iTesla

Leveraging the use of PMU data for *power system model identification through* **Modelica and FMI technologies**

Tetiana Bogodorova and Prof. Luigi Vanfretti

KTH Royal Institute of Technology / Statnett SF

tetianab@kth.se, luigiv@kth.se, luigi.vanfretti@statnett.no

KTH VETENSKAP OCH KONST

SmarTS Lab
Smart Transmission Systems Laboratory

Statnett

NASPI North American SynchroPhasor Initiative

Working Group Meeting
March 11-12, 2014 - Knoxville, TN, USA

Acknowledgment

- The work presented here is a result of the collaboration between RTE (France), AIA (Spain) and KTH (Sweden) within the EU funded FP7 iTesla project: http://www.itesla-project.eu/

- The following people have contributed to this work:
  - RTE: Patrick Panciatici, Jean-Baptiste Hyberger, Angela Chieh
  - AIA: Gladys De Leon, Milenko Halat, Marc Sabate
  - KTH: Wei Li, Tetiana Bogodorova, Maxime Baudette, Achour Amazouz, Joan Russiñol, Mengjia Zhang, Janelle (Le) Qi, Francisco Gómez

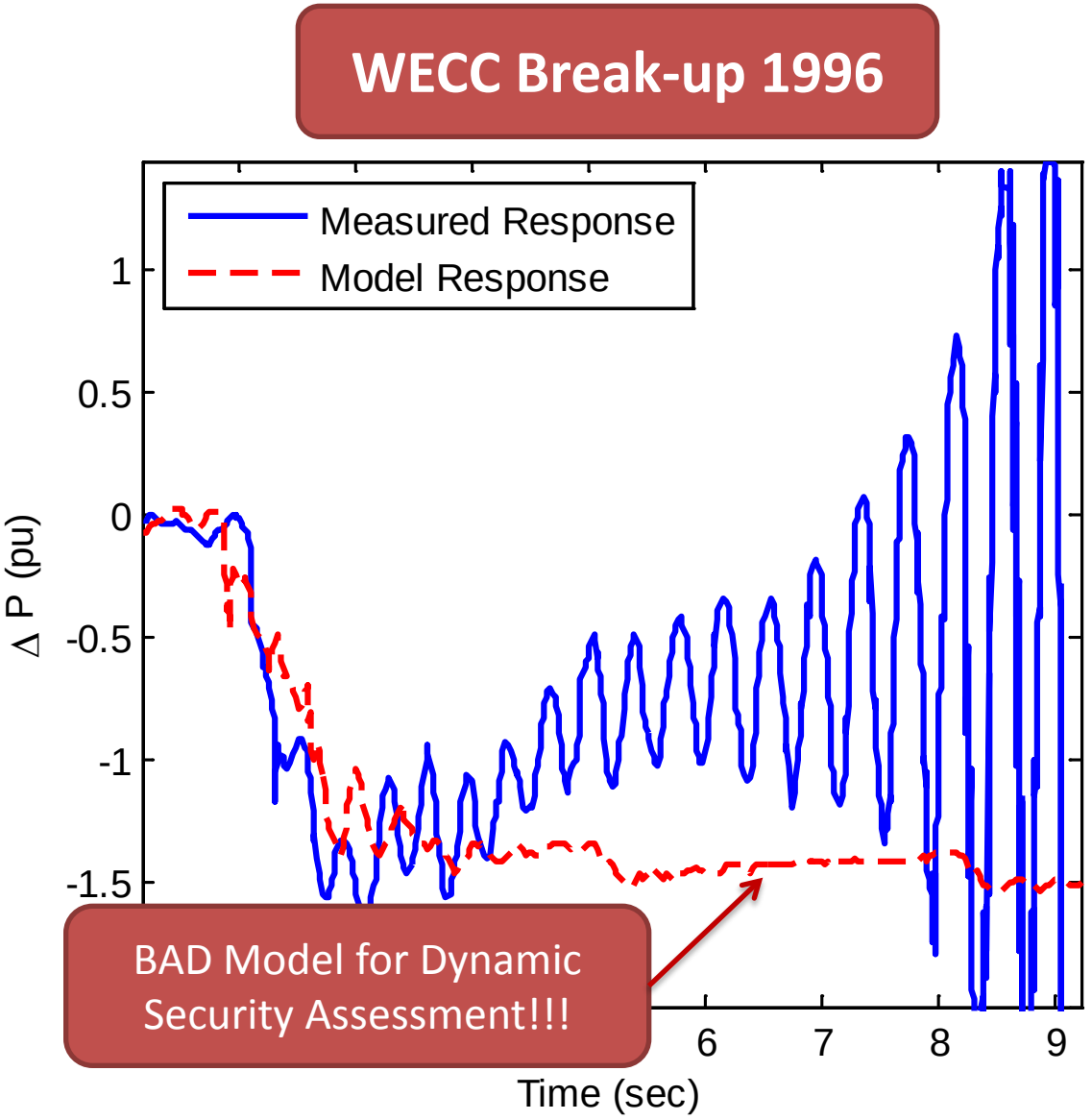
- Special thanks for 'special training' and support from
  - Prof. Fritzson *and his team* at Linköping University
  - Prof. Berhard Bachmann and Lennart Ochel, FH Bielefeld

Outline

- Background – Why Model Validation?
- The power system model validation loop
- Power System Modeling
  - Status quo and consequences
  - What is Modelica?
  - The iTesla Power Systems Modelica Library
  - The FMI Standard for tool-independent model exchange and co-simulation
- Flexible SW for Model Validation and Calibration
  - SW architecture requirements
  - RaPId: a proof of concept using Modelica and FMI Technologies
- Case Studies and Customized Methodologies:
  - Identification of generator parameters
  - Identification of parameters for aggregated load
- Conclusions

Why "Model Validation"?

