Unambiguous Power System Modeling and Simulation using Modelica Tools

Dr.-Ing. Luigi Vanfretti

E-mail: luigiv@kth.se
Web: http://www.vanfretti.com

Docent and Assistant Professor
Electric Power Systems Dept.
KTH Royal Institute of Technology
Stockholm, Sweden

Scientific Advisor
R&D Division (FoU)
Statnett SF
Oslo Norway

7th MODPROD Workshop on Model-Based Product Development
Linköping University
Feb. 5, 2013
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/

• Previous results from the FP7 Pegase project have been provided by RTE and used in this work.

• 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
Outline

• Background: *domain specific* state of the art
  – Power systems dynamics and domain tools for large-scale simulation
  – Modeling limitations, inconsistency and consequences
  – Model exchange in the IEC CIM era

• Unambiguous power system modeling and simulation
  – Building blocks for power system simulation: iTelsa PS Modelica Library
  – Consistent model sharing across different modeling tools
  – Adequate Modeling (... a case for controls and protections)
  – Large scale model simulation (... and initialization issues)
  – Using FMUs for model sharing and exploiting generic solvers

• Perspectives on the use of Modelica for
  – Unified CIM-compliant Multiple-Time Scale Power System Modeling
  – Smart Grids: cyber-physical modeling of power systems
Large Scale Power Systems
To operate large power networks such as these planners and operators need to analyze variety of operating conditions – both off-line and in near real-time (power system security assessment). Different SW systems have been designed for this purpose.

But, the dimension and complexity of the problems are increasing due to growth in electricity demand, lack of investments in transmission, and penetration of intermittent resources. **New tools are needed!**

Current and new tools will need to perform simulations:
- Of complex hybrid model components and networks with very large number of continuous and discrete states.
- Models need to be shared, and **simulation results need to be consistent** across different tools and simulation platforms.
Power system dynamics

- Lightning
- Line switching
- SubSynchronous Resonances, transformer energizations...
- Transient stability
- Electromagnetic Transients
- Long term dynamics
- Daily load following

Time scales:
- Electromagnetic Transients: $10^{-7}$ seconds
- Lightning: $10^{-5}$ seconds
- Line switching: $10^{-4}$ seconds
- SubSynchronous Resonances: $10^{-3}$ seconds
- Transient stability: $10^{-2}$ seconds
- Long term dynamics: 1 second
- Daily load following: $10^2$ seconds
Power system *dynamics*

- **Lightning**
- **Line switching**
- **SubSynchronous Resonances, transformer energizations**
- **Electromechanical Transients**

**Electromagnetic Transients**

Interaction between the electrical field of capacitance and magnetic field of inductances in power systems.

Ex: lightning impact, line switching

May produce: overvoltages, overcurrents, abnormal waveforms, electromechanical transients
Power system dynamics

Electromechanical Transients

Interaction between the electrical energy stored in the system and the mechanical energy stored in the inertia of rotating machines

Ex: Power oscillations

May produce: system breakup.
Power system dynamics challenges for simulation

- Lightning
- Line switching
- SubSynchronous Resonances, transformer energizations...
- Transient stability
- Long term dynamics
- Daily load following

The presence of very small time scales and large amount of discrete switches.

Difficult to simulate very large networks.

This is usually dealt with by discretizing the model and solving it using **discrete solvers**.
Power system dynamics challenges for simulation

Generally there are no discrete events. (Ad-hoc DAE solvers)

Models are simplified (averaged) to allow for simulation of very large networks. Ad-hoc solvers have been developed to reduce simulation time, but usually the “model” is “interlaced” with the solver.

- Lightning
- Line switching
- SubSynchronous Resonances, transformer energizations...
- Transient stability
- Long term dynamics
- Daily load following

Models are simplified (averaged) to allow for simulation of very large networks. Ad-hoc solvers have been developed to reduce simulation time, but usually the “model” is “interlaced” with the solver.

Generally there are no discrete events. (Ad-hoc DAE solvers)
Power system dynamics challenges for simulation

- Lightning
- Line switching
- SubSynchronous Resonances
- Transformer energization
- Transient stability
- Long term dynamics
- Daily load following

The models are simplified further by neglecting most dynamics (replacing most differential equations by algebraic equations).

(Ad-hoc DAE solvers)
Power system phenomena and domain specific simulation tools

Ad-hoc
Initialization of Dynamic States

Steady State (Power Flow)

Lightning

Line switching

SubSynchronous Resonances, transformer energizations...

Transient stability

Long term dynamics

Daily load following

Simulation

10^{-7} 10^{-6} 10^{-5} 10^{-4} 10^{-3} 10^{-2} 10^{-1} 1 10 10^{2} 10^{3} 10^{4} seconds

Broad range of time constants results in specific domain tools for simulation.

