

A Tool to ease Modelica-based Dynamic Power System Simulations

Raul Viruez¹ Silvia Machado¹ Luis María Zamarreño¹ Gladys León¹ François Beaudé²
Sébastien Petitrenaud² Jean-Baptiste Heyberger²

¹Aplicaciones en Informática Avanzada S.L., Sant Cugat del Vallès, Spain.

{viruezz, machados, zamarrenolm, leonge}@aia.es

²Réseau de Transport d'Électricité, Paris, France.

{francois.beaude, sebastien.petitrenaud, jean-baptiste.heyberger}@rte-france.com

Abstract

Developments made during the EU FP7-funded project *iTesla* towards automatic ways of transforming power systems from proprietary format to Modelica, served as a proof of concept for the adoption of Modelica as a common and standardized language for power system modelling and simulation. This work is a continuation of the progress made during the *iTesla* project. This paper presents a tool developed with the main purpose of providing users with an easy way to generate power system networks in Modelica and perform time-domain simulations. The tool is validated by generating Modelica systems for IEEE cases and comparing simulation outputs with a reference commercial tool (Eurostag).

Keywords: *Modelica, open source software, power system modelling, power system dynamics, CIM.*

1 Introduction

The work presented here is a continuation of the developments made during the *iTesla*¹ project towards the automatic transformation of power system networks for performing phasor time-domain simulations using Modelica. The specification for the automatic transformation was presented in (Vanfretti et al., 2016) and this work presents its full implementation in a user-friendly and open source tool.

One of the most challenging objectives of the *iTesla* project was to conduct accurate pan-European security assessments taking into account system dynamics in addition to static analysis. This task requires the execution of time-domain simulations considering a large number of contingencies. But one of the main obstacles for running such simulations is that each European TSO relies on its own (proprietary) data format in order to describe dynamic models.

To address this issue, *iTesla* partners agreed to use Modelica² as the standard language for power system dynamic modelling. The use of Modelica as a common language for power system dynamic simulations began with

a previous European project named PEGASE³, where simple systems were studied. During the *iTesla* project several software modules were developed to automatically transform power system models from different proprietary formats, used by TSOs, to Modelica. Also, a Modelica library containing electrical and logical models was implemented (Bogodorova et al., 2013) and validated against domain-specific simulation tools (PSS/E and Eurostag). This library called iPSL (*iTesla* Power System Library), is open source and can be found at the project repository⁴.

The developments made during the *iTesla* project allowed the successful conversion of several European power systems ranging from a dozen to hundreds of buses, and a similar number of generator machines, from PSS/E or Eurostag format to Modelica. Time-domain simulations, including contingencies, were performed on the automatically converted systems, thus achieving one of the main objectives of the project. The *iTesla* project served as a *proof of concept* for the usage of Modelica as a standard language for power system modelling, but many manual tasks were still required for each study conducted.

At the end of the project, the authors of the present work decided to further expand the *iTesla* software developments exploiting some of the current results, the expertise gained in Modelica, and the team working strength, for the development of a simulation tool based on Modelica. This user-friendly tool, from now on referred to as Power Systems on Modelica (PSM) tool, automatically generates and simulates power systems in Modelica. The tool is intended to be fully compatible with the iPSL library (iPSL has been enlarged and improved for this purpose), but the user will not be limited to the library models, since any Modelica model may also be freely added.

One of the main goals pursued with the development of such a tool, which will be released open source, is to enable users to easily convert CIM⁵ electrical networks to Modelica, in order to ease the transition to an open equation-based language for power system modelling. This paper presents the automatic model generation

¹*iTesla*: Innovative Tools for Electrical System Security within Large Areas. <http://www.itesla-project.eu/>

²Modelica® and the Modelica Association. <https://modelica.org/>

³PEGASE: Pan European Grid Advanced Simulation and State Estimation. <http://www.fp7-pegase.com/>

⁴iPSL repository. <https://github.com/itesla/ipsl>

⁵ENTSO-E Common Information Model (CIM) for grid model exchange. <https://www.entsoe.eu>

developed for the tool and shows some preliminary results.

