Available online at www.sciencedirect.com
ScienceDirect
SoftwareX 5 (2016) 84–88
SoftwareX
www.elsevier.com/locate/softx

iTesla Power Systems Library (iPSL): A Modelica library for phasor time-domain simulations

L. Vanfretti[a,b], T. Rabuzin[a], M. Baudette[a,*], M. Murad[a]

[a] SmarTS Lab., KTH Royal Institute of Technology, Stockholm, Sweden
[b] Research and Development Division, Statnett SF, Oslo, Norway

Received 10 February 2016; received in revised form 28 April 2016; accepted 3 May 2016

Abstract

The iTesla Power Systems Library (iPSL) is a Modelica package providing a set of power system components for phasor time-domain modeling and simulation. The Modelica language provides a systematic approach to develop models using a formal mathematical description, that uniquely specifies the physical behavior of a component or the entire system. Furthermore, the standardized specification of the Modelica language (Modelica Association [1]) enables unambiguous model exchange by allowing any Modelica-compliant tool to utilize the models for simulation and their analyses without the need of a specific model transformation tool. As the Modelica language is being developed with open specifications, any tool that implements these requirements can be utilized. This gives users the freedom of choosing an Integrated Development Environment (IDE) of their choice. Furthermore, any integration solver can be implemented within a Modelica tool to simulate Modelica models. Additionally, Modelica is an object-oriented language, enabling code factorization and model re-use to improve the readability of a library by structuring it with object-oriented hierarchy. The developed library is released under an open source license to enable a wider distribution and let the user customize it to their specific needs. This paper describes the iPSL and provides illustrative application examples.

Keywords: Modelica; Power system; Simulation

Code metadata

Current code version	v1.0.0
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-16-00027
Legal Code License	MPL 2.0
Code versioning system used	git
Software code languages, tools, and services used	Modelica
Compilation requirements, operating environments & dependencies	Modelica Standard Library (MSL), Modelica IDE and corresponding C compiler
If available Link to developer documentation/manual	https://github.com/ElsevierSoftwareX/SOFTX-D-16-00027
Support email for questions	info@itesla-ipsl.org

1. Motivation and significance

Since the 1970s [2], a wide range of power system models have been developed for various simulation and analysis purposes within the electric power systems domain. These models are generally tightly integrated to the software platforms for which they are developed for. Therefore, the use of such models results in ambiguous model sharing [3], e.g. in the form of incomplete block diagram exchange. Moreover, only few simulation tools facilitate users to access and/or alter the built-in models, impacting greatly the tool's flexibility and transparency.

The European Commission (EC) adopted the Third Energy Package [4] of legislative proposals in September 2007 to enable a competitive and integrated energy market. The Euro-

---

* Corresponding author.
E-mail address: baudette@kth.se (M. Baudette).

http://dx.doi.org/10.1016/j.softx.2016.05.001

pean Network of Transmission System Operators for Electricity (ENTSO-E) was established as a result of the European regulation [5] to bring a stronger coordination among the European Transmission System Operators (TSOs.) The motive was to use a common transmission model that efficiently deals with "interdependent physical loop-flows and having regard to discrepancies between physical and commercial flows". This common model should be used by ENTSO-E to support "common network operation tools and to ensure coordination of network operation in normal and emergency conditions".

The Common Information Model (CIM) and its evolution, Common Grid Model Exchange Standard (CGMES) is the most prominent data model used for power systems information exchange between TSOs [6]. According to European regulation, in the future, the CIM and CGMES will require to be extended to provide static and dynamic model information. As of today, dynamic information exchange is considered in the CIM through the IEC 61970-302 and IEC 61970-457 standards. However, this approach has limitations with respect to dynamic modeling cross-platform consistency. Also, the mathematical representation is not explicitly shared, but replaced by a pictorial block diagram representation with parameter definition.

Following the ENTSO-E R&D Road Map, the goal of the ***iTesla*** project [7] was to develop a common toolbox supporting the future operation of the pan-European power grid. The iTesla Power Systems Library (iPSL), as one of the deliverables of the project, seeks to tackle the aforementioned issues by adopting an open modeling language, i.e. Modelica.

