

Modelica Implementation and Software-to-Software Validation of Power System Component Models Commonly used by Nordic TSOs for Dynamic Simulations

Mengjia Zhang¹ Maxime Baudette¹ Jan Lavenius¹ Stig Løvlund² Luigi Vanfretti^{1,2}

¹School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden {mengjia, baudette, janlav, luigiv}@kth.se

²Research and Development Division, Statnett SF, Oslo, Norway luigi.vanfretti@statnett.no

Abstract

Following the European Network of Transmission System Operators for Electricity (ENTSO-E) R&D Road Map, the *iTesla*¹ project aims to develop a common toolbox to support the future operation of the pan-European power grid. This toolbox was developed to allow unambiguous power system dynamic model exchange between simulation tools and performing simulations using Modelica models for Pan-European dynamic security assessment.

The work presented here comprises the development of Modelica classes for power grid components used by Nordic TSOs to model the Nordic synchronous grid. The performance of these Modelica models has been validated through different test cases implemented in both Power System Simulator for Engineering (PSS/E), as the reference domain-specific tool, and a Modelica simulation environment. The results from dynamic simulations with the presence of different perturbations have been compared to qualitatively and quantitatively the PSS/E results to validate the Modelica implementation, obtaining consistency simulation results between both tools. This paper describes the methodology used to develop and to perform model-to-model validation of the newly implemented Modelica models.

Keywords: Modelica, Simulation, Software validation, Power systems

1 Introduction

Ever since the 1970s (Schulz, 1975), a wide range of power system models has been developed to be used for different simulation analysis purposes. However, these models are tightly interlaced with the integration routine of particular software platforms. Therefore, the analysis of these models becomes solver- and implementation-

dependent, which results in difficulties for unambiguous model sharing, simulation, portability, maintenance and verification (Vanfretti et al., 2013).

This fact was an important design issue considered in the *iTesla* project. The *iTesla* toolbox was meant to offer multiple features for dynamic security assessment, based on extensive dynamic simulations of the power grid within different modules. To enable vendor-independent simulation and model consistency, the Modelica language was chosen for power system modeling and simulation in *iTesla*.

Statnett, the Norwegian TSO participated in *iTesla*. Statnett uses the Common Information Model (CIM) (IEC 61970-301, 2013) to specify the topology and parameters of their grid models and PSS/E (Siemens PTI Inc., 2005) for dynamic analysis. Statnett contributed to the *iTesla* by providing grid models and scenarios to test and validate the *iTesla* toolbox.

PSS/E is an integrated set of programs broadly used for power system transmission network steady-state and dynamic studies. Statnett has used it for several decades, gaining significant experience with grid models developed with PSS/E component models, making PSS/E a the software of choice for dynamic simulations. However, PSS/E internal models are integrated with its trapezoidal solver, in other words, the models can not be simulated by using different solvers or in other simulators. In addition, because the dynamic model has been developed over the course of many years, and because of its lack of portability, it is difficult to perform unambiguous model exchange and simulation using other tools. Thus, from the point of view of long-term model maintenance and portability, it is highly desirable to have the PSS/E models separated from the solver and implemented using a standardized modeling language to allow for model exchange between tools.

To guarantee dynamic model consistency against PSS/E and reassure Statnett's analysts that models implemented in Modelica would perform equally as in PSS/E (or better), a software-to-software validation approach was chosen for validation. To achieve this, different grid compo-

¹Innovative Tools for Electrical System Security within Large Areas; *iTesla* is a collaborative R&D project co-funded by the European commission 7th Framework Program. www.itesla-project.eu

ment models from PSS/E specified by Statnett were implemented in Modelica and validated against simulated response of the reference models obtained from PSS/E.

Modelica is a standardized non-proprietary, object-oriented, equation-based language (Modelica Association, 2014a), offering a great versatility to model complex physical systems. Different from modeling methods used in widely spread power system analysis tools, instead of using assignment declaration, the Modelica language provides a framework where models can be developed by explicitly declaring the equations containing all the models' dynamic. This facilitates the development of new models, maintaining existing ones, or even checking existing implementations.

Because Modelica is a standardized language, there is a broad range of modeling and simulation tools that are Modelica compliant, such as: OpenModelica, JModelica, Dymola, MapleSim, SimulationX, Mathematica, SystemModeler, etc. This facilitates unambiguous model and information exchange between tools offering different features. In addition, the Functional Mock-up Interface (FMI) standard (Modelica Association, 2014b) can be used to simulate exported Modelica code in other software which are widely used in today's industry such as MATLAB². The paper is organized as follows: Section 2 discussed the method for components model implementation by covering modeling procedures and showing two model examples. Section 3 introduces the validation process. Section 4 presents the results obtained from two sample validation tests. The last section gives general conclusions.