The paper is organized as follows. Section 2 presents the PSM tool and how the automatic model generation is achieved. In section 3 specific cases are used to validate the tool. And finally, section 4 presents the conclusions and future work.

2 PSM tool

The PSM tool⁶ is intended to generate power system networks in Modelica (*.mo files) from CIM files containing static system models and state variables from a given snapshot, plus dynamic data given in XML files. A power flow computation is run over the network to obtain the steady-state of the system. A Dynamic Data Repository (DDR) is used for defining which dynamic models, either from iPSL library or user-defined, should be used to map network equipment, and which parameters must be used to instantiate those dynamic models. The users also have the possibility to perform time-domain simulations using a Modelica solver engine (either OpenModelica or Dymola).

The tool has been designed to ensure modularity, allowing users to either run processes individually or as a full workflow. The software is built as a modular Java application. Modules can be used directly from the command line or from a simple JavaFX user interface. Various power flow engines and Modelica simulation engines can be used, and further user-defined interfaces may be added later on.

The general architecture of the tool is shown in Figure 1. The different modules involved in the conversion process are: 1) CIM importer, 2) Power flow computation, 3) Modelica file generation, 4) Dynamic simulation definition, and 5) Time-domain simulation.

The tool first converts CIM data files into the iTesla Internal Data Model⁷ (IIDM), on which a power flow is computed. Then another module generates the Modelica file connecting to the Dynamic Data repository to retrieve dynamic data. The user has the possibility of introduce events to be studied in the dynamic simulation, and finally run time-domain simulations with a Modelica engine (currently OpenModelica or Dymola) selecting the desired simulation parameters.

The individual modules are described below.

2.1 CIM importer

The main goal of this module is to import a network system model file in CIM format and convert it to IIDM. The module only supports CIM-compliant files (at the time this work was prepared: ENTSO-E CIM Profile 1). The user

⁶PSM tool will be released open source in a specific repository starting June 2017. The exact link will be published in the iPSL repository <https://github.com/itesla/ipsl>.

⁷The IIDM allows to import, export and edit power system models in the iTesla platform. Specifications on IIDM format can be found at the GitHub repository for iTesla Power System Tools. <https://github.com/itesla/ipst>

is free to manually write/update CIM files, as long as the generated file is CIM-compliant⁸.

2.2 Power flow computation

This module is in charge of computing the power flow over the IIDM network obtained in the previous step. The tool has been designed to work with two different alternatives for power flow computation:

- HELMTM-Flow power flow engine⁹.
- HADES power flow engine¹⁰.

The tool is intended to allow easy switching between these two engines. The values obtained from the power flow computation are re-injected into the IIDM network before moving on to the next module, i.e. the Modelica file generation. The user also has the possibility of deactivating the power flow computation and generating the Modelica file using input values from CIM.

2.3 Modelica file generation

This module is responsible for the generation of the Modelica (.mo) file. The tool enables the user to specify dynamic models (relying on the iPSL library and/or user-defined libraries) and parameters for each static item in the given network setup, as well as to provide default parameters for each dynamic model. Default dynamic models can also be defined for each static object type.

As depicted in the general architecture of the tool (Figure 1), this module connects to the DDR populated with the system dynamic data and all necessary models.

The DDR is based on a set of XML files that store:

- Which mappings from static network elements to dynamic models to use in the dynamic system model building (and its connections).
- The definitions required for building full model initialization simulations of complex dynamic models.
- The parameter sets used for dynamic model instantiations. Global declarations and system wide equations.

Full model initializations (see the blue box at the bottom of the *Modelica file generation* module in Figure 1) refer to the derivation of relevant and coherent initial values for all model variables based on power flow outputs and external parameters. This is done by performing very short-time simulations of the specific models using either OpenModelica or Dymola. This initialization is performed

⁸Although CIM format can be used for dynamic models exchange, it is currently hardly used for this purpose in the power system community. Therefore the DDR option was considered more appropriate for PSM.

⁹For more information on HELMTM-Flow, see <http://elequant.com/>. This product is based on the Holomorphic Embedding Load Flow Method (Trias, 2012).

¹⁰RTE official power flow engine.