To the knowledge of the authors, there have been two previous efforts in the use of Modelica for power system modeling and simulation: the ObjectStab [8] library, and the SPOT [9] library that evolved to both the PowerSystems [10] library and the Electric Power Library by Modelon AB [11]. While the authors could have extended these libraries, it is worth noticing that the libraries publicly available as open source software have not been actively developed for the past 5 years. There are, also, social aspects of resistance to change [12] that motivated the authors to develop a new library. To ease the transition to a Modelica library, it was decided to adopt well-established practices in the power system domain. Therefore, the models integrate initialization routines from a power flow solution to compute a guess of the variables' starting value. The power flow solution is obtained from external tools or from the information contained in the Steady-State Hypothesis (SSH) and the State Variable (SV) CIM profiles. This enables to inherently integrate the library within the current industrial approach (tooling and workflow). In addition, the models developed are validated against trusted domain-specific power system tools where simulation results of the iPSL implementation are appraised against a domain-specific tool [13,14]. This has been done in order to overcome the resiliency of the power system domain to embrace new practices and for users to gain confidence in the library.

2. Software description

The iPSL is a Modelica package built from a collection of power system component models. These components are of different types, to model a power system network, comprising power generation, transmission, and power consumption. In addition to models of actual physical equipment, the library includes several auxiliary models for equipment such as control and stabilization systems. As a whole, the iPSL fully enables modeling of networks containing complex equipment models typically found in power systems. Currently, the simulation performance of Modelica tools is not comparable with the tools used in the power system domain for large-scale models such as regional or national grids. However, it has been shown that the further work in the back-end of Modelica tools can lead to substantial improvements in performance [15].

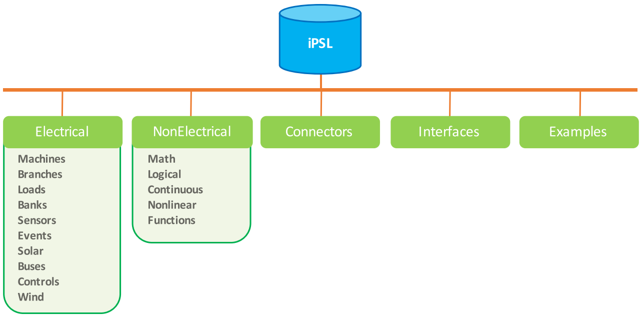


Fig. 1. Structure of the iPSL.

2.1. Software architecture

The library is organized in different sub-packages providing an easy understanding structure for the user. It is divided into five main packages:

- The *Examples* package contains a number of different sample power systems showing capabilities and use of other models in the library.
- The *Electrical* package contains the components of a power network such as electrical machines, transmission lines, loads, excitation systems, turbine governors, etc.
- The *NonElectrical* package is comprised by functions, blocks, and specialized models used to build the aforementioned power system component models. They are transfer functions, logical operators, etc., that perform specific operations which are not available in the MSL.
- The *Connectors* package consists of specifically developed Modelica connectors to harmonize the models in the library. For example, *PwPin* is the connector for voltage and current quantities in phasor representation.
- The *Interfaces* package contains models used for data conversion (to exchange data between the library and other Modelica libraries).

An overview of the top-level structure is shown in Fig. 1.

2.2. Software functionalities

The iPSL is one of the components of the iTesla platform dedicated for dynamic simulations of small, medium and large power systems as shown in Section 3. It was developed in Modelica to exploit several of the advantages offered by this flexible modeling language. In particular, the acausal modeling [16]

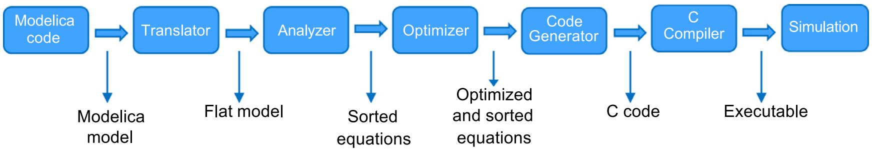


Fig. 2. Translation and compilation of Modelica models [16].

permits the use of equality sign as in the formal mathematical notation and not as in the assignment statements of many other programming languages. Thus, the dynamic behavior of the models in the library is described with exact differential equations from theory.

