Available online at www.sciencedirect.com
ScienceDirect
SoftwareX 5 (2016) 144–149
SoftwareX
www.elsevier.com/locate/softx

Original software publication

RAPID: A modular and extensible toolbox for parameter estimation of Modelica and FMI compliant models

Luigi Vanfretti[a,b], Maxime Baudette[a,*], Achour Amazouz[c], Tetiana Bogodorova[a], Tin Rabuzin[a], Jan Lavenius[a], Francisco José Goméz-López[a]

[a] *SmarTS Lab., KTH Royal Institute of Technology, Stockholm, Sweden*
[b] *Research and Development Division, Statnett SF, Oslo, Norway*
[c] *Control System Design, Rolls-Royce Deutschland, Blankenfelde-Mahlow, Germany*

Received 12 February 2016; received in revised form 25 July 2016; accepted 26 July 2016

Abstract

This paper describes the RApid Parameter IDentification toolbox (RAPID), developed within the EU FP7 ***iTesla*** project. The toolbox was designed to carry out parameter identification on models developed using the Modelica language, focusing in particular on power system model identification needs. The toolbox has been developed with modularity and extensibility in mind, using MATLAB/SIMULINK as a plug-in environment, where different tasks of the identification process are carried out. The identification process uses different optimization algorithms to improve the fitting of the model's response to selected criteria. The modular architecture of RAPID gives users complete freedom to extend and adapt the software to their needs, e.g. to implement or link external solvers for simulation or optimization. The compatibility with Modelica models is brought by the use of technologies compliant with the Functional Mock-up Interface (FMI) standard.

Keywords: FMI; Modelica; Model validation; System identification

---

Code metadata

Current code version	v0.9
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-16-00026
Legal Code License	LGPL 3.0
Code versioning system used	git
Software code languages, tools, and services used	Matlab
Compilation requirements, operating environments & dependencies	Matlab/Simulink R2011b to R2014b (MathWorks) with: - Optimization Toolbox - Global Optimization Toolbox - Statistics Toolbox - Signal Processing Toolbox - Image Processing Toolbox FMI Toolbox (Modelon)
If available Link to developer documentation/manual	https://github.com/ElsevierSoftwareX/SOFTX-D-16-00026/blob/master/README.md
Support email for questions	

1. Motivation and significance

Model validation, and parameter identification methods and tools are of paramount importance in many engineering fields,

---

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

http://dx.doi.org/10.1016/j.softx.2016.07.004

whose origins can be traced back to Gauss' development of the least squares method [1] in 1795. The purpose of such methods and tools is to ensure that the mathematical description of a dynamical system corresponds to its real behavior for a given validity domain (i.e. the model is "good enough" for a given "application"). The system's behavior is exposed in measurements and knowledge about the system's structure. In the case of power systems, Phasor Measurement Units (PMU) collect high-speed, time-synchronized data from multiple locations and containing multiple signal types, also known as synchrophasors [2]. Synchrophasor data gives engineers the opportunity to extract knowledge about the system's dynamic response. With the advent of more advanced information technology, data processing and knowledge extraction methods are now accessible for different domain experts. This opens research and development opportunities in model validation for power systems, which has been identified as an important PMU application [3].

Today's methods and tools for model identification and validation have their theoretical underpinning in the field of system identification, which has been recently unified with the development of prediction error methods [4]. This recent development establishes a framework that allows one to make a clear separation between (a) the model structure being employed to represent the system, (b) the estimation method used to identify the chosen model structure, (c) the algorithms used for the solution, and (d) their actual implementation in computer software.

Understanding this framework from the field of system identification and paying special attention to domain-specific needs (i.e. power system requirements), have provided the catalysts for the development of the Rapid Parameter Identification (RaPId) toolbox. The authors' aspiration with the design and implementation of this tool is to provide a framework/prototyping environment to solve system identification problems in power systems, while at the same time designing and implementing a software architecture that structurally achieves an actual separation [5] according to the modern principles of system identification (i.e. (a) through (d) above).

