# OpenIPSL Documentation

*Release 1.0.0*

Maxime

Jan 12, 2018

# Table of Contents

Welcome to **OpenIPSL** - The Open-Instance Power System Library.

This documentation is the main source of information for **users** and **developers** working with (or contributing to) the **OpenIPSL** project.

# CHAPTER 1

# OpenIPSL in short

The OpenIPSL or Open-Instance Power System Library is a Modelica library, fork of of the iTesla Power System Library developed and maintained by the SmarTS Lab research group, collaborators and friends (contributions are welcome!).

The library contains a set of power system component models and test power system networks adopting the "phasor" modeling approach. Time domain simulations can be carried out using a Modelica-compliant tool, which may also allow to do other computations on the model, for example, linearization for eigen-analysis and other purposes.

All models in a Modelica library require *initial guess* values that should come from a solution of the steady state of the overall model. From these values, a Modelica tool solves the initialization problem for all algebraic and differential - state variables. There are many ways to provide *initial guess* values, but following the standard "workflow" in power system analysis practice, all models in OpenIPSL are programmed in such way that by introducing a power flow solution (from another tool), the *initial guess* is computed as a parameter within each model and are provided into the initial equations that are used to solve the overall initialization problem. See this paper for a more detailed explanation.

Almost all of our models have been developed to provide the same response than a "reference" power system simulation tool, e.g. PSAT and PSS/E. You can use these tools to create a power flow solution for your network. If you do not have access to these tools or do not want to use them, there are several power flow solvers available on Github. Future work in the OpenIPSL effort will include to generate Modelica "records" from open source power flow solvers such as GridCal or PyPSA.

## 1.1 Questions or need help?

Check out our *Talk to Us* page for different ways of getting in touch.

### 1.1.1 Publications

The OpenIPSL library emerged as a fork of iPSL, which was been developed as part of a research project involving several partner institutions. The list of publications written about OpenIPSL library and its models can be found below.

**Note:** If you use OpenIPSL in your work or research, we kindly ask you to cite (preferably), our SoftwareX paper.

### SoftwareX paper:

- L. Vanfretti, T. Rabuzin, M. Baudette, and M. Murad, iTesla Power Systems Library (iPSL): A Modelica library for phasor time-domain simulations, SoftwareX, Available online 18 May 2016, ISSN 2352-7110, DOI: 10.1016/j.softx.2016.05.001.

Otherwise, you cite one of the following papers, according to a specific use of OpenIPSL:

- M. Zhang, M. Baudette, J. Lavenius, S. Løvlund, and L. Vanfretti, "Modelica Implementation and Software-to-Software Validation of Power System Component Models Commonly used by Nordic TSOs for Dynamic Simulations," In Proceedings of the 56th Conference on Simulation and Modelling (SIMS 56), 7-9 October 2015, Linköking, Sweden. Link.
- M. Sabate, G. Leon, M. Halat, J.B. Heyberger, F.J. Gomez and L. Vanfretti, "Aspects of Power System Modeling, Initialization and Simulation using the Modelica Language," IEEE PowerTech 2015. Link
- M.A. Adib Murad, F.J. Gomez, and L. Vanfretti, "Equation-Based Modeling of FACTS using Modelica," IEEE PowerTech 2015. Link.
- M.A. Adib Murad, F.J. Gomez, and L. Vanfretti, "Equation-Based Modeling and Simulation of Three-Winding, and Regulating Transformers using Modelica," IEEE PowerTech 2015. Link.
- L. Vanfretti, T. Bogodorova, and M. Baudette, "Power System Model Identification Exploiting the Modelica Language and FMI Technologies," 2014 IEEE International Conference on Intelligent Energy and Power Systems, June 2-6, 2014, Kyiv, Ukraine. Link.
- L. Vanfretti, T. Bogodorova, and M. Baudette, "A Modelica Power System Component Library for Model Validation and Parameter Identification," 10th International Modelica Conference 2014, Lund, Sweden, Mar. 10 – 12, 2014. Link.
- T. Bogodorova, M. Sabate, G. León, L. Vanfretti, M. Halat, J.B. Heyberger, and P. Panciatici, "A Modelica Power System Library for Phasor Time-Domain Simulation," IEEE ISGT Europe 2013, Oct. 6 – 9, 2013, Lyngby, Denmark. Link.
- L. Vanfretti, W. Li, T. Bogodorova, and P.Panciatici, "Unambiguous Power System Modeling and Simulation using Modelica Tools", IEEE PES General Meeting 2013. Link.

## 1.1.2 User Guide

### Get Started with OpenIPSL

The OpenIPSL was developed to be a familiar alternative to traditional power system analysis tools. There are, however, some particularities to the nature of the library, presented in this Section.

### Package Content

The OpenIPSL has been developed as a Modelica library for power system simulations. The repository contains the following:

- **The OpenIPSL package:** A Modelica library for power system component models.

- **The Application Examples:** A set of power system network models implemented in separate Modelica packages.

**Note:** The repository does **not** provide any external software packages. The simulation of the models requires separate Modelica-compliant software packages.

### Power Flow Solvers / Software

There is no power flow solver associated to the library. When building a use case, initialization of all variables must be performed with a power flow software. The users are free to choose their power flow software of choice.

Please note that during the development of the library, the power flow software used corresponds to that of the reference model, for example, for models coming from PSAT, we created the same power system model in PSAT and in Modelica using our library. The power flow solution from PSAT was then reported into the corresponding "power flow solution parameters" corresponding to each model.

Within the iTesla Toolbox (i.e. the whole platform), there are different power flow tools implemented and also an "Automatic Translator" that takes a file with the parameters of a model (as defined by a domain specific tool) and populates the power flow parameters of the models. Therefore, there was no need to develop a power flow solver.

