Electrical Power System Modelling in Modelica – Comparing Open-source Library Options

Dietmar Winkler

University College of Southeast Norway, dietmar.winkler@usn.no

Abstract

The past modelling of electrical powers systems used to be a the domain of a few major power system modelling tools.

Commercial tools[1] like DIgSILENT PowerFactory [3], POWERSYS EMTP-RV [17], PSCAD™ [14], Siemens PSS/E® [20] are just some examples. All of them very powerful but with their own proprietary format for the models created makes it hard to exchange validated models of power systems.

In order to disconnect the dependency of the power system *model* from the power system *tool* a project funded by the European Commission was started called "iTesla – Innovative Tools for Electrical System Security within Large Areas" [7]. One of the results of the project was the creation of a open-source modelling library called `iPSL` [24].

The library was created using the open-source Modelica modelling language which in turn allows to be used with several different Modelica modelling tools. The strong focus during the project was the validation of the models contained in the library. Most components stem from other tools (e.g., PSAT [15], PSS/E® [20]).

But the `iPSL` is by far not the first open-source power system library written in Modelica. Other libraries

which have been around before are: `SPOT` [1], `ObjectStab` [11] and `PowerSystems` [6] (an updated subset of `SPOT`). In addition the `iPSL` library has been "forked" as `OpenIPSL` [21] by SmartTS Lab which is one partners of the original "iTesla" project.

This paper is going to investigate the differences of the different available power system libraries. Point out their specific strength and weaknesses with respect to user-friendliness, robustness, physical representation and validation.

Keywords: electrical power systems, modelling, simulation, modelica, open-source, libraries

[1] Listed in alphabetical order.

1 The Libraries

1.1 `SPOT`

The "Modelica Power Systems Library SPOT" [25] is one of the first Modelica libraries to model power systems both in transient and steady-state mode. As a note, the capitalised spelling was changed to `Spot` in the Modelica package in order to follow Modelica naming conventions.

1.1.1 History

The theory behind the library was first presented at the *Modelica Workshop 2000* in "Advanced Modeling of Electromagnetic Transients in Power Systems" [1].

1.1.2 Concept

The `SPOT` library can be described as a general purpose library suitable to model and simulate power systems in DC and AC. Due to the special role of 3-phase AC systems it received a special role in the library. AC systems are periodically driven systems. An inherent limitation for efficient integration of such systems is the necessary small step size

Traditionally, power systems have always been treated differently depending on the size. Whilst smaller systems can be investigated using general transient methods, larger systems are normally restricted to the "power-flow" approximation. Normally different simulation tools are used in order to serve the different methods.

The `SPOT` library treats both cases within one common framework and brings the power-flow approximation closer to the general case. This is achieved by use of a transformed representation of the electrical equations. The transformed equations contain a steady-state or power-flow limit, obtained by choosing a synchronously rotating reference system and omitting the time-derivative. That implementation leads to a considerable increase of simulation speed for linear or linearised symmetric systems, compared to the direct representation. The increase in simulation speed is only available for linear systems or as long as when sources do not contain harmonics. As the electric equations are valid in reference systems with arbitrary angular orientation, the standard cases 'inertial' (non-rotating) and 'synchronous' (rotating with electrical frequency) system can simply be obtained by an appropriate parameter choice. The same goes for the choice of Frequency, where $50\,Hz$ and $60\,Hz$ is an option.

1.1.3 Library Structure

The structure of the SPOT library is a bit special since it is shipped as two packages. One containing the library and its components itself, see Figure 1, and an additional examples package.

