RaPID - Rapid Parameter Identification
An open source software for model identification and validation leveraging Modelica and FMI Technologies

Tin Rabuzin on behalf of:
E-mail: rabuzin@kth.se

Prof. Dr.-Ing. Luigi Vanfretti
E-mail: luigiv@kth.se
Web: http://www.vanfretti.com

luigiv@kth.se
Associate Professor, Docent
Electric Power Systems Dept.
KTH
Stockholm, Sweden

Luigi.Vanfretti@statnett.no
Special Advisor in Strategy and Public Affairs
Research and Development Division
Statnett SF
Oslo, Norway
Outline

• Background and Motivation
  – Modelica and Power System Modeling
  – Why do we need Model Validation?
  – Software Requirements

• RaPId Overview

• Use Cases
  – Generator Aggregation
  – Excitation system identification
  – N44 - Small Signal Model Calibration

• Conclusions and Recommendations
Present challenges, limitations and possible solution

POWER SYSTEM MODELING
• The order of computations is decided at modelling time

<table>
<thead>
<tr>
<th>Acausal</th>
<th>Causal</th>
</tr>
</thead>
<tbody>
<tr>
<td>R*I = v;</td>
<td>i := v/R;</td>
</tr>
<tr>
<td>v := R*i;</td>
<td>v := R*i;</td>
</tr>
<tr>
<td>R := v/i;</td>
<td>R := v/i;</td>
</tr>
</tbody>
</table>

• Most tools make no difference between “solver” and “model” – in many cases solver is implanted in the model

• There is no guarantee that the same standardized model is implemented in the same way across different tools

• Even in Common Information Model (CIM) v15, only block diagrams are provided instead of equations

• Models are black boxes whose parameters are shared in a specific “data format”

• For large models this requires translation into the internal data format of each program

ENTSO-E has recognized the Modelica approach and it will be used in IOP Tests in July
Modelica and Power Systems

- **Modelica** is an open standardized modeling language among all Modelica compliant IDEs

- **iPSL** is an open-source Modelica library for power systems
  - It contains a set of **power system components** for **phasor time domain** modeling and simulation
  - Models have been **validated** against a number of reference tools

- **iPSL** allows:
  - **Unambiguous** model exchange
  - Formal **mathematical description** of models
  - Exploitation of **object-oriented** paradigms
  - **Separation** of **models** from IDEs and **solvers**
Modelica model of Nordic44 system

- Modelica can be used to build models of various sizes
- Norwegian TSO Statnett provided a PSS/E model of Nordic44 system
- The same model was implemented in Modelica and validated against a reference software, PSS/E
Assume that you have a “good enough” model, then what?

WHY POWER SYSTEM MODEL VALIDATION?
iTesa tools aim to perform “security assessment”

The quality of the models used by off-line and on-line tools will affect the result of any SA computations

- **Good model**: approximates the simulated response as “close” to the “measured response” as possible

Validating models helps in having a model with “good sanity” and “reasonable accuracy”:

- Increasing the capability of reproducing actual power system behavior (better predictions)
What is required from a SW architecture for model validation?

- Support “harmonized” dynamic models
- Process measurements using different DSP techniques
- Perform simulation of the model
- Provide optimization facilities for estimating and calibrating model parameters
- Provide user interaction
A model validation and parameter identification SW

THE RAPID TOOLBOX
What is RaPlId?

- **RaPlId** is a toolbox providing a general framework for parameter identification.

- Any model made available through a Functional Mock-Unit (FMU) in the Simulink environment, is characterized by a certain number of parameters whose values can be independently chosen.

- **RaPlId** attempts to tune the parameters of the model so as to satisfy the user-defined fitness function.
Coupling Models with Simulation & Optimization: FMI and FMUs

- **FMI** stands for Functional Mock-up Interface:
  - **FMI** is *a tool independent standard* to support both model exchange and co-simulation of dynamic models using a combination of xml-files and C-code, originating from the automotive industry.

  The FMI Standard is now supported by 40 different simulation tools.

- A **Functional Mock-up Unit (FMU)** is a model which has been compiled using the FMI standard definition.
How does RaPId work?

1. Output (and optionally input) measurements are provided to RaPId by the user.

2. At initialization, a set of parameters is pre-configured (or generated randomly by RaPId).

3. The model is simulated with the parameter values given by RaPId.

4. The outputs of the model are recorded and compared to the user-provided measurements.

5. A fitness function is computed to judge how close the measured data and simulated data are to each other.

2'. Simulations continue until a min. fitness or max no. of iterations (simulation runs) are reached.
• **RaPID** was developed in **MATLAB**.
  – The MATLAB code acts as *wrapper* to provide interaction with several other programs (which may not need to be coded in MATLAB).

• Optimization process can be set up and ran from the **GUI** or more advanced users can simply use **MATLAB scripts** for the same purpose.

• **Plug-in Architecture**:
  – **Completely extensible and open architecture** allows advanced users to add:
    • Identification methods
    • Optimization methods
    • Specific objective functions
    • Solvers (numerical integration routines)

• A number of optimization algorithms are available:
  – Particle Swarm Algorithm (PSO)
  – Genetic Algorithm (GA)
  – Naïve method
  – Knitro Algorithm
Implementation Overview

Call from the GUI or the CLI. The settings and data structure (RaPlIdObject) is passed.