### Running Time-Domain Simulations

There is no Modelica-compliant tool provided to run the time domain simulations. Users of the OpenIPSL are free to choose their tool of choice.

Please note that the development of OpenIPSL has been carried out using OpenModelica and Dymola. The library is, also, systematically checked with OpenModelica in the repository (see *Continuous Integration*). It might be possible to face issues when using other Modelica tools that we have not fully tested.

## 1.1.3 Community

### Contributing

We welcome all sorts of contribution to the project!

### Get Started

However you decide to contribute to our project, please follow the following instructions:

- Create a GitHub Account.
- Browse our Issues base.
- Fork our repository!
- Create a pull request to our repository to submit your changes.

**Note:** Your contribution can only be considered when you have signed our Contributor License Agreement.

### Improve our Documentation

The documentation of the OpenIPSL is hosted on Read the Docs and is developed using the **reStructuredText** markup language.

The sources for compiling the documentations are placed in the `./docs/` folder of the repository.

The main goal of the documentation is to provide information on the development conventions adopted for this project, and provide a guide to the contributors of the project.

#### Useful Links

Here is a collection of resources to learn about the reStructuredText markup language:

- OpenIPSL Documentation
- Get Started with Read the Docs
- Some infos about reStructuredText
- Some more infos about reStructuredText

### Improve the Library

The library has been in development for a couple years now, and comprises many models for power system simulation. We are, however, always interested in adding new models to the library, or improve the existing models.

Please always document your contribution in an issue in our **'Issue'_** Tracker, feel free to create a new issue if none of the existing one fit your contribution. This will allow the team managing the repository to review the suggested changes, and discuss the motivation behind them.

**Note:** Structural changes to the library should be discussed with the team prior to any submission

When submitting new models to the library, please also include the following in the associated issue:

- A short report describing your component model, as well as the validation test.
- The source files that you used to validate your model against a reference tool.
- If the model is entirely new, and there is NO reference model to check against, some measurements could be useful…
- The data files to plot the comparison figures and compute the quantitative validation metrics.

**Note:** We prepared a `submission template` to help you prepare your submission.

#### Useful Links

- Issues Tracker
- MapleSoft Modelica Syntax Checker

### License information

The OpenIPSL is released under the same license as the original iPSL project.

### OpenIPSL

All changes applied in this fork have the following copyright statement:

### iPSL:

The library original's copyright statement:

### Our users

Coming Soon.

### Testimonials

Coming Soon.

The team behind the OpenIPSL is proud to share this project as an open source project. We hope to create a strong community around this library for Power System modeling.

Let's build the future of Power System modeling. To get involved, share your ideas, and collaborate with us, head to our *Contributing Home*.

### Project Contributors

### The iPSL

The original library (iTesla Power System Library, iPSL) was developed by the consortium behind the iTesla project, with the following partners:

- RTE: http://www.rte-france.com/
- AIA: http://www.aia.es/en/energy/
- SmarTS Lab, KTH: https://www.kth.se/en
- DTU: http://www.dtu.dk/english

Since the end of the iTesla project, the project has been forked into the OpenIPSL project by the SmarTS Lab research group. Our goal is to foster a larger community and include external collaborators to the development of the library.

## The OpenIPSL

### SmarTS Lab's people

SmarTS Lab is a research group at KTH Royal Institute of Technology, Stockholm, Sweden. You can learn more about our research on our departments webpage or on Luigi's website. We were participating in the iTesla project and several people have contributed to this Modelica library. Here is a complete list of the contributor (alphabetical order):

- Achour Amazouz
- Maxime Baudette
- Tetiana Bogodorova
- Francisco Gomez
- Giuseppe Laera
- Jan Lavenius
- Md. Ahsan Adib Murad
- Joan R. Mussons
- Le Qi
- Tin Rabuzin
- Luigi Vanfretti
- Mengjia Zhang

### Other Collaborators

Thanks to the following people who have contributed patches, new code or given us helpful suggestions:

- Dietmar Winkler (HSN, Norway, @dietmarw)

### Talk to Us

The team of collaborators involved in the development of the OpenIPSL can be reached at openipsl@googlegroups.com.

For more information about contributing to this repository, please head to *Contributing*.

## 1.1.4 Technical Documentation

### Continuous Integration

The OpenIPSL repository is connected to Travis CI that provides **Continuous Integration** services to the repository.

In the current implementation, the CI server performs:

- Modelica syntax checking for all classes in OpenIPSL
- Modelica syntax checking for all *Application Examples*

An upcoming extension of the CI implementation will also include regression testing to ensure the validation status of all models.

### Development Guidelines

The team behind OpenIPSL strives to achieve a harmonized development of the library to facilitate its usability and integration into other services.

### Naming Convention

Here are some naming conventions that we try to obey:

- Names of the parameters and variables should correspond to those found in documentation.
- Subscript should be denoted in Modelica with underscore, e.g. T_r or K_1 **Note!** Some names have first upper case letters and others lower case.
- Deviation signals such as delta omega, should be written as domega in Modelica.
- Parameters which are not to be entered by user and variables not to be seen by user should be **protected**
- Units should be written in parentheses in the description of the parameter or variable (see e.g. of the power flow)

### Initialization Routines

The models in the library are initialized from a power flow solution. All model for power system component should take in, as parameters, the power flow solution for the bus it is attached to.

The full initialization of the components' internal variables and states is achieved by a set of internal `initial equations` that are to be derived by the developer of the model.

In a effort to harmonized the presentation of the power flow parameters, a model should extend `pfComponent` that will provide all the necessary parameters. The `pfComponent` also provide the support for the common `sysBase` component that provides a single instance of the system's common parameters.

### Connectors

...

## 1.2 Indices and tables

- genindex
- modindex
- search