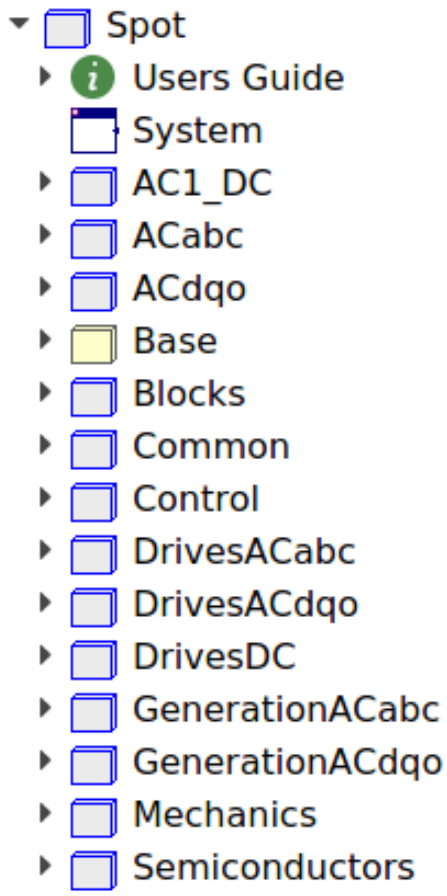

Figure 1. Structure of the SPOT library

The library tries to sort the main application areas into:

- 1-phase and DC applications
- 3-phase periodic signals, i.e., "abc"
- 3-phase transformed, i.e., "dq0"

Components that are common to all the main application areas only appear once. For others one has to decide which system sub-package to choose from in order to create the appropriate simulation model. This split into several sub-packages of identical components but with different equation systems is one of the short-comings from the usability point of view. Something that has been improved in derivatives of the SPOT library.

1.1.4 Licence

The SPOT library has been released under the "Modelica License 1.1" [22], a licence similar to the BSD license.

1.1.5 State of Development

The last official feature release "v0.706" dates back to the 11th September 2007 and was compatible with the Modelica 2 language specification. Some years later the library was cleaned up and upgraded to Modelica 3 language specification and using the Modelica Standard Library 3.2.1. This version was called "v0.706.1" and did not contain any additional functional changes.

The original Author, Hans Jürg Wiesmann, died in 2015 and the SPOT library is no longer officially developed. However some successors using parts of or all of the SPOT library are available and are described later on.

1.2 ObjectStab

The ObjectStab library [12] is a Modelica Library for Power Systems Voltage and Transient stability simulations.

1.2.1 History

This library was just like the theory behind SPOT presented at the *Modelica Workshop 2000* in "ObjectStab Library – A Modelica Library for Power System Stability Studies" [11].

1.2.2 Concept

Where the SPOT library was more a general purpose power systems library, the ObjectStab library concentrates mainly on electrical transmission and generation systems. Standard assumptions for multi-machine transient stability simulations are made. For example generator stator and network time constants are neglected and voltages and currents are assumed to be sinusoidal and symmetrical. Most components are modelled according to the guidelines form the book given in "Power System Dynamics and Stability" [13].

1.2.3 Library Structure

The structure of the ObjectStab library is shown in Figure 2.

It contains:

- Generators with constant frequency and voltage as slack or PV nodes, or using 3rd or 6th order dq-models with excitation and prime mover control systems.
- Transmission lines in π-link or series impedance representation.
- Reactive power compensation devices; shunt reactors, shunt capacitances and series capacitances according to [13].
- Fixed ratio transformers.
- On-load tap changing transformers (OLTC) modelled as detailed discrete models or using their corresponding continuous approximations according to [19].