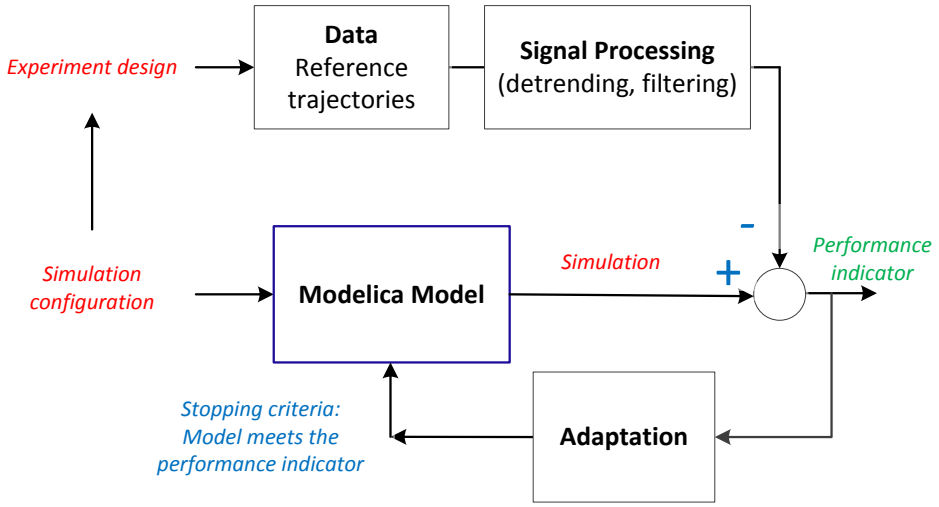
- **iTesla tools aim to perform "security assessment"**
- The quality of the models used by off-line and on-line tools will affect the result of any SA computations
  - *Good model*: approximates the simulated response as "close" to the "measured response" as possible
- Validating models helps in having a model with "good sanity" and "reasonable accuracy":
  - Increasing the capability of reproducing actual power system behavior (better predictions)



The Model Validation Loop

- The major ingredients of the model validation loop below have been incorporated into the proposed general framework for validation:

Validation Loop



- **A model can <u>NEVER</u> be accepted as a final and true description of the actual power system.**
  - *It can only be accepted as a suitable ("Good Enough") description of the system for specific aspects of interest (e.g. small signal stability (oscillations), etc.)*
  - Model validation can provide ***confidence*** on the fidelity and quality of the models for replicating system dynamics when performing simulations.
  - Calibrated models will be needed for systems with more and more dynamic interactions

Power System Modeling

limitations, inconsistency and consequences

- Causal Modeling:
  - Most components are defined using causal block diagram definitions.
  - User defined modeling by scripting or GUIs is sometimes available (casu
- Model sharing:
  - Parameters for black-box definitions are shared in a specific “data forma
  - For large systems, this requires “filters” for translation into the internal
- Modeling inconsistency:
  - For (standardized casual) models there is no guarantee that the model e same way in different SW
  - This is even the case with CIM dynamics, where no formal equations are definition is provided.
  - User defined models and proprietary models can’t be represented with platform
- Modeling limitations:
  - Most SWs make no difference between “model” and “solver”, and in ma *implanted* within the solver (inline integration, eg. Euler or trapezoidal so)

An equation based modeling language can help in avoiding all of these issues!

- Consequence:
  - It is almost impossible to have **the same model** in different simulation platforms.
  - This requires usually to re-implement the whole model from scratch (or parts of it) or to spend a lot of time “re-tuning” parameters.

This is very costly!

Modelica

The Next Generation Modeling Language

i.e., Modelica is **not** a tool

Declarative language

Equations and mathematical functions allow acausal modeling, high level specification, increased correctness

Multi-domain modeling

Combine electrical, mechanical, thermodynamic, hydraulic, biological, control, event, real-time, etc...

Everything is a class

Strongly typed object-oriented language with a general class concept, Java & MATLAB-like syntax

Visual component programming

Hierarchical system architecture capabilities

Efficient, non-proprietary

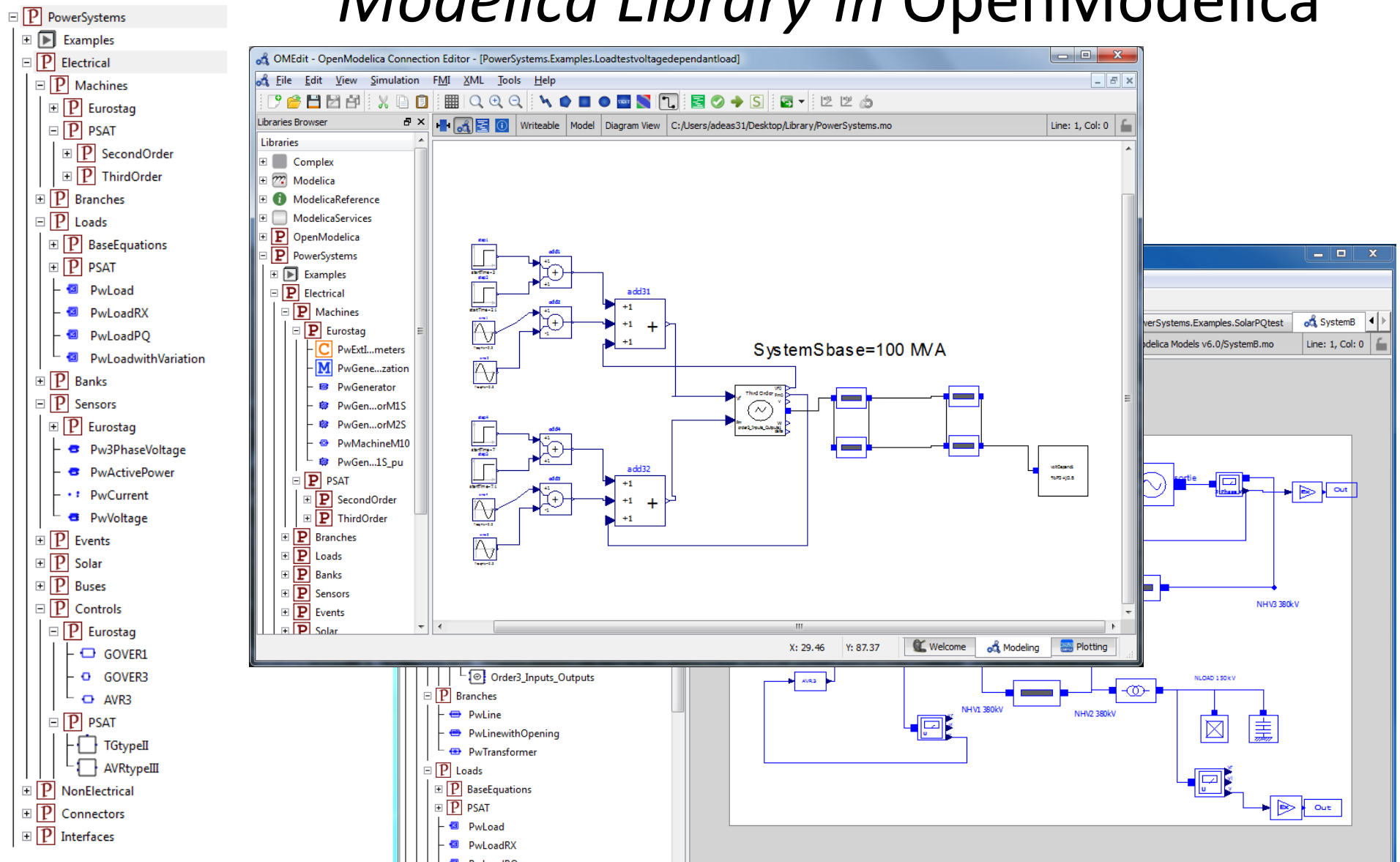
Efficiency comparable to C; advanced equation compilation, e.g. 300 000 equations, ~150 000 lines on standard PC