2. Software description

The RaPId toolbox was developed to automate the model validation, calibration, and parameter identification process for models exchanged according to the FMI standard [6]. The toolbox is provided with the following inputs:

1. A Matlab/Simulink [7] model with a Functional Mock-up Unit (FMU)[1] that encapsulates the dynamic model of interest;
2. A reference response (e.g. a time-series from measurements) that defines the variable(s) to be used in the cost function to be minimized;

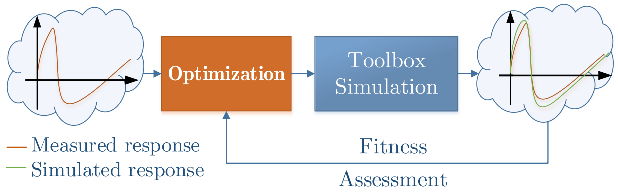


Fig. 1. Operating principle of the RaPId toolbox showing how the toolbox, through an iterative process, calibrates the model based on a fitness criterion between the model's response and a reference response.

3. A parameter set to be estimated/calibrated; for each parameter, its initial value, and a range defined by its minimal and maximal value;
4. The selected optimization and simulation algorithm (available or newly added), and its specific settings;
5. Type of cost function (i.e. objective function) (available or newly added).

Note: the user needs to consider the non-linearities of the studied system(s) and pay careful attention when configuring the experiment settings [5].

The toolbox will automatically load the initial parameter set, simulate the model, and record the output of the model (from a list of preliminary chosen output variables), as shown in Fig. 1. RaPId will attempt to tune the parameters of the model so as to minimize the given criteria (e.g. by defining a curve fitting criterion between the simulated model's output and the provided reference as an objective function). The toolbox is flexible, it allows the user to select any number of reference signals. The fitness function (i.e. objective function) can be defined by the user, or the user can utilize a set of default criteria.

2.1. Software architecture

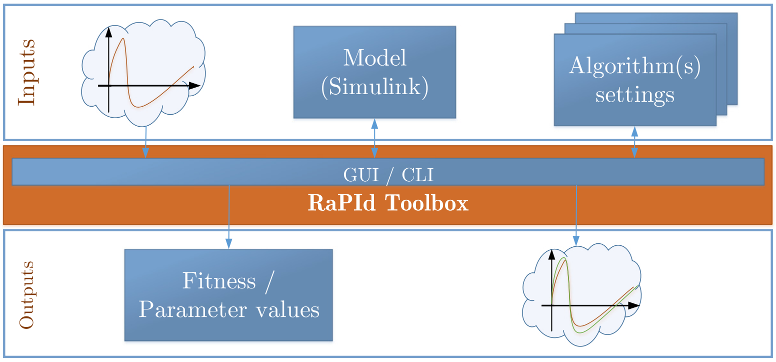
The toolbox has been designed as a modular software package (see Figs. 2(a) and 2(b)). It includes a graphical user interface (GUI) (implemented in the *gui* folder; to launch with *runRapidGui*), allowing its use as a standalone tool. The toolbox can also be called through a command line interface (CLI) in Matlab, enabling its integration in scripts for additional interactions with other toolboxes or software.

The software code is divided into several modules that carry out individual tasks of the automated process, and handle the communication between the modules. The core code, in *functions/rapid.m*, manages the experiment settings and the overall process execution. The other modules provide the following services: optimization, model simulation, and objective function computation. RaPId is made available with several implementations of each of these modules. The optimization services available include support for both conventional algorithms, such as *fmincon* in Matlab [8], and also heuristic algorithms, such as a particle swarm optimization (PSO) algorithm implemented by the authors. The simulation services available were implemented to utilize FMI technologies [9], which allow to use the solvers natively available in Matlab/Simulink. The FMI compliance is ensured via the FMI Toolbox [10] that brings compatibility with FMU 1.0 and 2.0.