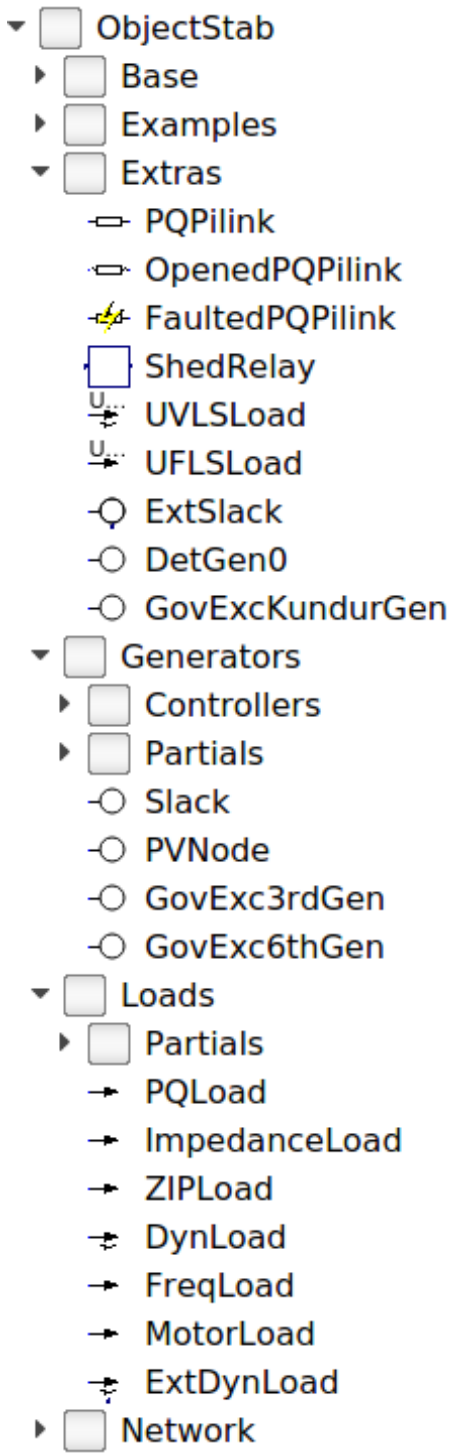

Figure 2. Structure of the `ObjectStab` library

- Static and dynamic loads, including induction motor according to [13] and generic exponential recovery loads according to [9].
- Busses.
- Faulted lines and busses including fault impedance.

1.2.4 Licence

The `ObjectStab` library has been released under the "Modelica License 1.1" [22], a licence similar to the BSD license.

1.2.5 State of Development

The last official feature release "v1.0c" dates back to the 25th September 2002 and was compatible with the Modelica 1 language specification. That release is still using the Modelica Standard Library 1.6 and `ModelicaAdditions 1.5`, something that makes it hard to use with current Modelica tools.

In 2014 and 2015 some effort was made to upgrade the `ObjectStab` library to current Modelica, i.e., make it work with Modelica Standard Library 3.2.1. This work was executed by Dietmar Winkler and with some help of the late Hans Jürg Wiesmann. However due to the dependencies of Petrinet models that are no longer available in the current Modelica Standard Library and the lack of any correct and Modelica standard conform state machine successor, some models will not work fully in standard Modelica tools.

There have been some non-official updated versions of the `ObjectStab` library distributed to individuals to by the original Author but those have never been published officially and contained changes were not documented.

The current state of the development version can be obtained from [12].

1.3 `PowerSystems`

The `PowerSystems` library [26] is intended for the modelling of electrical power systems at different levels of detail both in transient and steady-state mode.

1.3.1 History

Given that latest version of `ObjectStab` was published in 2002 and the the `SPOT` library has not been further developed since 2007 a new attempt to revive power system modelling using Modelica was presented at the *10th International Modelica Conference* in the paper "Flexible modeling of electrical power systems – the Modelica PowerSystems library" [6]. That paper presented the `PowerSystems` library which could be called a slimlined successor to the `SPOT` library. Interesting is that the original developer of `SPOT` was helping developing this new `PowerSystems` library.

1.3.2 Concept

The `SPOT` library was offering nice features like use of per-unit systems or SI-unit system depending on choice. At the same time the multiplication of components for the different phase-systems was creating a lot of code-duplications.

The `PowerSystems` library introduces a nice replaceable phase-system which allows to have one instance of a component but with different equation systems activated. This makes the library more compact and user-friendly. It

can be used to calculate simple power balances but also allows for usage of quasi-static models up to the treatment of detailed transient effects. The main focus are electric power network applications, something that lead to only using only part of the SPOT library

1.3.3 Library Structure

The structure is shown in Figure 3 and if compared with Figure 1 the heritage is quite obvious.

