

MANGO Classes-Modelica Classes of the Norwegian Grid for iTesla and SW-to-SW Validation

August 18, 2014

MENGJIA ZHANG



KTH Electrical Engineering

Master's Degree Project
Stockholm, Sweden 2014

XR-EE-EPS 2014:009

February 2014 to August 2014

**MANGO Classes-Modelica Classes of the Norwegian Grid for iTesla
and SW-to-SW Validation**
— Diploma Thesis —

MENGJIA ZHANG¹

Electrical Power Systems Division

School of Electrical Engineering, KTH Royal Institute of Technology, Sweden

Supervisor

Maxime

KTH Stockholm

Examiner

Dr. Luigi Vanfretti

KTH Stockholm

Stockholm, August ??, 2014

¹mengjia@kth.se

Abstract

This thesis is focus on the implementation of dynamic Norwegian power grid components in Modelica language. Currently, due to the available model and data format, analysis and simulation of Norwegian network are performed through transitional power system simulator PSS/E. The sponsor of this thesis project, the Norwegian transmission system operator, Statnett SF is carrying out a program, the Smart Operation program which will utilize the Modelica language to describe dynamic components of power grids.

To guarantee the dynamic Norwegian grid models can be properly represented in Modelica language, software-to-software validations are performed between PSS/E and Dymola. Several small scale power system models were developed in order to carry out complete validation tests. A reasonable degree of similarity have been achieved when performing the validation. After that, two sub-set models of real Norwegian grids were developed to perform further validation for the components.

Acknowledgments

There are a lot of people I would like to thank for help me during this thesis work. Firstly, I'd like to thank my supervisor and examiner Prof. Luigi Vanfretti. I am very appreciate that he accepted me to carry out a small project to build Modelica SVC models last summer. I have to thank him very much for trusting me and providing me with the necessary training in order to start working, because at that time I was brand new about Modelica and had no working experience. Further more, I'd like to thank him for being patient with me, providing guidance generously, and encouraging me to explore and carry out my own idea during working. Now, the little summer project ended up with this thesis work. I have learned really a lot from the whole procedure and have gained so much fun out of it.

Secondly, I'd like to thank my direct supervisor Maxime Baudette. To me it is really appreciate that he scheduled the regulate meeting for keeping good track of my work and answering my questions when I encountered some obstacles which I lacked the knowledge to solve. I'd like to thank him a lot for giving me very appreciate suggestions towards the my report and presentation of the project.

Finally, I'd like to thank my colleagues in SmartLab group at EPS (Electrical Power System)department of KTH (The Royal Institute of Sweden). Thank them for listening to my idea and having good discussion with me. It help me to make a good judgment and inspired my about the way to realize the idea. Also I would like to thank my parents and all of my friends. Thank them for providing good accompany when sometimes I was really depressed by my work. Thank them for listening to my complain and encourage me to carry on.

Mengjia Zhang
June

Contents

List of Tables	ix
List of Figures	xi
Notation	xiii
1. Introduction	1
1.1. Background	1
1.2. Motivation to Use Modelica in Power System	2
1.3. Objectives	3
1.4. The outline of the thesis	4
2. Software Environment	5
2.1. Introduction to PSSE	5
2.1.1. Power System Model and Simulation Procedure	5
2.1.2. Integration Algorithm	7
2.2. Introduction to Modelica	7
2.2.1. Dymola, the Modelica simulation environment	7
2.2.2. Modeling environment	11
2.2.3. Modelica libraries	12
3. Model Development	15
3.1. Electrical Element	15
3.1.1. Load	15
3.1.2. Transformer	18
3.1.3. Synchronous generator	24
3.1.4. Initialization	31
3.2. Nonelectrical Element	34
3.2.1. Excitation System	34
3.2.2. Power System Stabilizer	42
3.2.3. Governor System	44
4. Software implementation	49
4.1. Developed library	49
4.2. Build Class	49
4.2.1. Variables Declarations	50
4.2.2. Initialization	53

4.2.3. Equations	54
5. Validation Results	61
5.1. Small Scale Power System Tests	61
5.1.1. Validation of Genrou	65
5.1.2. Validation of Gensal	69
5.1.3. Validation of IEEEET2 and IEESGO	71
5.1.4. Validation of SCRXS and HYGOV	79
5.1.5. Validation of SEXS and IEEEEST	79
5.1.6. Validation of IEEEET1, STAB2A and LFDRAL	86
5.2. Real World Power System	86
5.3. AKD	86
5.3.1. System Setup	86
5.3.2. Validation result	86
6. Conclusion	95
A. Appendix	97
A.1. Dynamic parameter	97
B. Appendix	101
References	103

List of Tables

2.1. Available default variable types	11
2.2. Modelica Classes Overview	11
2.3. Equation Types	12
3.1. Composite Load Parameters	17
3.2. Frequency Dependent Composite Load Parameters	18
3.3. Two-windings Transformer Parameters	24
3.4. Three-windings Transformer Parameters	24
3.5. OLTC model Parameters	24
3.6. Standard Synchronous machine parameters of Genrou	29
3.7. GENSAL	30
3.8. IEEEET1	37
3.9. IEEEET2	38
3.10. SEXS	40
3.11. SCRX	41
3.12. IEEEEST	43
3.13. STAB2A	44
3.14. HYG0V	47
3.15. IEESGO	48
5.1. Power Flow Result	62
5.2. Combination of Dynamic Devices	63
5.3. Initialization	65

<h2>List of Figures</h2>

1.1. Power System Model Consist of Non- and Electrical Subsystems	3
2.1. PSS/E simulation procedure[9].	6
2.2. Translation stages from Modelica code to executing simulation[12]	8
2.3. Over view of Dymola (1)	8
2.4. Over view of Dymola (2)	9
2.5. Over view of Dymola	10
2.6. Brief view of MSL and PowerSystems Library.	13
2.7. Drag-and-drop components from library to build system[2]	13
3.1. Constant Power Load Characteristic	17
3.2. Constant Current Load Characteristic	17
3.3. Two Winding Transformer Model[9]	19
3.4. Phase shift function of transformer	20
3.5. Three Winding Transformer Model[22]	20
3.6. Diagram of ULTC Tap Logic[22]	21
3.7. Example of Integration Timer OLTC1[9]	23
3.8. Synchronous and Rotor Coodinate	26
3.9. Saturation mine function	28
3.10. Overview Models and Relationship	31
3.11. The equivalent circuit for self and shunt exciter	35
3.12. Block Diagram of IEEEET1	36
3.13. Block Diagram of IEEEET2	38
3.14. Block Diagram of SEXS	40
3.15. Block Diagram of SCRX	40
3.16. Equivalent circuit of SCRX with <i>crowbar</i>	41
3.17. Equivalent circuit of SCRX with reverse polarity bridges	41
3.18. Block Diagram of IEEEEST	43
3.19. Block Diagram of STAB2A	44
3.20. Hydro Turbine Diagram[25]	44
3.21. Hydro Turbine Diagram 2[25]	45
3.22. Block Diagram of HYGOV	46
3.23. Block Diagram of IEESGO	47
4.1. The package going to be developed	50
4.2. Example of Parameter Dialog	50

5.1. Test system diagram	62
5.2. Different perturbation cases.	63
5.3. Diagram in Dymola of the 1 _{st} test system.	65
5.4. Validation result of 1 _{st} test system (1).	66
5.5. Validation result of 1 _{st} test system (2)	68
5.6. Diagram in Dymola of the 2 _{nd} test system.	69
5.7. Validation result of 2 _{nd} test system (2)	70
5.8. Bus voltage's angle shift between sending and receiving ends of the transformer. 71	
5.9. Diagram in Dymola of the 3 _{rd} test system.	71
5.10. Validation result of 3 _{rd} test system (1)	74
5.11. Validation result of 3 _{rd} test system (2)	75
5.12. Validation result of 3 _{rd} test system (3)	76
5.13. Validation result of 3 _{rd} test system (4)	77
5.14. Validation result of 3 _{rd} test system (5)	78
5.15. Diagram in Dymola of the 4 _{th} test system.	79
5.16. Validation result of 4 _{th} test system (1)	80
5.17. Validation result of 4 _{th} test system (2)	81
5.18. Validation result of 4 _{th} test system (3)	82
5.19. Validation result of 4 _{th} test system (4)	83
5.20. Validation result of 4 _{th} test system (5)	84
5.21. Validation result of 4 _{th} test system (6)	85
5.22. Diagram in Dymola of the 5 _{th} test system.	86
5.23. Validation result of 5 _{th} test system (1)	87
5.24. Validation result of 5 _{th} test system (2)	88
5.25. Diagram in Dymola of the 6 _{th} test system.	89
5.26. Validation result of 6 _{th} test system.	90
5.27. Test system diagram	91
5.28. Validation result of AKD test system(1).	92
5.29. Validation result of AKD test system(1).	93
5.30. Validation result of AKD test system(3).	94

Notation

$\Delta\omega$	per unit speed deviation from synchronous
δ	angle between quadrature axis and terminal voltage
ω	per unit speed of generator
ψ'_d	direct axis flux linkages
ψ'_q	quadrature axis flux linkages
E'_d	Voltage proportional to quadrature axis flux linkages
E'_q	Voltage proportional to direct axis flux linkages
e_d, e_q	Armature voltage, direct and quadrature axis components
E_{fd}	generator field voltage(one per unit is the value for 1 per unit terminal voltage on the air gap line, open circuit)
i_d, i_q	Armature current, direct and quadrature axis components
P	active power
P_e	electrical power
Q	reactive power
T_e	electrical torque
T_m	mechanical torque
v_t	terminal voltage
$x(t), x$	vector of differential variables
$y(t), y$	vector of differential variables
BDF	Backward Differentiation formula
DAE	Differential and Algebraic Equations
ODE	Ordinary Differential Equations
PSAT	Power System Analysis Toolbox

Notation

PSS/E Power System Analysis for Engineer

SMIB Single Machine Infinite Bus

SPS SimPowerSystem

1.1. Background

The operation of Pan-European electricity network is becoming more and more complex. That is because there are growing contribution of less predictable renewable energy sources, the rapidly development of new control technologies, and the progressive construction of a single European electricity market etc. The network is working closer to its operation limits which yields the need for revision of operational rules and procedures. As a result, different transmission system operators (TSOs) increased coordinate and harmonize operation procedure[1].

While the problem is that currently different TSOs are using various types of software to perform power system analysis (e.g. PSS/E, Eurostag...). And transitionally the network model are built with predefined models which are programmed inside the simulation programs. The mathematical representations of the models may be implicit and the corresponding data formats are program dependent. Inconsistent simulation records and analysis results occur across different software platforms even if the system models are set up to be identical. That means the security assessment decision for a system from two software could be different or contract. It is difficult to verify the correctness of the models or to perform modification to any existing models. Since on one hand the detail modeling methods are either hidden or described imprecisely by the software document, and on the other hand it is tedious work due to the solver-dependency of the models. A user who attempt to do verification or make any modification to the models has to start from scratch. Firstly, to get the correct mathematic representation of the models and then to understand the source code including calculation algorithm.

As a consequence, European Commission and several TSOs co-found a project named iTESLA which stands for: Innovative Tools for Electrical System Security within Large Area. iTESLA was launched in January 2012 and will run until end of 2015. The overarching goal of the iTESLA project is to develop and validate an open inter-operable toolbox able to support the future operation of the Pan-European grid [1]. One of the challenge which the tool box has to take up is to perform accurate security assessment taking into account the dynamics of the system using time-domain simulations. As it has been explained above, the simulation of nowadays complex network model requires the tool to have following capacities[5]:

1. A clear separation between the model and the solver.

2. Facilitate the reuse of the modeling knowledge.
3. The capability of handling the simulation of hybrid system which include both continuous and discrete time variabilities.

Thanks to the nature of Modelica language, it is picked up as the base program and modeling language used in the project. This thesis project is a sub-project of the iTesla project and it is found by Statnett, the Norwegian transmission operator. Currently, Statnett is using CIM and PSS/E data format to specify the dynamic components of the power grids. The revision of Norwegian grid models will be carried out in this thesis, and validation tests will also be performed in order to ensure the proper performance of the developed power grid models.

1.2. Motivation to Use Modelica in Power System

In power system community, numerous efforts have been made to enable consistent dynamic models exchange. However, due to the inconstancy of modeling and computation method across different simulation platforms, there are difficulties in the exchange of the models between simulation tools. The labor can be saved if the modeling method of the dynamic components are open and independent of the simulator engine. The nature of the equation base, declarative programming language Modelica satisfies all the requirements for modeling complex power system with such consistent format of dynamic power grid model[15].

Thanks to utilize equations, the causality are eliminated when developing new models. This feature of Modelica enable the better reuse and exchange of the developed power grid models. And because Modelica language adopts the declarative programming method, the modeling and simulation are completely separated. Thus the modeler can save huge effort from designing the simulate procedures and calculation algorithm. And can only focus on fully understanding the model and get the correct equation representations of the dynamic models respect to physical aspect.

What's more, there are now more than ten simulation platforms can support Modelica language both commercially and free of charge such as OpenModelica, JModelica, Dymola, MapleSim, SimulationX, Mathematica, SystemModeler etc. The more powerful part is that, the Modelica code can be exported and be simulated in MATLAB/Simulink once the Flexible Mock-up Interface (FMI) is installed. This feature enables the future optimization and the on-line identification of the system models. The simulator used throughout this thesis is Dymola. In the future, the models will be checked and be simulated with OpenModelica.

The principal of modeling power system in Modelica is briefly explained here and the detail of modeling method will be introduced through the rest of the thesis. In Modelica each of the power system component is modeled by mathematic equations respect to the physical rules. The direction of the power flow obey the physical constrains without causality. The system can be divided into electrical sub-systems which contain all of the electrical components who participate in power exchange; and the nonelectrical sub-systems which are consist of the control and protection system who take the measured signal from the network and affect the system behavior by regulating the output signal to the network. Physically, the electric power

system converts primary energy, *e.g.* heat, into electricity and then transmit it through system network and finally delivery to power consumption load. The power system network is a set of interconnection power grid devices, *e.g.* transformers, transmission line. Instantaneous balance should be kept between power generation and consumption which is the key of power flow. While the regulators influence the power flow by varying the signal to control the electrical devices. In this sense, the power system model can be viewed as shown in Fig. 1.1.

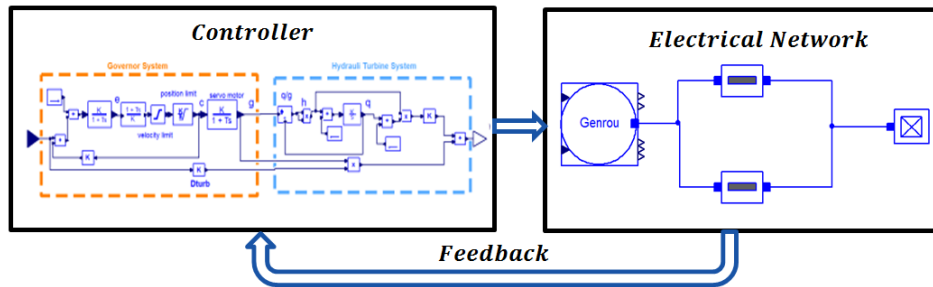


Figure 1.1.: Power System Model Consist of Non- and Electrical Subsystems

1.3. Objectives

The aims of this project are to develop a corresponding Modelica class for every component model which is used in dynamic Norwegian grid PSS/E model. Firstly, it is necessary to study the materials and to get a full understanding of the models. After that, the modeling method should be summarized and explained clearly in order to facilitate the reuse of models. The developed models will be integrated into PowerSystems library in Dymola and will be used to perform dynamic simulation of the grid.

However, before jumping to the stage of dynamic simulation, power flow of the system and initialization of each components should be carried out. Since this thesis will only focus on the analysis of transient stability of power system utilizing the developed components but not power flow. And there is not power flow function available in Modelica. PSS/E will be used to perform power flow and the results will be exported from PSS/E and serve as input parameters during initializing the Modelica model.

To promise the correct performance of the developed Modelica class, validation tests should be carried out between Modelica and PSS/E. Firstly, the models should be validated within small scale power system models where full tests of each components should be carried out. by full tests, it means that within the tests, every single special function of the model should be activated and the records should illustrated the proper behaviors of it. When it comes to the stage of comparing the dynamic behaviors of the system by using the two software, a reasonable degree of similarity should be achieved from the validation tests.

Secondly, the models should be further validated by increasing the scale of system models. In this project, two subset models of Norwegian grid will be built with developed Modelica classes. The final tasks, within this project must be synchronized and collaborated with

another master thesis work (MANGO Models), which will develop different power system models utilizing the Modelica classes validated in this project.