Used with permission of Prof. Peter Fritzson:

Open Source Modelica Consortium

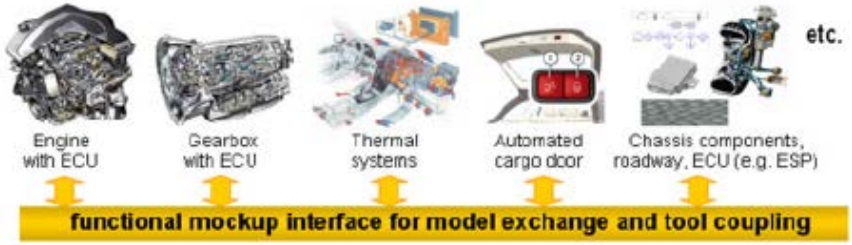
iTesla Power Systems

Modelica Library in OpenModelica



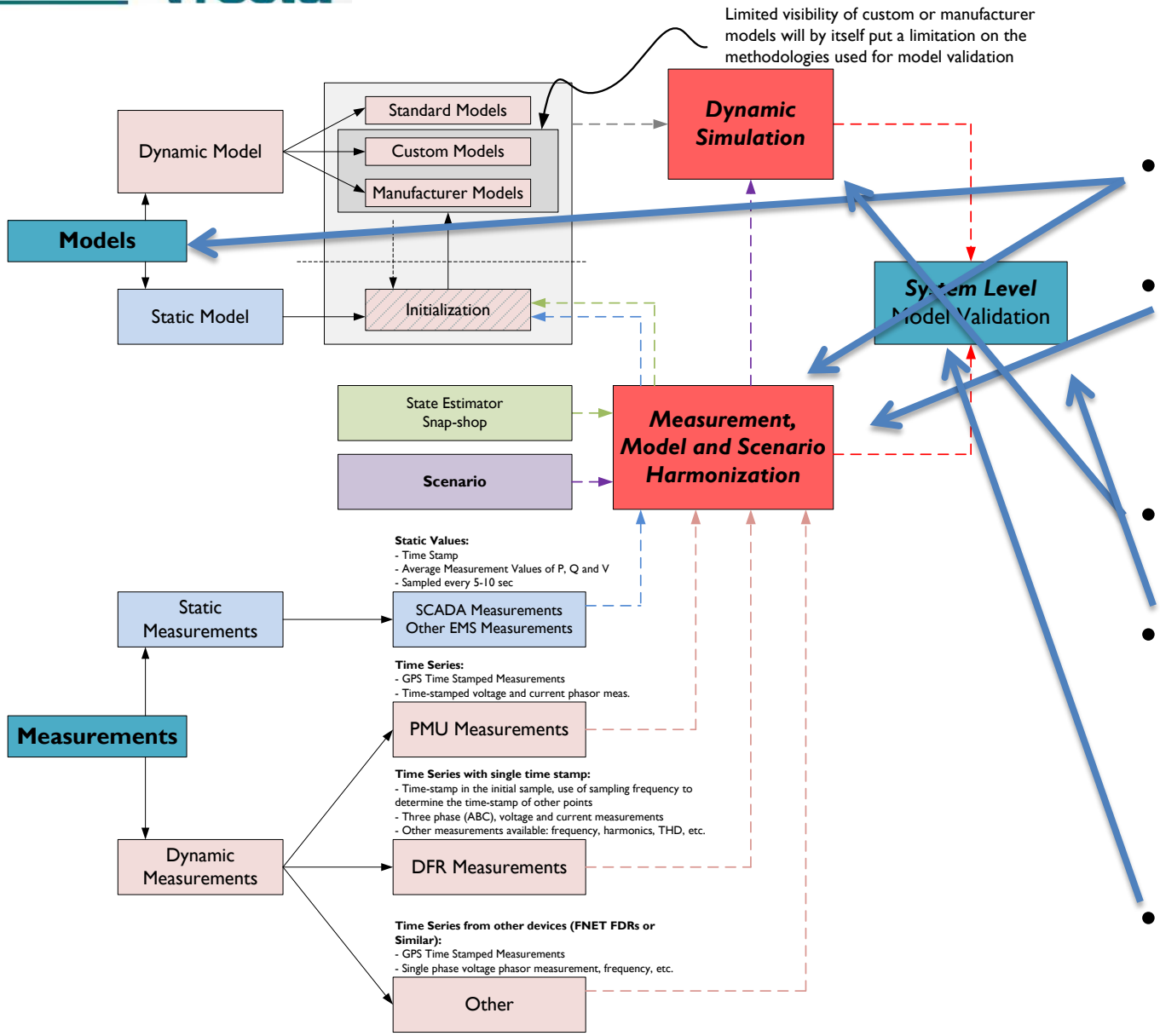
FMI and FMUs

- FMI stands for flexible 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
- FMU stands for flexible mock-up unit
  - An FMU is a model which has been compiled using the FMI standard definition
- For what?



  - Model Exchange
    - Generate C-Code of a model as an input/output block that *can* ***be utilized by other modeling and simulation environments***
  - Co-simulation
    - Couple two or more simulations in a co-simulation environment.
- **The FMI Standard is now supported by 35 different tools.**

Requirements of a SW architecture for model validation and calibration



- 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

(RaPId) **Ra**pid **P**arameter **Id**entification Toolbox

Software Architecture using Modelica and FMI Technologies

iTesla

iTesla WP2 Inputs to WP3: Measurements & Models

HARMONIZED MODELICA MODEL:
Modelica Dynamic Model Definition for Phasor Time Domain Simulation

MODELICA

SCADA/EMS Snapshots + Operator Actions

PMU and other available HB measurements

EMTP-RV and/or other HB model simulation traces and simulation configuration

iTesla Cloud or Local Toolbox Installation

.mo files

Internet or LAN

FMU

FMU compiled by another tool

.mat and .xml files

.mat and .xml files

Model Validation Software

User Target (server/pc)

Data Conditioning

MATLAB SIMULINK

MATLAB/Simulink
(used for simulation of the Modelica Model in FMU format)