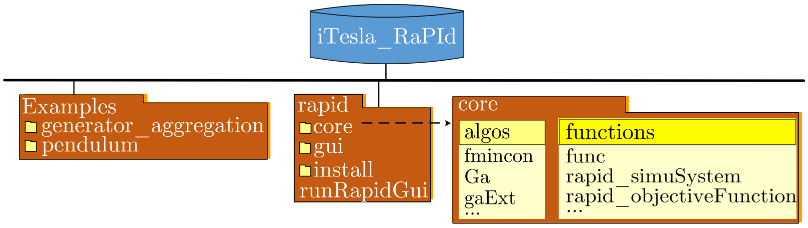
The user can also easily implement new modules. In particular, the optimization facilities use a plug-in architecture

---

[1] A FMU is a compiled model following the FMI standard. The FMI standard is now supported by more than 30 tools, see http://www.fmi-standard.org/tools —OpenModelica supports FMU generation from Modelica models, see http://www.openmodelica.org.



(a) Software architecture.



(b) Software architecture implementation.

Fig. 2. Architecture of the RaPId toolbox (a) showing its interaction with external components detailing the inputs to provide and the outputs obtained from the toolbox. The implementation (b) is done using a nested folder structure.

that can be extended to include additional optimization algorithms. This is achieved by defining, in *algos/*, a function that integrates the objective function and the user options. Further integration in the toolbox is achieved by extending *functions/rapid.m* with a new case, and by editing the GUI in *gui/rapidMainWindow.fig* in GUIDE.

2.2. *Software functionalities*

The RaPId toolbox was developed as a software prototype and as one of the modules of the ***i****Tesla* platform (http://www.itesla-project.eu/) involving model validation, calibration and parameter estimation. It provides the necessary methods to carry out:

- Component level validation and parameter estimation, i.e. given the component reference (measurements) and knowledge (assumptions) on the component model, perform parameter identification and compare with the original values and perform cross validation to calibrate the parameters values in the model;
- Development of aggregate models/model reduction, i.e. substitute a set of components with an aggregated model for which the estimated parameter values will give a minimal error according to the cost function;
- Large networks validation, i.e. estimate a parameter set for several components of large interconnected networks.

The iterative process is automatic, i.e. a given model response is automatically matched against a given reference (measured) response. The use of FMI technologies adds compatibility with Modelica models, such as the ones developed also within the ***i****Tesla* project [11], as several Modelica tools support the FMI standard for model export.

3. Illustrative example

In order to avoid going into domain-specific complexities, this section provides a simple illustrative example. Actual power system identification examples can be found in [12].

In power system studies, as well as in other domains, aggregation (or model order reduction) is a common practice that aims to simplify a system model by replacing several units by a single one, while preserving the original aggregate behavior. The goal of this “use case” is to illustrate this by performing parameter identification of an aggregated generator model using the RaPId Toolbox. The setup of the experiment is as follows:

- The reference model is shown in Fig. 3. It is a basic power system model with a generator, a load and several lines, to which a plant consisting of 2 generators is connected. All generators are represented by the *GENROE* model (see [11]);
- Fig. 3 shows the aggregated model where the two generators of the original plant are replaced by an equivalent single generator plant;
- The Modelica models are initialized using the power flow solution imported from PSS/E (a power system analysis tool) (see [13]);
- The minimization criterion (i.e. objective function) was set as the sum of the Euclidean distances between the aggregate model’s response and the reference model’s response. The sum considers the three following outputs (at bus *GEN1*): terminal voltage, active power, and reactive power;
- An identical small signal perturbation was introduced in both models to excite the systems’ dynamics (outlined in orange on Fig. 3), thereby making the excitation voltage

