (Model Driven Physical Systems Operation) – project id: ITEA 2 – 11004.

The project econnect Germany is funded by the federal ministry for economy and technology due to a decision of the federal parliament.

## References

- [1] C. Kral, A. Haumer: Modelica libraries for dc machines, three phase and polyphase machines, *Modelica Conference 2005*, Hamburg, March 2005. [https://www.modelica.org/events/Conference2005/online\\_proceedings/Session7/Session7a2.pdf](https://www.modelica.org/events/Conference2005/online_proceedings/Session7/Session7a2.pdf)
- [2] A. Haumer, C. Kral, J.V. Gragg, H. Kapeller: Quasi-Static Modeling and Simulation of Electrical Circuits using Complex Phasors, *Modelica Conference 2008*, Bielefeld, March 2008. <https://www.modelica.org/events/modelica2008/Proceedings/sessions/session2d3.pdf>
- [3] B. Bachmann, H. Wiesmann: Advanced Modeling of Electromagnetic Transients in Power Systems, *Modelica Workshop 2000*, Lund, September 2000. <https://modelica.org/events/workshop2000/proceedings/Bachmann.pdf>
- [4] D. Oeding, B.R. Oswald: *Elektrische Kraftwerke und Netze*, Springer, 7th edition 2011.
- [5] R. Franke, J. Doppelhamer: Integration of Advanced Model Based Control with Industrial IT. In: *Assessment and Future Directions of Nonlinear Model Predictive Control*. R. Findeisen, F. Allgöwer, L. Biegler (editors). Springer Verlag 2007.
- [6] R. Franke, S. Kautsch, M. Blaumann, L. Vogelbacher: Integration von erneuerbaren und konventionellen Energien – Intraday-Optimierung, Pooling und vorausschauende Zielsollwertführung, BWK 9/2013.