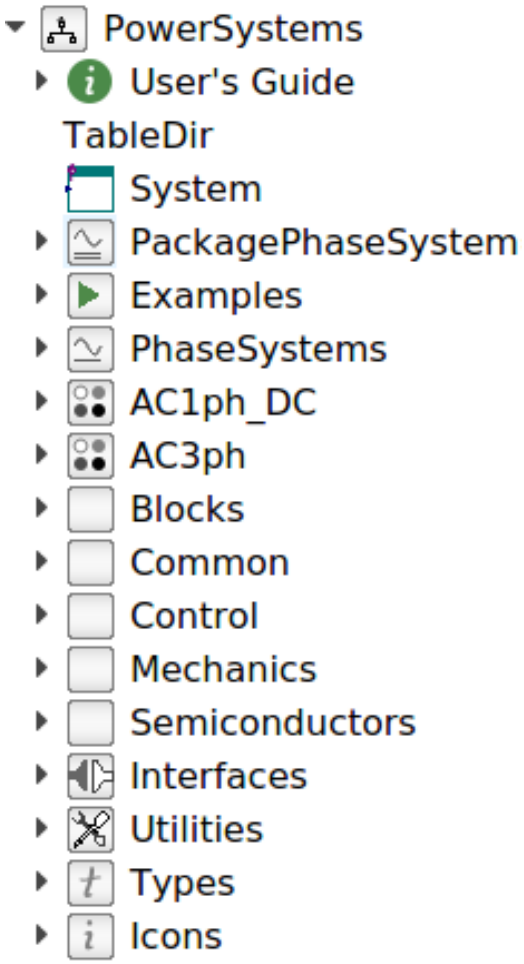

Figure 3. Structure of the PowerSystems library

One can see that the PowerSystems library got rid of unnecessary duplications of sub-packages (by use of the phase-systems feature). Also the Examples sub-package is now part of the library as is much more common in Modelica library, rather than having them as external package.

1.3.4 Licence

The PowerSystems library has been released under the "Modelica License 2" [23].

1.3.5 State of Development

The last official feature release "v0.6.0" is from the 18th January 2017 and is under active development on GitHub [26] by Rüdiger Franke and contributors. The progress and issues can be followed publicly.

1.4 iPSL

The iPSL stands for "iTesla Power System Library" [8] and is a Modelica library developed as part of the iTesla project [7].

1.4.1 History

Developed as part of the "iTesla – Innovative Tools for Electrical System Security within Large Areas" - project (supported by the European Commission under the 7th Framework Programme). The original project ran from 2012 to 2016.

1.4.2 Concept

The library contains a set of power system component models for phasor time domain simulations especially. The goal was to build a model library that allows to use different tools to compare results in order to validate power system models. Hence instead of building up models using generic electrical equations to describe the physical behaviour, reference models that are used other existing power system tools are implemented (e.g., EUROSTAG [4], PSAT [15], PSS/E® [20]). That way the results can first be compared to the different tools they originate from and then also with each other.

1.4.3 Library Structure

The structure of the iPSL is shown in Figure 4.

The structure demonstrates that instead of having one model for a specific machine, e.g., generator, there are several models based on the originating implementation reference.

This is quite a special way to build up a modelling library and brings some caveats which will be discussed later.

1.4.4 Licence

The iPSL has been released under the "Mozilla Public License Version 2.0" [5].

1.4.5 State of Development

The last official feature release "v1.1.1" is from the 9th June 2017 and is under active development on GitHub [8]. The development progress and issues can be followed publicly.

1.5 OpenIPSL

The OpenIPSL stands for Open-Instance Power System Library and is a fork of of the iTesla Power System Library - iPSL, currently developed and maintained by the SmarTS Lab research group, collaborators and friends.

1.5.1 History

When the original iTesla project was completed on 31st March 2016, part of iTesla consortium continued the development of the iTesla Power System Tool [18] which