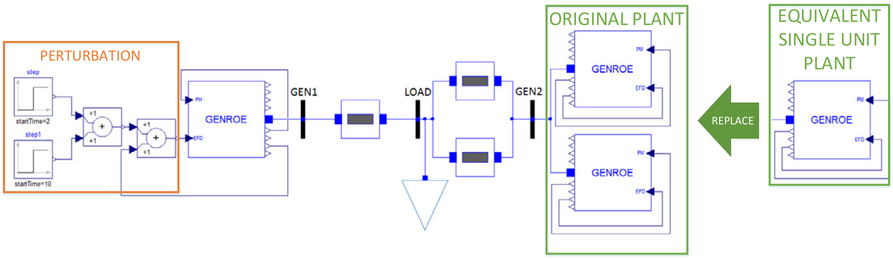


Fig. 3. Reference and aggregated models used for the parameter identification process. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

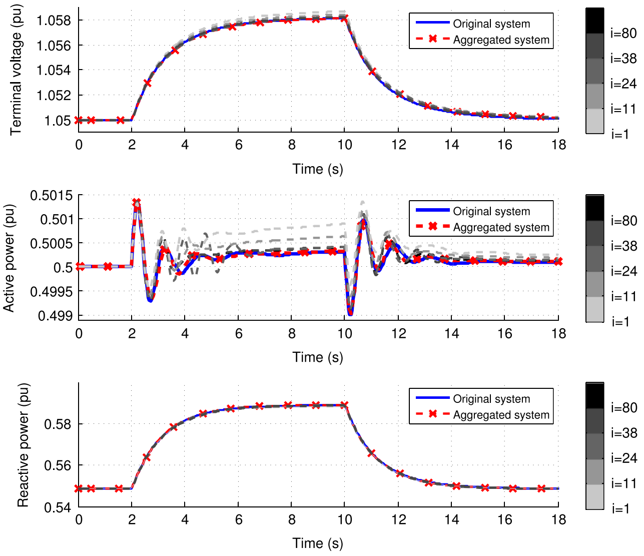


Fig. 4. Comparison between the responses of the reference model and the aggregated model submitted to excitation voltage perturbations at $t = 2$ s and $t = 10$ s. Partial results are shown at different iterations of the process (see color scales on the right side). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(i.e. *EFD*) of the generator on the left-hand side increase temporarily;

- Using the three simulated outputs (at bus *GEN1*), 12 parameters of the aggregated plant were identified using the PSO algorithm.

The calibration process is iterative and evaluates the different parameter sets until the model's response is close enough to the original plant's response (i.e. satisfying the minimization criteria). Fig. 4 shows a comparison of the aggregated model's response against the reference system. Both models are perturbed through a step-up at $t = 2$ s and step-down at $t = 10$ s. signal in the field voltage of the generator at bus *GEN1*. It can be noted that as the simulations with a new parameter set progress, the responses get closer to the reference response.

Once the optimization process is over, the toolbox returns a list of numerical values for all the parameters. These values are within a valid practical range, because the RAPID experiment was configured to restrict each parameter to a range of typical values for synchronous machines. Furthermore, it can be noted that the calibrated aggregated plant response closely matches that of the original plant (see Fig. 4).

4. Impact

Power systems are becoming highly complex cyber-physical networks and model validation tools have been highlighted as a major and urgent need for the analysis of power systems of different complexity and scale [3]. There are several tasks that would benefit from more flexible tools, such as the estimation and validation of parameters of a single generating unit [14], the optimization of control parameters [15] through the identification of aggregate model parameters [16], and the validation of overall power grid dynamics [3].

Indeed, a good power system model requires not only a correct mathematical representation of the physical components, but also the correct parametrization corresponding to the modeled physical system. Therefore, Transmission System Operators (TSOs) have recently focused on maintaining a database of parameters representing the actual power system, while the mathematical representations of most power system components started to be developed in the 1970s. This task is challenging for several reasons. First, the information that is available from the grid is incomplete. In addition, intellectual property right issues for technology do not allow one to openly exchange modeling details about some of the power system components. Second, the usage of diverse simulation tools by different TSOs translates into information loss when exchanging only parametrization data. Consequently, great efforts have recently been made to develop the common information model (CIM) [17]. However, there are several limitations to CIM and the Common Grid Model Exchange Standard (CGMES) [18] that do not guarantee unambiguous model exchange [19]. Recent work on this front that considers the use of Modelica to define "user defined models" [20] might help to improve these limitations in the future.

