

Power System Model Identification

Exploiting the Modelica Language and FMI Technologies

Luigi Vanfretti, Tetiana Bogodorova¹, Maxime Baudette¹

1: Electric Power Systems Department, Smart Transmission Systems Lab. (SmarTS Lab.)

KTH Royal Institute of Technology, Stockholm, Sweden.

2: Statnett SF, Research & Development, Oslo, Norway.

Abstract—This article provides an overview of the work performed at SmarTS Lab on power system modeling and system identification within the FP7 *iTesla* project. The work was performed using Modelica as the modeling language for phasor time domain simulation and FMI (Flexible Mock-up Interface) Technologies for coupling Modelica models with simulation and optimization tools. The article focuses on use case examples of these Modelica models in an FMI driven environment to perform parameter identification.

Index Terms—Modelica, FMI, model identification, parameter estimation, system identification.

I. INTRODUCTION

Modeling power system components is important for different studies. The level of details in the component models is directly correlated to the type of studies performed. In contrast to Electro-Magnetic Transient (EMT) studies, stability assessment studies only require a simplified representation of the components valid for a low bandwidth, which are used in positive sequence phasor time domain simulation. This is motivated by the fact that the primary concerns for stability assessment are dynamics in a relatively low frequency range. This simplified representation not only preserves the necessary dynamics of the components for stability analysis, but it also enables the study of large networks due to the lower mathematical complexity of each component model.

A. Motivation: Model Calibration

Power system component models may have been developed and validated against a reference component in specific use case and have a certain validity domain. In commercial tools dedicated to power systems analysis this step is performed by the vendor. The validity of such models used within other use cases is directly linked to the fidelity of the parametrization.

Available model parameters may not give the expected results. One reason can be that the model has been validated against a different component or that the parameters have changed over time. A calibration phase is thus necessary to ensure that the model used will represent correctly the component it is used for. Calibrating a model consists of several phases of parameter tuning until the response of the model to certain types of common perturbations matches the measured response of the actual component to the same perturbations.

Another case in which model calibration can be used is when replacing several components by an aggregate model. The aggregation process is not straightforward as there is no unique mathematical method to derive the parameters of

an aggregate model. The calibration of the aggregate is thus necessary to validate the representation.

B. Modelica and FMI Technologies

The work presented in this paper is part of the FP7 *iTesla* project [1], where a power system component library for phasor domain simulation has been developed. The Modelica language was chosen for this task because it offers a formal mathematical language and because it separates the model from the solver [2]. Further details on power system modeling using Modelica are available in [3], [?].

The Modelica language also interacts well with technologies compliant with the Functional Mock-up Interface (FMI) standard. This standard allows for model exchange between different tools which support and implement the standard. The models are exchanged in the form of standardized compiled objects. The standard offers the possibility to exchange single models or structures containing both a model and a specific solver. It is for example implemented in Modelica development environments such as OpenModelica from the Open Source Modelica Consortium [5], Dymola from Dassault [4], and JModelica.org from Modelon [6] for both import and export; as well as co-simulation.

The FMI standard [7] offers great possibilities for model reuse within different software tools. As such, the models developed and used in Modelica are not locked in the development environment, contrarily to traditional tools for power system modeling. Traditional tools can be restrictive in the sense that the user is bound to their dedicated modeling and simulation environment. With Modelica and FMI, the models can be used in other software tools, opening a great array of possibilities. The experiences presented in this paper use models built with components from the aforementioned Modelica power system library, and are integrated through FMI technologies with simulation and optimization tools.

C. Paper Organization

The remainder of this paper is organized as follows. The software environment is described in Section II. The results of a parameter identification experiment are shown in Section III and the results of an aggregate load model identification experience are presented in Section IV. Conclusions are drawn in Section V

II. MODEL IDENTIFICATION USING MODELICA AND FMI

Within workpackage 3 of the FP7 *iTesla* project, the Rapid Parameter Identification toolbox (RAPID), is being developed.

This section explains how Modelica models are coupled with simulation and optimization tools in MATLAB using different FMI Technologies. A future publication will report in detail the architecture and functions of RAPID. The main function of RAPID is to perform identification on a selected set of model parameters as illustrated in Fig. 1. The validity of a set of parameter values is assessed by evaluating the fitness of the simulated response of the model with the set of parameters, and compared to a reference measured response provided as input to RAPID. Sets of parameters are generated using optimization algorithms built upon different techniques. The toolbox is being developed using a plugin architecture to add custom or external optimization tools.