According to that, the tasks can be specified as following:

Part 1: Modeling

1. Fully study the documents of PSS/E to understand the modeling method of each components.
2. Summary the over whole mathematic equations and assumptions of all the models.
3. Solve initialization problems of the developed models.
4. Implement the models according to Modelica specification.

Part 2: Validation

1. Set up power system models in PSS/E.
2. Run power flow in PSS/E and record the results.
3. Set up identical system models in Dymola.
4. Initialize Modelica model using power flow records from PSS/E.
5. Run dynamic simulation in the two softwares.
6. Plot and compare results in matlab for each pair of systems.

The models can be accepted only if their behavior show a reasonable degree of similarity both in steady state and in dynamic simulation. The detail of validation method will be introduced in Chapter 5. If the results from Dymola are diverse form those from PSS/E, the whole procedures should be go through all over again, but one can reduce the labor by studying the results and locating the mistakes. A brief introduce of debugging procedures of the models will be covered in Chapter 7.

1.4. The outline of the thesis

The remaining parts of the thesis is organized as follows. Firstly, a brief introduction of the software PSS/E and the specification of Modelica language are given in Chapter 2. Chapter 3 covers the mathematic representation of the dynamic Norwegian grid components. After that, the readers are supposed to have the basic idea of how the implementation should be performed. Chapter 4 describes some examples to further explain the detail of implementation method of the components. The results of validation are analyzed in Chapter 5. The discussion based on the validation results will be presented in Chapter 6. The thesis is ended by the conclusion and future work in Chapter 7.

2

Software Environment

This chapter gives a brief introduction to the main software environment used in this thesis. The reference model are from wild used traditional power system simulator PSS/E, Power System Simulator for Engineer. The first part of this chapter provides a brief introduction of PSS/E including a general explanation of the adopted network modeling method, the applied simulation engine algorithm and the dynamic analysis procedure of the software. Secondly, Modelica language together with its simulation platform Dymola will be introduced. This section contains brief explanation of the layout and basic function of Dymola. Following that, a few summaries of Modelica language specification will be provided.

2.1. Introduction to PSSE

PSS/E package is an integrated set of computer programs for studies of power system transmission network and generation performance in both steady-state and dynamic conditions. Considering timescale, PSS/E focus on the analysis of electromechanical transient behavior of the generators and their controls named transient stability analysis.

2.1.1. Power System Model and Simulation Procedure

Power system in PSS/E is modeled by a set of algebraic equations for voltages and currents phasor at the fundamental frequency. The dynamic behavior of the system depends on the differential equations of the flux linkage, rotor speed, and controllers of generators. The transmission system of the network is modeled by linear algebraic equations of fundamental frequency voltage and current vector derived from Kirchhoff's laws. The nonlinearity is introduced by the characteristics of generators and loads. The interface between the electrical components and the network is the certain voltage versus current characteristics determined by dynamic components.

During power flow, the equations of the network are expressed as constant admittance matrix. Both transmission lines and transformers are modeled by equivalent series and shunt admittance, but the difference is that a complex nominal tap ratio introduced by transformers can allow modification of voltage phasor at both ends. While during transient study, the dynamic generators and the load demands should also be included. In order to include them, PSS/E makes the generators and load conversion. The synchronous generators are modeled by a specification of equivalent Norton source whose instantaneous values are governed by the dif-

ferential equations of rotor speed and flux linkages. And loads are consist of constant current and admittance load with certain characteristics. After conversion of loads and generators, the admittance matrix is updated and power flow should be solved again to eliminated the little mismatch if there is any.

Before switching into dynamic analysis, the system should be initialized with the results from power flow. During this stage, the time derivate of all state variables are set to be zero. Together with power flow results, the equations can be set up to solve the initial values of each state variables and the set points of the controllers. The goal is to find the equivalent operation point of the system which severs as start point of dynamic simulation.

The main skeleton of PSS/E only contains logic for data input, output, numerical integration, and electric network solution, but contains no logic related to differential equations of dynamic equipments. They are written in subroutines. Whenever there are needs to calculate the numerical values of time derivatives for one dynamic equipment, the corresponding subroutine which contains such calculation logic is then called by the main skeleton for initialization stage or for dynamic simulation within each time step. The procedure is shown as in Fig2.1.

The users are allowed to model their own components but beside determine the equations

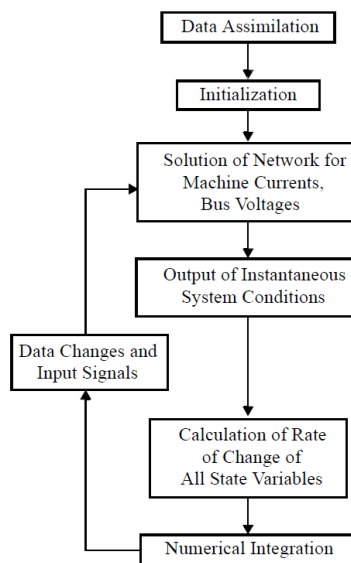


Figure 2.1.: PSS/E simulation procedure[9].

there are several other tasks should be completed. Firstly, one should identify the state variables of the model. And then determine the procedure for computing the time derivatives of each states, additionally,the input output causality and the date locations of the model should be identified. Finally, the user should describe the model in FORTRAN or FLECS. The simplification and assumption in the modeling method is then sticked to the model, the further user can only accept it otherwise one have to follow the same whole procedure in order to make some modification of the model.

2.1.2. Integration Algorithm

The integration technique applied in PSS/E is an explicit methods which belongs to the family of the Second-Order Runge-Kutta method[9]. PSS/E solver applies integration algorithms to the values taken from the last time step to calculate the values of the next time step. It is obviously that, the nature of this computation process is discrete and can be viewed as a transfer function. Thus it was expressed in Z-transform form which is a power series representation of a discrete-time sequence. The equation is as shown below.

$$F(Z) = \frac{h(3Z - 1)}{2Z(Z - 1)} \quad (2.1)$$

Apply inverse Z-transform straight forwards to the equation, one can get the following time domain equation which yields a Two-step Adams-Bashforth method.

$$x_k = x_{k-1} + \frac{h}{2}(3f_{k-1} - f_{k-2}) \quad (2.2)$$

2.2. Introduction to Modelica

Modelica is an equation-based object-oriented modeling language which capable to describe both continuous and discrete dynamic of physical systems in a convenient way by differential, algebraic and discrete equations. Modelica enables the accusal equation-based description of the complete physical system by a set of interconnected multi-domain subsystems where the mathematical equations describing the dynamic behavior of the subsystems and the interconnection rules governing the information communication between them. No fixed causality is imposed on the model, since the mathematical equations are valid regardless of the causality context[14]. The language enables separation between the modeling environment and the numerical solver. The description of the system is described with equations instead of assignments. The compiler uses the source Modelica code to build several execution process for the simulation engine. The whole process is illustrated as in Fig. 2.2.

The Modelica source code contains both graphical and textual description of the system. During the stage of compiling, an analysis is performed for the internal representation converted from the source code. The resultant flat set of equations, constants, variables, and function definition is then processed to an optimizer module. The output of the module contains only a minimal set of equations needed to be solved. Those explicit form independence equations are then converted to assignment statements. Finally, C code for simulation is generated.[12]

2.2.1. Dymola, the Modelica simulation environment

Currently, there are numerous Modelica simulation platforms available both commercially and free of charge. In this thesis Dymola is chosen. Dymola, the Dynamic Modeling Laboratory, from Dynasim company[3] is a general-purpose simulation environment for Modelica. Dymola incorporates a set of advanced solvers that can perform the simulation of complex hybrid systems. The basic parts of Dymola tool are:a Modelica compiler with a symbolic optimizer for reducing equation size; a simulation engine including solver which can handle hybrid DAE

2. Software Environment

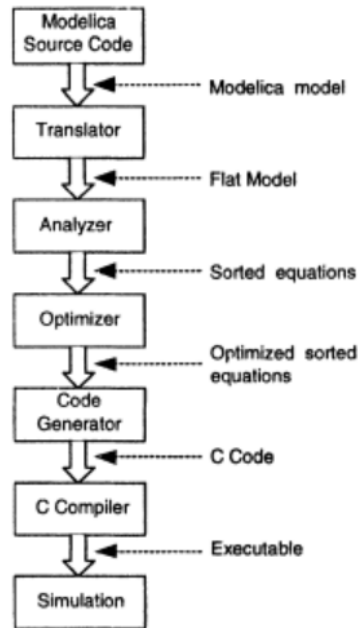


Figure 2.2.: Translation stages from Modelica code to executing simulation[12]

equations systems; a graphic user interface; a text editor. Fig. 2.3 shows an overview of Dymola environment.

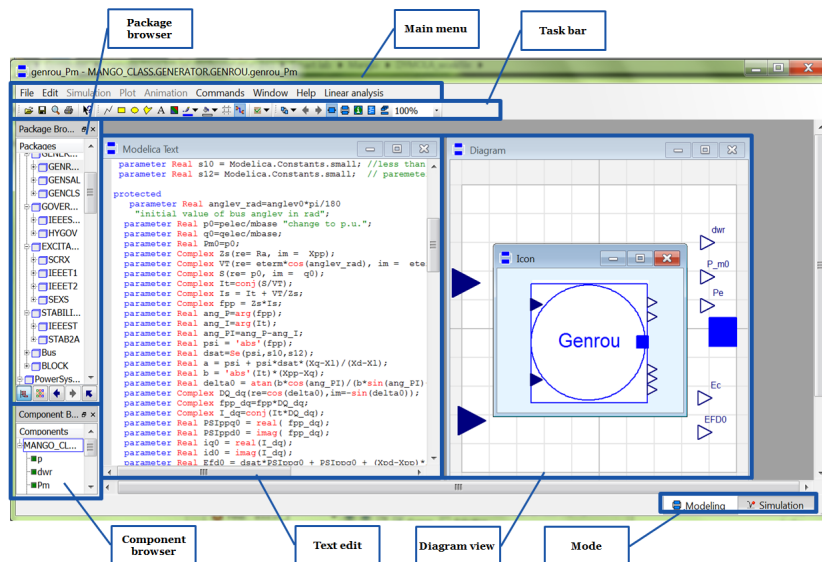


Figure 2.3.: Over view of Dymola (1)

The device class in Dymola is consist of connected sub-components (hierarchical structure) and/or is described by equations. Models are added by drag-and-drop from the library package browser to the diagram view. In Dymola, the user can easily switch between the modeling mode and simulation mode. The diagram above shows the mode of modeling where one can create, edit, modify and brow models. The user can switch to simulation mode by click the

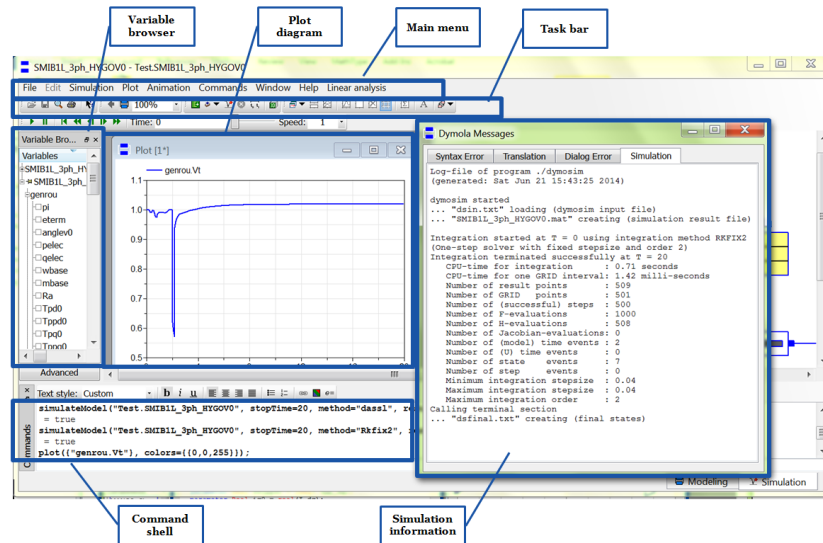


Figure 2.4.: Over view of Dymola (2)

icon on the low right corner. The diagram is shown in Fig. 2.4. After define a class, it is necessary to perform **check** to the model. The **check** function in Dymola is used to make sure that the model is correct respect to some aspects such as balance between the number of the unknowns and equations, correctness of the syntax and the existences of all the paths.

When the model has been checked successfully, the code is ready to be converted into C-code, and then be compiled into a executable file. The information (the number of the equations, unknowns, states, parameters, and variables) about the structure of the model will be shown in the translation log. Additionally, a statistical result of the number and the sizes of the of linear and nonlinear equations before and after symbolic manipulation will be shown. After translation, the model can be simulated. When switch to the simulation mode, the user should make setup the simulation by choosing the desire solver, time step, tolerance, plot interval and simulation time length. If not specified, Dymola will choose the solver DASSL by default. The tolerance and the plot interval are 0.0001 and 500 respectively. The dialog for set up the simulation is in Fig. 2.5.

After performing the simulation, the variables are accessible through variable browser which contains parameters and variables of the simulated model. Except those defined with prefix **protected**, all of the variables will be listed hierarchal in the browser and can be plotted by drag-and-drop the variables to the plot diagram. Up to 99 different simulation results can be shown simultaneously in the Variable Browser. Furthermore, the information about the executed simulations such as CPU time, function and Jacobian evaluations, and the number of events and result points are available in the prompt up simulation log.

2.2.1.1. Dymosim and solvers

Dymosim stands for Dynamic model simulator and it the executable code generated by Dymola for computing initial value and performing simulation of the model[4]. There are several

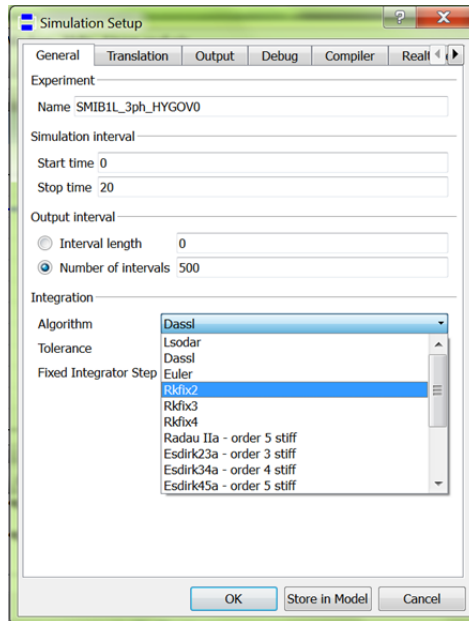


Figure 2.5.: Over view of Dymola

different environments can support Dymosim which is a stand-alone program, among which Dymola is especially suitable to be used in conjunction with it. Dymola transforms the definition of the system into a state space description which is then handled by the integrators of Dymosim.

Currently, Dymosim provides ten different variable step-size integration methods. An integration method approximates the solution internally by a polynomial of order k . Some methods such as Euler, Rkfix use a fixed order while other methods such as DASSL, LSODAR vary the order during simulation. The bigger the order, the larger the integration step-size could be which impose a better efficiency of computation. We will only talk about two solvers here which are Rkfix2 and DASSL.

Rkfix2 is a 2nd order Runge-kutta fixed step solver. The method can be expressed in following form. This method can only handle the ODE type, no stiff system model. Although our power system model contains DAEs, Dymola converts the model to explicit ODE form first before handing it over to solver for simulation. All of the tests in this thesis project have selected Rkfix2 as the solver since this is the one most similar to the PSS/E solver.

$$x_{k+1} = x_k + hf(t_k + \frac{1}{2}h, x_k + \frac{1}{2}hf(t_k, x_k)) \quad (2.3)$$

DASSL is considered to be the most successful simulation algorithm on the market nowadays. It is a variable-step, variable-order method which implements the backward differentiation formula (BDF) from order one to five in DAE format. It can solve DAEs directly and can deal with stiff system. It is currently set as the default simulator in Dymola. DASSL will be used for checking the performance of the model in general way for example checking the tendency of behavior but is not selected to solve the tests in this thesis. Because, a variable-step size might loose or catch more points of the records, which might introduce difference

between the results. Thus in order to control unknown factors, the method which is most similar to the PSS/E solver, Rkfix2 is applied.

2.2.2. Modeling environment

The fundamental structuring unit of modeling in Modelica is the class. Every object is an instant of the corresponding class which defined its data and behavior. A class is consist of a name, a list of declaration of its attribution, and several equations under the equation section.