Modelica also enables the use of object-oriented paradigms, allowing class inheritance to factorize the code of the models. Also, classes are instantiated to build more complex models, such as power system networks, thus facilitating the management of parameters sets.

Finally, Modelica is a standardized modeling language supported by several Modelica compliant IDEs offering a broad range of alternatives to the user. These IDEs provide the development environment and the simulation facilities. Thus, there are several steps that these IDEs perform in order to simulate a model, in particular the model is compiled into C-code and binded to the selected solver. In theory, considering Fig. 2, IDEs should not alter the model equations substantially in any of the listed steps. Thus, the repeatability of the simulations and the separation of the iPSL models from the IDEs and the solvers are ensured. In contrast to traditional domain-specific tools, where second order (trapezoidal) integration method is used, the choice of the integration method depends on the IDEs and is not locked to the model.

3. Illustrative examples

The goal of such a library is to perform dynamical studies of custom-built and more complex power system networks. In this Section, the Nordic 44-Bus test system is used to demonstrate the feasibility of building medium size networks. This system is an equivalent representation of the Nordic grid and is originally implemented in PSS/E [17]. It consists of 44 buses, 61 generators with various control systems (exciter, turbine governor, stabilizer), 67 lines and 43 loads, all of which have been implemented in iPSL. Each of the generation unit model encapsulates the actual generator model and its associated control loops (see Fig. 3).

The implementation of Nordic 44-Bus test system in Modelica using iPSL components is shown in Fig. 4.

The validation of this test network is performed by applying various small and large disturbances in both Modelica and PSS/E, with the following simulation scenarios: (a) a line opening between Bus 5103 to Bus 5304 and (b) a bus fault at Bus 3100 at time 2 s for a duration of 200 ms with negligible fault impedance. The simulation results from both software packages are compared in Fig. 5.

In addition, simulation performance for both disturbances is evaluated and compared in the Dymola Modelica IDE and PSS/E. The Runge Kutta Order 2 solver with a fixed time step of 0.01 s and a tolerance of 0.0001 is used in both tools. The integration times for both cases are shown in Table 1.

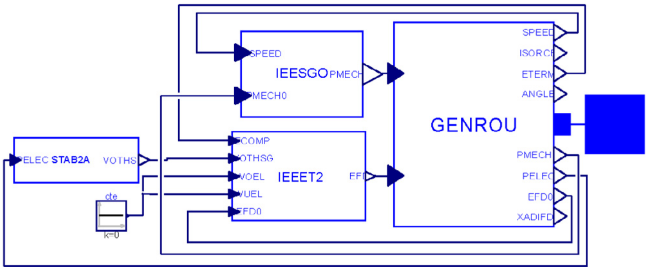


Fig. 3. Generator model including excitation, governor and stabilizer (e.g. modeling a power plant with similar units and controls, but with their own respective parameters).

Table 1
Integration times for the disturbances simulated with Nordic44 model.

Fault	PSS/E	Dymola
Line fault	4.934	468 s
Bus fault	4.027	372 s

The nonconformity in simulation performance arises due to the difference in the development and implementation philosophies of the two tools. Power system domain tools, such as PSS/E, use custom solver implementations and heuristic methods to optimize power system simulation performance. On the other hand, Modelica tools are different, they are general purpose tools and can be used for multi-domain simulations. Thus, additional work has to be performed on the back-end of Modelica tools to enable them to have comparable performance with respect to domain-specific tools to simulate large-scale power systems [15].