Read settings and optimization method

RaPlIdObject
  - experimentData
  - experimentSettings
  - algorithmSettings
  - parameterNames
  - fmuInputNames
  - fmuOutputNames

rapid.m

Call optimization method

x_algo.m
Implementation Overview

**rapid_objectiveFunction.m**
- Calculate fitness factor according to the selected criterion

**rapid_simuSystem.m**
- Simulates the system with the vector of parameters from optim. algorithm

**func.m**
- Call simulation
- Pass the simulation results

**x_algo.m**
- Provide param. vector
- Iteratively generate vector of parameters according to the optim. algorithm

**Return fitness factor**
Parameter and Mode Estimation

USE CASES
Excitation system identification

Problem Formulation

• This use case deals with the parameter identification of the excitation system.

• Estimation is based on the real data acquired on the hydro-power plant Mostar.

• Measurements were acquired during the disturbance to the voltage reference of the Automatic Voltage Regulator (AVR).

• The disturbance was in form of successive 5% step increase and decrease of the voltage reference.

• It will be illustrated how estimation can be performed with limited information:
  - No approx. exciter parameters known.
  - Governor model is unknown.
  - Plant and system configurations surrounding the generator are unknown.
The simple model of the power system was built in Modelica.

The generator whose excitation parameters were identified is connected to the infinite bus through the line.

The load is connected to the generator bus.

The model of the excitation system is a simplified model based on the excitation system manufacturer’s recommendations.
As it could be seen on the previous two slides, no turbine governor has been used in the model of the power system.

If the measurement of the active power is observed, in addition to the electromechanical mode of oscillation, the slower mode related to the turbine governor can be observed.

The bandpass filter was applied to the signal to isolate the electromechanical mode of the oscillation.
Excitation system identification

Simulation and Results
Generator Aggregation

Problem Formulation

Generator model

Original Model

Aggregated Model
Generator Aggregation

Calibration results
7 parameters calibrated in 3 consecutive simulations
• Previous examples used time domain response of the systems to perform validation

• In this example, in addition to the time domain response, small signal characteristic of the system will be used as well

• RaPID will perform the linearization of the system and extract the mode of the system with currently set parameters

• The fitness function (performance indicator) which is used with small signal analysis is an Euclidean distance between the measured and the pole obtained from the linearization of the system:

\[
P_I = \|s_{model} - s_{ref}\| = \sqrt{(\sigma_{model} - \sigma_{ref})^2 + (\omega_{model} - \omega_{ref})^2}
\]

• In RaPID, it is also possible to perform validation using both the time domain and small signal performance integrator. This is done by merging the two criterias into one using weighting coefficients:

\[
P_I = w_1 P_{I\text{small signal}} + w_2 P_{I\text{time domain}}
\]
The calibration of the generator inertia in the N44 system has been carried out on the marked generator.

The disturbance is introduced to the system in form of line opening between buses 3244 and 6500.

Three signals are used for parameter estimation:
- Terminal voltage magnitude
- Terminal voltage angle
- Active power transfer over the faulted line

The calibration is carried out with the following setting of performance indicator:

\[ PI = w_1 PI_{small}^{signal} + w_2 PI_{time}^{domain}, \quad w_1 = 1000, \quad w_2 = 1 \]

The large difference between two weighing factors is due to the numerical difference between the two performance indicators (small signal and time domain).

The true value of the estimated generator inertia is 3.556 and the starting guess is 4.556.
Nordic44 – Small Signal Model Calibration

Estimation Results

- The figure shows the evolution of the parameter value with respect to the number of iterations being carried out.
- The total number of iterations is 70 and the estimated parameter value is 3.5546 which is very close to the true value of 3.556.
- The dots marked in red are parameter values currently giving optimum and the ones in blue are just attempts by the algorithm.
- The total number of iterations is 70 and the estimated parameter value is 3.5546 which is very close to the true value of 3.556.
- The dots marked in red are parameter values currently giving optimum and the ones in blue are just attempts by the algorithm.
- The objective function evolution shows the same behaviour of the estimation process from a different perspective.
- Convergence of the estimation was reached in a definite time and the objective function is close to the value of 0.
- Time domain results show a very good fit of data.
- However, since the small signal analysis was used in parameter estimation, one should also look at the changes in the stability plane.
- This figure shows all of the modes of the system.
- The position of true poles, poles with the parameter set to the initial value and poles with the estimated value of the parameter.
- However, the estimated process was focused only on one mode of oscillations shown on this figure.
- We can now conclude that the estimation procedure was successful since both time domain results and the small signal analysis of the system with optimal and true value give a close match of the two systems.
Validating power system models requires to develop new methods and new tools itself:

- The tools for model validation can be built independent from a specific power system simulator, thanks to the development of the Modelica library which allows to run the models with different tools and using FMUs.
- Model validation tools developed in this approach will provide additional flexibility to couple in a modular fashion: simulation, optimization and signal processing tools.

Model validation results, including validation metrics and parameters are sent back to the dynamic database which updates these specifications.
Conclusions and Looking Forward

• Modeling power system components with Modelica (as compared with domain specific tools) is very attractive:
  – Formal mathematical description of the model (equations)
  – Allows model exchange between Modelica tools, with consistent (unambiguous) simulation results

• The FMI Standard allows to take advantage of Modelica models for:
  – Using Modelica models in different simulation environments
  – Coupling general purpose tools to the model/simulation (case of RaPlId)

• There are several challenges for modeling and validating “large scale” power systems using Modelica-based tools:
  – A well populated library of typical components (and for different time-scales)
  – Support/linkage with industry specific data exchange paradigm (Common Information Model - CIM)

• Rapid provides a general framework for validation of models available through the FMI interface:
  – Models can be validated at different levels
  – Its architecture is completely modular
  – It is not tied to the domain specific tools
RaPiD and iPSL! Now Available as OSS!

Download at:
https://github.com/SmarTS-Lab/iTesla_RaPiD

Download at:
https://github.com/itesla/ipsl