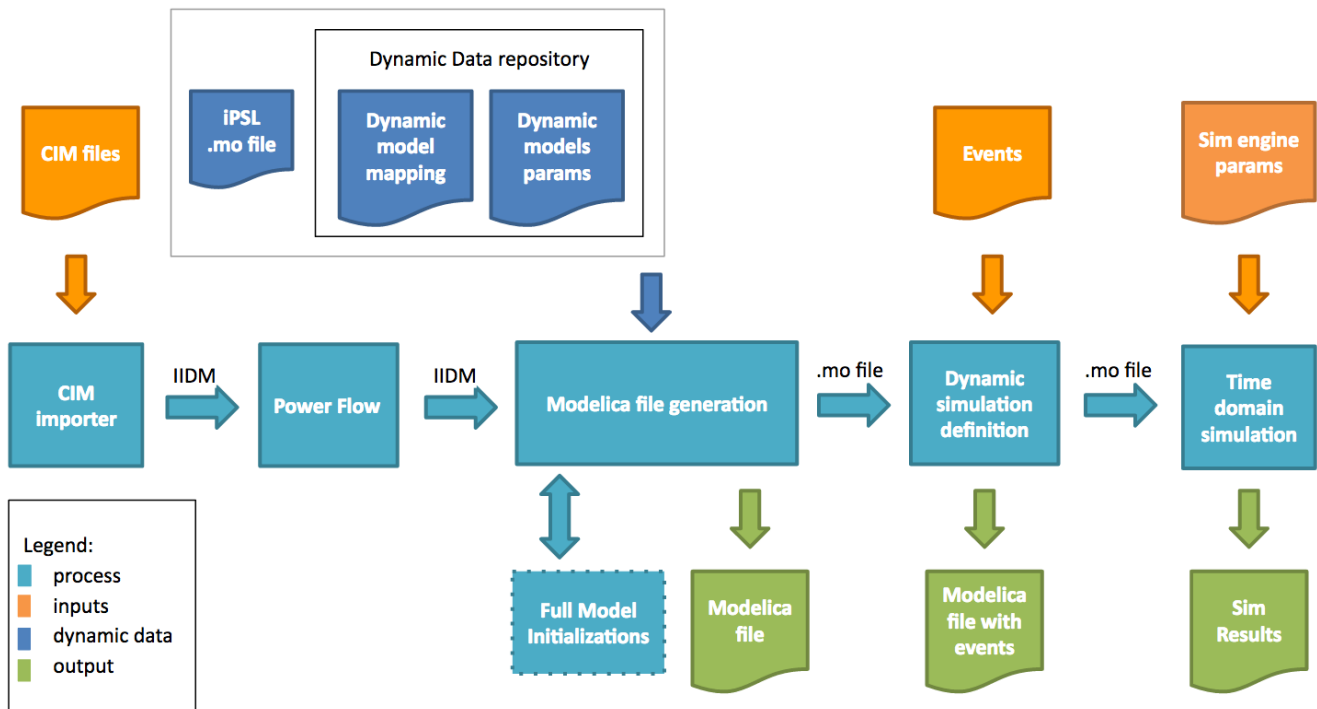


Figure 1. General Architecture of the PSM tool

only for component models requiring explicit initialization, which is the case of some generator machines and may be the case for specific control models. The main advantage of using *full model initialization* instead of *initial equation* inside Modelica models (the common initialization strategy in Modelica) is that the use of external (local) initializations can significantly increase simulation speed since it reduces the size of the initial system to be solved (especially for large systems) and it is designed to be very scalable. The use of *initial equation* may cause boundary problems and increase the number of equations, resulting in a decreased performance. See (Vanfretti et al., 2016) for more details on initializations schemes.

2.4 Dynamic simulation definition

This module is intended to define the simulation scenario. The user can define events to be triggered at specific times during the simulation. This module also allows defining load variations, capacitor changes, etc. As a result, one or several Modelica files are generated in addition to the Modelica base case generated in the previous step.