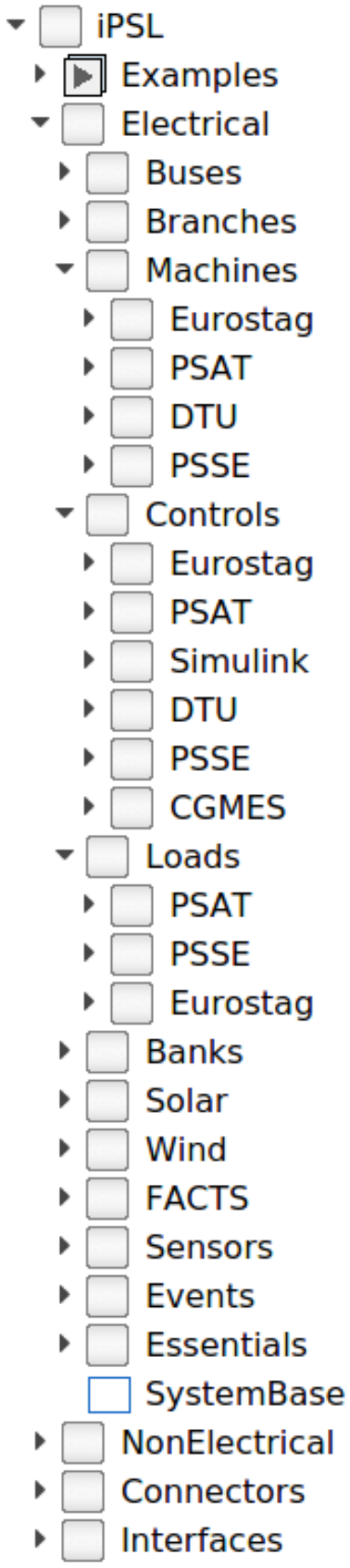

Figure 4. Structure of the `iPSL`

utilises the `iPSL`. Hence the `iPSL` continued to be developed in such a way to fulfil the needs of the iTesla Power System Tool which not necessarily fits the needs of general power system researcher and teachers.

The developers attached to the SmartTS Lab decided therefor to fork the `iPSL` under the new name `OpenIPSL`. Approximately 75 % of the original `iPSL` developers continue the development of `OpenIPSL`.

1.5.2 Concept

In principle, the concept of `OpenIPSL` is of course the same as the `iPSL`. But in contrast its to be used as a research library with maximum compatibility with OpenModelica [2] (to provide a free/libre and cost-free alternative for power system dynamic simulation)and to provide as many as possible typical "test networks" for use in research and teaching. Not having to serve a certain tool interface, it is now also free to adapt the models to a more object-oriented coding style.

1.5.3 Library Structure

The structure of the `OpenIPSL` is shown in Figure 5.

Due to the recent split from the `iPSL` the general structure is still very similar. The difference is mainly some better usage of Modelica language features and cleaner diagram representation. In addition the SMART Lab group removed all sub-packages that are based on models from tools where they do not have access to and/or are lacking any documentation on the implementation background.

1.5.4 Licence

The `OpenIPSL` has been released under the “Mozilla Public License Version 2.0” [5].

1.5.5 State of Development

The last official feature release “v1.0.0” is from the 16th December 2016 and is under active development on GitHub [**oipsl-lib**]. The development progress and issues can be followed publicly.

2 Example Application

In order to demonstrate how the libraries appear visually when modelling an example power system was chosen. The following figures show the “Example 13.2” from [10] which is used to analyse the transient stability of a power system which experiences a three-phase to ground fault at one of the busses.

Figure 6 shows the `OpenIPSL` version, Figure 7 shows the `PowerSystems` version and Figure 8 shows the `ObjectStab` version.

The representation from the `iPSL` would be very much alike the `OpenIPSL` though lately also dynamic displays of the dynamic power-flow values were added to the `OpenIPSL` on the diagram. Similarly the model from `SPOT` would have looked identically to that of the `PowerSystems` library, since the graphic properties have not been changed much.