2.2.2.1. Attributions

The attribution of a Modelica class contains: constants, parameters, and variables. By default, all of the variables are continuous time variables which might evolves their values continuously during simulation. The specific *time variability* for variables can be declared by prefixes *constant*, *parameter*, *discrete*. Constants will never change their value once they have been defined, *e.g.*, $\pi = 3.1415926\dots$. The value of the variables declared as *parameter* can be assigned by users before simulation or after the stage of initialization, but will remain constant during time-dependent simulation. The *discrete* variables can change their values only at event instants during simulation. Furthermore, *time* is a global built-in variable, which can be used without declaration.[13]. There are four types of variable by default which are shown in Table2.1:

Table 2.1.: Available default variable types

Types	Description
Real	floating point, <i>e.g.</i> 1.0, $-2.3e - 5$
Integer	integer, <i>e.g.</i> 1, -4, 333
Boolean	boolean, <i>e.g.</i> false, true
String	string, <i>e.g.</i> "from file:"

For the sake of making the code to be more readable and easier to be maintained, special keywords as shown in Table2.2 are utilized to describe specific class under some conditions [13].

Table 2.2.: Modelica Classes Overview

Types	Description
type	Class to define variable types
connector	Class to define interfaces
model	Class to define model components
block	Like model class, but only public input, output or parameter
function	no internal states, with algorithm section and function-call syntax
package	Class to define library, no equations
record	Class to group variable declarations, no equations

2.2.2.2. Equations/ Algorithm

Equations are added in the equation section after the **equation** keyword. Differential equations are expressed with the **der**-operator. It denotes the time derivative of the expression following in brackets, the state variable. The states are integrated by the numerical solver to find the solution of the problem. Each state requires an initial condition. Order of equations and which of the variables are located on the left or right hand side of the equality sign is irrelevant. Each component must have a set of equations that uniquely defined its behavior based on its interfaces and initial conditions. There may be the following types of equations:

Table 2.3.: Equation Types

Type	Example
initial equation	$\text{der}(w)=0$
Equality equation	$V = R \times I$
Connection equation	connect (pwpin1.n,pwpin2.p)
Conditional equation	if-then; when-end when

2.2.3. Modelica libraries

2.2.3.1. Modelica standard library

Modelica standard library (MSL) is a free (standard conform) library developed by Modelica Association. The brief view of the library is shown in the left side of Fig.2.6. It is freely available in the source code and can be modified to be used in commercial software. The elements in the library can be used to model multi-domain system which can include: mechanical (1D/3D), electrical (analog, digital, machines), thermal, fluid, control systems and hierarchical state machines. To build a system, one can drag-and-drop the components from the library to the graphical edit screen as shown in Fig.2.6 Also numerical functions and functions for strings, files and streams are included[2]. This thesis utilize several basic blocks such as transfer functions and some numerical functions to build the components of control and protection system.

2.2.3.2. PowerSystems library

The PowerSystems library is developed under *iTesla* project shown in Fig. 2.6. It is written in Modelica language within in Dymola. The elements in the library can support time-domain simulation of small or medium scale power system models. The dynamic components models developed in this thesis project will finally be integrated into this library. Some of the simple models such as **Pwline** and **Pwpin** are utilized in this thesis. **Pwline** is a pi-equivalent model of transmission line. And **Pwpin** is the connector for setting up connection between electrical components.

Connectors are special types of Modelica classes defining an interface to a model which rules the physical properties at the boundaries of the models. A connection between two connectors will generate equations based on two types of connection rules: equality and sum-to-zero. For

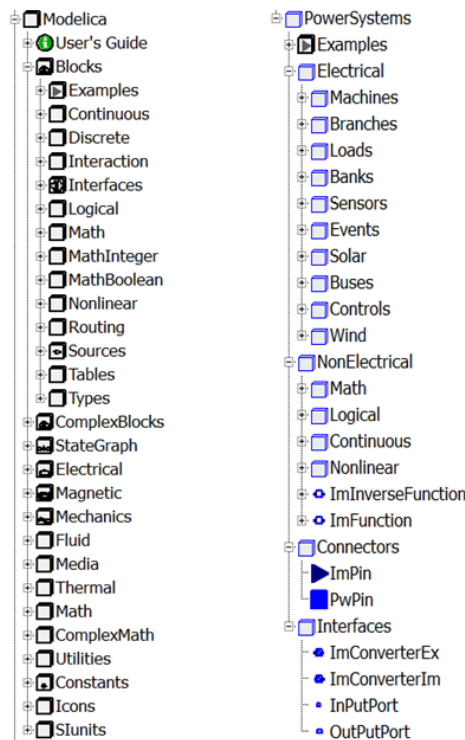


Figure 2.6.: Brief view of MSL and PowerSystems Library.

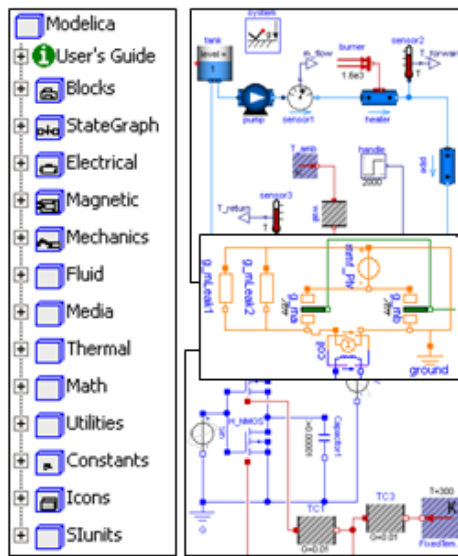


Figure 2.7.: Drag-and-drop components from library to build system[2]

all connector variables declared with prefix **flow**, only one equation will be generated per connection set *i.e.* $i_1 + i_2 + i_3 = 0$. Those without prefix by default is set to be potential variables, and $n - 1$ equations based on equality rule will be generated per set. Connection between connectors are regards as equations. The direction of data flow in the connection is treated without causality. But when the flow directs towards inside of the component,

2. Software Environment

the sign of the variables are considered as positive by convention. Additionally, one can also specify causal connection by declaring connector as *input* or *output*.

PwPin defines voltage and current as complex variables. Four variables are declared within **PwPin** connector, and they are real and imaginary part of potential variables voltage and flow variables current. For other components like synchronous machines regulator, **Realinput** from Modelica library are adopted, and only potential variables are declared for signal communication.

3

Model Development

As it has been introduced in last chapter, the fundamental structuring unit in Modelica is a class. Since Modelica is an equation-based languages. The key to build a Modelica class is to get the correct mathematical representation of the dynamic model and then write the equations down according to Modelica specification. In this chapter, the detail explanation of the mathematical representation of each device will be provided. We start from electrical element such as load, transformer, generator and then move to nonelectrical components, the regulators. Once the equations have been set up, there is only one step remain before the model is completed. That is to solve the initialization problems of the model. Here the transitional way of initializing is applied where the dynamic devices are initialized with the results from power flow and it is a straight forwards procedure without iteration. The method will also be explained at the end.

3.1. Electrical Element

3.1.1. Load

Although there is a composite load classes available in Dymola PowerSystem library, it is different from the one using in PSS/E. In order to get a better result during modeling validation stage, a new class of composite load is developed respecting exactly the same modeling method used in PSS/E.

Since during steady-state study the total demand level are of more interest and the network bus voltage are at or close to nominal voltages, as a result it is recommended to specify the load as constant MVA load. However, when it switches to dynamic simulation, the constant MVA load characteristic increases the stiffness and unrealistic of the network model. In order to make the load more realistic and reduce the stiffness of the system, PSS/E converts the constant MVA load to be a mix of the three components with user defined portion ratio where the resulted constant current and constant admittance load is added to any existing load represented by those characteristics.

The converting process following the rules below:

$$\begin{aligned}
 S_I &= S_i + \frac{aS_p}{V} \\
 S_Y &= S_y + \frac{bS_p}{V^2} \\
 S_P &= S_p \times (1 - a - b)
 \end{aligned} \tag{3.1}$$

where:

- S_p Original constant MVA load
- S_i Original constant current load
- S_y Original constant shunt admittance load
- S_P Final constant MVA load
- S_I Final constant current load
- S_Y Final constant shunt admittance load
- a, b Load transfer fractions, $(a + b) < 1$
- V Magnitude of bus voltage when load conversion is made

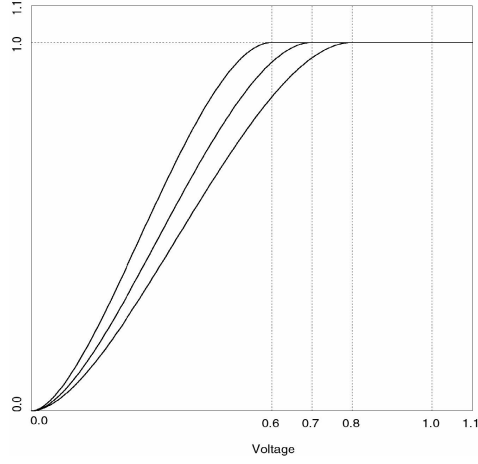
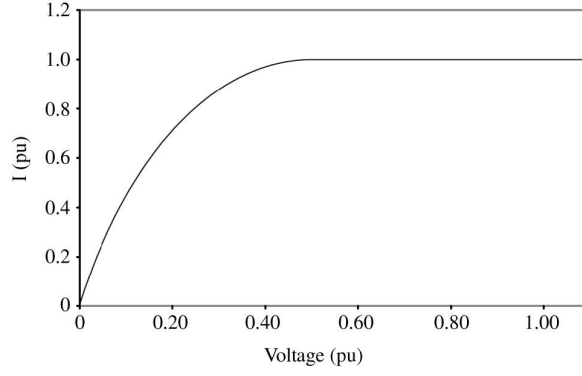
Furthermore, the constant load characteristic becomes unrealistic when the bus voltage drops too much from the nominal value of it (normally approximately 0.8 pu). In PSS/E this threshold value can be specified through the solution parameter **PQBRAK**. Fig.3.1 describes the behavior of the constant power load as functions of bus voltage respecting to different **PQBRAK** with values of 0.6, 0.7, 0.8. Similarly, the characteristic for constant current load should be modified as long as the bus voltage exceeds about 0.5 pu. The modified characteristic of the constant current load is shown as in Fig.3.2. When the voltage is below the threshold values, the corresponding load should be shed according the specific behavior as defined above. It can be modeled as to times the original load with a shedding factor.

The manuals of PSS/E do not provided the exactly equations which rule the specific behavior shown in those two figures. However, with these two available figures it is possible to extra the data points lying along the curves with MATLAB and then find the equations which match the curves. In this thesis, only two curves are extracted, one is the shedding factor function $K_P = f(V)$ for constant power load with **PQBRAK** of 0.7, the other is that of constant current load, $K_I = f(V)$.

$$K_P = \begin{cases} 0.4881 - 0.4999 \cos(3.964V) + 0.1389 \sin(3.964V) & V < 0.7 \\ 1 & V \geq 0.7 \end{cases} \tag{3.2}$$

$$K_I = \begin{cases} 2.657V^{0.769} \exp(-1.502V^{1.769}) & V < 0.5 \\ 1 & V \geq 0.5 \end{cases} \tag{3.3}$$

As it is the same way in PSS/E, the newly developed Modelica composite load class recognizes load components with constant power (P, MVA), constant current(I,MVA) and constant admittance (Z,MVA) characteristics. Users can specify each component in MW and Mvar. For constant current and constant admittance components, the values should represent the consumed MW and Mvar under per unit bus voltage. Furthermore, the converting procedure are also included in the new model. The user should specified certain Load transfer fraction for constant current and constant admittance load. The Modelica code is available in Appendix


Figure 3.1.: Constant Power Load Characteristic

Figure 3.2.: Constant Current Load Characteristic

while the parameters for the composite load is shown in Table3.1:

Table 3.1.: Composite Load Parameters

Column	Parameter	Type	Description	Unit
1	S_p	Comp.	Original constant power load	p.u
2	S_i	Comp.	Original constant current load	p.u
3	S_y	Comp.	Original constant shunt admittance load	p.u
4	a	Comp.	Coverting fraction for constant I load	-
5	b	Comp.	Coverting fraction for constant Z load	-
6	PQ_{BRAK}	Real	Constant power characteristic threshold	p.u.

The voltage and current characteristics for the load should be then specified as following:

$$\begin{aligned}
 P + jQ &= \tilde{V}\tilde{I} \\
 P &= K_P \Re(S_P) + K_I \Re(S_I)V + \Re(S_Y)V^2 \\
 Q &= K_I \Im(S_P) + K_I \Im(S_I)V + \Im(S_Y)V^2
 \end{aligned} \tag{3.4}$$

Where \tilde{V} and \tilde{I} are the complex value of load bus voltage and current respectively. The power exchange information is contained in the potential and flow variables of the **PwPin** connector (v_r, v_i, i_r, i_i) . And the voltage and current phasor are expressed as:

$$\begin{aligned}\tilde{V} &= v_r + jv_i \\ \tilde{I} &= i_r + ji_i\end{aligned}\tag{3.5}$$

Defining that the positive direction of flow variables is flowing inwards to the model thus the active and reactive power value with positive sign illustrate that the power is being consumed. As a consequence, the active and reactive power of the load can be expressed as in Equation 3.6. Then it is understandable that when it comes to determine the sign for the power exchange of generator which is defined as power source, the sign should be negative for both active and reactive power.

$$\begin{aligned}\tilde{P} &= v_r \times i_r + v_i \times i_i \\ \tilde{Q} &= v_i \times i_r - v_r \times i_i\end{aligned}\tag{3.6}$$

Further more, there is also a frequency dependent load going to be developed. In PSS/E document, there is not clearly claim about how the model calculates the frequency of the load bus. But in Modelica, the frequency of the load bus is calculated directly by differentiate the bus angle. It is lucky that the behavior of the Modelica model matches that of PSS/E model fairly well. The load frequency model make the constant current and MVA components of all loads dependent on bus frequency according to following rules:

$$\begin{aligned}P &= P_o \frac{\omega^m}{\omega_o^m} \\ Q &= Q_o \frac{\omega^n}{\omega_o^n} \\ I_p &= I_{po} \frac{\omega^r}{\omega_o^r} \\ I_q &= I_{qo} \frac{\omega^s}{\omega_o^s}\end{aligned}\tag{3.7}$$

Thus for a frequency dependent load model (LDFRAL), there are four more parameters should be specified by user.

Table 3.2.: Frequency Dependent Composite Load Parameters

Column	Parameter	Type	Description	Unit
1	m	Real	Real power load frequency exponent	-
2	n	Real	Reactive power load frequency exponent	-
3	r	Real	Real current load frequency exponent	-
4	s	Real	Reactive current load frequency exponent	-

3.1.2. Transformer

In this section the modeling method of positive-sequence transformer will be introduced. The detail of derivation procedure of the transformer equivalent circuit will not be covered. For

further information, one can refer to [6][9]. There are two types of transformers going to be developed. They are a two-winding transformer and a three-winding transformer. The transformer models can include the function of phase shift and on load tap changer.

Two Windings Transformer

The standard PSS/E two-winding transformer circuit is shown Fig.3.3. Letter with subscripts i stands for variables at primary side, and those with subscripts j represents the quantities at secondary side. t is the per-unit turns ratio and \bar{X}_{eq} is the equivalent impedance of the transformer branch. For the sake of reducing the number of variables, in PSS/E it is assumed that changing tap position does not alter the flux field map of the transformer. Based on that the winding leakage reactances are proportional to the number of turns squared as tap position is adjusted. After making proper manipulation of the equations, the equivalent impedance can be calculated as the sum of the winding leakage reluctances times the number of turns on the secondary side. Thus X_{eq} is a constant value as long as the tap ratio on the secondary side is fixed. For transformers without tap changer, that is always the case.

But for those with tap changer, it is assumed that variable-tap winding can only be located on the primary side, while the secondary side is always equipped with a fixed-tap winding. As a result, the term X_{eq} can be specified as a parameter. Currently the values of these two parameters have to be pre-calculated by users according to the value of measured impedance across the windings, alternate voltage based, and the specified nominal secondary turns ratio. The equations for calculations can be found in [9].