FMI Toolbox for MATLAB
(with Modelica model)

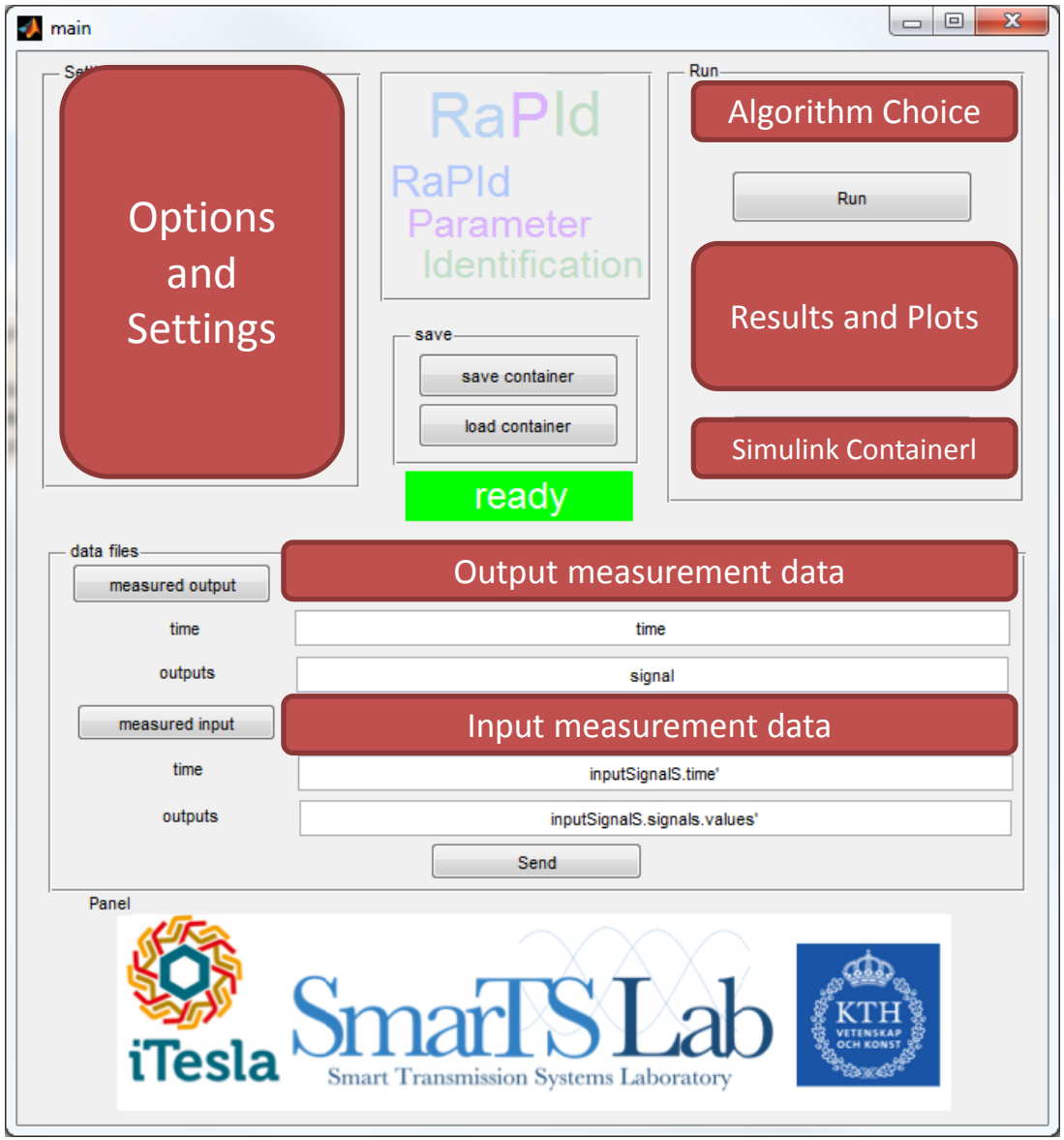
Model Validation Tasks:

Parameter tuning, model optimization, etc.

User Interaction

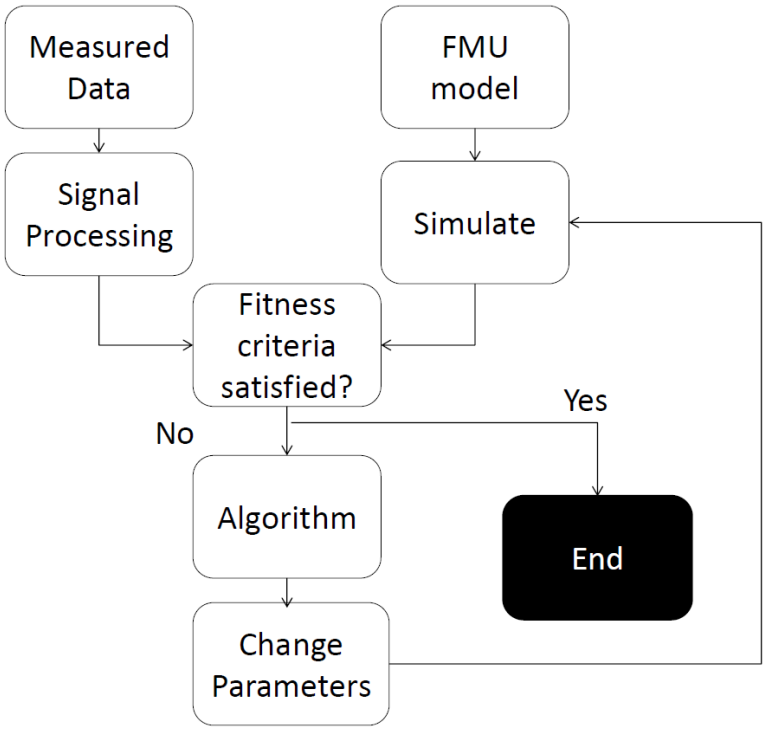
MATLAB

RaPId Interface



- RAPID has been developed in MATLAB, where the MATLAB code acts as wrapper to provide interaction with several other programs.
- Advanced users can simply use MATLAB scripts instead of the interface.
- **Plug-in Architecture:**
  - Completely extensible and open architecture allows advanced users to add:
    - Identification methods
    - Optimization methods
    - Specific objective functions
    - Solvers (integration routines)

Using RaPId



1. Collect the measurement data
2. Create the power system model in Modelica
3. Compile an FMU from the Modelica model
4. Create a Simulink model using the FMU block from the FMI Toolbox
5. Start the RaPId Toolbox
6. Input the settings required by RaPId

What is an FMU?