4. Impact

iPSL is a library for building power system networks and performing dynamic time-domain simulations, like many other domain-specific tools. Traditional studies can be performed: synchronous machine dynamic analysis, electro-mechanical studies, control design, etc. However, none of the domain-specific tools (to the knowledge of the authors) offers the flexibility of iPSL. This flexibility is brought mainly by the use of Modelica.

The unambiguous exchange of dynamic models and their parameters is becoming of utmost importance for the coordination between TSOs. The extensibility offered by some Modelica environment is used to demonstrate how the iPSL can be coupled to existing standards for the information exchange (CIM and CGMES), to provide details on models dynamics [18].

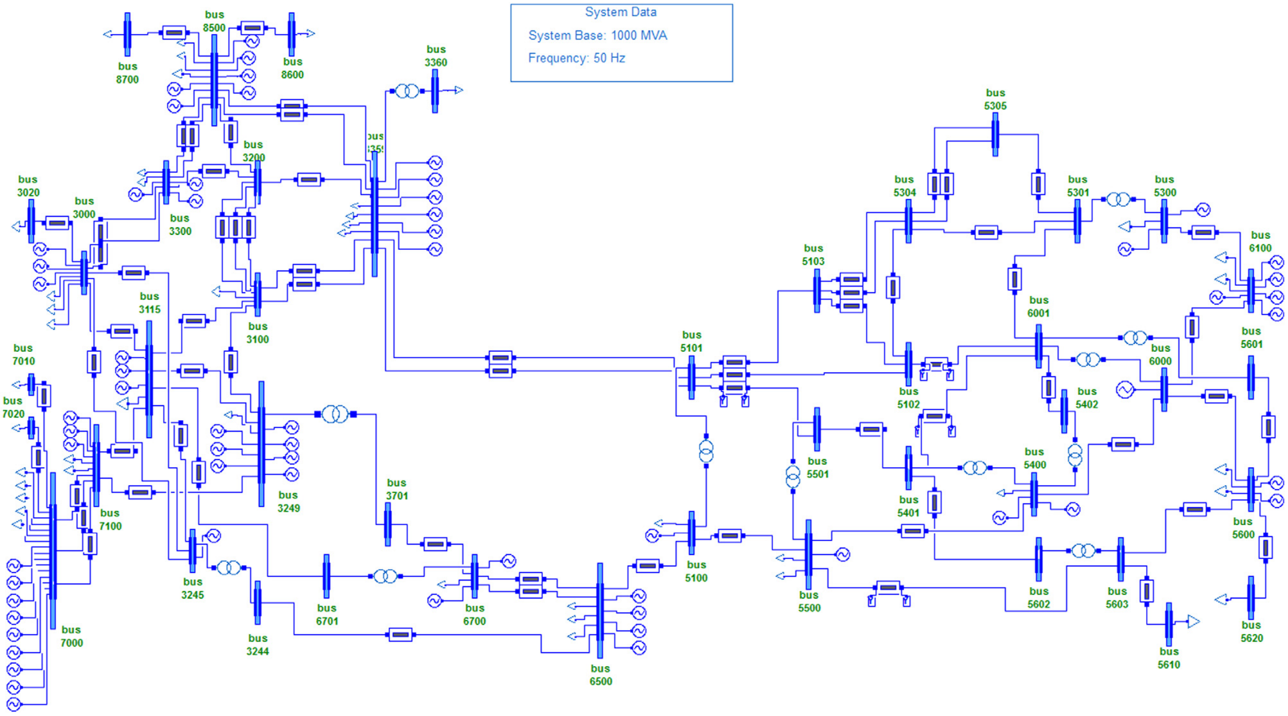
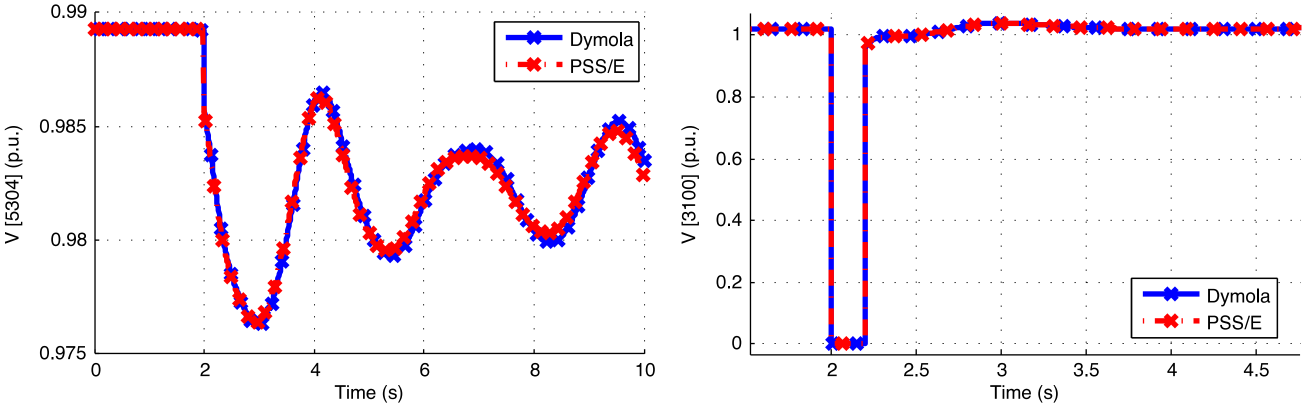