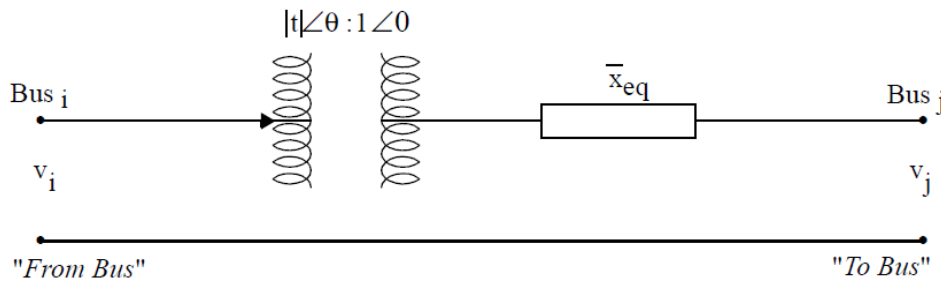


Figure 3.3.: Two Winding Transformer Model[9]

The equations of two-winding transformers are the same as those of transmission line if the tap ratio is real unit one. If the phase shift is introduced then the tap ratio becomes a complex number \tilde{t} which shifts both current and voltage in one direction with the shift angle θ . Introducing a shift angle to both voltage and current components in the same direction can be interpreted as viewing the original voltage and current phasor from a new coordinate which is rotated with the shift angle in reverse direction. That means, it is the same as changing the reference frame. To make it more clear, it is shown in Fig.3.4. As a consequence, the transformer can be modeled as the equations for an original transmission line plus the equations of Park's transformation for changing the coordinate.

Three Windings Transformer

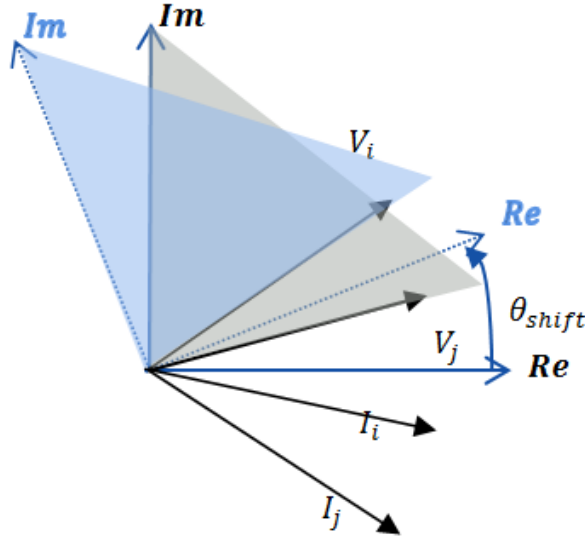


Figure 3.4.: Phase shift function of transformer

The three winding transformer is modeled as three two-winding transformers connected their secondary winding side together at a common star point bus as shown in 3.5. The model automatically calculates the three equivalent leakage impedances of each windings from the per-unit between windings leakage impedances obtained from short circuit tests. The calculation follows the Equation3.8. The complex off-nominal taps $t_n^{j\phi_n}$ ($n = 1, 2, 3$) only associate with the primary side winding which is connected to network bus, while the secondary winding side always has a unit tap ratio.

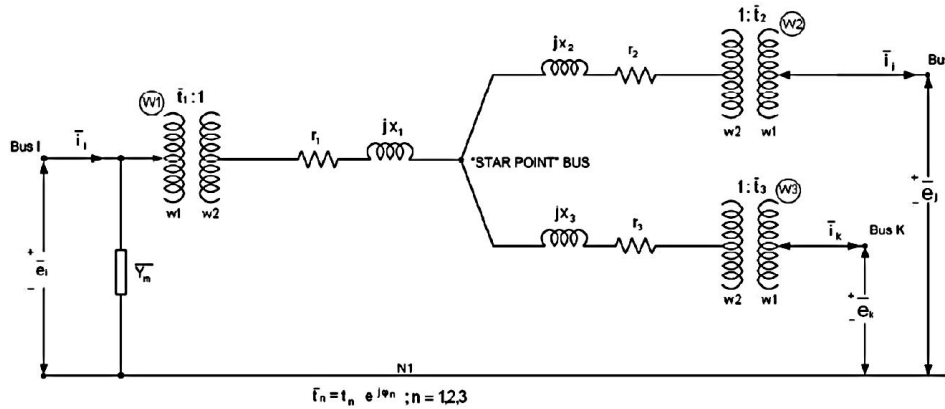


Figure 3.5.: Three Winding Transformer Model[22]

$$\begin{aligned}
 Z_1 &= \frac{Z_{12} + Z_{13} - Z_{23}}{2} \\
 Z_2 &= \frac{Z_{12} + Z_{23} - Z_{13}}{2} \\
 Z_3 &= \frac{Z_{23} + Z_{13} - Z_{12}}{2}
 \end{aligned} \tag{3.8}$$

Take the advantage of physical connector model in Modelica, the three windings transformer can be built by directly connecting the secondary side of three two-winding transformers together and the corresponding equations will be generated automatically.

3.1.2.1. On load tap changer OLTC

The above transformer are static models if without tap changer. To include the tap changer action, a discrete state variable should be introduced to represent the voltage dependent tap ratio. The modeling method of OLTC used in this thesis follows a well-accepted discrete time OLTC model which has been presented in [24], and has also been discussed in [23]. The key of the model is the logic describe the tap ratio as a function of time and controlled bus voltage. When adopting the well-accepted tap changer logic of the model, some modifications are necessary to be made. Since there are slightly different between the behavior of OLTC using in PSS/E and that of the well-accepted model. The modified switching logic is shown as in Fig.3.6 and the corresponding mathematical representation is provided in Equation 3.9.

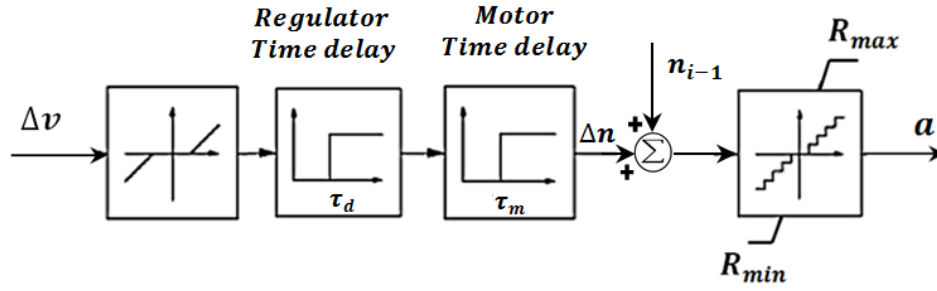


Figure 3.6.: Diagram of ULTC Tap Logic[22]

With this logic, the measured voltage magnitude at the controlled bus is compared with the switching band at each time step. In order to avoid unnecessary vibration caused by measurement noise, a user defined tolerance, ΔB , is introduced to both upper and lower limit of the voltage band. Only if the measured value of the controlled bus voltage is outside the control band more than the tolerance, the variables which are defined to indicate the desired moving direction of the tap changer will then change the values which will activate the timers (one for each direction, up or down) to count the elapse time since that instance. Every time, when the indicator change its value, the corresponding timer will be refreshed and reactive. When the elapse time is longer then the regulator delay, τ_e , and the signal indicate the same direction for tap changer, then the information is passed to the next logic. The regulator delay, τ_e is a fixed value in our case, although unreal, reader can refer to [24] for more information about the model with adaptive delay.

The motor will not take action directly, since there exist mechanical delay, τ_m in the motor. To include it, the tap action is considered to be taken every certain time interval according to the mechanical delay as long as the signal indicate a movement. This is recorded by a counter which will increase its value for one increment every time interval. The motor monitors the counter and will take action in one direction every time when the corresponding counter increase its value.

Up to now we can say that the model is almost the same as the well-accepted one. However, the OLTC model include one more logic to avoid frequent action of OLTC during the situation of voltage vibration. There is an example in PSS/E document illustrating the situation which is shown in Fig.3.7. As it is shown that a voltage which is out-of-range for 13 sec and returns within the bandwidth for 5 sec. Therefore, a tap signal is not sent until another 22 sec (30 - (13 - 5)) elapses with the voltage below the minimum value[9]. To solve this problem, one more if-then condition should be added to make sure that the timer is renewed only if the indicator e changes sign and the time length between the last change is larger than the regulator's delay τ_e . Additionally, PSS/E allows for a subsequent delay for sending further signals to the tap changer motor. The user can define the value for a extra delay, T_{SD} . Note that delay T_{SD} must be greater than T_C to be effective.

$$e = \begin{cases} 1, & v - V_{max} > DB \text{ and } a_{i-1} < R_{max} \\ -1, & v - V_{min} < -DB \text{ and } a_{i-1} > R_{min} \\ 0, & \text{otherwise,} \end{cases} \quad (3.9)$$

$$c^1 = \begin{cases} c_{t-\Delta t}^1 + \Delta t, & e = 1 \text{ and } c_{t-\Delta t}^1 \geq 0 \\ c_{t-\Delta t}^1 - \Delta t, & e = -1 \text{ and } c_{t-\Delta t}^1 \leq 0 \\ 0, & \text{otherwise,} \end{cases} \quad (3.10)$$

$$c^2 = \begin{cases} c_{t-\Delta t}^2 + \Delta t, & c_{t-\Delta t}^2 \geq 0 \text{ and } c_t^1 > \tau_d \\ c_{t-\Delta t}^2 - \Delta t, & c_{t-\Delta t}^2 \leq 0 \text{ and } c_t^1 < -\tau_d \\ 0, & \text{otherwise,} \end{cases} \quad (3.11)$$

$$n = \begin{cases} 1, & e = 1 \text{ and } c_t^2 > \tau_m \\ -1, & e = -1 \text{ and } c_t^2 < -\tau_m \\ 0, & \text{otherwise,} \end{cases} \quad (3.12)$$

$$r_i = r_{i-1} + n\Delta r \quad (3.13)$$

$$a = \begin{cases} r_i, & R_{max} \geq r_i \geq R_{min} \\ R_{max}, & R_{max} < r_i \\ R_{min}, & R_{min} > r_i \end{cases} \quad (3.14)$$

- e Measuring element output
 c^1 Regulator time delay element output
 c^2 Mechanical time delay element output
 n tap ratio step counter
 r tap ratio step counter
 a tap ratio position in p.u.
 τ_d adjustable time delay of regulator
 τ_m Tap changer motor mechanism time delay
 DB adjustable deadband of regulator
 V_{max} Upper limit of the voltage control band
 V_{min} lower limit of the voltage control band
 R_{max} Upper limit of the off-nominal turns ratio in p.u.
 R_{min} lower limit of the off-nominal turns ratio in p.u.

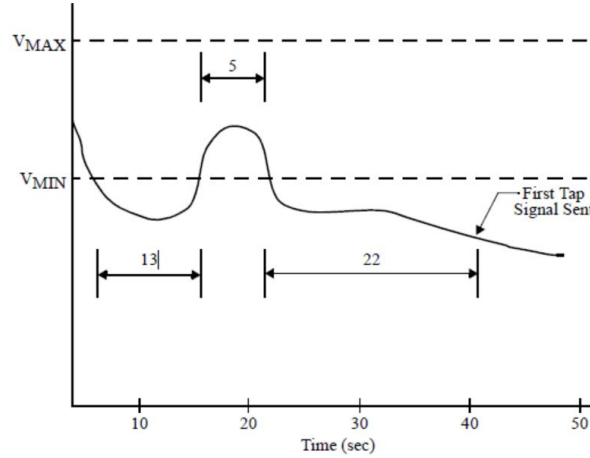


Figure 3.7.: Example of Integration Timer OLTC1[9]

Now all of the models related to transformers have been discussed which can be briefly summarized as following: the transformer models can be built based on the basic π -equivalent transmission line model if the tap ratio is unit and real. To include phase shift, a complex value of tap ratio is introduced, but in the developed model, the phase shift function is represented as viewing the primary side quantities in a reverse shifted coordinate according to the phase shift angle. Further more, to include the function of OLTC, several discrete variables should be introduced to complete the logic of tap changer. Finally, a tree winding transformer model can be built by simply connect the secondary side of three two winding transformers together the difference is that the parameters of equivalent impedance of each

winding is calculate automatically by the three winding model using the value of between winding impedance. The parameters set of the models are summarized as following:

Table 3.3.: Two-windings Transformer Parameters

Column	Parameter	Type	Description	Unit
1	X_{eq}	Comp.	Equivalent impedance	p.u
2	t	Real	Off-nominal tap ratio	p.u
3	θ	Real	Phase shift angle	p.u

Table 3.4.: Three-windings Transformer Parameters

Column	Parameter	Type	Description	Unit
1	Z_{12}	Comp.	Leakage impedance between winding 1 and 2	p.u
2	Z_{23}	Comp.	Leakage impedance between winding 3 and 2	p.u
3	Z_{13}	Comp.	Leakage impedance between winding 1 and 3	p.u
4	t_1	Real	Off-nominal tap ratio of winding 1	p.u
5	θ_1	Real	Phase shift angle of winding 1	p.u
6	t_2	Real	Off-nominal tap ratio of winding 2	p.u
7	θ_2	Real	Phase shift angle of winding 2	p.u
8	t_3	Real	Off-nominal tap ratio of winding 3	p.u
9	θ_3	Real	Phase shift angle of winding 3	p.u

Table 3.5.: OLTC model Parameters

Column	Parameter	Type	Description	Unit
1	DB	Real	Dead-band of regulator	p.u
2	V_{max}	Real	Upper limit of the voltage control band	p.u
3	V_{min}	Real	lower limit of the voltage control band	p.u
4	R_{max}	Real	Upper limit of the off-nominal turns ratio	p.u
5	R_{min}	Real	lower limit of the off-nominal turns ratio	p.u
6	T_D	Real	Time delay of the regulator	s
7	T_M	Real	Mechanical time delay of the motor	s
8	T_C	Real	Time delay of the regulator to avoid voltage vibration	s
9	T_{SD}	Real	Extra user define time delay	s

3.1.3. Synchronous generator

Synchronous machine plays an essential role in modern power system and thus correct presentation of these machines are crucial in power system dynamic studies. Synchronous machine consists of two essential elements, the field and the armature. For the sake of physical practice, the field winding is put on the rotor side while the armature is put on the stator side. When carrying direct current, a rotating field winding can produces a magnetic field rotates with the same speed which can induces alternating voltages in the armature windings. On the other hand, when carrying balanced three phase currents, the armature will produce a

rotating air-gap magnetic field. During normal operation, these two fields are rotating with the same speed thus the balance is kept between mechanical torque τ_m and electro-magnetic torque τ_e . During this process, the mechanical power is transferred to electric power injected into the system network. A set of differential equations can describe the dynamic behavior of the machine. For the sake of solving these equations, Park's transformation are applied to eliminate the time varying inductances and transfer the three phase stator variables to two phase in rotor reference frame thus the machine quantities became static. [6]

Since the electricity is generated by synchronous machine via electromagnetic relationship where the mechanical power provided by a prime mover driving the rotor to rotate and the resultant rotating magnetic field can induce the alternative voltage in the stator which is connected to the system network. Under balance operation the electrical quantities have the same fundamental frequency which is determined by the rotating speed of the generators rotor thus all of the machines should stay synchronism during steady state. The speed of the system is the synchronous speed. All the network variables are expressed according to the reference frame rotates at the synchronous speed, and this network coordinate is named synchronous reference frame.

Every synchronous machine is modeled in its individual rotor reference frame which is rotate at the rotor speed. Electric source is represented by equations of flux behavior in orthogonal dq -axes. When the system is subjected to a disturbance, speed deviation from synchronous speed occurs which be caused be unbalance between power generation and consumption. For the sake of analysis, all of the machine variables have to be transferred to synchronous reference frame.

In time domain analysis, power system is modeled by a set of hybrid nonlinear differential algebraic equations (*DAE*) Which take the form 3.15.

$$\begin{cases} \dot{x} = f(x, y, h, u, t) \\ 0 = g(x, y, h, u, t) \end{cases} \quad (3.15)$$

Where x is a vector of differential variables including generator fluxes linkage, rotor speed deviation, and control signals of machines' regulators. And y is the vector of algebraic variables, *e.g.*, phasors of network voltage, currents. To include the actions of controllers or protection schemes typically the step tap changer action of transformers, discrete time states u are introduced to the system. h is a set of controllable parameters, *e.g.* some if-then conditions which rules the behavior of the discrete variables u . Differential equations f describe all dynamics from electromechanical oscillations to load evolution while algebraic equations g represents the network constraints. Classically, g utilize the current-injection model which expresses the whole network by setting up the relation between the phasors of bus voltage and the current injections at the network[7].

The basic assumption of synchronous machines modeling are very well-known and served as ground in this thesis during model development procedure. One can refer to [6] for detail information, thus they will not be repeated here. Take the advantage of Park's transformation, the time variability of mutual impedances are eliminated. The models are thus represented in machine's rotor axis, named the direct and quadrature axis, dq -axis. The dynamic behavior

of the machines are mainly affected by rotor flux linkage transients and magnetic saturation during non-steady-state with 0 to 10HZ. And the reference model Genrou and Gensal from PSS/E are inherently correct representation of electromagnetic synchronizing and damping effects over the entire frequency band.