In addition, power system model validation tools available today [3] have several limitations, such as: (a) narrow application scope (e.g. only for single component/purpose model validation), a monolithic architecture that (b) binds the whole identification process to a specific method/algorithm (e.g. Kalman Filter [16]) and/or (c) simulation tool, and (d) none of the available tools address the important issue of unambiguous model exchange.

The RAPID toolbox seeks to offer a flexible solution that is compliant with unambiguous model exchange by the use of standardized software components, such as the FMI for model exchange. Furthermore, the toolbox was built to be easily extended, allowing the user to integrate its own optimization algorithms, and cater to specific identification problems.

As illustrated in this paper, another case where the model calibration can be used is when simplifying models, where an aggregate model replaces several components. The aggregation process is not straightforward as there is no unique mathematical method to derive the parameters of an aggregate model. Its calibration is thus necessary to validate the simplified model.

This is a typical task for many TSOs who need to aggregate an entire power plant (composed by several generators and their automatic control systems) into a single generator representation in order to decrease the overall state-space dimension of the power grid model [21]; mostly to simplify analyses and result interpretation, but also to reduce simulation time. Although there are some tools for such purpose, these depend either on an ambiguous model description [22] or are bound to a specific simulator [16]. As this use case demonstrates, the use of RAPID does not impose these requirements.

The development of an open source tool is relevant in the context of today's needs in power systems, and it is the humble attempt of the authors to make an open source contribution that helps addressing them. The authors also wish to democratize the use of Modelica and FMI for tasks related to power system modeling, simulation, and model validation, by providing a useful toolbox supporting such technologies.

5. Conclusions

Power systems are becoming highly complex cyber-physical networks, and model validation tools have been highlighted as a major and urgent need for the analyses of power system descriptions of different complexity and scale [3].

This paper gave an overview of the RAPID toolbox developed within the EU FP7 ***iTesla*** project. It is a humble attempt of the authors to advance the state-of-the-art in today's power system model identification practices, by making this project available as open source. The authors acknowledge that there might be an immense array of possible features, optimization methods, etc., that could have been implemented already—therefore, the authors have decided to release this software using an OSS license so that further development can be made regardless of the authors' time resources.

The software will be maintained by the authors during 2016 and until funding is available for its development by the first author's research team. Therefore, to allow the software to grow, the authors would like to encourage users to provide their own application examples or any other contributions towards the further development of the toolbox.

Acknowledgments

The support of the following funding bodies is gratefully acknowledged:

- Statnett SF, the Norwegian Transmission System Operator, through projects Symptom and RT-VS.
- EC funding through the FP7 ***iTesla*** project.
- Nordic Energy Research through the STRON g^2rid project.
- The STandUP for Energy collaboration initiative, through the SRA grant supporting the first author.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.softx.2016.07.004.

References

[1] Abdulle A, Wanner G. 200 years of least squares method. Elem Math 2002;57(2):45–60. http://dx.doi.org/10.1007/PL00000559.

[2] Almas MS, Baudette M, Vanfretti L, Løvlund S, Gjerde JO. Synchrophasor network, laboratory and software applications developed in the STRONg^2rid project, in: 2014 IEEE PES general meeting — conference & exposition. 2014. p. 1–5 http://dx.doi.org/10.1109/PESGM.2014.6938835.

[3] NASPI Synchrophasor Technical Report, Model Validation Using Synchrophasors. Tech. Rep. (2013).