FMU, is a common standardized interface file format for model exchange between different software tools.

A Customized Calibration Methodology *using RaPId*

Example Two-Stage Procedure:

1. Run sophisticated algorithm (stochastic (Bayesian) Particle filter(PF) or meta heuristic Particle Swarm Optimization(PSO)) for a few iterations.

2. Start from the found close to optimum solution with a very simple (gradient descent-based Naive or simplex Nelder–Mead method) algorithm in order to find optimal value.

METHODOLOGY FOR PARAMETER IDENTIFICATION USING RAPID

Algorithm 1 Identification process using RaPId

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: $MSEmax$.

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 $MSE < MSEmax$ 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: $MSEmax$,
- maximum iterations: $IterMax$

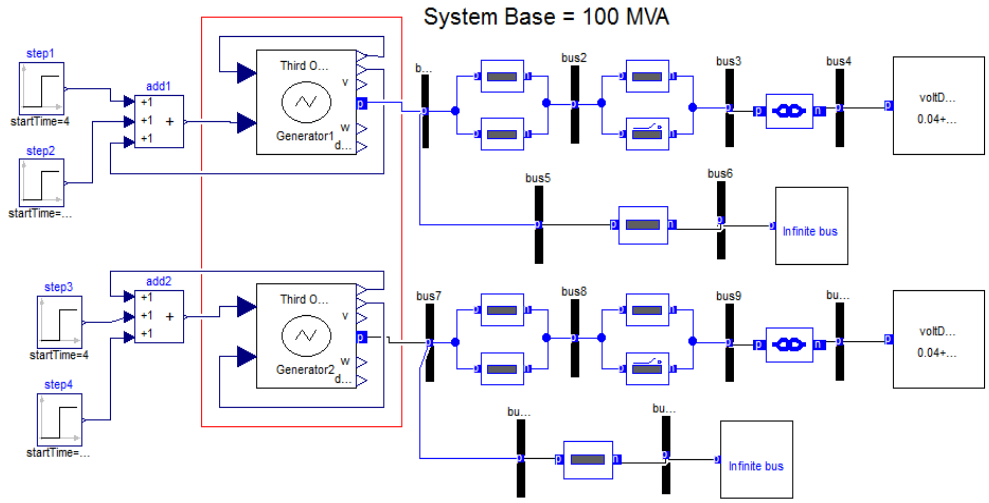
Iteration:
9. while $i<IterMax$ begin
10. call Naive(or NM) algorithm
11. call (Simulink system (with $*.fmu$))
12. calculate MSE w.r.t. $output\ variables$ and $outputs$
13. if $MSE < MSEmax$ then
14. evaluate ($estimated\ parameters$, MSE)
15. break; else continue; end

Output:
$estimated\ parameters$, MSE

Note: * see Table I;
** Chose the maximum number of particles to initialise PSO

Generator Parameter Identification
Case Study

Modelica Model with Two Simultaneous Experiments

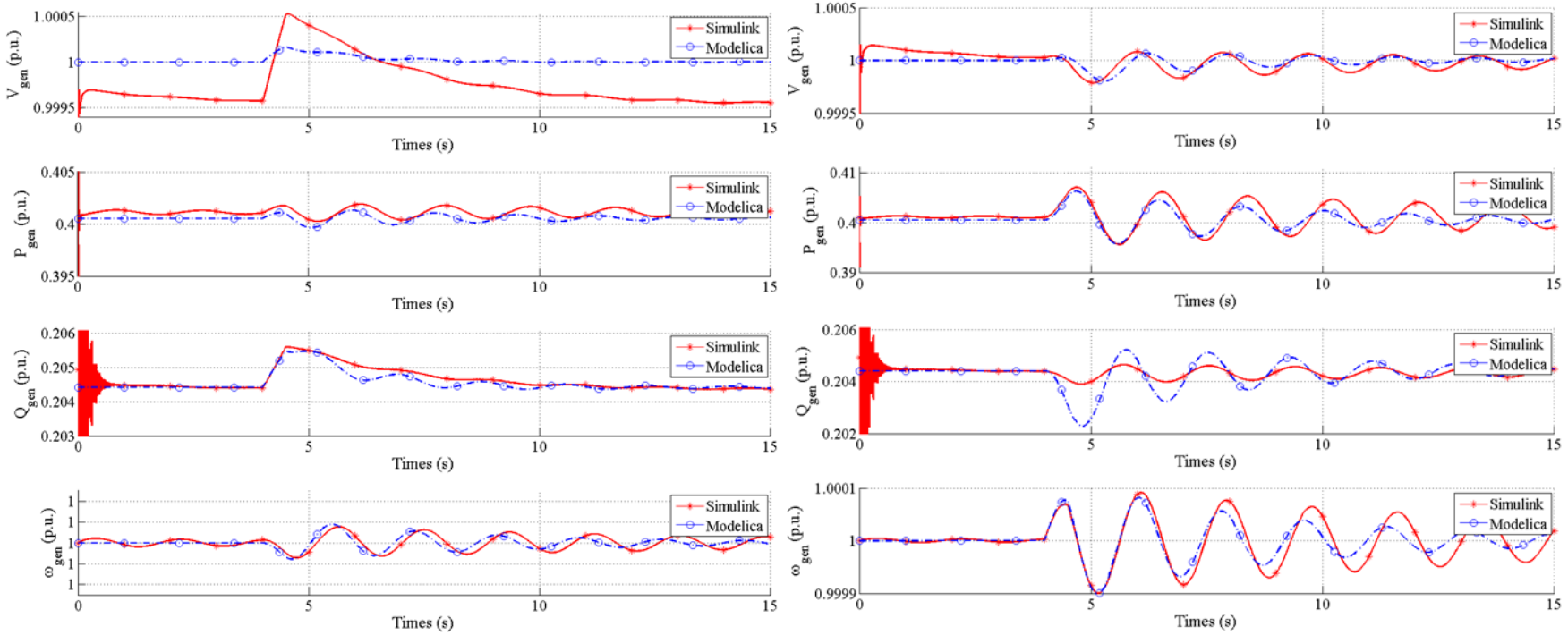


Estimated Parameters

Paremeter	Value
Armature resistance (Ra)	0.0010156
Direct axis reactance (Xd)	4.2924
Direct axis transient reactance (Xd')	1.37
Direct axis transient time const. (Td')	2.6156
Quadrature axis reactance (Xq)	5.3994
Inertia coefficient (M)	14.9005
Damping ratio (D)	0.0088415

- Aim:
  - To identify the **corresponding parameters of a low-order model generator** that preserve the main *electromechanical* dynamics of the full-order generator model.
- Methodology:
  - Two-step: PSO + NM
- Measurements:
  - Synthetic measurements from a EMT model simulation.
  - 4 outputs: voltage magnitude, rotor speed, active and reactive powers.