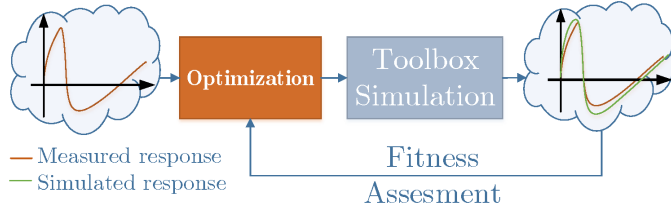


Fig. 1. RAPID's working principle

RAPID uses MATLAB as an integration layer. The top-level MATLAB code acts as a wrapper to provide interaction with several other programs. The role of each language and/or software is as follows:

Modelica: The toolbox performs parameter identification on Modelica models. The language is thus used for developing test systems including several power system components. An example is shown in Fig. 3, for details see [2], [3].

FMU: A Flexible Mock-up Unit (FMU) is product of the compilation of a model according to the FMI standard. It is a C-object containing the methods for simulation. FMUs are used as input to RAPID. Its parameters will be optimized.

MATLAB: It is used to combine all the software behind the toolbox's simulation and optimization functions. It provides a Graphical User Interface (GUI) and a Command Line Interface (CLI) for the user of the toolbox. Built-in parameter optimization algorithms are also developed in MATLAB, and several MATLAB toolboxes are used within RAPID.

SIMULINK: It is used to configure the models used by RAPID for simulating new sets of parameters and generating an output used for fitness assessment.

FMI Toolbox: It is a MATLAB toolbox from Modelon AB [8] providing FMI standard support to the MATLAB/SIMULINK environment. It provides an FMU import Simulink block used to simulate the aforementioned FMUs. It enables setting new values to parameters of the model.

Extra/Optional: The toolbox provides a plugin capability to integrate new optimization algorithms through a wrapper. As such, the KNITRO [9] optimization software has been interfaced with the toolbox.

RAPID has been used for several power system identification applications and the results of two of them are

TABLE I
PARAMETERS FOR IDENTIFICATION AND MEASURED OUTPUT

Model Parameters	R_a	Armature resistance
	X_d	Direct axis reactance
	X'_d	Direct axis transient reactance
	T'_d	Direct axis transient time constant
	X_q	Quadrature axis reactance
	M	Inertia coefficient
Model Outputs	D	Damping ratio
	$ V $	Voltage magnitude
	ω	Rotor speed
	P	The machine active power
	Q	Reactive power

presented in this paper. The first example deals with parameter identification on a synchronous generator. The second example deals with parameter identification of aggregate load models.

III. COMPONENT MODEL PARAMETER IDENTIFICATION EXPERIENCE

The goal of this experience is to identify the parameters of a simplified dynamic model of a generator that can match the electro-mechanical dynamics of the reference model. The experiment setup made for identification of the generator parameters is the following:

- The reference system is modeled in SPS/SIMULINK. It represents a generator and a load connected to the grid through short transmission lines (1 km). The generator used as reference is modeled through a detailed EMT-type high-order model.
- An identical power system model is implemented in Modelica. The generator, which will be calibrated to reproduce the reference response, is represented using a low-order generator model. For this purpose a 3rd-order synchronous generator model was chosen. The power flow solution for initializing the Modelica models is imported from PSAT [10], an external tool.
- Seven generator parameters were estimated by measuring four outputs (see Table I.)

RAPID receives the model's simulated response through an FMU block provided by the FMI Toolbox for MATLAB/SIMULINK (see Fig. 2)