2 Model Development and Implementation

Seventeen of the component models are reported in this work, however, note that additional models are being implemented. The models include synchronous machines and their regulators, transformers, and composite load models. They are listed in Table 1.

The challenge of the implementation was to derive the dynamic equations for each model from the PSS/E documentation. Because the actual model equations are confined in PSS/E's source code and are not accessible to end-users, it is necessary to fully study the documentation of the software in order to understand the models and collect the correct equations to represent their dynamics. After collecting or deriving the equations (presumably used in PSS/E), there are mainly three steps towards modeling each components.

²The FMI Toolbox for MATLAB links FMI compliant models to the MATLAB/SIMULINK environment <http://www.modelon.com/products/fmi-toolbox-for-matlab/>.

Table 1. Developed Components

<i>Model type</i>	<i>Name / description</i>
Generator	GENROU, GENSAL, GENCLS
Transformers	2-winding & 3-winding with phase-shift and OLTC
Load model	ZIP load includes PSS/E's specified characteristics
FACTS devices	SVC (Static var compensator)
Exciter	IEEET1, IEEET2, SCR, EXST1, SEX, URST5T, ESST4B, ESAC2A
Governor	HYGOV, GAST, IEESGO, GGOV1
Stabilizer	IEEEST, STAB2A

2.1 Modeling steps

- 1. Define the input/output interface of the model.** There were two types of connector used, **Pwpin** and other single variable connectors (Bogodorova et al., 2013). **Pwpin** is a special type of connector which contains four real variables to express voltage and current phasors. Electrical components were connected by **Pwpin**.
- 2. Define model attributes.** Model attributes contain parameters, variables and equations. Other than the standard parameters given by model specifications, some auxiliary parameters should also be declared for initialization proposes.
- 3. Provide initial values for variables.** A power system model is commonly defined by equations (1-3)

$$\dot{\bar{x}} = f(\bar{x}_d, \bar{p}, \bar{u}) \tag{1}$$

$$0 = g(\bar{x}_{da}, \bar{p}, \bar{u}) \tag{2}$$

.....

$$0 = h(\bar{x}_{na}, \bar{p}, \bar{u}) \tag{3}$$

where, \bar{x}_d and \bar{x}_{da} are the vector of dynamic and algebraic state variable for dynamic components, respectively, while \bar{x}_{na} is the vector of algebraic variables of the network, e.g. $|\bar{V}|$ and $\angle \bar{V}$. \bar{p} is the vector of parameters of the model and \bar{u} defines a vector of inputs, such as perturbations to which the model can be subjected, e.g. ΔV_{ref} and ΔP_m .

Power system domain specific tools find \bar{x}_{na} by solving (3). The values obtained are used to solve for \bar{x}_d and \bar{x}_{da} , by setting (1) to zero. On the other hand, Modelica compilers attempt to solve for x by setting (1) to zero and simultaneously solving the resulting non-linear algebraic equations (1-3).

The difference in the solution approach needs to be carefully considered when implementing a typical power system model. Observe that due to the complexity of the power system model, it is common

that zero start values will cause singularity or convergence problems during the initialization process. Thus it is necessary to calculate the initial values explicitly. The calculations of these initial values are case dependent (i.e they are different for each model component).

These steps are explained in detail using two model examples in the following sections. They are a round rotor generator model and a three-winding transformer model.

2.2 Model example: Round rotor generator model (GENROU)

Every synchronous generator contains a rotor and a stator. The rotor of the GENROU model, shown on Fig. 1, is represented using two equivalent circuits per rotor axis and the armature fluxes are modeled as a function of the field voltage and the armature current (Schulz, 1975). Electric-magnetic saturation effects are represented by d- and q-axis mutual conductances, as a function of ψ'' . The block diagram of the rotor circuit is available in the PSS/E manual and in (Schulz, 1975).

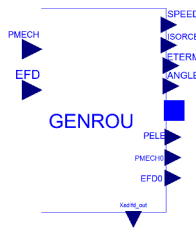


Figure 1. Genrou icon view

There is a need to use a change of coordinates because the dynamics of each synchronous machine is expressed in the rotor reference frame, thus to interface with the network, Park's transformation should be applied. The corresponding modelica implementation and complete equation set of the model is attached in Appendix, please refer to Schulz (1975) and Kundur et al. (1994) for the meaning of the symbols. Additional calculations should be performed in order to obtain the initial value of each state. Steady state operation is assumed during initialization, which means that there are not transient changes in the system. Thus, a power flow solution provides suitable initial values for \bar{x}_{na} by solving (3) for the power system.