Comparing the Full-Order Model
with the Identified Reduced-Order Model



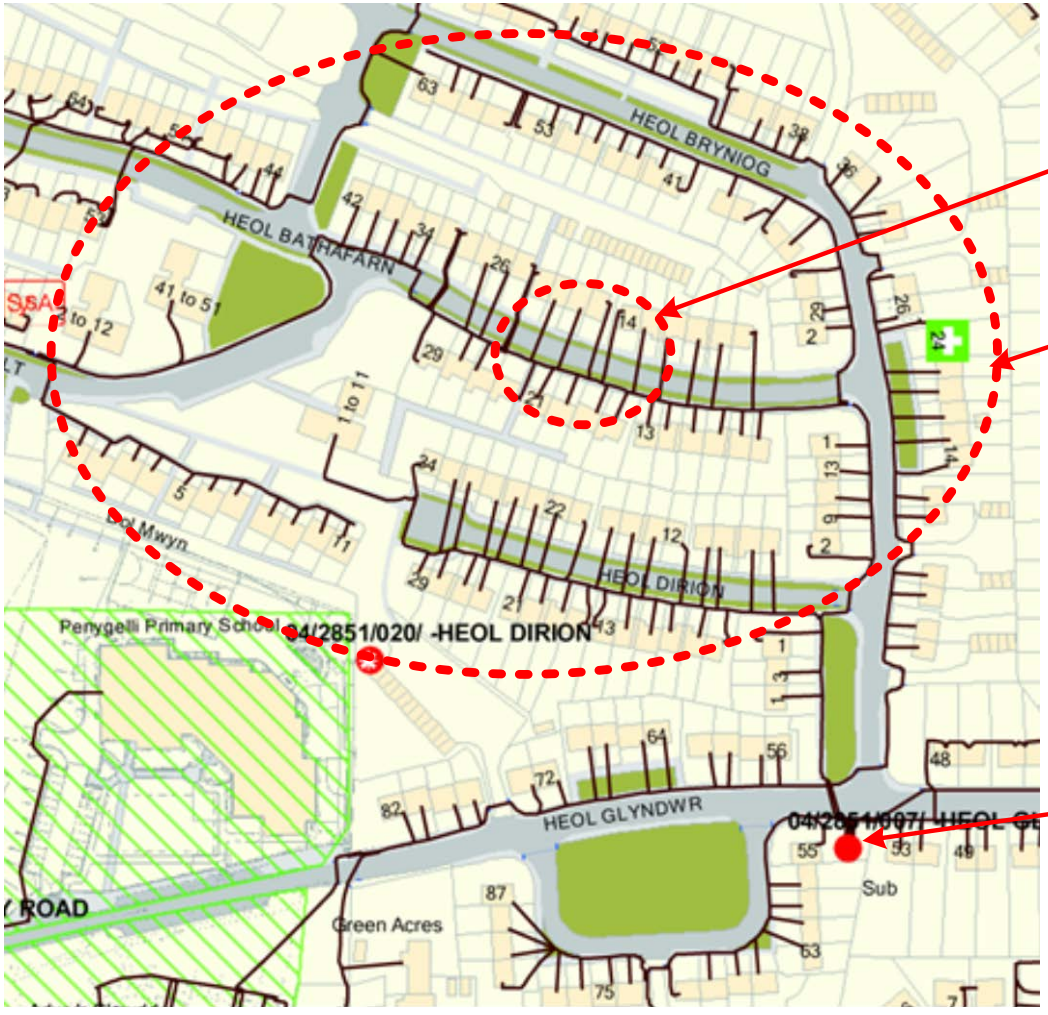
a) Field voltage perturbation

b) Torque perturbation

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

Aggregate Load Model Identification Case Study

Aggregate Load Model Identification of a Feeder in Scottish Power Distribution Grid



Aggregate Load Model Identification

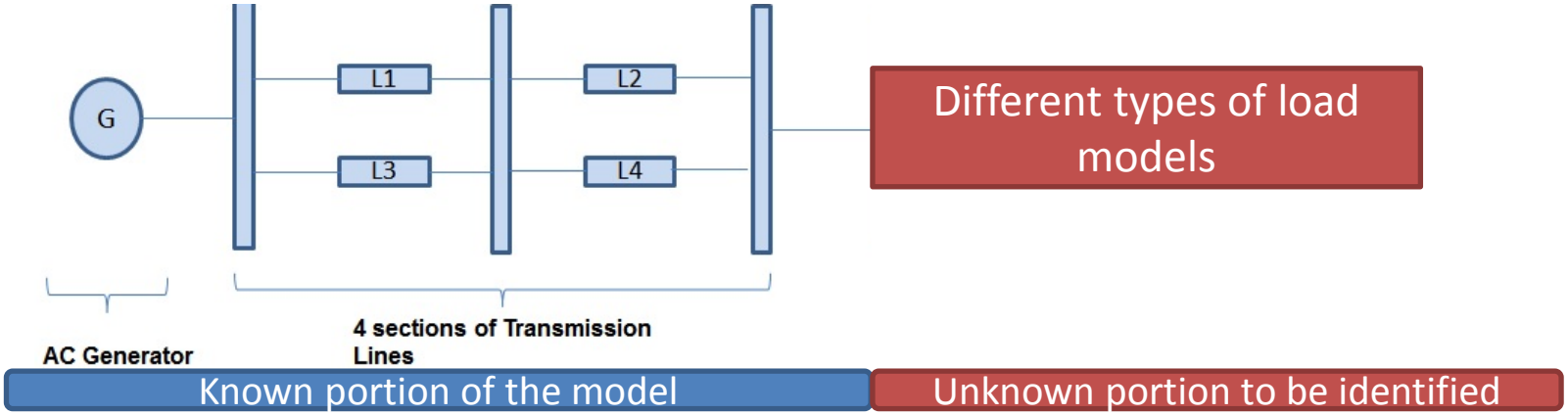
Aim and Methodology

Aim

- We would like to represent the feeder as a load with an aggregate model to reduce the modeling complexity in a large scale power system simulation.