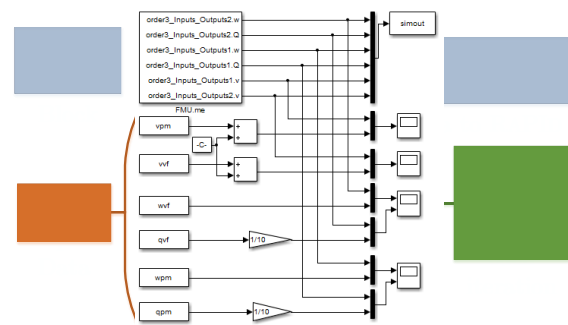


Fig. 2. Simulink model with the FMU of the Modelica model

TABLE II
METHODOLOGY FOR PARAMETER IDENTIFICATION USING RAPID

Algorithm 1 Identification process using RaPIId	
Input: Measurements InputDataFile [time, outputs], Modelica model(*.fmu), output variables* estimated parameters* (range of possible values)	
Initialize PSO algorithm:	
-	PSO algorithm settings**, number of iterations: $M = 1$.
-	Iteration counter i , stopping criteria: MSE_{max} .
Iteration:	
1.	while $i < M$ begin
2.	call PSO algorithm
3.	call (Simulink system (with *.fmu))
4.	return (estimated parameters)
5.	calculate MSE w.r.t. output variables and outputs
6.	if $MSE < MSE_{max}$ then
7.	evaluate (estimated parameters, MSE)
8.	break; else continue; end
Initialize Naive (or NM) algorithm:	
-	estimated parameters, Naive (or NM settings)
-	stopping criteria: maximum MSE: MSE_{max} ,
-	maximum iterations: $IterMax$
Iteration:	
9.	while $i < IterMax$ begin
10.	call Naive (or NM) algorithm
11.	call (Simulink system (with *.fmu))
12.	calculate MSE w.r.t. output variables and outputs
13.	if $MSE < MSE_{max}$ then
14.	evaluate (estimated parameters, MSE)
15.	break; else continue; end
Output: estimated parameters, MSE	
Note: * see Table I;	
** Chose the maximum number of particles to initialise PSO	

RAPID allows the user to choose from different optimization algorithms, and to combine them. This example exploits gradient-based methods and a meta-heuristic algorithm [11]. For the gradient descent (or direct search) algorithm to perform well, a good starting point (which is close to the optimum) is required.

A suitable starting point was obtained using the Particle Swarm Optimization (PSO) [12] algorithm. In practice, it was sufficient to find the particle with best fitness after one iteration only. The methodology is summarized in Table II.

The experiment was carried out for two perturbations designed to excite the dynamics of the system. The load parameters of the model of Fig. 3 were assumed as known values. The first perturbation is a pulse of 0.5 sec. of duration of 1% the nominal torque in the shaft of the generator. The second one is a pulse of 0.5 sec. of duration of 1% nominal field voltage of the machine. The perturbations in the torque will primarily excite mechanical dynamics, enabling the identification of mechanical parameters. When perturbing the system with respect to field voltage, the direct axis time constant influences the machine voltage dynamics and has impact on output of active and reactive powers (P, Q).

The results of identification experience are presented in Fig. 4 and Table III. The values of the identified parameters are shown in Table III. The numerical values for all the parameters were constrained before running the algorithms to a valid range of real-valued numbers, typical of synchronous

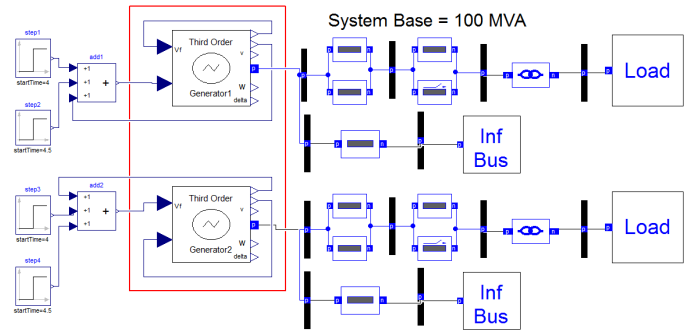
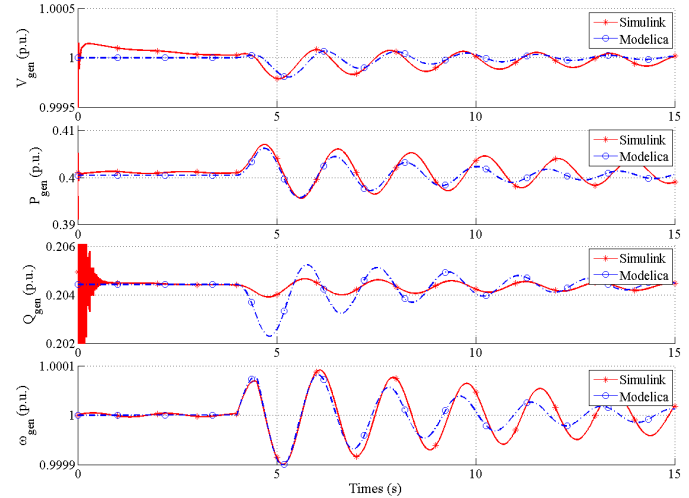
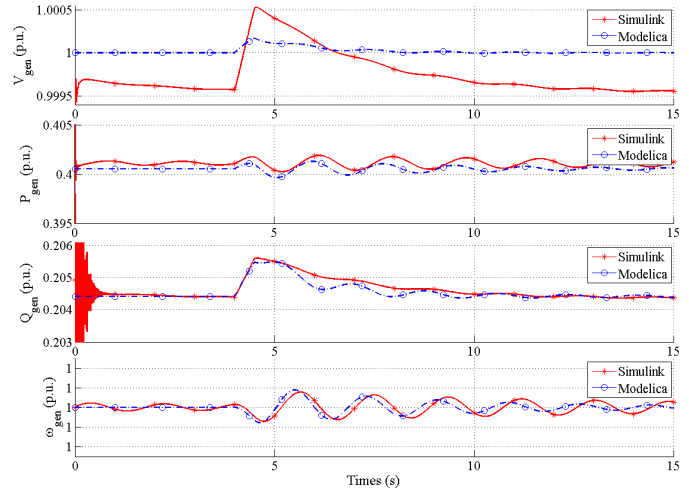