The power flow solution was obtained using PSS/E and provided to the model explicitly. The procedure to provide initial guess value can be viewed as the chain in Fig. 2, which starts from the components connected to system bus such as generators, flexible AC transmission system (FACTS) devices, or loads until the last equipment in the chain. The initial values of each state and set points are calculated in a reverse manner considering that the generator should provide the power quantities at desired voltage

level according to power flow calculation³.

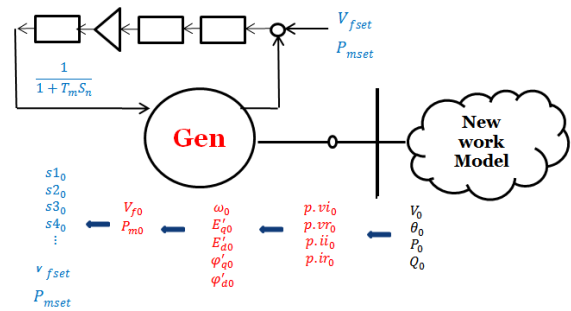


Figure 2. Procedure to provided initial guess values

In the case of initializing GENROU, the solution of the equations set was reduced to the solution of equation (4) in the rotor reference frame.

$$(X'_q - X''_q)i_{q0} + \psi''_{q0}K = 0 \quad (4)$$

where: $K = (1 + S_e |\psi''_0| \frac{X_q - X_l}{X_q - X_l})$

In order to take advantages of power flow and derived quantities (P_0, Q_0, I_0, V_0), it is necessary to express the equation above in the synchronous reference frame. In addition because changing the reference frame will not influence the magnitudes of the phasor, the value of ψ''_0 without saturation can be expressed in terms of voltage and current phasor at the generator bus (with the assumption that $X''_d = X''_q$). As a result, equation 4 can be written as:

$$\begin{aligned} -K|\psi''_0| \cos(\theta_{\psi''_0} - \delta_0) &= |I_{r0}|(X'_q - X''_q) \cos(\theta_{i0} - \delta_0) \\ j|\psi''_0| \angle \theta_{\psi''_0} &= |V_0| \angle \theta_{v0} + (R_a + jX''_q)|I_0| \angle \theta_{i0} \end{aligned}$$

from which the following initial guess value can be computed:

$$\delta_0 = \arctan\left(\frac{|I_0|(X'_q - X''_q) \cos(\delta_{\psi''_0} - \theta_{i0})}{|I_0|(X'_q - X''_q) \sin(\delta_{\psi''_0} - \theta_{i0}) - K|\psi''_0|}\right) + \delta_{\psi''_0}$$

where $\delta_{\psi''_0} = \theta_{\psi''_0} + \frac{\pi}{2}$. By substituting the value of δ_0 into the remaining equations, the initial values of all the states were obtained.

The initial values of states variables $\bar{x}_d, \bar{x}_{da}, \bar{x}_{na}$ can be specified with an equation through the initial equation construct or by setting the (fixed=true, start=x0) attribute when initializing variables (Fritzson, 2003). If nothing is specified, the default value would be zero and if (fixed=false) is set, the Modelica compiler will set a guess value. For parameters, setting (fixed=false) tells the compiler to implicitly compute them during initialization. At initialization, all the derivatives of the states (der(x)) and the parameters with (fixed=false) are treated as unknown variables. It is thus necessary to provide adequate initial values and initial equations to keep a balance between the number of unknown variables and equations.

³This is the typical method for initialization used in power system software, however, this method is used here only to provide initial "guess" values for Modelica to solve the initialization problem.

2.3 Model example: 3-winding Transformer with OLTC and phase-shifter

The three-winding transformer model is comprised by three two-winding transformer models with their secondary side interconnected, as shown on Fig. 3. Each two-winding transformer is equipped with both a phase-shifter and an on load tap-changer (OLTC) function. It is used to control the voltage magnitude of the user specified bus. Including the OLTC in a power system model results in a hybrid model. A hybrid model is comprised by continuous and discrete (and possibly event-driven) equations.