2.5 Time-domain simulation

This module is in charge of running time-domain simulations on the Modelica files generated with the tool, allowing the user to choose between two Modelica simulation engines: OpenModelica or Dymola. PSM fully relies on these engines to run simulations, and thus one of them must be previously installed before the user starts running the PSM tool.

OpenModelica is an open source Modelica environ-

ment, freely downloadable¹¹. Instead, Dymola is a commercial environment¹². The user may also directly open the generated Modelica file in any preferred Modelica environment for edition and simulation.

In PSM, the user has access to the definition of standard simulation setup (simulation interval, output interval, integration method, integration tolerance, fixed integrator step, etc.) given by the available simulation solver: OpenModelica or Dymola. The *Time-domain simulation* module will generate output files in the standard Modelica output format (MATLAB binary format *.mat), which is provided by both supported simulation engines, as well as in *.csv format. This will allow the user to import simulation results into the preferred Modelica environment or other software for plotting and analysis of results.

2.6 User interface

The tool also includes a basic graphical user interface with functionalities allowing the user to select which processes to run, on which data sets, and displaying progress and logs, without having to manually run command line actions.

3 Tool's validation

Different power system networks are being used to test and validate PSM. Some of these systems were already used for testing developments made during the iTesla project and the results are presented in (Vanfretti et al., 2016) and (León et al., 2015). In this work, validation was

¹¹OpenModelica. <https://openmodelica.org/>

¹²Dymola by Dassault Systèmes. <http://www.3ds.com>

also performed using IEEE¹³ cases (IEEE14, IEEE30, IEEE57, and IEEE118), which have not yet been used before.

All IEEE cases have been generated in Modelica format using PSM, and for this all necessary dynamic models from Eurostag were included in iPSL and DDR files were generated with the corresponding dynamic data. Eurostag equivalent models were created in Modelica using the exact same equations used by this proprietary software. This section presents the results obtained with the IEEE57 and IEEE118 cases.

The results obtained with the different generated Modelica systems are compared against the exact same networks built in Eurostag, taking this software as a reference. Validations are done both graphically and numerically. The numerical assessment is carried out using the Root Mean Square Error (RMSE) as done in previous works (Vanfretti et al., 2016). The assessment metric chosen to accept the results as valid is that the absolute RMSE is $\leq 10^{-03}$ for all compared values (voltage magnitudes in p.u. and angles in radians in all buses).

3.1 IEEE57 case

The IEEE57 test case is composed of 57 buses, 7 generators, 63 lines, 17 transformers, 42 loads, and 3 compensation banks. The generators are modelled in Modelica as synchronous machines defined by external parameters. This Eurostag's model, together with the machine described using internal parameters, were developed and included in iPSL during the iTesla project. The control systems present in this IEEE case are a voltage controller ExcSEXS, an stabilizer PSS13E2B, and a governor GovSteam0¹⁴. These models were developed in Modelica and Eurostag specifically for PSM, and have been included in iPSL¹⁵. Two types of events were introduced to validate the results of time-domain simulation performed with the Modelica model, using Dymola, against Eurostag.

Figure 2 shows the voltage magnitude at a given specific bus when a bus fault event is introduced (at this bus) at time $t=1$ seconds, lasting 0.2 seconds. As can be seen in the figure, the curves obtained with the Modelica system generated with PSM and simulated using Dymola (blue dashed line), and the Eurostag system (red solid line) match very well. The numerical assessment test was carried out obtaining a RMSE for all voltage magnitudes and angles below the defined threshold (10^{-03}).

Figure 3 shows the response of the IEEE57 test case to a line fault lasting 0.1 seconds occurring at time $t=1$ seconds, where again, the results obtained with Modelica system using Dymola (blue dashed line) and Eurostag (red solid line) are very similar. The RMSE obtained is below