Genrou represents a solid rotor generator with field and one iron circuit on d -axis and two iron circuits on q -axis while Gensal models the salient pole generator with field circuit on d -axis and only one amortisseur circuit on both d -axis and q -axis. Both of the two models include the effect of saturation. One state variable should be included for each rotor circuit, thus Genrou is a 6th order round rotor generator and Gensal is a 5th order salient rotor generator.

3.1.3.1. Change reference frame

As it has been described in ??, individual synchronous machines variables are transferred to rotor reference frame, however the network quantities are expressed in synchronous reference frame. Figure 3.8 illustrates the relationship between two coordinate.

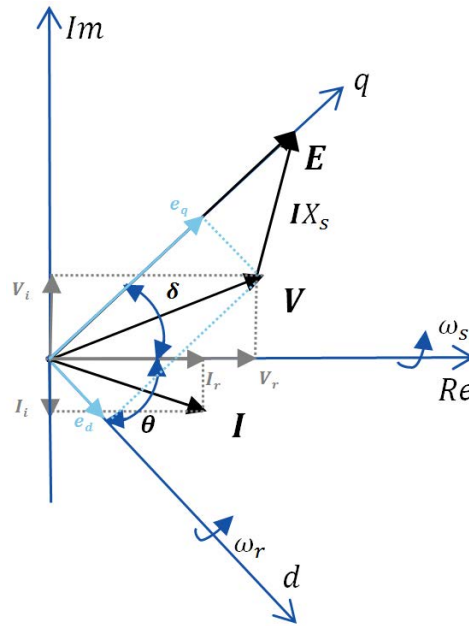


Figure 3.8.: Synchronous and Rotor Coordinate

where generator is represented by a voltage source E behind a dynamic impedance X_s . The rotor angle here is angular spatial positions of the generator rotor shaft. The synchronously rotating reference axes \Re - \Im axes leading the dq -axes by angle $\theta = \frac{\pi}{2} - \delta$. Thus relationship between the quantities expressed in \Re - \Im axes and dq -axes are as following:

$$\begin{bmatrix} V_r \\ V_i \end{bmatrix} = \begin{bmatrix} \sin(\delta) & \cos(\delta) \\ -\cos(\delta) & \sin(\delta) \end{bmatrix} \times \begin{bmatrix} e_d \\ e_q \end{bmatrix} \quad (3.16)$$

3.1.3.2. Equation of Motion

The electromagnetic torque T_e induced by the stator three phase currents can be expressed by the current and flux linkages in the dq axis:

$$T_e = \psi_d i_q - \psi_q i_d \quad (3.17)$$

While mechanical torque T_m can take the initial value or provided by corresponding turbine governor. As it is described in ??, since T_e revolves the rotor, a unbalance between T_e and T_m can result in increasing (or decreasing) rotating energy of generator rotor and thus cause the speed deviation of the generator. To include the effect from damping torque, a term proportional to speed deviation is added. Hence, the equation of motion is:

$$\begin{aligned} 2H \frac{d\Delta\omega_r}{dt} &= T_m - T_e - D_e \frac{d\Delta\omega_r}{\omega_r} \\ \frac{\delta}{dt} &= \omega_0 \Delta\omega_r \end{aligned} \quad (3.18)$$

T_m = mechanical torque in p.u

T_e = electromagnetic torque in p.u

ω_r = angular velocity of the rotor, electrical. rad/s

δ = rotor angle in p.u

t = time, s

3.1.3.3. Stator voltage Equations

The elect-magnetic relationship between the generator terminal voltage, current and flux linkage is set up by stator voltage equations. With the flux transients are neglected in the models the stator voltage is as following:

$$\begin{aligned} u_d &= -\psi_q \omega_r - R_a i_d \\ u_q &= +\psi_d \omega_r - R_a i_q \end{aligned} \quad (3.19)$$

The R_a hasn't been taken into account in the reference models from PSS/E , additionally it is assumed that $X_d'' = X_q''$. While these parameter is adjustable in the newly developed models but with a default value zero of R_a and $X_d'' = X_q''$ by default.

3.1.3.4. Magnetic Saturation

The essential data used to describe the magnetic saturation effect for the rotor and stator iron is contained in *Open-circuit Characteristic Curve*, the OCC for generator. The OCC is gained by plotting the open-circuit stator terminal voltage as a function of the field current. At low level of flux, the linear relationship holds between flux (terminal voltage) and the field current, named "air-gap line". However, when flux gets relatively higher (at high level of terminal voltage), the stator iron get saturated resulting in apparently higher magnetic reluctance. Additional field current is required to drive the flux through the iron. The departure of the OCC line away from the "air-gap line" is an indicate of the degree of iron saturation[21]. The Typical generator open circuit saturation curve is shown below.

The input requirements for characterizing generator saturation for most commercial-grade stability programs are in terms of a parameter called S . The saturation factor is assumed to have the quadratic relation with the input voltage (or flux) as shown in Figure 3.9:

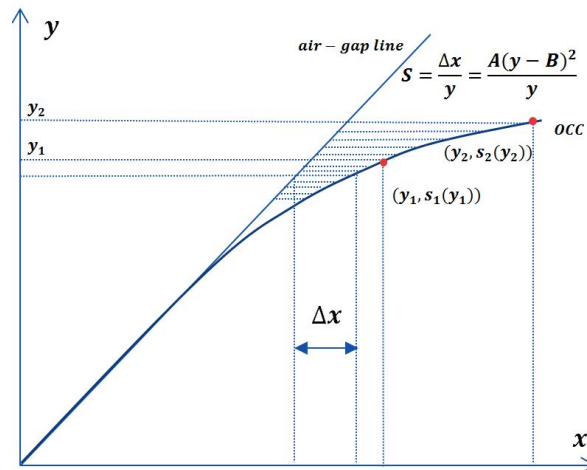


Figure 3.9.: Saturation curve function

The y axis can stand for induced voltage or the flux by the field current represented by x axis, and A and B are such that the points $(1.0, S_{1.0})$ and $(1.2, S_{1.2})$ lie on the curve as shown in Figure 3.9. With the method of undetermined coefficients, the value of A and B can be calculated by solving the equations of the two points $(1.0, S_{1.0})$ and $(1.2, S_{1.2})$. Where $S_{1.0}, S_{1.2}$ are considered to be constant parameter for the generator used to determine certain saturation behavior when the stator terminal voltage is at 1.0 and 1.2 p.u. as in Figure 3.9.

3.1.3.5. Electro-magnetic Equations

The above algebraic equations are common for both salient and solid pole generator. However, the electro-magnetic behavior of them should be modeled differently. In the modeling of salient pole generators, saturation has normally been assumed to occur only in the direct axis. In solid iron rotor turbine generators, saturation can be significant in both the direct and quadrature axes[18]. The following equations corresponding to solid rotor generator, while the equations of Gensal will be discussed later. The generator open-circuit saturation curve is used to determine the saturation factors S . The definition of S has been discussed in 3.1.3.4. The following equations are expressed with standard synchronous machine parameter shown in Table 3.6.

Table 3.6.: Standard Synchronous machine parameters of Genrou

Column	Variable	Description	Unit
1	D	Damping coefficient	pu
2	H	Inertia constant	s
3	r_a	Armature resistance	pu
4	x_l	Leakage reactance	pu
5	x_d	d-axis synchronous reactance	pu
6	x'_d	d-axis transient reactance	pu
7	x''_d	d-axis sub-transient reactance	pu
8	x_q	q-axis synchronous reactance	pu
9	x'_q	q-axis transient reactance	pu
10	x''_q	q-axis sub-transient reactance	pu
11	T'_{d0}	d-axis open circuit transient time constant	s
12	T''_{d0}	d-axis open circuit sub-transient time constant	s
13	T'_{q0}	q-axis open circuit transient time constant	s
14	T''_{q0}	q-axis open circuit sub-transient time constant	s
15	s_{10}	saturation behavior parameter	-
16	s_{12}	saturation behavior parameter	-

$$\dot{E}'_q = \frac{1}{T'_{d0}} (E_{fd} - X_{ad}I_{fd}) \quad (3.20)$$

$$\dot{E}'_d = \frac{1}{T'_{q0}} (-1)(X_{aq}I_{1q}) \quad (3.21)$$

$$\dot{\psi}_{kd} = \frac{1}{T''_{d0}} (E'_q - \psi_{kd} - (X'_d - X_l)i_d) \quad (3.22)$$

$$\dot{\psi}_{kq} = \frac{1}{T''_{q0}} (E'_d - \psi_{kq} + (X'_q - X_l)i_q) \quad (3.23)$$

$$X_{ad}I_{fd} = \frac{(X'_d - X''_d)(X_d - X'_d)}{(X'_d - X_l)^2} [E'_q - \psi_{kd} - i_d(X'_d - X_l)] \\ + i_d(X_d - X'_d) + E'_q + S_e(|\psi''|)\psi''_d \quad (3.24)$$

$$X_{aq}I_{1q} = \frac{(X'_q - X''_q)(X_q - X'_q)}{(X'_q - X_l)^2} [E'_d - \psi_{kq} + i_q(X'_q - X_l)] \\ - i_q(X_q - X'_q) + E'_d - S_e(|\psi''|)\frac{X_q - X_l}{X_d - X_l}\psi''_q \quad (3.25)$$

$$\psi''_d = \frac{E'_q(X''_d - X_l) + \psi_{kd}(X'_d - X''_d)}{X'_d - X_l} \quad (3.26)$$

$$\psi''_q = \frac{-E'_d(X''_q - X_l) - \psi_{kq}(X'_q - X''_q)}{X'_q - X_l} \quad (3.27)$$

$$|\psi''| = \sqrt{(\psi''_d)^2 + (\psi''_q)^2} \quad (3.28)$$

$$\psi_d = \psi''_d - X''_d i_d \quad (3.29)$$

$$\psi_q = \psi''_q - X''_q i_q \quad (3.30)$$

For salient pole generator, there are in total three rotor circuits. In order to represent the model in detail, one state variable is introduced for each of the rotor circuit. Additionally, the magnetic saturation is only account for d axis. The following equations can is the correct representation of its electro-magnetic behavior. The standard parameters is shown in Table3.7.

Table 3.7.: GENSL

Column	Variable	Description	Unit
J	T'_{do}	d-axis transient open-circuit time constant	sec
J+1	T''_{do}	d-axis sub-transient open-circuit time constant	sec
J+2	T''_{qo}	q-axis sub-transient open-circuit time constant	sec
J+3	H	Inertial,H	sec
J+4	D	Speed damping	
J+5	X_d	d-axis reactance	p.u.
J+6	X_q	q-axis reactance	p.u.
J+7	X'_d	d-axis transient reactance	p.u.
J+8	X''_d	d-axis sub-transient reactanc	s
J+9	X_l	leakage reactance	s
J+10	$S(1.0)$	Saturation factor	p.u.
J+11	$S(1.2)$	Saturation factor	p.u.

$$\dot{E}'_q = \frac{1}{T'_{d0}} (E_{fd} - X_{ad}I_{fd}) \quad (3.31)$$

$$\dot{\psi}_{kd} = \frac{1}{T''_{d0}} (E'_q - \psi_{kd} - (X'_d - X_l)i_d) \quad (3.32)$$

$$\dot{\psi}''_q = \frac{-1}{T''_{q0}} (\psi''_q + (X'_q - X''_q)i_q) \quad (3.33)$$

$$X_{ad}I_{fd} = \frac{(X'_d - X''_d)(X_d - X'_d)}{(X'_d - X_l)^2} [E'_q - \psi_{kd} - i_d(X'_d - X_l)] + i_d(X_d - X'_d) + E'_q + S_e(E'_q)E'_q \quad (3.34)$$

$$\psi''_d = \frac{E'_q(X''_d - X_l) + \psi_{kd}(X'_d - X''_d)}{X'_d - X_l} \quad (3.35)$$

$$\psi_d = \psi''_d - X''_d i_d \quad (3.36)$$

$$\psi_q = \psi''_q - X''_q i_q \quad (3.37)$$

3.1.4. Initialization

The solution of the power flow provides network's initial condition including the voltage magnitude, angle, active and reactive power injection at each node. Since no dynamic devices are include during power flow calculation, before switching from steady-state study to dynamic analysis, extra routines are need to set state and algebraic variables to their initial condition inside the dynamic equipment models.

The machines' state variables should be set to satisfy the boundary condition determined by the power flow solution, and initial value of the regulators' states can then be calculated with initial input and output requirement. The overall models relationship can be seen as in Figure3.10. Thus the initialization should start with the components connected directly to the network bus and then the regulators one by one. The power flow solution with the node phasor voltage, active and reactive power will be taken directly from PSS/E.

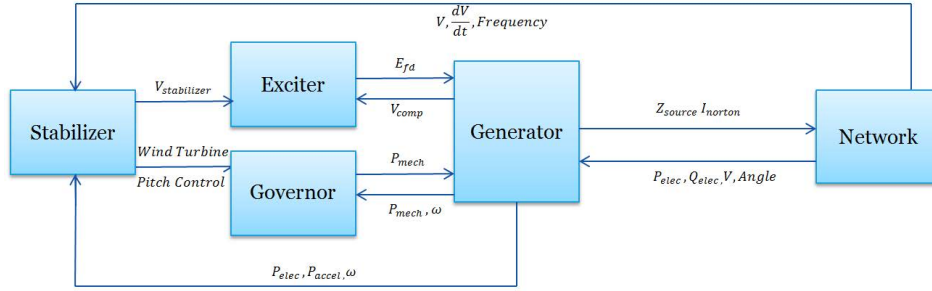


Figure 3.10.: Overview Models and Relationship

3.1.4.1. Genrou

The detail of initialization generator model Genrou and Gensal are discussed here. At steady state all the derivate of state variables are equal to zero, and additionally, all the generator stay synchronous $\omega_r = \omega_{ref} = 1$, the speed derivate is zero $\Delta\omega_r = 0$. For model Genrou, the following equation set can be obtained from Equation3.20-3.30 :

$$\begin{aligned}
 0 &= \frac{P_m - D\Delta\omega_r}{\omega_r} - T_e \\
 0 &= \omega_0 \Delta\omega_r \\
 0 &= \frac{1}{T'_{d0}} (E_{fd} - X_{ad} I_{fd}) \\
 0 &= \frac{1}{T'_{q0}} (-1) (X_{aq} I_{1q}) \\
 0 &= \frac{1}{T''_{d0}} (E'_q - \psi_{kd} - (X'_d - X_l) i_d) \\
 0 &= \frac{1}{T''_{q0}} (E'_d - \psi_{kq} + (X'_q - X_l) i_q)
 \end{aligned} \tag{3.38}$$

which implies that

$$0 = P_{m0} - T_{e0} \quad (3.39)$$

$$0 = E'_{fd0} - X_{ad0}I_{fd0} \quad (3.40)$$

$$0 = X_{aq0}I_{1q0} \quad (3.41)$$

$$0 = E'_{q0} - \psi_{kd0} - (X'_d - X_l)i_{d0} \quad (3.42)$$

$$0 = E'_{d0} - \psi_{kq0} + (X'_q - X_l)i_{q0} \quad (3.43)$$

and we can also have:

$$X_{ad0}I_{fd0} = i_{d0}(X_d - X'_d) + E'_{q0} + S_e|\psi''_0|\psi''_{d0} \quad (3.44)$$

$$0 = -i_{q0}(X_q - X'_q) + E'_{d0} - S_e|\psi''_0|\frac{X_q - X_l}{X_d - X_l}\psi''_{q0} \quad (3.45)$$

By doing proper manipulating, the equation above can be reduced to:

$$\begin{aligned} (X'_q - X''_q)i_{q0} + \psi''_{q0}K &= 0 \\ K &= (1 + S_e|\psi''_0|\frac{X_q - X_l}{X_d - X_l}) \end{aligned} \quad (3.46)$$

With the stator voltage equation described in Section 3.1.3.3, all of initial value the variables can be determined if the magnitude of the flux linkage and current are specified. However the voltage and current phasor $V_t \angle \theta_v$, $I_{t0} \angle \theta_{i0}$ given by the power flow solution is expressed in synchronous reference frame. But the initial rotor position is also an unknown. Proper manipulating should be made in order to solve the equations. By transferring the quantities from dq -axes to network coordinate and applying the Simpson's formulas, the Equation 3.46 can be manipulated to be expressed with δ_0 , and the quantities on synchronous reference frame $I_{t0} \angle \theta_{i0}$, and $\psi''_0 \angle \theta_{p0}$:

$$\begin{aligned} (X'_q - X''_q)(i_{r0} \cos(\delta_0) + i_{i0} \sin(\delta_0)) + K(\psi_{r0} \cos(\delta_0) + \psi_{i0} \sin(\delta_0)) &= 0 \\ \implies & \\ |I_{t0}|(X'_q - X''_q) \cos(\theta_{i0} - \delta_0) + K|\psi''_0| \cos(\theta_{p0} - \delta_0) &= 0 \end{aligned} \quad (3.47)$$