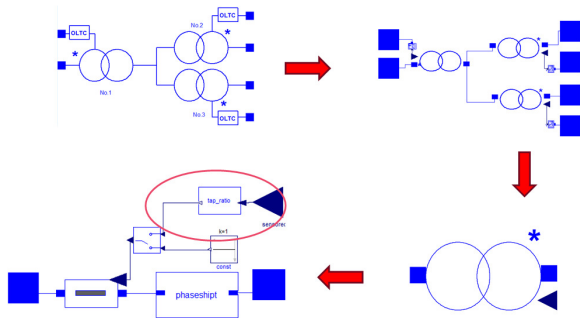


Figure 3. 3-winding transformer Icon and diagram view

The tapping logic of the OLTC is depicted in the block diagram in Fig. 4.

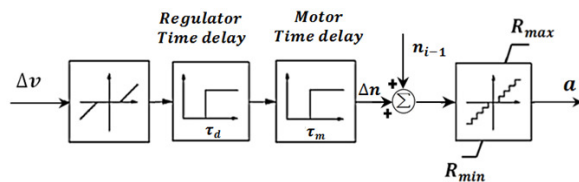


Figure 4. OLTC tapping logic block diagram

The corresponding Modelica code representation of this logic is listed in the Appendix. To keep the voltage at the controlled bus in the desired range, the OLTC adjusts the transformer turn ratio within the possible range in discrete steps. The model has internal delays to avoid unnecessary tapping actions.

A phase shifter is used to control active power flow by introducing a shift angle to both voltage and current phasors across the transformer. Ideally, there are no power losses due to the phase shifter. Hence, it is common practice to model the shift angle by introducing a variable impedance. However, in this work, a new method was adopted where the shift angle was introduced by modeling the receiving end phasor variables in a new axis which was separated from the sending end axis by the shift angle. The diagram is shown in Fig. 5. The receiving end phasor variables were obtained by applying Park’s transformation to the sending end phasor variables.

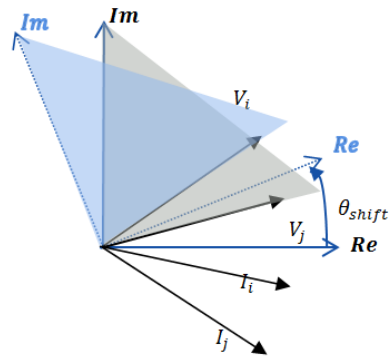


Figure 5. The diagram of phase shifter function

3 Software-to-Software Validation

The adopted software-to-software (SW-to-SW) validation approach consists in simulating an identical test system in both software environments and comparing simulation results qualitatively and quantitatively. For this approach to be successful, the solver chosen for the Modelica implementation should be equivalent to the PSS/E solver. The PSS/E documentation (Siemens PTI Inc., 2005) describes a trapezoidal integration method which corresponds to a second-order Runge-Kutta method, leading to the choice of “Rkfix2” for the simulations in Dymola (Dassault Systemes, 2015). A minimal difference between the two will guarantee the fidelity of the Modelica implementations. All of the models listed in Table 1 have been validated against PSS/E. To reduce the complexity of the validation procedure, identical small scale test systems were built in both Modelica and PSS/E for testing purposes.

Validation results of the components demonstrate both the correctness of the initialization and the transient behavior of the dynamic components under different disturbances. The validation tests were designed to activate each function inside the models and are, thus, case dependent. Different perturbations were applied to enable limiter and force the systems into a transient state after a few seconds from the simulation start.

In the sequel, the following sections briefly introduce the applied test scenarios and then go through the validation results of two of the test systems to prove the successful validation of the models.

3.1 Test system and validation scenarios

Validation tests are carried out using a simple system. In PSS/E it is not possible to test a controller (e.g exciter) without connecting it to a generator. Thus a single generator model was validated first and then each control system was validated using the validated generator. The basic scenarios used to study the transient performance of a generator and controllers are shown Fig. 6.

Most of the components were validated using the simple test system shown in Fig. 6. The perturbations intro-

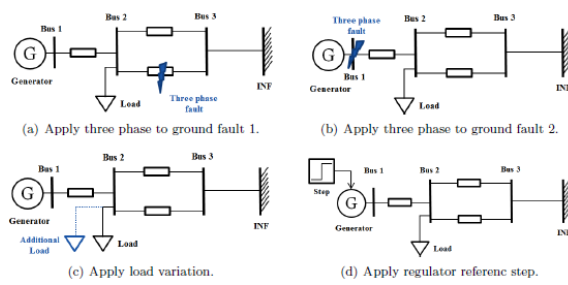


Figure 6. Test system and validation scenarios

duced were events such as three phase to ground faults, load variations and step changes applied to the regulators' set points. All of the perturbations were modeled in the same way in Modelica as they are in PSS/E.