Fig. 4. Nordic 44-Bus test system in Modelica (screenshot from the Dymola IDE).



(a) Line.

(b) Line.

Fig. 5. Dynamic response of Nordic 44-Bus system in Modelica IDE and PSS/E.

The open-source nature of iPSL advocates more demonstration and implementation projects taking advantage of the open, flexible, and extensible Modelica language. In addition, the models' description is fully accessible, thus, providing full awareness of the models' limitations to system analyst. This, in turn, will allow the users to extend the library to suit their specific needs and contribute to the expansion of the iPSL.

The nature of a standardized language also facilitates repeatability in different IDEs. This means that TSOs of different countries are able to get equivalent simulation results regardless of which simulation tool they use, under the assumptions that they use the same integration solver implementation and parameters.

Finally, Modelica is tightly related to the Functional Mock-up Interface (FMI) [19], a standard for model exchange and co-simulation. It allows the export of Modelica models, such as the ones in iPSL, to non-Modelica tools and vice-versa [16]. Thus, advantages of other non-Modelica tools can be exploited using the FMI standard. For example the iPSL is used with Matlab to perform model calibration, validation and parameter estimation. Such possibility is of great importance in the field of power system simulation and modeling [20].

5. Conclusions

With the increase in complexity of the cyber–physical networks of power systems, the need for dynamic model exchange between the TSOs is evolving. Nowadays, TSOs are in need for a language which provides the common understanding of the behavior of various devices and control systems.

This paper provides an overview of the iPSL developed as a part of the EU funded FP7 ***iTesla*** project. The library is developed to advance established practices in the modeling and simulation of power systems. With iPSL, an attempt was made to address issues related to the use of traditional tools and to contribute towards enabling dynamic model exchange. Thus, it was decided that the software should be released using an OSS license so that the further development of the library can be made regardless of the initial developers.

The library was actively developed and updated until the end of the ***iTesla*** project (April 2016). Further development of the software by the first author's research team will be dependent on research funding. Therefore, to allow the software to grow, the authors would like to encourage users to contribute their own models or in any other means towards the further development of library.

Acknowledgments

The financial support from the EU funded FP7 ***iTesla*** project and the contribution from the following parties are gratefully acknowledged:

- RTE — http://www.rte-france.com
- AIA — http://www.aia.es/en/energy
- DTU — http://www.dtu.dk/english.

The iPSL project has been forked in the OpenIPSL project for its maintenance by the SmarTS Lab research team which is also available in Github at https://github.com/SmarTS-Lab/OpenIPSL.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.softx.2016.05.001.

References

[1] Modelica Association, Modelica - A Unified Object-Oriented Language for Systems Modeling - Language Specification. www.modelica.org.