Methodology:

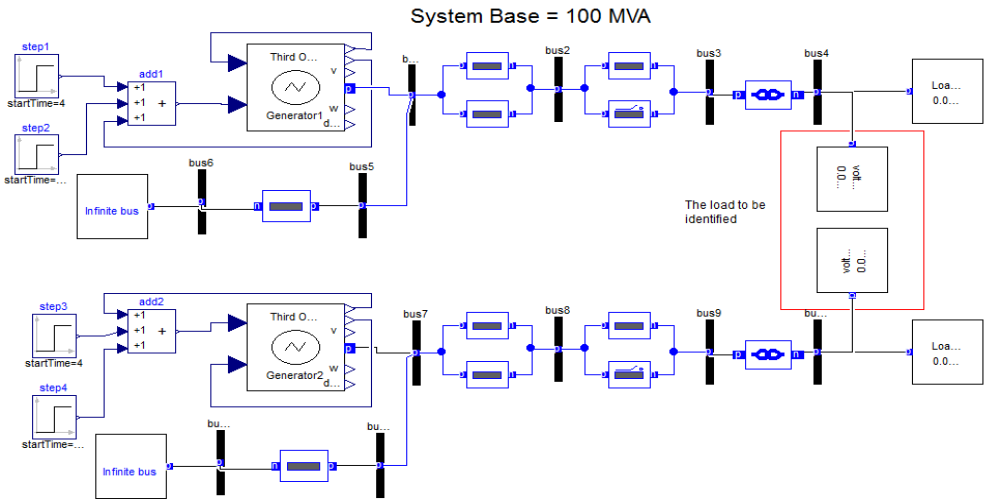
- Identification set-up:
  - Unknown Aggregate Load Model
  - All other components are known.
- The identification process is executed with different load models



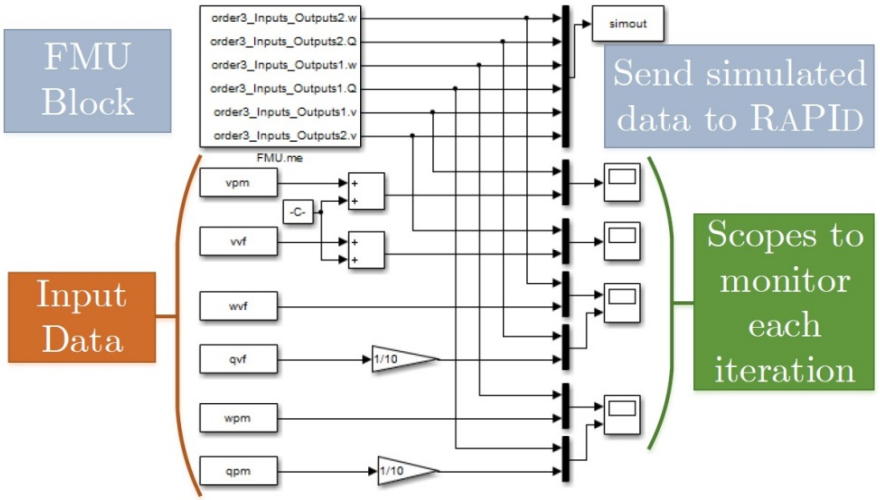
- The decision on which load type to use is based on a numerical criteria (Mean Squares Error / Best Fit)
- Experiments used for the identification process:
  - 1% torque perturbation at the generator (excites electromechanical dynamics)
  - 1% filed voltage perturbation at the generator (excites voltage dynamics)

Modelica Model and FMU-Simulink Model for RaPId

- The Modelica model is set up to perform the simulation of both experiments at the same time.



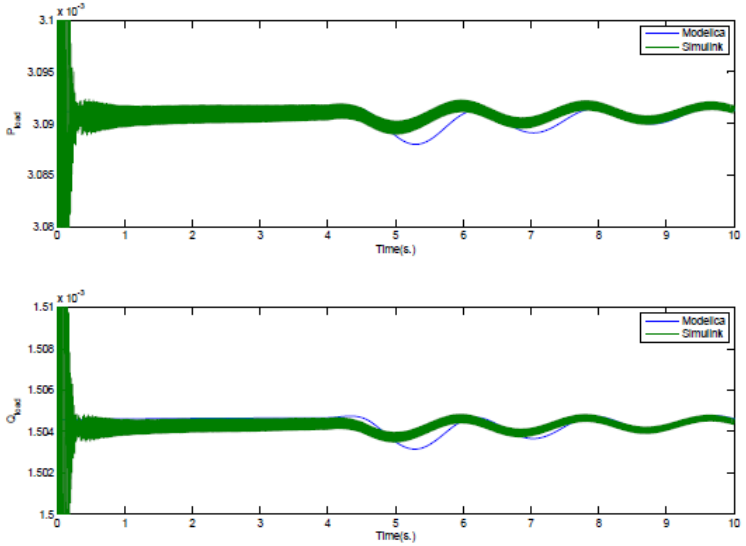
- The Modelica model is imported to a FMU Block in Simulink and the optimization process is carried out using RaPId



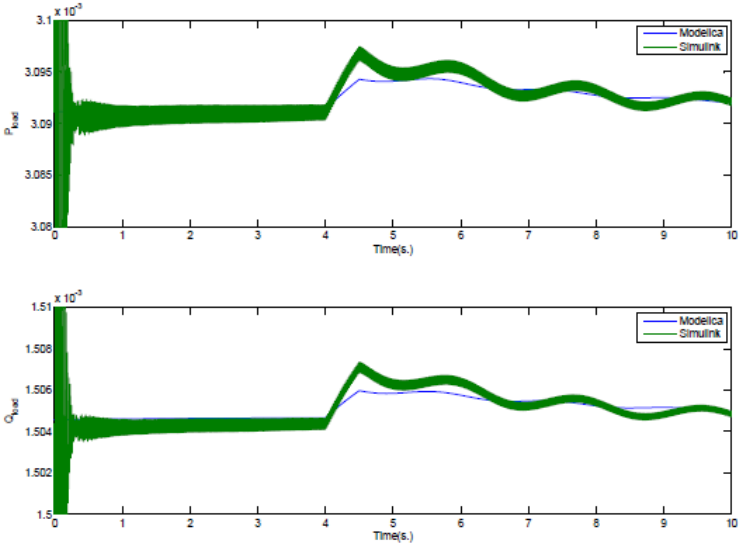
Exponential Recovery Load Model

Exponential recovery estimated aggregate load model parameters

Parameter	Value
Active power time constant (T_p)	1.3198
Reactive power time constant (T_q)	0.69108
Static active power exponent (α_s)	9.5074
Dynamic active power exponent (α_t)	2.3919
Static reactive power exponent (β_s)	8.5691
Dynamic reactive power exponent (β_t)	2.9145
Mean squared error	1.8627e-007