3.2 Some Validation Results

The dynamic behavior of the power system a few seconds after a disturbance is driven by the stored energy in the rotors' inertias, transients in the field windings, and the controls of generators (Kundur et al., 1994). Therefore signals such as generator speed deviation, a bus voltage magnitude, and the injected active and reactive power P and Q of a generator are recorded and compared. When there exist regulators in the system, the output quantities of these devices are compared and analyzed. The applied perturbations are listed in Table 2. Results are shown in Fig. 7, where a comparison of the records of four state variables of the generator are presented. It shows that the behavior of the developed Modelica model matches perfectly with that of the PSS/E reference model in both steady state and transient conditions.

Table 2. Applied perturbations scenarios to validate the GENROU test system

Time	Changes Applied
0-2 s	Running under steady state
2 s	Vary the system load with constant P/Q ratio
2.15 s	Restore the original load
10 s	Apply a 3 phase fault to ground in the middle of one of the parallel lines
10.15 s	Clear the fault by tripping the line
20 s	Stop the simulation

Quantitative assessment as proposed in (Rogersten et al., 2014) is performed to measure the validity of a model response using numerical metrics. The assessment is carried out by calculating root mean square (RMS) of the difference between simulation outputs using the equation:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2} \quad (5)$$

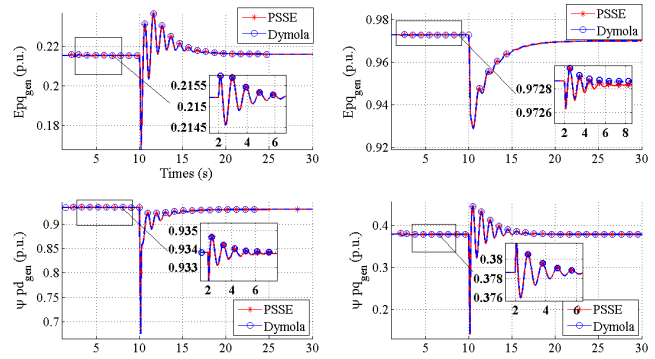


Figure 7. Validation results for the GENROU model

where x_i and y_i are the discrete measurement points at time t_i for software (a) and (b) respectively. The value is calculated on an interval of 2 s divided into total number of time steps in PSSE. In this case, the time step used is 0.001 s and the time interval included the period of pre-fault, fault, and post-faults such that the RMSE value reflects the response of the model against its reference. For example, for the test case shown in Fig. 8, there were seven changes applied and six variables were recorded to compute the RMSE values to validate the model quantitatively.

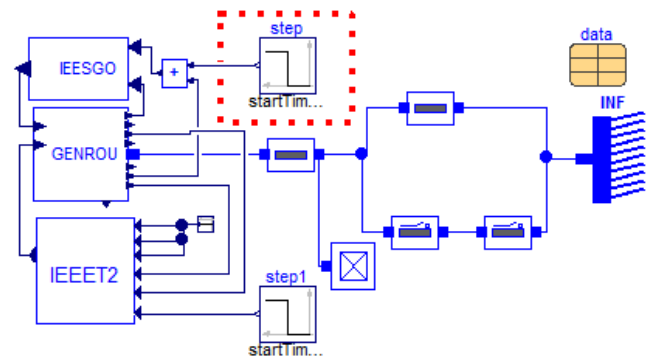


Figure 8. Test for GENROU, IEEESGO, IEEET2

For this case, all the perturbations scenario applied at 0.3 s and they are listed in Appendix Table 4. Results are shown in Appendix Table 5. This quantitative assessment has been carried out for each test case and the results show that the behavior of the developed Modelica model matches perfectly that of the PSS/E reference model in both steady state and transient conditions.

The test system used for the validation of the three winding transformer is shown in Fig. 9, where the remote and local loads were connected through a three-winding transformer. An infinite bus was connected to the primary side of the transformer, a local generation unit was connected to the secondary side and a composite load was connected to the tertiary side of the transformer. The local power plant was modeled using a GENROU model equipped with an IEEET1 exciter and a STAB2A stabilizer.