Fig. 3. Modelica model used in the identification process (component to be identified bounded in red)



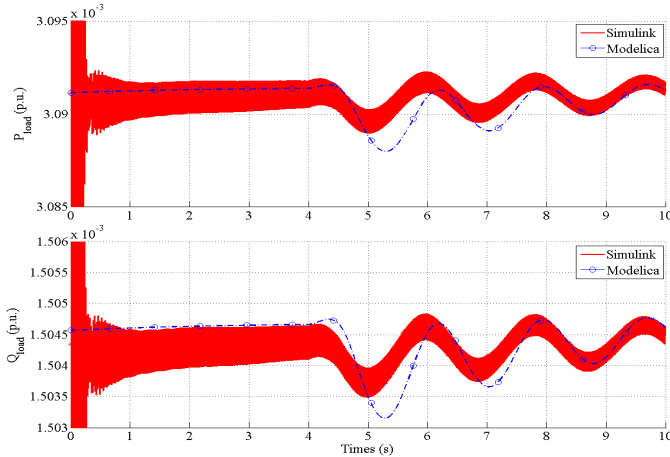
(a) Torque perturbation



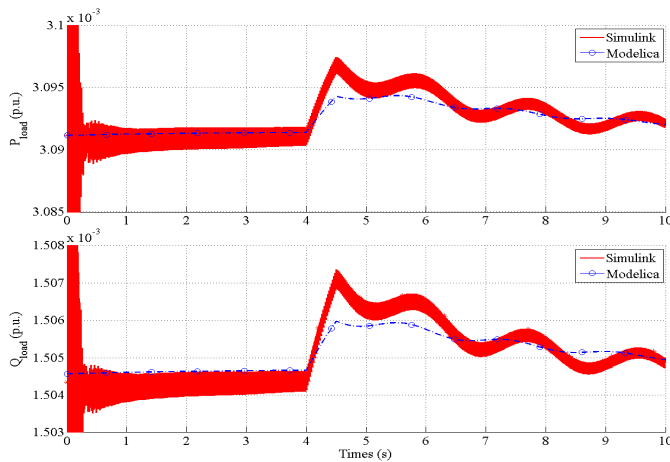
(b) Field voltage perturbation

Fig. 4. Comparison between the reference (Simulink) and the identified (Modelica) model responses with a perturbation at $t=4$ sec.

machines. Thus, the resulting parameter values are within practical and realistic. Fig. 4 shows a graphical comparison between the simulations in SPS/SIMULINK and Modelica. It shows that the responses match with an acceptable error. The key here is to remember that a reference signal (from a high order model (SPS/SIMULINK)) is matched to a low-



(a) Torque perturbation



(b) Field voltage perturbation

Fig. 7. Exponential recovery aggregate load model results

coefficient are above 7, thus the variation of the active power and reactive power is affected by the voltage variation at the load.

 TABLE VII
PARAMETERS - AGGREGATE ZIP LOAD MODEL

Parameter	Est. Value
Constant Z coefficient for active power (k_{pz})	1
Constant I coefficient for active power (k_{pi})	0
Constant P coefficient for active power (k_{pp})	0
Constant Z coefficient for reactive power (k_{qz})	1
Constant I coefficient for reactive power (k_{qi})	0
Constant P coefficient for reactive power (k_{qp})	0
Mean squared error	5.1415e-007

The ZIP load model has the constraint:

$$k_{pz} + k_{pi} + k_{pp} = 1 \quad (1)$$

RAPID allows to consider constraints in the form ($min < x < max$). To be able to take into account the equality constraint in (1), a change of variables was used. Instead of working in the Cartesian coordinate frame, the identification was held in the spherical coordinate frame, setting the constraints to $0 < r < 1$, $0 < \Theta < \pi/4$, $0 < \Phi < \pi/4$,

which force the solution space to the first 1/8 of the sphere. So the final expressions used to represent the constraints are:

$$\begin{aligned} k_{pz} &= (r_p \sin(\Theta_p) \cos(\Phi_p))^2, \\ k_{pi} &= (r_p \sin(\Theta_p) \cos(\Phi_p))^2, \\ k_{pi} &= (r_p \cos(\Theta_p))^2. \end{aligned}$$

The results presented in Table VII show a dominant constant impedance behavior of the aggregate load. The MSE results for this load are high, thus it means that the difference between the reference and the Modelica aggregate load model is the highest. The identification results from different aggregate load models are compared in Table VIII.