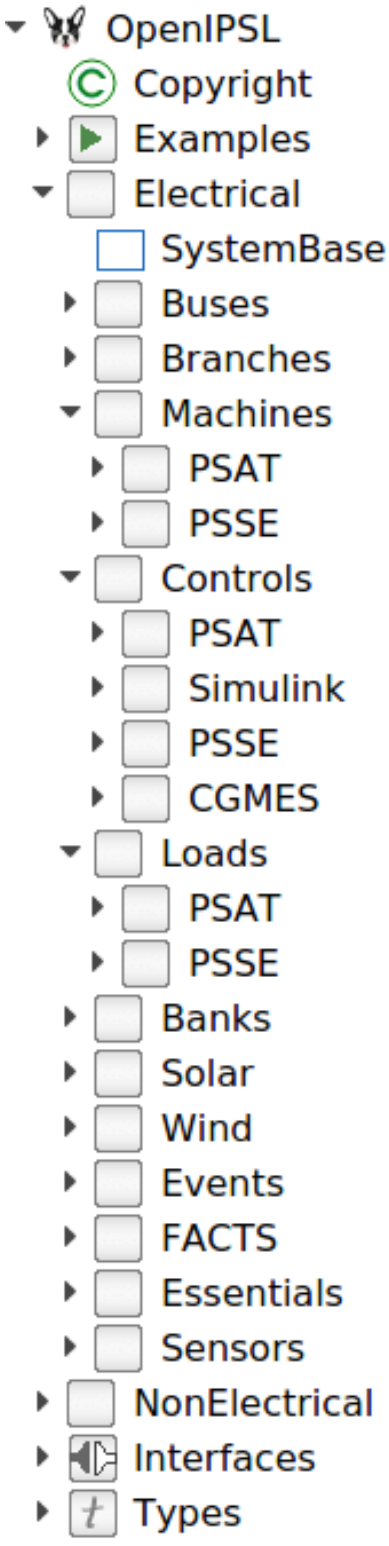

Figure 5. Structure of the `OpenIPSL`

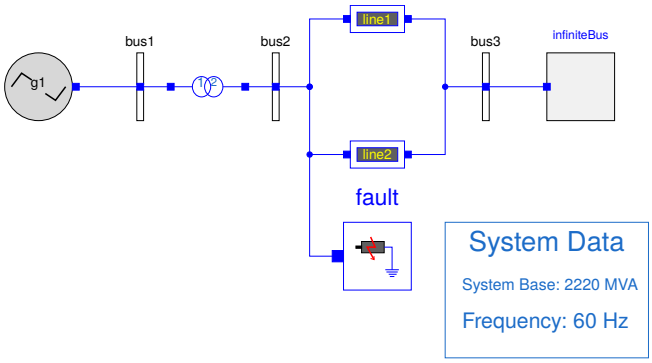

Figure 6. Kundur, example 13.2 with `OpenIPSL`

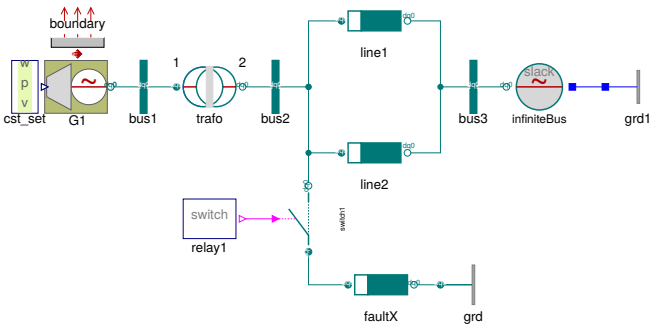

Figure 7. Kundur, example 13.2 with `PowerSystems` library

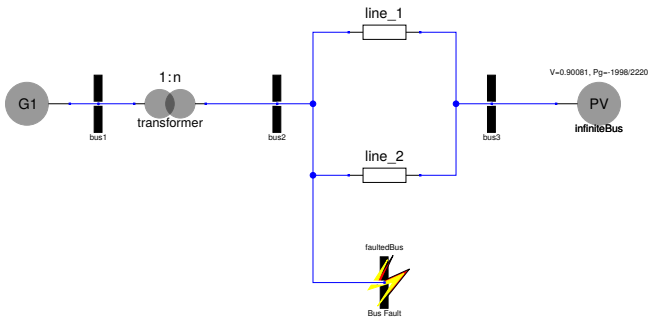

Figure 8. Kundur, example 13.2 with `ObjectStab` library

3 Usability

3.1 User-friendliness

One thing that becomes very apparent when working with the different power system libraries is the different focus on usability vs. compliance with other tools. Both the `SPOT` and `PowerSystems` library on the one end of the scale providing models of lines and machines with standard parameter sets as seen in real live. For the `PowerSystems` library even providing ready to use parameter for different standard line types. One can enter the length in *Km* and the material and will get the reactance and resistance values accordingly.