[4] Ninness B. Some system identification challenges and approaches, Vol. 42. Elsevier BV; 2009. p. 1–20. http://dx.doi.org/10.3182/20090706-3-FR-2004.00001.

[5] Ljung L. Perspectives on system identification. Ann Rev Control 2010; 34(1):1–12. http://dx.doi.org/10.1016/j.arcontrol.2009.12.001.

[6] Fritzson P. Introduction to modeling and simulation of technical and physical systems with modelica. Hoboken, NJ, USA: Wiley-Blackwell; 2011. http://dx.doi.org/10.1002/9781118094259.

[7] MATHWORKS, SIMULINK. URL http://mathworks.com/products/simulink/.

[8] MATLAB, Optimization toolbox. URL http://mathworks.com/help/optim/index.html.

[9] Functional mockup interface. URL https://fmi-standard.org/.

[10] Modelon AB, FMI Toolbox for Matlab. URL http://www.modelon.com/products/fmi-toolbox-for-matlab/.

[11] Vanfretti L, Rabuzin T, Baudette M, Murad M. iTesla power systems library (iPSL): A modelica library for phasor time-domain simulations, SoftwareX. http://dx.doi.org/10.1016/j.softx.2016.05.001.

[12] Bogodorova T, Vanfretti L, Turitsyn K. Bayesian parameter estimation of power system primary frequency controls under modeling uncertainties. IFAC-PapersOnLine 2015;48(28):461–5. http://dx.doi.org/10.1016/j.ifacol.2015.12.171.

[13] 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.

[14] Chow J, Glinkowski M, Murphy R, Cease T, Kosaka N. Generator and exciter parameter estimation of fort patrick henry hydro unit 1. IEEE Trans Energy Convers 1999;14(4):923–9. http://dx.doi.org/10.1109/60.815009.

[15] Liu G, Venkatasubramanian VM, Carroll JR. Oscillation monitoring system using synchrophasors. In: 2009 IEEE power energy soc. gen. meet. institute of electrical & electronics engineers (IEEE). 2009. p. 1–4. http://dx.doi.org/10.1109/PES.2009.5275209.

[16] Kalsi K, Sun Y, Huang Z, Du P, Diao R, Anderson KK, Li Y, Lee B. Calibrating multi-machine power system parameters with the extended kalman filter. In: 2011 IEEE Power Energy Soc. Gen. Meet. institute of electrical & electronics engineers (IEEE). 2011. p. 1–8. http://dx.doi.org/10.1109/PES.2011.6039224.

[17] Lambert E, Yang X, Legrand X. Is CIM suitable for deriving a portable data format for simulation tools? IEEE PES Gen Meet 2011;1–9. http://dx.doi.org/10.1109/PES.2011.6039354.

[18] ENTSO-E, Common Grid Model Exchange Standard (CGMES). URL https://www.entsoe.eu/major-projects/common-information-model-cim/cim-for-grid-models-exchange/standards/Pages/default.aspx.

[19] 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. institute of electrical & electronics engineers (IEEE). 2015. p. 1–5. http://dx.doi.org/10.1109/PESGM.2015.7286434.

[20] Annex: ENTSO-E. Annex F (normative) — use of modelica in the dynamics profile. Common Grid Model Exchange Specification (CGMES), Version 2.5, Draft IEC 61970-600 Part, Edition 2, 2016. Document version 1: July 8 2016. To be draft in Sep 2016. Online: https://www.entsoe.eu/major-projects/common-information-model-cim/cim-for-grid-models-exchange/standards/Pages/default.aspx.

[21] Galarza R, Chow J, Price W, Hargrave A, Hirsch P. Aggregation of exciter models for constructing power system dynamic equivalents. IEEE Trans Power Syst 1998;13(3):782–8. http://dx.doi.org/10.1109/59.708632.

[22] Chow JH. Power system coherency and model reduction. In: Power electronics and power systems. New York: Springer; 2014. URL https://books.google.se/books?id=HGnABAAAQBAJ.