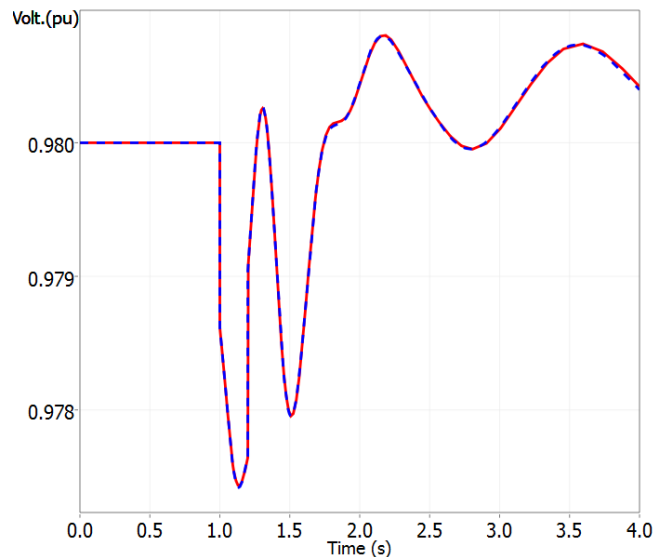


Figure 2. Simulation of the IEEE57 case with a bus fault of 0.2 seconds occurring at $t=1$ s. The blue dashed line corresponds to Dymola simulation results and the red solid line to Eurostag results.

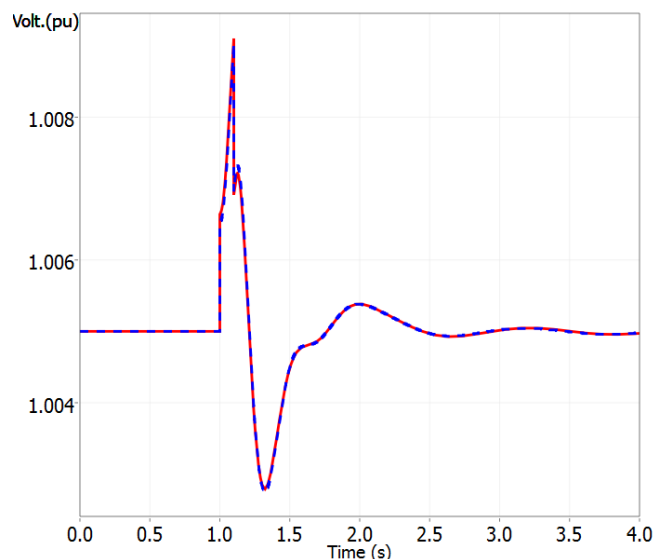


Figure 3. General Simulation of the IEEE57 case with an open line of 0.1 seconds occurring at $t=1$ s. The blue dashed line corresponds to Dymola simulation results and the red solid line to Eurostag results.

the threshold.

All simulations in Dymola were performed using the DASSL integration method with a tolerance of 10^{-06} . Typical times for running the type of simulations shown for the IEEE57 case are ~ 300 seconds to simulate 4 actual seconds. These simulations were run in a machine with the following characteristics: Intel® Xeon® CPU E5-2690 2.60 GHz (2 processors), 48 GB of installed memory (RAM) and 64-bit Operating System. Eurostag simulation for the same system takes only a few seconds to complete.

¹³Power Systems Test Case Archive from the University of Washington. <https://www2.ee.washington.edu/research/pstca/>

¹⁴Control systems were built following standard IEC 61970-302.

¹⁵All these Modelica models can be found at the iPSL repository. <https://github.com/itesla/ipsl>

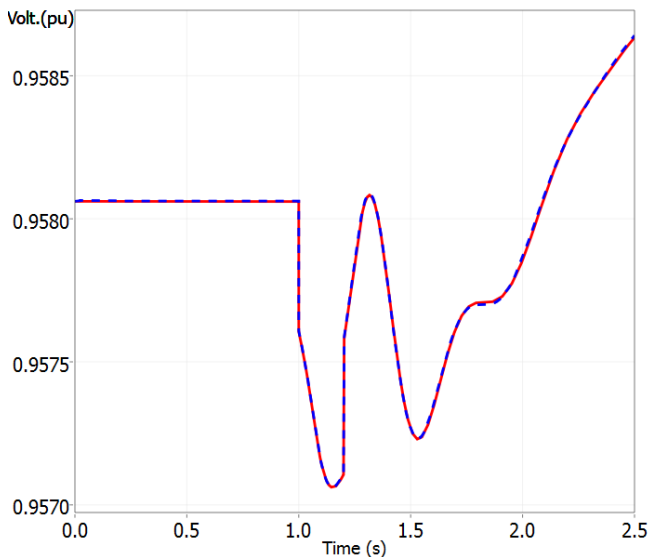


Figure 4. General Simulation of the IEEE118 case with a bus fault of 0.2 seconds occurring at $t=1$ s. The blue dashed line corresponds to Dymola simulation results and the red solid line to Eurostag results.