3.1.4.2. Gensal

Following the same procedure, but with Equation from 3.31-3.37, the following can be obtained:

$$\psi''_{q0} + (X'_q - X''_q)i_{q0} = 0 \quad (3.48)$$

$$E'_{q0} = \psi''_{d0} + (X'_d - X''_d)i_{d0} \quad (3.49)$$

Note that E'_{q0} lies on q axis, thus the space position of E'_{q0} indicate the rotor angle of the generator. If we expresses Equation 3.49 as: $(X''_q = X'_d)$

$$E'_{q0} = \psi''_{d0} + (X'_q - X''_q)i_{d0} + (X'_d - X'_q)i_{d0} \quad (3.50)$$

The trick is that the term $(X'_d - X'_q)i_{d0}$ will not influence the space position of E'_{q0} . Then drop this term and add the condition 3.48. Set the magnitude of the term $\psi''_{d0} + (X'_q - X''_q)i_{d0}$ to be F . The initial rotor angle can be found:

$$\begin{aligned} jF &= \\ j\Psi_{\mathbf{dq}0} + (X'_q - X''_q)\mathbf{I}_{\mathbf{dq}0} &= \\ \mathbf{U}_{\mathbf{dq}0} + (R_a + jX''_q)\mathbf{I}_{\mathbf{dq}0} + (X'_q - X''_q)\mathbf{I}_{\mathbf{dq}0} & \end{aligned} \quad (3.51)$$

Applying Park's transformation then :

$$jF \times (\cos(\delta_0) + j \sin(\delta_0)) = jF \angle \delta_0 = |V_{t0}| \angle \theta_{v0} + (R_a + jX''_q)|I_{t0}| \angle \theta_{i0} \quad (3.52)$$

The magnitude of E'_{q0} can be calculated by taking the q stator voltage equation:

$$\begin{aligned} u_{q0} &= \psi''_{d0} - X'_d i_{d0} - R_a i_{q0} \\ \implies & \\ u_{q0} &= E'_{q0} - X'_d i_{d0} + R_a i_{q0} \end{aligned} \quad (3.53)$$

3.1.4.3. Summary

Finally the initialization procedure can be summarized as following:

- With P_{t0}, Q_{t0} , and $V_{t0} \angle \theta_{v0}$ given, the variables of interface connector of electrical elements can be initialized as following. Additionally for generator, the magnitude of the bus terminal current I_{t0} is needed:

$$S_0 = P_{t0} + jQ_{t0} \quad (3.54)$$

$$|I_{t0}| \angle \theta_{i0} = \left(\frac{S_0}{|V_{t0}| \angle \theta_{v0}} \right)^* \quad (3.55)$$

$$v_{r0} = |V_{t0}| \cos(\theta_{v0}) \quad (3.56)$$

$$v_{i0} = |V_{t0}| \sin(\theta_{v0}) \quad (3.57)$$

$$i_{r0} = \Re(I_{t0}) \quad (3.58)$$

$$i_{i0} = \Im(I_{t0}) \quad (3.59)$$

- Since change reference frame will not modify the magnitude of the quantities, with the stator voltage equation expressed in synchronous coordinate, ψ''_0 without saturation can be calculated as: (with the assumption that $X''_d = X''_q$)

$$j|\psi''_0| \angle \theta_{p0} = |V_{t0}| \angle \theta_{v0} + (R_a + jX''_q)|I_{t0}| \angle \theta_{i0} \quad (3.60)$$

- Thus all of the quantities needed to calculate δ_0 is known. For Gensal θ_{p0} is the same as δ_0 , while for Genrou following calculation should be performed:

$$\delta_0 = \arctan\left(\frac{|I_{t0}|(X'_q - X''_q) \cos(\theta'_{p0} - \theta_{i0})}{|I_{t0}|(X'_q - X''_q) \sin(\theta'_{p0} - \theta_{i0}) - K|\psi''_0|} \right) + \theta'_{p0} \quad (3.61)$$

$$\theta'_p = \theta_p + \frac{\pi}{2} \quad (3.62)$$

- With δ_0 known, the dq components of stator voltage and current are given by

$$u_{d0} = |V_{t0}| \sin(\delta_0) \quad (3.63)$$

$$u_{q0} = |V_{t0}| \cos(\delta_0) \quad (3.64)$$

$$i_{d0} = |I_{t0}| \sin(\delta_{i0} + \phi) \quad (3.65)$$

$$i_{q0} = |I_{t0}| \cos(\delta_{i0} + \phi) \quad (3.66)$$

$$\phi = \theta_{v0} - \theta_{i0} \quad (3.67)$$

- The remaining machine quantities are computed by using Equation 3.43, 3.45, 3.20, 3.30:

$$P_{m0} = T_{e0} = P_{t0} + R_a(I_{t0})^2 \quad (3.68)$$

$$E_{fd0} = i_{d0}(X_d - X'_d) + E'_{q0} + S_e |\psi''_0| \psi''_{d0} \quad (3.69)$$

$$(3.70)$$

With P_{m0} and E_{fd0} known, all of the regulators can be properly initialized. Since all the nonelectrical elements are represented by certain transfer function. The initialization is really devices dependently. The detail of each device will be discussed later.

3.2. Nonelectrical Element

3.2.1. Excitation System

As it has been shown in Figure 3.10, the field current of synchronous generator are supplied and automatically adjusted by its excitation system. A suitable control of field current can contribute to effective control of voltage and enhance the stability of power system.

Basically, the excitation system can be modeled by following components: voltage sensor, regulator (amplifier, stabilizing circuit), exciter. The modeling of excitation system utilize the block diagram representation. The input signal to the excitation system is the voltage at generator terminal, the output is the desired field voltage of the generator. Firstly, the sensor filters the signal and compares it with the desired value, then the error is processed and amplified by the regulator. The voltage regulator output provided the desired power level of the exciter which then supply the suitable dc power of the generator.

The voltage sensor can be viewed as a simple transfer block with a time constant T_R which represents rectification and filtering of the synchronous machine terminal voltage, and can be neglected by setting T_R to be zero. The regulator is consist of process and amplifier. The amplifier is represented by a gain and a time constant. Because of power-supply limitation, the state of the amplifier is subjected to "non-windup" limits V_{rmax} and V_{rmin} (or E_{max} and E_{min}). A description of such limits is provided in the appendix. Furthermore, the stabilizing circuit of the excitation system usually utilizes a feedback from output through series transformers. Take the *Laplace* transformation of the transformer equations, one can get the needed transfer block.[6]

Depends on form of excitation system, the exciter could be: DC exciter (self or separately (shunt) excited), AC exciter; rectifiers (controlled or non-controlled). Due to magnetic saturation effect, the output of the exciter is a nonlinear function of the its field current. The

saturation factor is defined as the same way explained in section 3.1.3.4.

The circuit for self and shunt exciter are shown in the Figure 3.11. For field circuit, The separately exciter takes the solid loop, while the self exciter takes the blue dotted current loop. The transfer function of the two circuit are: where R is the field resistance, and L is the

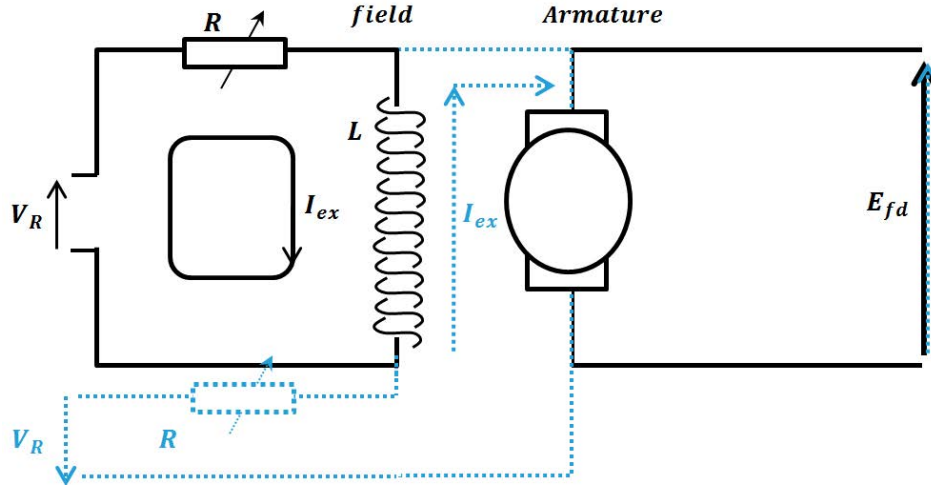


Figure 3.11.: The equivalent circuit for self and shunt exciter

unsaturated inductance. V is the output from voltage regulator.

$$RI_{ex} + L' \frac{I_{ex}}{dt} = E_{fd} \quad (3.71)$$

$$RI_{ex} + L' \frac{I_{ex}}{dt} = E_{fd} + V_c \quad (3.72)$$

where L' is the saturated value of the exciter field inductance, and may be taken for practical purposes to be a function of unsaturated inductance L and the S_e . S_e follows the same definition as it is described in section 3.1.3.4, the only difference is that the input of the function is field voltage E_{fd} as in Equation 3.74. Usually, the field voltage values E_1 and E_2 and the corresponding value $S_e(E_1)$ and $S_e(E_2)$ are used to specify the certain saturation behavior.

$$L' = \frac{L}{1 + S_e} \quad (3.73)$$

where:

$$S_e = \frac{\Delta I}{E_{fd}} \quad (3.74)$$

Rearranging the Equation 3.71:

$$\frac{L}{R} \frac{dE_{fd}}{dt} = \frac{V}{R} - E_{fd} - S_e E_{fd} \quad (3.75)$$

$$\frac{L}{R} \frac{dE_{fd}}{dt} = \frac{V}{R} - \left(1 - \frac{1}{R}\right) E_{fd} - S_e E_{fd} \quad (3.76)$$

By setting :

$$T_e = \frac{L}{R} \quad (3.77)$$

$$K_e = \begin{cases} 1 & \text{for separately exciter} \\ 1 - \frac{1}{R} & \text{for shunt exciter} \end{cases} \quad (3.78)$$

Practically, field resistance R is an adjustable value. The excitation systems might take the advantage of shunt exciter to adjust R in order to reduce V to be zero in steady-state operating conditions. However, the adjustment of the field resistance is assumed to be done for each new pre-event steady-state condition, which means the value of R is considered to be constant during dynamic simulation but the recalculation for its new value is need for each new initial condition.

Normally, the value of R is might vary little around unity, as a result, T_e can be taken as constant, but K_e must be adjusted to an appropriate value each time before switch to dynamic simulation.[9]. The models develop in this thesis can perform the automatic calculation of K_e and also for the relative parameter V_{rmax} and V_{rmin} when the user specified them as zero in the parameter interface. The automatic calculation of K_e is handle by following logic. The logic is the same as it is in PSS/E program:[9]

$$K_e = \frac{V_{rmax}}{10E_{fd}} - S_e(E_{fd}) \quad K_e = 0 \quad (3.79)$$

$$\begin{cases} V_{rmax} = S_e(E_2)E_2 + K_e & K_e > 0, V_{rmax} = 0 \\ V_{rmax} = S_e(E_2)E_2 & K_e \leq 0, V_{rmax} = 0 \end{cases} \quad (3.80)$$

3.2.1.1. IEEE T1, IEEE T2

Model IEEE T1 could represent a dc exciters or alternator exciters, while IEEE T2 usually will be use to model ac exciters. As it has been discussed before, when representing dc exciter systems, the constants K_e, V_{max}, V_{min} , should be entered as zero so that the values will be automatically adjusted. However, when representing alternator rectifier systems, the constant K_e should be set to unity, and normally V_{max}, V_{min} should be set to zero so that their values will be assigned by the routines inside the model.

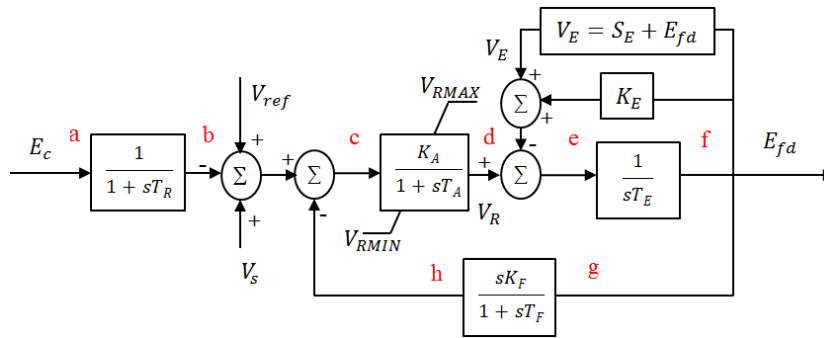


Figure 3.12.: Block Diagram of IEEE T1

Table 3.8.: IEEE T1

Column	Variable	Description	Unit
J	T_R	Voltage sensor time constant	sec
J+1	K_A	Voltage regulator gain	sec
J+2	T_A	Voltage regulator time constant	sec
J+3	V_{RMAX}	Power source upper limit	sec
J+4	V_{RMIN}	Power source lower limit	
J+5	K_E	Exciter equivalent gain	p.u.
J+6	T_E	Exciter equivalent time constant	p.u.
J+7	K_F	Feedback loop equivalent gain	p.u.
J+8	T_F	Feedback loop equivalent time constant	s
J+9	Switch	llallal	s
J+10	E_1	Field voltage value	p.u.
J+11	$S_E(E_1)$	Saturation factor	p.u.
J+12	E_2	Field voltage value	p.u.
J+13	$S_E(E_2)$	Saturation factor	p.u.

IEEE T1 The block diagram for the model is illustrated in Figure 3.12. And the parameter sheet is in Table 3.8.

Each block is initialized by assign a initial value to its state. In order to properly initialize the states, it is needed to apply inverse *Laplace* transform on the given block diagram. For the sake of analysis, the input and output of each block is identified by a letter. After transformation, the following relationship is obtained:

$$a = b + T_R \dot{b} \quad (3.81)$$

$$K_{AC} = d + T_A \dot{d} \quad (3.82)$$

$$e = T_E \dot{f} \quad (3.83)$$

$$K_F \dot{g} = h + T_F \dot{h} \quad (3.84)$$

When it is steady state, the time derivative of all the states are zero, and the output of the exciter should be the same as the field voltage at steady state. Without compensate circuit, the input of the exciter should be the initial value of measured generator bus voltage.

$$a_0 = V_{t0} \quad (3.85)$$

$$a_0 = b_0 \quad (3.86)$$

$$K_{AC0} = d_0 \quad (3.87)$$

$$e_0 = 0 \quad (3.88)$$

$$h_0 = 0 \quad (3.89)$$

$$f_0 = E_{fd0} \quad (3.90)$$

d_0 , the initial regulator output V_{R0} can be calculated as following. And the reference voltage

3. Model Development

is set as

$$V_{R0} = (S_e(E_{fd0}) + 1 + K_E)E_{fd0} \quad (3.91)$$

$$V_{ref} = \frac{V_{R0}}{K_A} - V_{S0} + V_{i0} \quad (3.92)$$

IEEET2 Differs from IEEET1, IEEET2 take the source used for the excitation system stabilizing feedback as proportional to the control element output, while IEEET1 takes a stabilization signal proportional to the main output E_{fd} of the exciter. The block diagram for the IEEET2 is shown in Figure3.13. And the parameter sheet is shown in Table3.9. Following the same procedure, IEEET2 can be initialized by performing following calculation:

Firstly, apply inverse *Laplace* transformation to the transfer block diagram.