The three other candidates (`ObjectStab`, `iPSL`, `OpenIPSL`) are on the other end of the scale. Here needs to provides parameters like reactance and resistance directly. Also one needs to take care of possible per-unit conversion due to different system or machine base oneself.

Both methods have their benefits and drawbacks.

E.g., implementing a text book example with given per-unit values would be easier more straight forward in ObjectStab, iPSL, OpenIPSL. The same goes for models taken from other power system simulation tools like PSS/E, where the parameter set is normally given in per-unit values and not physical dimensions.

Implementing a practical problem on the other hand would be easier using the PowerSystems library (and SPOT for that matter).

3.2 Use of Modelica features

When looking at the level usage of Modelica features then basically the SPOT and PowerSystems library are the winner. Here the full power of switched models (e.g., phase-system) and inheritance and library structure are fully put into use.

The ObjectStab library is simply too old to make use of features that have been implemented in later versions of the Modelica language. iPSL and OpenIPSL had the challenge to stay as close to the original reference models, which often meant system-flow oriented modelling and not making use of the power of equations.

3.3 Documentation

Unfortunately only the PowerSystems and SPOT library contain extensive and usable documentation of the models and a user's guide as part of the library.

The other library basically point either to the original presentation paper (ObjectStab) or to the reference tools where the models were taken from.

3.4 Robustness

The SPOT and PowerSystems library were developed as generic multi-purpose libraries (the latter less so). Unfortunately this leads to a certain amount of complexity which in turn again leads to a series of parameters that can/must be set. Not knowing the correct fitting parameters can easily lead to a non-solvable equation system and can be frustrating to the user.

The ObjectStab, iPSL and OpenIPSLhave a restricted set of parameters available (when compared with the other two) and therefor tend to lead to more robust simulation models with less numerical issues.

All of the libraries on the other hand depend on initial conditions that are in the proximity of a stable power system. Basically that meant one needs steady-state power flow calculations before hand to parameterise the systems. One can get lucky to be able to get the model running and converging towards a stable power flow but often this is not the case.

4 Validations

The PowerSystems and SPOT library do not name specific validation cased. The only reference for them are basically some real-life applications that were published.

The ObjectStab library notes that the component library has been validated using comparative simulations with EUROSTAG.

The iPSL and OpenIPSL basically only contain validated modes using software-to-software comparison. This was the original reason to build the iPSL in the first place. They contain models validated against CGMES[2], Simulink, EUROSTAG (iPSL only), PSAT and PSS/E.

5 Other options

Despite the open-source libraries presented here so far, the Modelica Standard Library itself contains the quasi-static library package "Modelica.Electrical.Quasitationary" that could potentially also be used for power systems modelling.

Another library that needs to be mentioned in this context is another successor of the SPOT library, the "ElectricPower library" [16]. It is a commercial product of Modelon and therefor not part of the comparison.

6 Outlook

When looking at the future of the presented library options, from my perspective both, the iPSL and the OpenIPSL, will continue to be actively developed. It will be interesting to see to what degree the iPSL's dependency of the iTesla Power Tool will restrict the "clean up" of the structure. This is something that the OpenIPSL might be able to use to its advantage of a more flexible development future.

As for the PowerSystems library, it is actively developed, has better documentation and is more user-friendly. Something that might push its application further especially for new-comers.

The ObjectStab library is most likely deprecated by iPSL and OpenIPSL alike.

The SPOT library has already two active projects carrying on. In a reduced form as PowerSystems library and in the commercial ElectricPower library,

References

[1] Bernhard Bachmann and Hansjürg Wiesmann. "Advanced Modeling of Electromagnetic Transients in Power Systems". In: *Modelica Workshop 2000 Proceedings*. Oct. 23, 2000, pp. 93–97.

[2] Open Source Modelica Consortium. *OpenModelica*. 2017. URL: https://openmodelica.org/.