3.2 IEEE118 case

The IEEE118 case was also used to test PSM. This system is composed of 118 buses, 177 lines, 9 transformers, 54 generators, 91 loads, and 14 compensation banks. The dynamic models are the same as those used in the IEEE57 with CMCONST governor added.

Figure 4 shows the comparison between the Modelica generated file and Eurostag for the voltage magnitude at a specific bus, when a bus fault is introduced at time $t=1$ sec, lasting 0.2 seconds.

Due to the size of the IEEE118 network, the number of equations in Modelica ($\sim 14,000$ equations) increases considerably compared to the IEEE57 case ($\sim 2,000$ equations), and this greatly increases the simulation time in any Modelica engine. For the IEEE118 results shown, it takes Dymola approximately 6 hours to run 2.5 actual seconds (on the same machine used for the IEEE57 case). This is clearly a limitation for achieving scalability to real networks, which are of the order $\sim 10,000$ nodes. Further work is needed in this direction¹⁶.

4 Conclusions and future work

This paper presents the implementation of a user-friendly and open source tool that allows users to automatically generate and simulate power system networks in Modelica. This work is the result of years of development made during the iTesla project and the extension of those developments in a post-iTesla collaboration work. This collaboration motivated by the authors' commitment

¹⁶OpenModelica developers are working on the integration of the Sundials IDA solver in order to significantly improve simulation times for large DAE systems. For more information see <https://www.openmodelica.org/>.

to open simulation tools to improve cooperation and exchange for performing dynamic power system simulations.

Modelica allows enhanced transparency and results in power system modelling and simulation, with all the necessary ingredients to become a standard language for power system modelling. However, there is a clear need for easier ways to transform power systems networks into the Modelica language. The proposed tool is designed to provide such possibility and bridge the gap between Modelica and power system simulations, in order to ease widespread adoption among the power system community.

For the development of this tool, the authors have proposed a simple architecture that, given the correct inputs, allows the user to generate a Modelica file and simulate it using a Modelica solver engine. The tool has been validated with a variety of power system networks, comparing simulation results with a reference commercial software (Eurostag). In particular, results obtained with two IEEE cases are presented here, showing very accurate results.

Nevertheless, further work on Modelica simulation engines is needed in order to achieve scalability. The networks studied are below or around a hundred nodes, but simulation time increases excessively with systems size and rapidly becomes unmanageable above 100 nodes.

PSM will be released as open source in a specific repository starting June 2017. The exact link will be published in the iPSL repository: <https://github.com/itesla/ipsl>.

References

- 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. *Smart Grid Technologies Europe (ISGT EUROPE), 2013 4th IEEE/PES*, pages 1–5, 2013. doi:10.1109/ISGTEurope.2013.6695422.
- G. León, M. Halat, M. Sabate, J-B. Heyberger, F.J. Gomez, and L. Vanfretti. Aspects of power system modeling, initialization and simulation using the Modelica language. *IEEE PES Innovative Smart Grid Technologies Europe*, pages 1–6, 2015. doi:10.1109/PTC.2015.7232504.
- A. Trias. The Holomorphic Embedding Load Flow method. *Power and Energy Society General Meeting, 2012 IEEE*, pages 1–8, 2012. doi:10.1109/PESGM.2012.6344759.
- L. Vanfretti, A. Adib Murad, F. Gómez, G. León, S. Machado, J-B. Heyberger, and S. Petitrenaud. Towards Automated Power System Model Transformation for Multi-TSO Phasor Time Domain Simulations using Modelica. *PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2016 IEEE*, pages 1–7, 2016. doi:10.1109/ISGTEurope.2016.7856341.