(a) Torque perturbation



(b) Field voltage perturbation

Exponential recovery aggregate load model results

Results Analysis

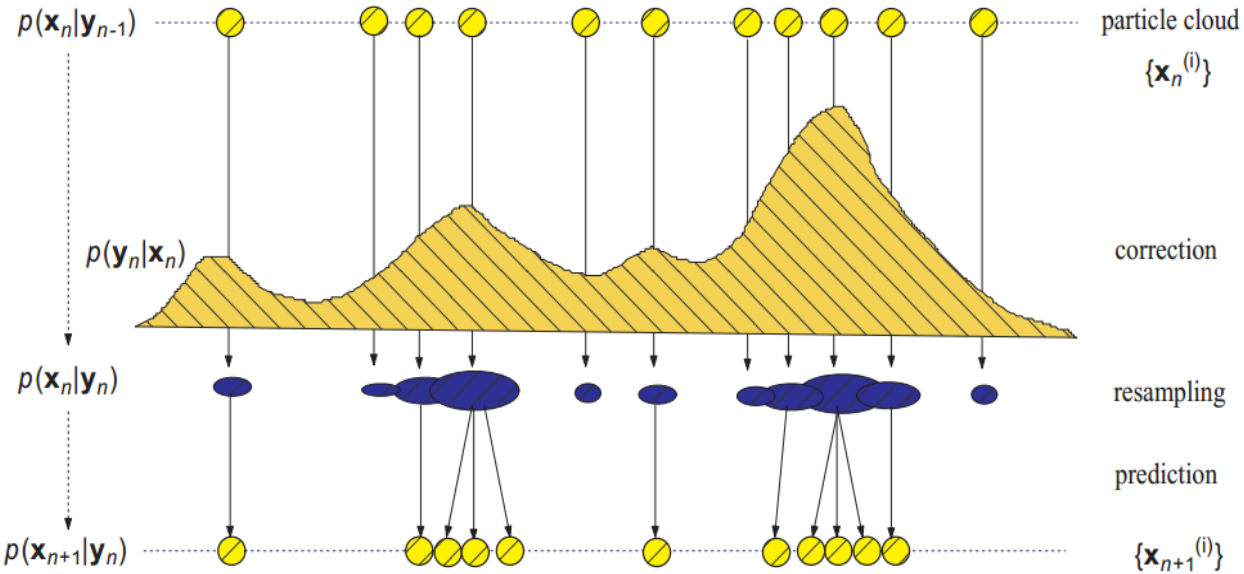
Load Aggregation Error Comparison

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 (MSE), the aggregate load model which matches the behavior of the data is the Exponential Recovery Load model.

Particle Filter

- Particle Filter - estimate the posterior density of the state-space by directly implementing the **Recursive Bayesian estimation** equations.



Recursive Bayesian estimation (a Bayes filter), is a general probabilistic approach for estimation unknown probability density function recursively over time using incoming measurements and a mathematical process model.

Result ⇨ Computation Time efficiency!

PSO vs PF: Results Analysis

To compare results from different combinations of algorithms, especially a newly implemented Particle Filter algorithm.

Use case:
Test for Voltage Dependent Load Model

Algorithm	Solution	Time, sec	Error (MSE)
PF and NM	[7.9815 7.2734]	16.207378	3.8042e-005
	[8.0380 7.2093]	28.264175	3.8042e-005
PSO and NM	[8.0 7.1667]	21.053648	3.8042e-005
	[8.0380 7.2093]	31.198479	3.8042e-005

✓ Particle Filter performed faster with the same precision as PSO
✓ Particle Filter returned more ”favorable” starting point for NM algorithm (according to time)

* All the results are very preliminary and have to be verified and averaged on a big number of runs.

Conclusions and

Looking Forward

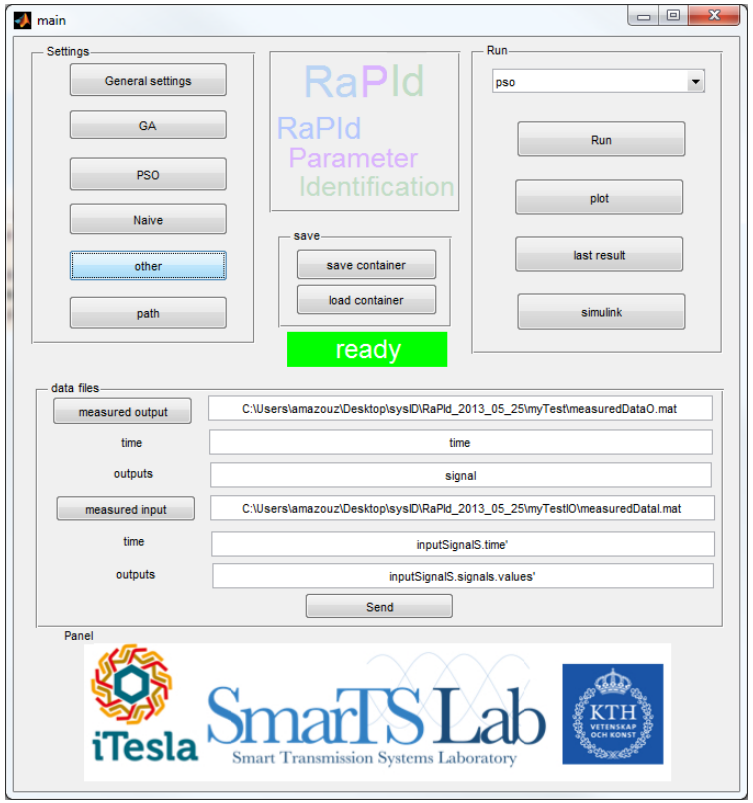
- Modeling power system components with Modelica (as compared with domain specific tools) is very attractive:
  - Formal mathematical description of the model (equations)
  - Allows model exchange between Modelica tools, with consistent (unambiguous) simulation results
- The FMI Standard allows to take advantage of Modelica models for:
  - Using Modelica models in different simulation environments
  - Coupling general purpose tools to the model/simulation (case of RaPId)
  - Model exchange with domain-specific tools
- PMU data for Model Validation:
  - With advent of PMU it is possible and convenient to create quite accurate models of power system and its components in conditions of uncertainty or lack of knowledge about the system.
- There are several challenges for modeling “large scale” power systems:
  - A well populated library of typical components (and for different time-scales)
  - Model builder from domain-specific tools “data files/base”
  - Support/linkage with industry specific data exchange paradigm (Common Information Model - CIM)

Thank you!
Questions?

tetianab@kth.se

luigiv@kth.se

luigi.vanfretti@statnett.no



SmarTS Lab
Smart Transmission Systems Laboratory