When the OLTC function of winding one is enabled, it

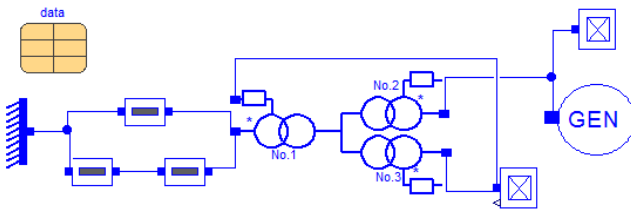


Figure 9. Test system for OLCT

controls the voltage at the load bus to stay within the desired range. The perturbations are designed to force the OLTC to activate following the logic shown in Fig. 4. They are listed in Table 3. The method to step Q in PSSE is not explained in the software documentation with sufficient detail, thus the value used in Q stepping in Dymola is not exactly the same as they are shown in Table 3. The actual value used in Dymola is obtained by trial-and-error to match the behavior in PSSE.

Table 3. Applied perturbations scenarios for OLTC validation

Time	Scenarios
0-20 s	Running under steady state
20 s-40 s	Increasing the Q load to 1.5 p.u.
40 s-80 s	Increasing the Q load to 2.2 p.u.
80 s-200 s	Increasing the Q load to 2.5 p.u.

The validation results are shown in Fig. 10 where the voltage magnitude at the three terminals of the transformer are compared. The comparison of Q stepping in both software is also shown in the figure. The small mismatch in the voltages are due to the differences in Q stepping. However, in this test, the aim is to verify the stepping logic of the OLTC instead of transient behavior, thus the results show that the Modelica OLTC model functions in the same way as the PSS/E reference model.

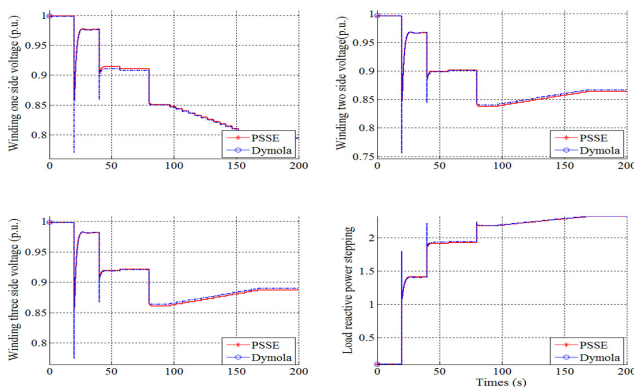


Figure 10. Validation results of OLCT

4 Conclusion

All the PSS/E component models specified by Statnett have been successfully implemented in Modelica. In addition, the developed models have been tested using OpenModelica and Dymola⁴. The validation results guarantee consistent simulation results among Openmodelica, Dymola, and PSS/E. This work can serve as a proof of the feasibility of using Modelica tools in power system for modeling and simulation as an alternative to PSS/E. This work also provides a proof of simulation capabilities of Modelica tools to handle power system problems in terms of the complex controls and initialization problems.

References

- Tetiana Bogodorova, Marc Sabate, Gladys Leon, Luigi Vanfretti, Milenko Halat, Jean-Baptiste Heyberger, and Patrick Panciatici. A modelica power system library for phasor time-domain simulation. In *2013 IEEE PES Innovative Smart Grid Technologies Europe (ISGT EUROPE)*, pages 1–5, 2013.
- Dassault Systemes. *Dymola*, 2015. <http://www.3ds.com/products-services/catia/products/dymola/>.
- Peter Fritzson. *Principles of object-oriented modeling and simulation with Modelica 2.1*. John Wiley & Sons, 2003.
- IEC 61970-301. *Energy management system application program interface (EMS-API) - Part 301: Common information model (CIM) base*, 2013.
- Prabha Kundur, Neal J Balu, and Mark G Lauby. *Power system stability and control*, volume 7. McGraw-hill New York, 1994.
- Modelica Association. *Modelica Language Specification*, 2014a. <https://modelica.org/documents>.
- Modelica Association. *FMI for Model Exchange and Co-Simulation Specification*, July 2014b. <https://www.fmi-standard.org/downloads>.
- R. Rogersten, L. Vanfretti, Wei Li, Lidong Zhang, and P. Mitra. A quantitative method for the assessment of vsc-hvdc controller simulations in emt tools. In *2014 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–5, Oct 2014.
- Richard P. Schulz. Synchronous machine modeling. *IEEE publication* 75, 1975.
- Siemens PTI Inc. *PSSE User Manual v30.2*, Siemens Power Transmission & Distribution. Power Technologies International, 2005.
- L. Vanfretti, W. Li, T. Bogodorova, and P. Panciatici. Unambiguous power system dynamic modeling and simulation using Modelica tools. In *2013 IEEE PES General Meeting*, pages 21–25, 2013.

⁴OpenModelica results will be shown in future publication.