[2] Dandeno PL, Hauth RL, Schulz RP. Effects of synchronous machine modeling in large scale system studies. IEEE Trans Power Appar Syst 1973;92(2):574–82. http://dx.doi.org/10.1109/TPAS.1973.293760.

[3] Vanfretti L, Li W, Bogodorova T, Panciatici P. Unambiguous power system dynamic modeling and simulation using modelica tools. In: IEEE PES Gen. Meet. IEEE; 2013. p. 1–5. http://dx.doi.org/10.1109/PESMG.2013.6672476.

[4] European Comission - Market Legislation. URL https://ec.europa.eu/energy/en/topics/markets-and-consumers/market-legislation.

[5] Regulation (EC) No 714/2009 of The European Parliament and of the Council of 13 July 2009. URL http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:211:0015:0035:EN:PDF.

[6] Uslar M, Specht M, Rohjans S, Trefke J, Gonzalez JM. The common information model CIM IEC 61970, 61968 and 1078 62325. Heidelberg: Springer; 2012. http://dx.doi.org/10.1007/978-3-642-25215-0.

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

[8] Navarro I, Larsson M, Olsson G. Object-oriented modeling and simulation of power systems using modelica. In: IEEE power engineering society winter meeting. conference proceedings (Cat. No.00CH37077). Institute of Electrical & Electronics Engineers (IEEE); 2000. http://dx.doi.org/10.1109/pesw.2000.850173. URL https://github.com/modelica-3rdparty/ObjectStab.

[9] Wiesmann H. Spot. URL https://www.modelica.org/libraries/spot.

[10] Modelica Association, Powersystems library. URL https://github.com/modelica/PowerSystems.

[11] Modelon AB. Electric Power Library. URL http://www.modelon.com/products/modelica-libraries/electric-power-library/.

[12] Ford JD, Ford LW, D'Amelio A. Resistance to change: The rest of the story. Acad. Manage. Rev. 2008;33(2):362–77. http://dx.doi.org/10.5465/amr.2008.31193235.

[13] Vanfretti L, Bogodorova T, Baudette MM. A modelica power system component library for model validation and parameter identification. In: Proc. 10th int. model. conf., KTH, electric power systems, 10th international modelica conference, Lund, Sweden. 2014. p. 1195–203. http://dx.doi.org/10.3384/ecp140961195.

[14] Zhang M, Baudette M, Lavenius J, Løvlund S, Vanfretti L. Modelica implementation and software-to-software validation of power system component models commonly used by nordic TSOs for dynamic simulations. In: 56th conf. simul. model. (SIMS 56). Linkoping University Electronic Press; 2015. p. 105–12. http://dx.doi.org/10.3384/ecp15119105.

[15] Casella F. Simulation of large-scale models in modelica: State of the art and future perspectives. In: Proceedings of the 11th international modelica conference, Versailles, France, September 21–23, 2015. Linkoping University Electronic Press; 2015. http://dx.doi.org/10.3384/ecp15118459.

[16] Fritzson P. Introduction to modeling and simulation of technical and physical systems with modelica. Hoboken, NJ, USA: John Wiley & Sons, Inc; 2011. http://dx.doi.org/10.1002/9781118094259.

[17] PSS/E: Power Transmission System Planning Software. URL http://www.pti-us.com/.

[18] Gomez FJ, Vanfretti L, Olsen SH. Binding CIM and modelica for consistent power system dynamic model exchange and simulation. In: 2015 IEEE power energy soc. gen. meet. IEEE; 2015. p. 1–5. http://dx.doi.org/10.1109/PESGM.2015.7286434.

[19] Functional Mockup Interface. URL https://fmi-standard.org/.

[20] Vanfretti L, Bogodorova T, Baudette M. Power system model identification exploiting the modelica language and FMI technologies. In: 2014 IEEE int. conf. intell. energy power syst. IEEE; 2014. p. 127–32. http://dx.doi.org/10.1109/ieps.2014.6874164.