Non-exhaustive list. There exists other proprietary and few OSS. Only two proprietary solutions for real-time simulation.
Power System Time-Scale Modeling

from this point on

Lightning

Line switching

SubSynchronous Resonances, transformer energizations...

Transient stability

Long term dynamics

Phasor Time-Domain Simulation

Daily load following

10^{-7} 10^{-6} 10^{-5} 10^{-4} 10^{-3} 10^{-2} 10^{-1} 10 10^{2} 10^{3} 10^{4}
seconds
February 19th 2011

Synchornized Phasor Measurement Data

20110219_0755-0825

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 (casual)

• Model sharing:
  – Parameters of block definitions are shared in a specific “data format”
  – For large systems, this requires “filters” for translation into the internal data format of each program

• Modeling inconsistency:
  – For (standardized casual) models there is no guarantee that the model definition is implemented “exactly” in the same way in different SW
  – User defined models and proprietary models can’t be represented without complete re-implementation in each platform

• Modeling limitations:
  – Most SWs make no difference between “model” and “solver”, and in many cases the model is somehow implanted within the solver

• 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.
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 (casual)

• Model sharing:
  – Parameters of block definitions are shared in a specific “data format”
  – For large systems, this requires “filters” for translation into the internal data format of each program

• Modeling inconsistency:
  – For (standardized casual) models there is no guarantee that the model definition is implemented “exactly” in the same way in different SW
  – User defined models and proprietary models can’t be represented without complete re-implementation in each platform

• Modeling limitations:
  – Most SWs make no difference between “model” and “solver”, and in many cases the model is somehow implanted within the solver

• 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.
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.

• Model sharing:
  – Parameters of block definitions are shared in a specific “data format”.
  – For large systems, this requires “filters” for translation into the internal data format of each program.

• Modeling inconsistency:
  – For (standardized casual) models, there is no guarantee that the model definition is implemented “exactly” in the same way in different SW.
  – User defined models and proprietary models can’t be represented without complete re-implementation in each platform.

• Modeling limitations:
  – Most SWs make no difference between “model” and “solver”, and somehow implanted within the solver.

• 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.

The Modelica language and Modelica Tools can help in avoiding all of these issues!

This is very costly!
Power System Modeling

limitations, inconsistency and consequences

• Modeling adequacy:
  – In general, the exclusive use of DAE solvers forces the representation of discrete devices using averaging or with other simplifications/tricks
  – This is an issue with complex controls and protections that require the handling of discrete events: this is not an issue for Modelica models and tools!
IEC CIM

• A set of standards in enable system integration and information exchange based on a common information model

• The CIM standards are based on a Unified Modeling Language (UML) based information model representing real-world objects and information entities exchanged within the value chain of the electric power industry
  – Provides common semantics for all information exchanges

• However:
  – Modeling of dynamics is still in progress (the CIM effort was started in 1996 in IEC) and there are many issues to resolve.
  – The modeling approach selected will use “IEEE Standard” models which are causal block-oriented model definitions
  – An equation-based definition of model dynamics could be more beneficial for model consistency: Modelica is a tangible choice
Modeling and simulation should not be ambiguous: it should be consistent across different simulation platforms.

For unambiguous modeling, model sharing and simulation, Modelica and Modelica Tools can be used.

We illustrate how we use Modelica and Modelica Tools towards this goal:

- Building blocks for power system simulation: iTelsa PS Modelica Library
- Consistent model sharing across different modeling tools
- Adequate Modeling (... a case for controls and protections)
- Large scale model simulation (... and initialization issues)
- Using FMUs for model sharing and exploiting generic solvers
iTesla Power Systems  
Modelica Library

• Power Systems Library:
  – A Power Systems library developed during the Pegase project in Scilab/Xcos was converted to Dymola (manually).
  – The library has also been tested (and modified) for OpenModelica and SystemModeler.
  – Components and systems are validated against Eurostag’s results.

• New components and time events are being added to this library in order to simulate new systems.
  – Efforts will be put in replicating all of Eurostag’s and PSAT models in the library, and adding new models of RES and other power electronic controlled devices
  – Automatic translator from domain specific tools to Modelica will use this library’s classes.