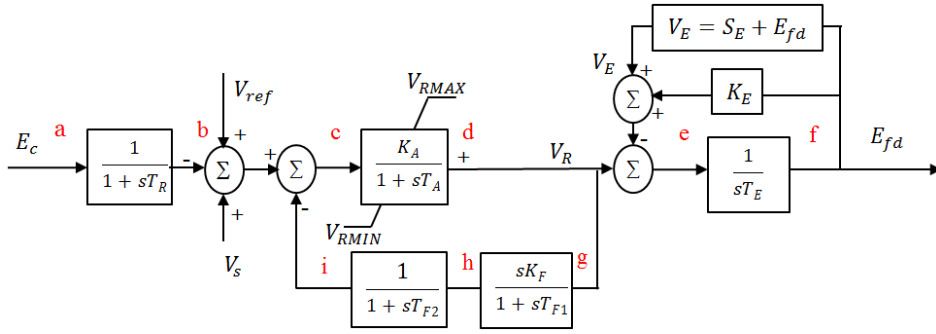


Figure 3.13.: Block Diagram of IEEET2

Table 3.9.: IEEET2

Column	Variable	Description	Unit
J	T_R	Voltage sensor time constant	sec
J+1	K_A	Voltage regulator gain	sec
J+2	T_A	Voltage regulator time constant	sec
J+3	V_{RMAX}	Power source upper limit	sec
J+4	V_{RMIN}	Power source lower limit	sec
J+5	K_E	Exciter equivalent gain	p.u.
J+6	T_E	Exciter equivalent time constant	p.u.
J+7	K_F	Feedback loop equivalent gain	p.u.
J+8	T_{F1}	Feedback loop equivalent time constant	s
J+9	T_{F2}	Feedback loop equivalent time constant	s
J+10	E_1	Field voltage value	p.u.
J+11	$S_E(E_1)$	Saturation factor	p.u.
J+12	E_2	Field voltage value	p.u.
J+13	$S_E(E_2)$	Saturation factor	p.u.

$$a = b + T_R \dot{b} \quad (3.93)$$

$$K_A c = d + T_A \dot{d} \quad (3.94)$$

$$e = T_E \dot{f} \quad (3.95)$$

$$K_F \dot{g} = h + T_{F1} \dot{h} \quad (3.96)$$

$$h = i + T_{F2} \dot{i} \quad (3.97)$$

And then set all derivate to be zero, and take the initial information from generator.

$$a_0 = V_{t0} \quad (3.98)$$

$$a_0 = b_0 \quad (3.99)$$

$$K_A c_0 = d_0 \quad (3.100)$$

$$e_0 = 0 \quad (3.101)$$

$$h_0 = 0 \quad (3.102)$$

$$i_0 = h_0 \quad (3.103)$$

$$f_0 = E_{fd0} \quad (3.104)$$

and thus

$$V_{R0} = (S_e(E_{fd0}) + 1 + K_E) E_{fd0} \quad (3.105)$$

$$V_{ref} = \frac{V_{R0}}{K_A} - V_{S0} + V_{t0} \quad (3.106)$$

3.2.1.2. SEXS, SCR X

When the detail design of the excitation system is not known, one can use SEXS and SCR X to model the general characteristics with a basic representation of the excitation power source (with the gain K , time constant T_e and limits E_{max} and E_{min}) and the transient gain reduction (with time constant T_A and T_B).

SEXS The transfer block of SEXS is quite simple shown as in Figure3.14. The parameter is in Table3.10. The calculation procedure and the results for the state initial values are

$$a + T_A \dot{a} = b + T_B \dot{b} \quad (3.107)$$

$$Kb = c + T_E \dot{c} \quad (3.108)$$

$$a_0 = b_0 \quad (3.109)$$

$$Kb_0 = c_0 \quad (3.110)$$

$$c_0 = E_{fd0} \quad (3.111)$$

$$V_{ref} = \frac{E_{fd0}}{K} - V_{S0} + V_{t0} \quad (3.112)$$

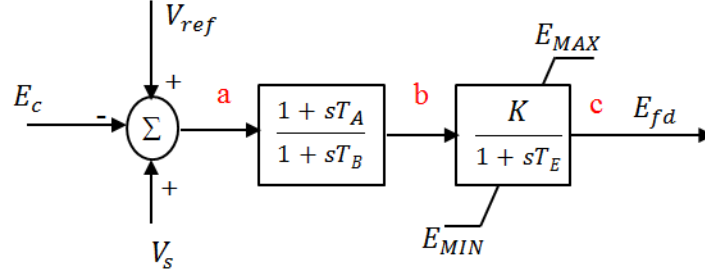


Figure 3.14.: Block Diagram of SEXS

Table 3.10.: SEXS

Column	Variable	Description	Unit
J	T_A/T_B	ratio of Power source transient gain reduction	p.u.
J+1	T_B	Power source transient gain reduction	sec
J+2	K	Power source gain	sec
J+3	T_E	Power source time constant	p.u.
J+4	E_{MAX}	Power source upper limit	sec
J+5	E_{MIN}	Power source lower limit	

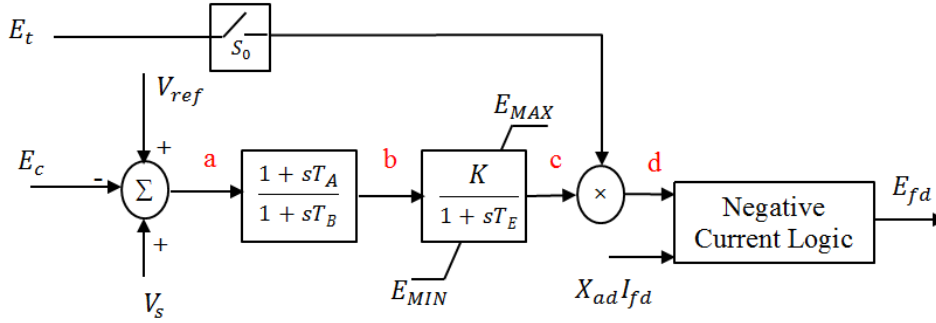


Figure 3.15.: Block Diagram of SCRX

SCRX is a more complex model than SEXS, it represents the system built with rectifier bridges which supplies the dc output directly to the field of the main synchronous generator. Since rectifiers cannot conduct in the reverse direction, the exciter current cannot be negative.

However, under some conditions like system short circuits, the induced current in the generator field winding may be negative which might result in dangerous high voltage if no path can be provided for this current. Thus, a *crowbar* is added to bypass the exciter to discharge the negative field current with the resistor of the *crowbar*, r_c . The circuit is shown in Figure 3.16.

$$E_{fd} = \frac{R_C}{r_{fd}} X_{ad} I_{fd} \quad (3.113)$$

As long as the field current of the synchronous machine becomes negative, the field shorting circuitry is active, E_{fd} is then set to zero, and the field circuit resistance is increased by the amount equal to the value of the crowbar field discharge resistor[20]. A typical value of r_c is

Table 3.11.: SCRX

Column	Variable	Description	Unit
J	T_A/T_B	ratio of Power source transient gain reduction	p.u.
J+1	T_B	Power source transient gain reduction	sec
J+2	K	Power source gain	sec
J+3	T_E	Power source time constant	p.u.
J+4	E_{MAX}	Power source upper limit	sec
J+5	E_{MIN}	Power source lower limit	sec
J+6	C_{switch}	llallal	s
J+7	r_c/r_{fd}	ratio of crowbar and field winding resistor	p.u.

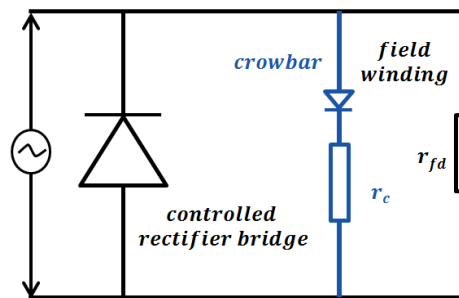


Figure 3.16.: Equivalent circuit of SCRX with *crowbar*

about ten times the field resistance, r_{fd} [9]. A *crowbar* is not the only solution to for providing field shorting circuitry.

In some other excitation system a controlled rectifier with two bridges connected in reverse polarity can allow bidirectional current and the equivalent resistor of the shoring circuit is relatively a small number. Under the condition when negative field current occur, the field circuit is directly shorted.

$$E_{fd} = 0 \tag{3.114}$$

Thus for SCRX, negative current logic can be summarized as:

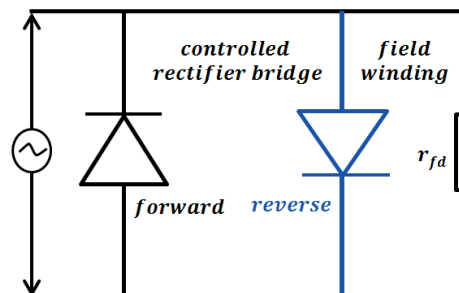


Figure 3.17.: Equivalent circuit of SCRX with reverse polarity bridges

$$E_{fd} = \begin{cases} V_{crow} & X_{ad}I_{fd} < 0 \\ V_R & \text{otherwise} \end{cases} \quad (3.115)$$

$$V_{crow} = \begin{cases} \frac{R_C}{r_{fd}} X_{ad}I_{fd} & \frac{R_C}{r_{fd}} > 0 \\ 0 & \frac{R_C}{r_{fd}} = 0 \end{cases} \quad (3.116)$$

Beside bidirectional current capacity, SCRX is also capable to represent either a bus fed or an independent fed rectifier system. When represent bus fed rectifier system, the regulator output should be proportional to the bus voltage which serves as the source voltage.

It is reasonable to assume that the field current is positive during steady state, since that will only occur under disturbance. If the rectifier system is fed independently, the initialization procedure of SCRX is the same with that of SEXS. However, when representing a bus fed rectifier system, the initialization of the model should be take cared specially. The initial bus fed signal should taken into account. Note that, the bus fed signal should be enable after a small instance of initialization.

$$c_0 = \frac{E_{fd0}}{Et_0} \quad (3.117)$$

$$V_{ref} = \frac{E_{fd0}}{KEt_0} - V_{S0} + V_{t0} \quad (3.118)$$

3.2.2. Power System Stabilizer

When differential the electrical torque of a synchronous machine around an operation point, one can find that it is consist of two components: *synchronizing torque* $T_S\Delta\delta$ in phase with angel deviation and *damping torque* $T_D\Delta\omega$ in phase with speed deviation. Insufficient $T_D\Delta\omega$ results in *oscillatory instability* which is through oscillations of increasing amplitude of rotor angle. To compensate the negative effect on *damping torque* from the excitation systems, power system stabilizer (PSS) is added into the system. The function of PSS is to introduces a supplementary signal into the voltage regulator with proper phase and gain adjustments to produce a component of damping that will be sufficient to cancel the negative effect from the excitors by modifying the normal reference signal of the voltage regulator.

3.2.2.1. IEEEEST

The transfer function of IEEEEST in Figure3.18 consist of a notch filter, two lead-lag blocks and a washout block. The parameter used in the diagram is explained in Table 3.12. By appropriately specifying the time constants of the two lead-lay blocks, desired phase lead can be introduced to that of the input signal. Except the block of wash-out, all of the other block can be bypassed by setting the relative parameters to be zero. The effect of the washout block may be canceled by setting $T_5=T_6=20$.

IEEEEST is a quite general model, a wild range of information can be pick up as input signal, *e.g.*, active power, speed deviation, or voltage magnitude. In *PSS/E* the user should specified the source of the input signal, *e.g.*, which quantity from either generator terminal or a remote bus. However in Dymola, one just need to put a sensor to measure the desired information, and input it to IEEEEST through connector directly.

Initialization procedure of the model are:(None of the limit is hit during the initial state)

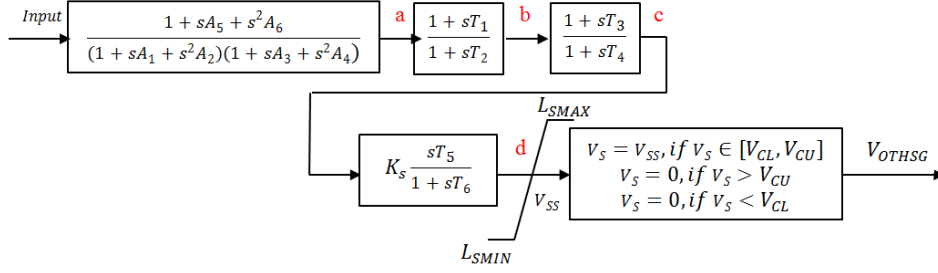


Figure 3.18.: Block Diagram of IEEEEST

Table 3.12.: IEEEEST

Column	Variable	Description	Unit
J	A_1	Filter coefficient	-
J+1	A_2	Filter coefficient	-
J+2	A_3	Filter coefficient	-
J+3	A_4	Filter coefficient	-
J+4	A_5	Filter coefficient	-
J+5	A_6	Filter coefficient	-
J+6	T_1	Lead time constant	sec
J+7	T_2	Lag time constant	sec
J+8	T_3	Lead time constant	sec
J+9	T_4	Lag time constant	sec
J+10	T_5	Washout time constant	sec
J+11	T_6	Washout time constant	sec
J+12	K_S	Stabilizer gain	sec
J+13	L_{SMAX}	Upper limit	p.u.
J+14	L_{SMIN}	Lower limit	p.u.
J+15	V_{CU}	Upper limit(if zero then ignored)	p.u.
J+16	V_{CL}	Lower limit(if zero then ignored)	p.u.

$$a + T_1 \dot{a} = b + T_2 \dot{b} \quad (3.119)$$

$$b + T_3 \dot{b} = c + T_4 \dot{c} \quad (3.120)$$

$$K_S T_5 \dot{c} = d + T_6 \dot{d} \quad (3.121)$$

$$a_0 = P_{t0}, V_{t0} \text{ or } \Delta\omega (= 0) \quad (3.122)$$

$$a_0 = b_0 \quad (3.123)$$

$$b_0 = c_0 \quad (3.124)$$

$$d_0 = 0 \quad (3.125)$$

3.2.2.2. STAB2A

Models STAB2A is designed specially for modeling the stabilizer which take the generator electrical power as supplementary signal. The block diagram and the parameter is provided

3. Model Development

as following. Follow the same procedure of initialization, one can find that the initial value of all of the states are zero.

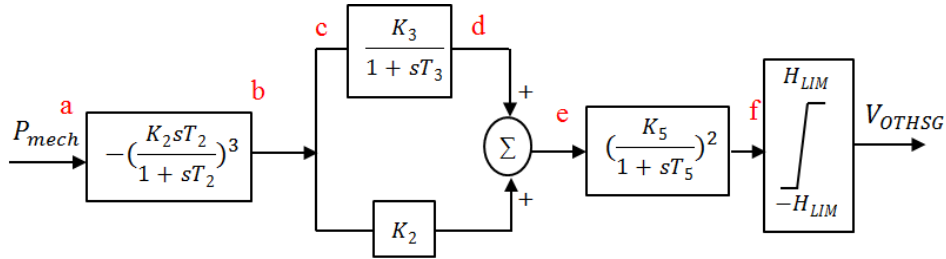


Figure 3.19.: Block Diagram of STAB2A

Table 3.13.: STAB2A

Column	Variable	Description	Unit
J	K_2	Block gain	-
J+1	T_2	Block time constant	sec
J+2	K_3	Block gain	-
J+3	T_3	Block time constant	sec
J+4	K_4	Block gain	-
J+5	K_5	Block gain	-
J+6	T_5	Block time constant	sec
J+7	H_{LIM}	Output limit	p.u.

3.2.3. Governor System

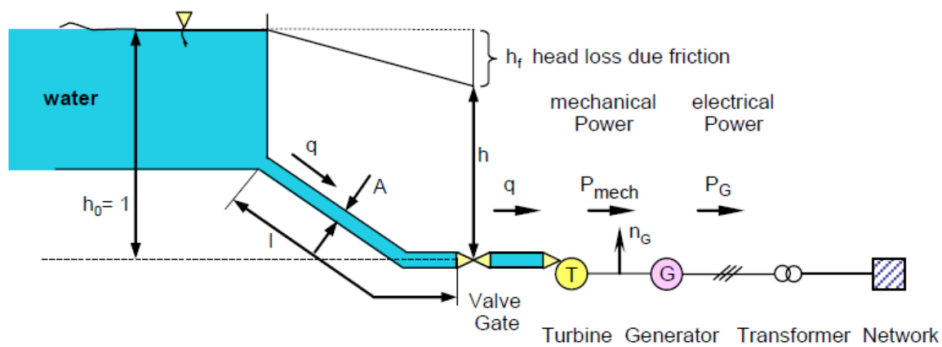


Figure 3.20.: Hydro Turbine Diagram[25]

Figure 3.20 shows the basic diagram for a Hydro turbine unit. The symbol used in the figure is explained as:

- h_0 static head of water column
- l length penstock
- A area penstock
- q urbine flow rate
- h head at the turbine admission
- $n_G(\omega_G)$ generator speed

The The water flow from down from high evaluation contains kinetic energy, the hydraulic turbines convert the energy of water into mechanical power required by synchronous generator. The governor of the turbine control the gates and thus the water flow. Due to the inertia of the water the suddenly opening of the gate will not increase the turbine flow immediately instead it will result in reduction of the water pressure on the turbine thus the decreasing mechanical power output. It is called the water column characteristic. The *acceleration of water column* can accounted as a nonlinear function between the velocity of the water and the change in the head. [6]. The block diagram in Figure 3.21 completely describe the nonlinear water column and turbine characteristic.