 TABLE VIII
LOAD AGGREGATION ERROR COMPARISON

Type of load	Mean squared error
Exponential Recovery	1.8627e-007
Voltage Dependent	2.8357e-007
Frequency Dependent	3.3287e-007
ZIP	5.1415e-007

Based on the maximum fitness criteria (minimum MSE), the aggregate load model that gives the most satisfactory match to the behavior of the measured data is the Exponential Recovery Load model. Further experiments can be performed in order to improve the aggregate load model response compared to the reference model. For example, instead of identifying load models independently, the identification could be performed using a combination of load models in parallel and identifying which percentage corresponds to each type of load.

V. CONCLUSION

This article provided an overview of the work of the authors on parameter and model identification for power system component and aggregate models. The work encompasses the development of a library of component models in Modelica, the development of an identification toolbox using those models and proof of concept experiments.

The results of the experiments presented in this paper are encouraging. Some shortcomings in the results are acknowledged, but the authors emphasize that the focus of this article was on showing the possibilities to exploit Modelica and FMI technologies through the RAPID toolbox and the Modelica power system library developed for power system model identification.

This first experience using Modelica and FMI technologies offers opportunities for further work. RAPID can help as a tool to develop methods to determine the choice of parameters to be identified, development of optimization methods and the design of experiments for power system identification.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of Dr. Alan Collinson of SP Energy Networks who provided the original model of the feeder used for load model aggregation in Section IV.

L. Vanfretti was supported by Statnett SF, the Norwegian Transmission System Operator, the STandUP for Energy collaboration initiative, the -EU funded FP7 *iTesla* project and Nordic Energy Research through the STRONG²rid project.

T. Bogodorova was supported by the EU funded FP7 *iTesla* project.

M. Baudette was supported by the EU funded FP7 *iTesla* project and by Statnett SF, the Norwegian Transmission System Operator.

The support of the funding bodies mentioned above is gratefully acknowledged.

REFERENCES

- [1] *iTesla*: Innovative Tools for Electrical System Security within Large Areas. [Online]. Available: <http://www.itesla-project.eu/>
- [2] L. Vanfretti, W. Li, T. Bogodorova, and P. Panciatici, "Unambiguous Power System Dynamic Modeling and Simulation using Modelica Tools," *IEEE PES General Meeting 2013*.
- [3] T. Bogodorova, M. Sabate, G. León, L. Vanfretti, M. Halat, J. Heyberger, and P. Panciatici, "A Modelica Power System Library for Phasor Time-domain Simulation." *IEEE PES Innovative Smart Grid Technologies (ISGT)*, october 2013.
- [4] Dymola. [Online]. Available: <http://www.dymola.com>
- [5] P. Fritzson, P. Aronsson, H. Lundvall, K. Nystrom, A. Pop, L. Saldmli, and D. Broman, "The openmodelica modeling, simulation, and software development environment," *Simulation Notes Europe*, no. 45, pp. 8–16, 2005.
- [6] JModelica.org - an extensible Modelica-based open source platform for optimization, simulation and analysis of complex dynamic systems. [Online]. Available: <http://www.jmodelica.org/>
- [7] Functional Mock-up Interface. [Online]. Available: <https://fmi-standard.org/>
- [8] FMI Toolbox for Matlab. Accessed: 2013-11-22. [Online]. Available: <http://www.modelon.com/products/fmi-toolbox-for-matlab/>
- [9] KNITRO - a solver for nonlinear optimization. [Online]. Available: <http://www.ziena.com/knitro.htm>
- [10] F. Milano, "An open source power system analysis toolbox," *IEEE Transactions on power systems*, vol. 20, pp. 1199 – 1206, 2005.
- [11] O.Nelles, *Nonlinear System Identification*. Springer, 2010.
- [12] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proceedings, IEEE International Conference on Neural Networks, 1995.*, vol. 4, 1995, pp. 1942–1948 vol.4.