• New regulator macroblocks and medium size power systems have been built and simulated in Dymola, OpenModelica and SystemModeler.
class PwLine

parameter Real R "Resistance";
parameter Real X "Reactance";
parameter Real G "Shunt conductance";
parameter Real B "Shunt susceptance";

equation
R*(p.ir-G*n.vr+B*n.vi)-X*(n.ir-B*n.vr-G*n.vi)=(n.vr-p.vr);
R*(n.ir-B*n.vr-G*n.vi)+X*(n.ir-G*n.vr+B*n.vi)=(n.vi-p.vi);
R*(p.ir-G*p.vr+B*p.vi)-X*(p.ir-B*p.vr-G*p.vi)=(p.vr-n.vr);
R*(p.ir-B*p.vr-G*p.vi)+X*(p.ir-G*p.vr+B*p.vi)=(p.vi-n.vi);
end PwLine;

classname PwTransformer

PwPin p:
PwPin n;
parameter Real R "Resistance";
parameter Real X "Reactance";
parameter Real G "Shunt conductance";
parameter Real B "Shunt susceptance";
parameter Real r "Transformation ratio";

equation
r*(G*n.vr-B*n.vi-n.ir)=p.ir;
r*(G*n.vi+B*n.vr-n.ir)=p.ii;
R*p.ir-X*p.ii=r*r*p.vr-r*n.vr;
R*p.ii+X*p.ir=r*r*p.vi-r*n.vi;
end PwTransformer;
New components in iTesla

- Here are some of the new components and time events built in Modelica for the iTesla Project:
  - Capacitor Bank with Modification.
  - Line opening, Load modification

```modelica
class PwCapacitorBank
  PwPin p
  parameter Real nsteps "number of steps";
  parameter Real Go "active losses (p.u.) in each element";
  parameter Real Bo "reactive power (p.u.) in each element";
  parameter Real t1 "time for Bank Modification";
  parameter Real nmod "number of step to swigh on/off (+/-)"
  Real G;
  Real B;
  Real nt;

  equation
  if (time > t1) then
    nt = nsteps + nmod;
  else
    nt = nsteps;
  end if;
  G=nt*Go;
  B=nt*Bo;
  p.vr = (p.ir*G + p.ii*B)/(G*G + B*B);
  p.vi = (p.ir*B + p.ii*G)/(G*G + B*B);
end PwCapacitorBank;
```

```modelica
class PwLoadwithVariation
  PowerSystems.PwPin p
  parameter Real Vo_real "Initial voltage at node in p.u.";
  parameter Real Vo_img "Initial voltage at node in p.u.";
  parameter Real Po "Initial Active Power in p.u.";
  parameter Real Qo "Initial Reactive Power in p.u.";
  parameter Real t1 "Time of Load variation";
  parameter Real P2 "Active load variation";
  parameter Real Q2 "Reactive load variation"
  Real Vo;
  Real P;
  Real Q;
  Real R;
  Real X;
  Real a;

  equation
  Vo = sqrt(Vo_real*Vo_real + Vo_img*Vo_img);
  if (time > t1) then
    P = Po + P2;
    Q = Qo + Q2;
  else
    P = Po;
    Q = Qo;
  end if;
  a = P/Q;
  R = (abs(Vo)*abs(Vo)/P)*(a*a/(1+a*a));
  X = R/a;
  p.vr=R*p.ir-X*p.ii;
  p.vi=X*p.ir+R*p.ii;
end PwLoadwithVariation;
```
Example 1 [from Pegase] in Dymola
Example 1

OpenModelica
Example 1

System Modeler
• Modeling adequacy: “Example 2”
  – Complex controls and protections may require the handling of discrete events
  – This case illustrates the special protection used in a generator’s excitation current
  – Two discrete time delays are modeled explicitly and sequentially
This figure shows a larger-scale power system network extending "Example2" with 20 more generators. Total of 22 generators + controls approx. 440 continuous states + approx. 100 discrete states.
(Larger) Model Simulation

• This figure shows a larger-scale power system network extending “Example2” with 20 more generators.
  – Total of 22 generators + controls approx. 440 continuous states + approx. 100 discrete states

• **Initialization:**

• In the previous the initialization of dynamic states has been extracted from Eurostag.

• Default Dymola initialization is used here. Not possible to initialize/simulate using Scilab/Xcos.