5 Appendix

5.1 Modelica OLTC logic

```

//Check difference between scheduled
// and measured voltage.
if u - Vmax > 0 then
  voltageDiff = -1;
elseif Vmin - u > 0 then
  voltageDiff = 1;
else
  voltageDiff = 0;
end if;
//Set resetRegulatorTimer
//if difference exists.
when ResetTimer then
  reinit(MotorTimer,0);
end when;
//Let timer run if we have
// a large enough difference
when voltageDiff<>0 and StartTimer then
  tapDirection=
  if RegulatorTimer<=0 then voltageDiff
  else pre(tapDirection);
end when;

when RegulatorTimer<=0 or
  RegulatorTimer>=1 then
  MotorEnabled = if RegulatorTimer>=1
  then true else false;
end when;

if RegulatorTimer<=0 then
  der(RegulatorTimer)= if (voltageDiff<>0)
  then 1/Td else 0;
  ResetTimer=false;
  StartTimer=true;

elseif RegulatorTimer>=1 then
  der(RegulatorTimer)=
  if (tapDirection==voltageDiff)
  then 0 else -1/Td;
  ResetTimer=true;
  StartTimer=false;

elseif RegulatorTimer<1 then
  der(RegulatorTimer) =
  if (tapDirection == voltageDiff)
  then 1/Td else -1/Td;
  ResetTimer=false;
  StartTimer=false;
else
  der(RegulatorTimer) = 0;
  ResetTimer=false;
  StartTimer=true;
end if;

if MotorEnabled and MotorTimer < 1 then
  der(MotorTimer)=1/TC;
elseif MotorEnabled then
  der(MotorTimer)=
  if (TSD)>0 then 1/(TSD) else 1/TC;
else
  der(MotorTimer)=0;
end if;

```

```

tapMoved = integer(MotorTimer);

when change(tapMoved) and tapMoved <>0
  and pre(MotorEnabled) then
  m = pre(m) + dtap * pre(tapDirection);
end when;

if m>=Rmax then
  y = Rmax;
elseif m<=Rmin then
  y = Rmin;
else
  y=m;
end if;

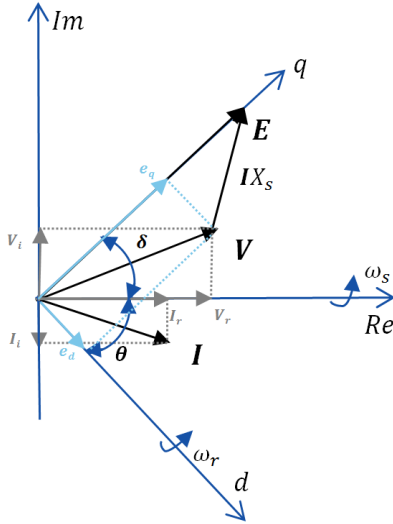
```

5.2 Dynamic equations of GENROU (Schulz (1975) and Kundur et al. (1994))

$$\begin{aligned} \dot{E}'_q &= \frac{1}{T'_{d0}} (E_{fd} - X_{ad} I_{fd}) \\ \dot{E}'_d &= \frac{1}{T'_{q0}} (-1) (X_{aq} I_{1q}) \\ \dot{\psi}_{kd} &= \frac{1}{T''_{d0}} (E'_q - \psi_{kd} - (X'_d - X_l) i_d) \\ \dot{\psi}_{kq} &= \frac{1}{T''_{q0}} (E'_d - \psi_{kq} + (X'_q - X_l) i_q) \\ X_{ad} I_{fd} &= \frac{(X'_d - X''_d)(X_d - X'_d)}{(X'_d - X_l)^2} [E'_q - \psi_{kd} - i_d (X'_d - X_l)] \\ &\quad + i_d (X_d - X'_d) + E'_q + S_e (|\psi''|) \psi''_d \\ X_{aq} I_{1q} &= \frac{(X'_q - X''_q)(X_q - X'_q)}{(X'_q - X_l)^2} [E'_d - \psi_{kq} + i_q (X'_q - X_l)] \\ &\quad - i_q (X_q - X'_q) + E'_d - S_e (|\psi''|) \frac{X_q - X_l}{X_d - X_l} \psi''_q \\ \psi''_d &= \frac{E'_q (X'_d - X_l) + \psi_{kd} (X'_d - X''_d)}{X'_d - X_l} \\ \psi''_q &= \frac{-E'_d (X'_q - X_l) - \psi_{kq} (X'_q - X''_q)}{X'_q - X_l} \\ |\psi''| &= \sqrt{(\psi''_d)^2 + (\psi''_q)^2} \\ \psi_d &= \psi''_d - X''_d i_d \\ \psi_q &= \psi''_q - X''_q i_q \\ 2H \frac{d\omega}{dt} &= T_m - T_e - D \frac{d\omega}{\omega} \\ \frac{d\delta}{dt} &= \omega_0 d\omega \\ T_e &= \psi_d i_q - \psi_q i_d \\ e_d &= -\psi_q - R_a i_d \\ e_q &= +\psi_d - R_a i_q \end{aligned}$$