[3] DIgSILENT. *PowerFactory*. URL: http://www.digsilent.de/index.php/products-powerfactory.html.

[4] Tractebel Engineering. *EUROSTAG*. URL: http://www.eurostag.be/en/products/eurostag/the-reference-power-system-dynamic-simulation/.

[5] Mozilla Foundation. *Mozilla Public License Version 2.0*. Jan. 3, 2012. URL: https://www.mozilla.org/en-US/MPL/2.0/.

[2] Common Grid Model Exchange Standard

[6] Rüdiger Franke and Hans Jürg Wiesmann. "Flexible modeling of electrical power systems–the Modelica PowerSystems library". In: *Proceedings of the 10 th International Modelica Conference; March 10-12; 2014; Lund; Sweden*. 096. Linköping University Electronic Press. Linkoping University Electronic Press, Mar. 11, 2014, pp. 515–522. DOI: 10.3384/ecp14096515.

[7] *iTesla - Innovative Tools for Electrical System Security within Large Areas*. URL: http://www.itesla-project.eu/.

[8] ITesla project. *iPSL on GitHub*. 2017. URL: https://github.com/itesla/ipsl.

[9] Daniel Karlsson and David J Hill. "Modelling and identification of nonlinear dynamic loads in power systems". In: *IEEE Transactions on Power Systems* 9.1 (1994), pp. 157–166.

[10] Prabha Kundur and Neal J. Balu. *Power system stability and control*. The EPRI power system engineering series. New York, NY: McGraw-Hill, 1994. 1176 pp. ISBN: 978-0-07-035958-1.

[11] Mats Larsson. *ObjectStab Library – A Modelica Library for Power System Stability Studies*. 2000.

[12] Mats Larsson. *ObjectStab on GitHub*. Sept. 25, 2002. URL: https://github.com/modelica-3rdparty/ObjectStab.

[13] Jan Machowski, Janusz Bialek, and Dr Jim Bumby. *Power System Dynamics and Stability*. Wiley, 1997. ISBN: 978-0-471-97174-0.

[14] Manitoba Hydro International Ltd. *PSCAD™*. URL: https://hvdc.ca.

[15] Federico Milano. *PSAT*. URL: http://faraday1.ucd.ie/psat.html.

[16] Modelon AB. *Electric Power Library*. Modelon. URL: http://www.modelon.com/products/modelica-libraries/electric-power-library/ (visited on 05/28/2016).

[17] POWERSYS. *EMTP-RV*. URL: http://www.emtp-software.com.

[18] iTesla PST group. *iTesla Power System Tools*. 2017. URL: http://www.itesla-pst.org/.

[19] P.W. Sauer and M.A. Pai. "A comparison of discrete vs. continuous dynamic models of tap-changing-under-load transformers". In: *Proceedings of NSF/ECC Workshop on Bulk power System Voltage Phenomena - III : Voltage Stability, Security and Control*. Davos, Switzerland, 1994.

[20] Siemens. *PSS/E*. URL: http://siemens.com/power-technologies/software.

[21] SmarTS Lab. *OpenIPSL on GitHub*. 2017. URL: https://github.com/SmarTS-Lab/OpenIPSL.

[22] The Modelica Association. *The Modelica License 1.1*. June 30, 2000. URL: https://modelica.org/licenses/ModelicaLicense1.1.

[23] The Modelica Association. *The Modelica License 2*. Nov. 19, 2008. URL: https://modelica.org/licenses/ModelicaLicense2.

[24] L. Vanfretti et al. "iTesla Power Systems Library (iPSL): A Modelica library for phasor time-domain simulations". In: *SoftwareX* 5 (2016), pp. 84–88. DOI: 10.1016/j.softx.2016.05.001.

[25] Hans Jürg Wiesmann. *SPOT library on GitHub*. Sept. 11, 2007. URL: https://github.com/modelica-3rdparty/Spot.

[26] Hans Jürg Wiesmann and Rüdiger Franke. *Power Systems library on GitHub*. Jan. 18, 2017. URL: https://github.com/modelica/powersystems.