• *Not clear how to deal with initialization for power system models.*
FMUs for Model Sharing and simulation with generic solvers

• Why?:
  – The iTesla project has adopted Modelica as its toolbox internal language for dynamic model description
  – There is a WP on model validation.
• Some TSOs are not allowed to provide explicit information of models or their description (e.g. wind turbines)

• FMUs could help – two options:
  – FMUs of specific devices are generated, and then incorporated into the overall model.
  – FMUs of a complete model can be generated and shared without revealing the model’s internal structure/equations

• We explore the second option.
Sharing FMUs with MATLAB/Simulink and the FMI Toolbox

- The model validation architecture needs to exploit the sys id. tools in Matlab.
- FMUs generated from Dymola.
- FMUs allow for model simulation in Matlab for self-contained integration of prototype model validation tool.
FMUs for simulation with generic solvers and in generic purpose environments

- The use of generic solvers is attractive for performing large numbers of simulations of large-scale power system models.
- It is also attractive not to depend on Matlab for the model validation tools.
- This example shows the use of Assimulo solvers within the JModelica.org framework.
- The JModelica.org framework for dynamic optimization is attractive for this kind of application (model validation and correction).
FMUs for simulation with generic solvers and in generic purpose environments

The use of generic solvers is attractive for performing large numbers of simulations of large-scale power system models. It is also attractive not to depend on Matlab for the model validation tools. This example shows the use of Assimulo solvers within the JModelica.org framework. The JModelica.org framework for dynamic optimization is attractive for this kind of application (model validation and correction).
Results & Lessons Learned

No.1:
• A Modelica library has been developed and tested in different Modelica tools.
• Modelica allows for unambiguous model sharing across different simulation software
  – This is natural thanks to the Modelica language

No.2:
• Modeling of complex power system controls and protections can take into account discrete events
  – Flexibility of modeling language and solvers to handle discrete events.

No.3:
• It is possible to simulate large models (although not very large) of power systems.
  – Proper initialization will allow to determine the suitability for real-life networks
  – Automatic conversion from domain specific tool will be required (too time consuming to model from scratch)

No.4:
• FMUs allow to use general purpose tools for specific power system applications (model validation)
• FMUs allow sharing of models without revealing the internal model definition/structure (useful for manufacturer specific models).
Opportunities for Modelica in Power Systems

• Main issue: guarantee security of supply when facing
  – increased amount of “variable” RES
  – limited transmission capacity coupled to growth and dependency on demand
  – cyber-security concerns/threats

• These and many other challenges will require:
  – New/different set of tools [Feynman]
  – New/different ways of thinking and dealing with new/old problems [teach an old dog new tricks]

• Two key areas where Modelica and Modelica Tools can help [immediately]:
  – Unified CIM-compliant unified modeling of power grids (in multiple time scales)
  – Smart Grids: cyber-physical modeling of power systems and computer systems
CIM is the (future) Grid Model

Business Canonical Models

CIM Grid Canonical Model

- Control centers.
- Grid development planning.
- Asset management.
- Call centers.
- Markets.

CIM represents holistic grid information domain, supporting the integration of major business systems that operate, analyze and plan the grid.

Substations
- Generating Plants
- Buildings
- Electric Vehicles

Windfarms
- Homes
- Industrial
- Etc.

Field Environment Canonical Models
Perspectives for Unified Grid Modeling

Multiple Time-Scale Unified Power System Modeling

Semantic Power System Modeling
Tool supporting CIM and ModelicaML

CIM
Grid's UML Semantic Model

ModelicaML
UML Profile for Modelica

IEC

Meta Models of components defined in MetaModelica
Queries for model instances from meta models
Model instance selected

Modelica (+FMU) Model Binder

Proprietary Component Models

Functional Mock-Up Interface

Model Instance Call

Model Instance Bind

Code Generation/Integration

To simulation engines

Steady State Analysis and Simulation

Steady State Simulation Engines
Power Flow Routines
Short Circuit Routines, other

Initialization of Dynamic Models
Specific Routines
Built-in Routines

Dynamic Simulation Engines

Generic Control Design and Optimization Tools

Modelica (.mo) and FMUs with Model Definition

MATLAB/Simulink
Mathematica/SystemModeler

OpenModelica

JMODELICA.ORG
Perspectives on Modelica for Smart Grids

- Meta Models of components defined in MetaModelica
- Modelica model definition of the Electrical Power System Model (+ICT components/networks)
- Model Graphical Editor OMEdit
- Queries for model instances from meta models
- Linkage of components
- To simulation engines
- Initialization of Power System Dynamic Models
- Steady State Solvers
  - Power Flow Routines
  - Short Circuit Computations
- Dynamic Simulation Engines
  - OMPython, Pysimulator, Mathematica, MATLAB
- Functional Mock-Up Interface
- Generic Tools
  - MATLAB/Simulink
  - Mathematica/SystemModeler
  - Other
- Applications and Specialized Tools
  - (e.g., Analysis Tools, Model Validation Tools)
- Control Design and Optimization Tools
  - MATLAB/Simulink
  - Mathematica/SystemModeler
  - Other
Thank you!

Questions?

Looking for people to join these efforts!

- PhD Position on “CIM-Compliant Unified Modeling, Simulation and Estimation of Smart Transmission Grids”
- Email me: luigiv@kth.se