$$\begin{bmatrix} V_r \\ V_i \end{bmatrix} = \begin{bmatrix} \sin(\delta) & \cos(\delta) \\ -\cos(\delta) & \sin(\delta) \end{bmatrix} \times \begin{bmatrix} e_d \\ e_q \end{bmatrix}$$

$$\begin{bmatrix} I_r \\ I_i \end{bmatrix} = - \begin{bmatrix} \sin(\delta) & \cos(\delta) \\ -\cos(\delta) & \sin(\delta) \end{bmatrix} \times \begin{bmatrix} i_d \\ i_q \end{bmatrix}$$


Figure 11. Synchronous and Rotor Coordinate

```

// swing equation
der(delta) = wbase * w;
der(w) = ((PMECH-D*w)/(w+1)-Te)/(2*H);
Te = PSId * iq - PSiq * id;
// transient voltage & flux
der(Epq) = (1 / Tpd0) * (EFD - XadIfd);
der(Epd) = (1 / Tpq0) * (-1) * Xaqllq;
der(PSIkq) = (1 / Tppd0) *
(Epq-PSIkq-(Xpd-Xl)*id);
der(PSIkq) = (1 / Tppq0) *
(Epd-PSIkq+(Xpq-Xl)*iq);
// flux linkage
PSId = PSIppd - (Xppd * id);
PSIq = PSIppq - (Xppq * iq);
PSIppd = (Epq * K3d) + (PSIkq * K4d);
PSIppq = (-Epd * K3q) - (PSIkq * K4q);
PSIpp = sqrt(PSIppd^2 + PSIppq^2);
// saturation effect
XadIfd = (K1d*(Epq-PSIkq-(Xpd-Xl)*id))
+ Epq + (id * (Xd - Xpd))
+ (Se(PSIpp, s10, s12) * PSIppd);
Xaqllq = (K1q*(Epd-PSIkq+(Xpq-Xl)*iq))
+ Epd - (iq * (Xq - Xpq))
- (Se(PSIpp, s10, s12) * PSIppq)
* (Xq - Xl) / (Xd - Xl);
    
```

5.3 Quantitative assessment

Table 4. Applied perturbations scenario to validate the system in Fig. 8

No.	Changed Applied
1	Three phase to ground fault ($Z=0.5 + 0.5j$ p.u.) in the middle of one parallel line at last for 0.5 sec
2	Three phase to ground fault ($Z=0.2 + 0.2j$ p.u.) at bus 2 last for 0.5 sec
3	Trip one parallel line
4	Step of Efd reference 0.01 p.u.
5	Step of Efd reference step -0.01 p.u.
6	Step of Pmech reference step -0.002 p.u.
7	Step of Pmech reference step 0.001 p.u.

Table 5. RMSE Calculations according to Equation(4)

	<i>Genrou(Pelec)</i>	<i>Genrou(Qelec)</i>	<i>IEEET2(Efd)</i>
1	5.6150e-13	1.4125e-11	5.8495e-11
2	4.9004e-13	3.1062e-13	6.1085e-13
3	4.3398e-14	1.2844e-18	7.3565e-13
4	3.9391e-14	3.1698e-16	7.7892e-13
5	4.7541e-14	5.5590e-15	4.2712e-13
6	2.9281e-14	7.9683e-15	3.9605e-13
7	1.0012e-14	5.1896e-15	6.4285e-13
	<i>IEEESGO(Pmech)</i>	<i>Genrou(Speed)</i>	<i>Genrou(Vt)</i>
1	1.6497e-14	3.3524e-15	2.5234e-12
2	4.6549e-16	1.4021e-17	1.6314e-14
3	1.7494e-18	3.3895e-19	1.4843e-17
4	1.0529e-15	1.5968e-20	9.1668e-15
5	2.6711e-16	1.7923e-19	6.5037e-16
6	1.6166e-16	5.3919e-20	9.4046e-16
7	6.1121e-16	1.5454e-17	3.6245e-16