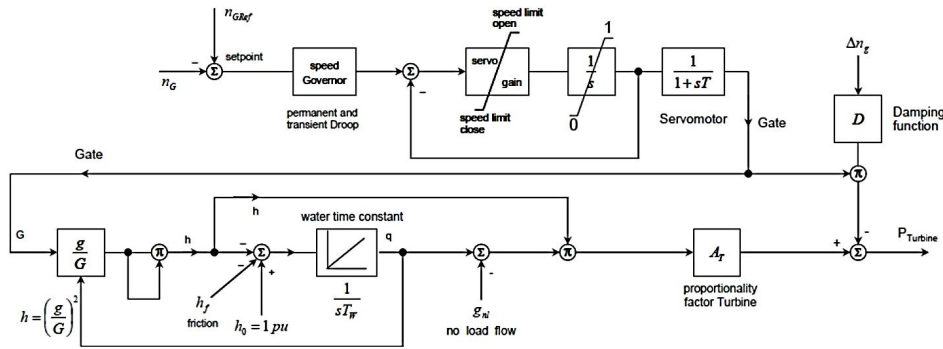


Figure 3.21.: Hydro Turbine Diagram 2[25]

3.2.3.1. HYGOV

The model HYGOV developed in this thesis utilized the diagram shown in Figure ?? which drops the part representing The head loss due to friction h_f . The model is shown as in Fig.3.22:

A_t is the turbine gain which relate the ideal gate opening (from no load to full load) to real gate opening (from fully closed to fully open). Specification of the parameter T_w should be care. By definition, T_w is referred to as the *water starting time*. It represents the time required for a head h to accelerate the water in the pen-stock from stand still to the flow rate q , but it may vary with loading level. But the model parameter T_w is calculated by the definition but using per unit base flows and heads. It is a fixed value for a given turbine-penstock unit. By doing this, the model automatically accounts for dynamic changes in its effective value. For detail of the changing base procedure, one can refer to book [6].

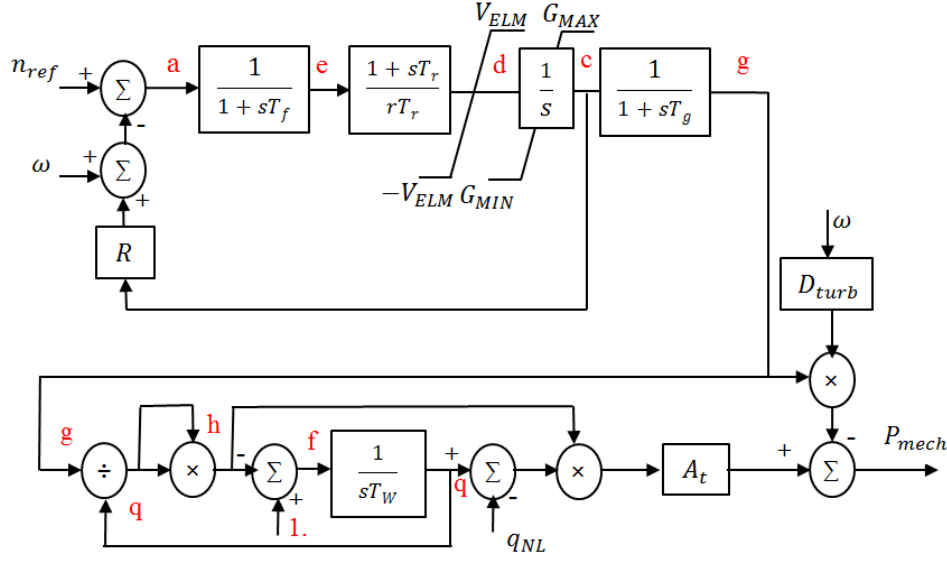


Figure 3.22.: Block Diagram of HYG0V

To initialize the model, following calculation should be performed. Where at steady state, there is no speed deviation and the model should provided the desired mechanical power to the generator which is obtained from initialization of the machine.

$$a = e + T_f \dot{e} \quad (3.126)$$

$$e + T_r \dot{e} = r T_r c \quad (3.127)$$

$$d = \dot{c} \quad (3.128)$$

$$c = g + T_g \dot{g} \quad (3.129)$$

$$\left(\frac{q}{g}\right)^2 = h \quad (3.130)$$

$$1 - h = f \quad (3.131)$$

$$f = T_W \dot{q} \quad (3.132)$$

$$A_t h (q - q_{NL}) - D_{turb} g \omega = P_{mech} \quad (3.133)$$

$$a_0 = 0 \quad (3.134)$$

$$e_0 = 0 \quad (3.135)$$

$$d_0 = 0 \quad (3.136)$$

$$c_0 = g_0 \quad (3.137)$$

$$\left(\frac{q_0}{g_0}\right)^2 = 1 \quad (3.138)$$

$$h_0 = 1 \quad (3.139)$$

$$f_0 = 0 \quad (3.140)$$

$$q_0 = \frac{P_{mech0}}{A_t h} + q_{NL} \quad (3.141)$$

$$n_{ref} = R \left(\frac{P_{mech0}}{A_t h} + q_{NL} \right) \quad (3.142)$$

Table 3.14.: HYG0V

Column	Variable	Description	Unit
J	R	Permanent droop	p.u.
J+1	r	Temporary droop	p.u.
J+2	T_r	Governor time constant	sec
J+3	T_f	Filter time constant	sec
J+4	T_g	Servo time constant	sec
J+5	\pm VELM	Gate velocity limit	p.u.
J+6	G_{MAX}	Maximum gate limit	p.u.
J+7	G_{MIN}	Minimum gate limit	p.u.
J+8	T_W	Water time constant	sec
J+9	A_t	Turbine gain	s
J+10	D_{turb}	Turbine damping	p.u.
J+11	q_{NL}	No load flow	p.u.

3.2.3.2. IEESGO

IEESGO is a simple configuration which can be use to represent both reheat steam turbine or an simplified hydro plant. The block diagram, parameter sheet and initial procedure are:

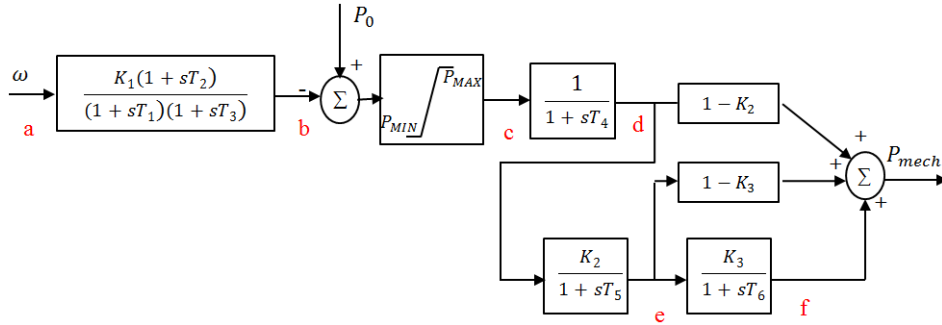


Figure 3.23.: Block Diagram of IEESGO

$$K_1(a + T_2\dot{a}) = (b + T_1\dot{b})(b + T_3\dot{b}) \quad (3.143)$$

$$c = d + T_4\dot{d} \quad (3.144)$$

$$K_2d = e + T_5\dot{e} \quad (3.145)$$

$$K_3e = f + T_6\dot{f} \quad (3.146)$$

$$a_0 = 0 \quad (3.147)$$

$$b_0 = K_1a_0 \quad (3.148)$$

$$c_0 = P_0 \quad (3.149)$$

$$d_0 = c_0 \quad (3.150)$$

$$e_0 = K_2d_0 \quad (3.151)$$

$$f_0 = K_3e_0 \quad (3.152)$$

Table 3.15.: IEESGO

Column	Variable	Description	Unit
J	T_1	Controller lag	Sec
J+1	T_2	Controller lead compensation	Sec
J+2	T_3	Governor lag	sec
J+3	T_4	Delay due to steam inlet volumes associated with steam chest and inlet piping	sec
J+4	T_5	Re-heater delay including hot and cold leads	sec
J+5	T_6	Delay due to IP-LP turbine, crossover pipes, and LP end hoods	p.u.
J+6	K_1	Maximum gate limit	p.u.
J+7	K_2	Minimum gate limit	p.u.
J+8	K_3	Water time constant	sec
J+9	P_{MAX}	Upper power limit	s
J+10	P_{MIN}	lower power limit	p.u.

4

Software implementation

By now the general description necessary mathematical equations of all of the components have been covered and the problem of initialization has been solved. As it has been introduced that the major tasks of implement models in Modelica, a equation-based language are firstly to get correct mathematical equations and secondly to solve the initialization problem of the model. After that, the remaining step is to rewrite all of the equations according to Modelica specification. In the chapter, some examples will be covered to explain the procedures of building a new Modelica class.

All of the model can be built either textually or by block diagram. Among them the electrical element which include phasor variables, *e.g.*, complex voltage, current, power is built integrally by textual Modelica code. On the other hand, the regulators is built by the diagrams. The diagram is established by drag-and-drop simple transfer block from MSL or Dymola PowerSystems library, *e.g.*, integrator, lead-lag, gain, relay etc. The code defining each simple transfer block class is the differential algebraic equation (DAE) set. The user can easily modify the block as he need.

4.1. Developed library

Before starting build individual class for each component, it is better to build a package and name all the needed components in advance. It can give one a brief idea of the number of the model going to be developed and the sort of them. The layout of the package is shown in Fig.4.1. As it has been introduced in last chapter, there are two detail generator models, two kinds of transformers, one composite load, and several regulators including four excitation models, two governors, and two stabilizers are going to be developed.

4.2. Build Class

A class is a Modelica key work use to define certain behavior of the Model. It contains several properties such as parameter, variables, and equations. Each component is an instance of a class whose behavior is defined by the class definition.

4. Software implementation

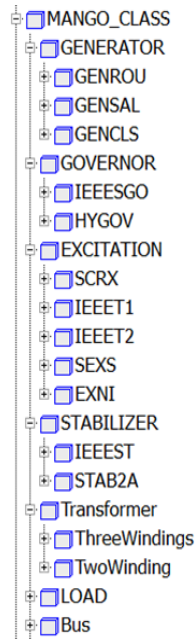


Figure 4.1.: The package going to be developed

4.2.1. Variables Declarations

A brief introduction of variables declarations have been covered in Section 2.2.2. The detail code of declarations for generator parameters is served as an example and shown below. The value for parameter can assigned in code will serve as default value. Without prefix, the declarations are treated as variables by default. If it is needed to view the monitor the value of the variables, it should be specified with keyword **public**. After being translated, the user interface of the model is like in Figure 4.2.

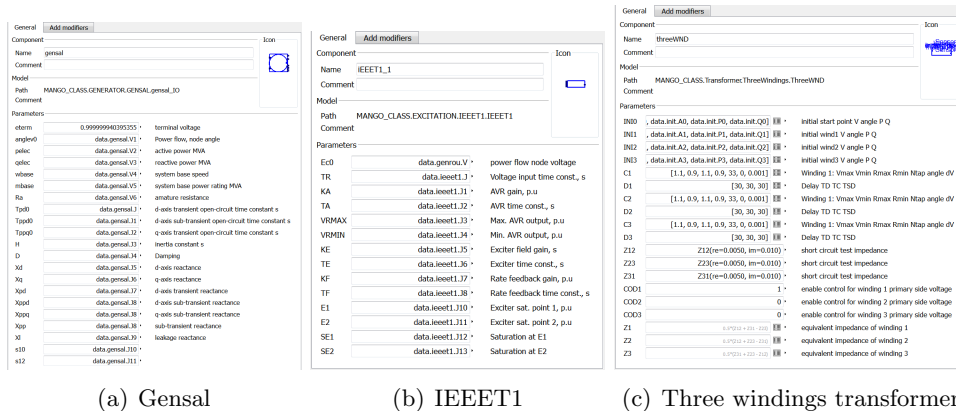


Figure 4.2.: Example of Parameter Dialog

Listing 4.1: Example of Parameter Declaration

```

1  constant Real pi=Modelica.Constants.pi;
2  parameter Real eterm "terminal voltage";
3  parameter Real anglev0 "Power flow, node angle in degree";
4  parameter Real pelec "active power MVA";
5  parameter Real qelec "reactive power MVA";
6  parameter Real wbase = 2*pi*50 "system base speed";
7  parameter Real mbase = 100 "system base power rating MVA";
8  parameter Real Ra = 0 "amature resistance";
9  parameter Real Tpd0 "d-axis transient open-circuit time constant s";
10 parameter Real Tppd0 "d-axis sub-transient open-circuit time constant s";
11 parameter Real Tpq0 "q-axis transient open-circuit time constant s";
12 parameter Real Tppq0 "q-axis transient open-circuit time constant s";
13 parameter Real H "inertia constant s";
14 parameter Real D "Damping";
15 parameter Real Xd "d-axis reactance";
16 parameter Real Xq "q-axis reactance";
17 parameter Real Xpd "d-axis transient reactance";
18 parameter Real Xpq "d-axis transient reactance";
19 parameter Real Xppd "d-axis sub-transient reactance";
20 parameter Real Xppq = Xppd "q-axis sub-transient reactance";
21 parameter Real Xpp = Xppd "sub-transient reactance";
22 parameter Real Xl "leakage reactance";
23 parameter Real s10;
24 parameter Real s12;

```

Listing 4.2: Example of Variable Declaration

```

1  public
2  Real Vt "Bus voltage magnitude";
3  Real anglev "Bus voltage angle";
4  Real I "terminal current magnitude";
5  Real anglei "terminal current angle";
6  // DQ Axis
7  Real id(start=id0);
8  Real iq(start=iq0);
9  Real ud;
10 Real uq;
11 //states
12 Real delta "load angle";
13 Real w "machine speed deviation, p.u.";
14 Real Epd;
15 Real Epq;
16 Real Te;
17 Real PSId;
18 Real PSIQ;
19 Real PSIdk;
20 Real PSIkq;
21 Real PSippd;
22 Real PSippq;
23 Real PSipp;
24 Real XadIfd;
25 Real Delta_XadIfd;
26 Real XaqIlq;
27 Real Delta_XaqIlq;
28 Real P;

```

4. Software implementation

29 Real Q;

The initialization value of the states can be considered to be as parameter, since once it has been defined, the value will not change during the simulation. It is usually the case that, the initial value are calculated by the default equation added for the parameter, and it is unnecessarily to be shown in the dialog. Thus one can type a prefix "protected" to keep those parameter inaccessible from the parameters dialog. The code is shown as:

Listing 4.3: Example of Initial Value Calculation

```
1 protected
2 //Initial value of bus anglev in rad
3 parameter Real anglev_rad=anglev0*pi/180;
4 //Initial value of bus active and reactive power in p.u."
5 parameter Real p0=pelec/mbase;
6 parameter Real q0=qelec/mbase;
7 //Initialize current and voltage components of synchronous reference frame.
8 parameter Real v0=eterm;
9 parameter Real vr0=v0*cos(anglev_rad);
10 parameter Real vi0=v0*sin(anglev_rad);
11 parameter Real ir0=(p0*vr0 + q0*vi0)/(vr0^2 + vi0^2);
12 parameter Real ii0=(p0*vi0 - q0*vr0)/(vr0^2 + vi0^2);
13 parameter Complex Zs(re=Ra, im =Xpp)"Equivation impedance";
14 parameter Complex VT(re=eterm*cos(anglev_rad), im=eterm*sin(anglev_rad));
15 parameter Complex S(re=p0, im=q0);
16 parameter Complex It=conj(S/VT);
17 parameter Complex Is = It + VT/Zs"Equivation current source";
18 //Flux linkage in synchronous reference frame
19 parameter Complex fpp = Zs*Is;
20 parameter Real ang_P=arg(fpp);
21 parameter Real ang_I=arg(It);
22 parameter Real ang_PI=ang_P-ang_I;
23 parameter Real psi = 'abs'(fpp);
24 //Include saturation factor during initialization
25 parameter Real dsat=Se(psi,s10,s12);
26 parameter Real a = psi + psi*dsat*(Xq-Xl)/(Xd-Xl);
27 parameter Real b = 'abs'(It)*(Xpp-Xq);
28 //Initialize rotor angle position
29 parameter Real delta0 = atan(b*cos(ang_PI)/(b*sin(ang_PI)-a)) + ang_P;
30 //Change reference frame
31 parameter Complex DQ_dq(re=cos(delta0),im=-sin(delta0));
32 //Flux linkage in rotor reference fram (dq axes)
33 parameter Complex fpp_dq=fpp*DQ_dq;
34 parameter Complex I_dq=conj(It*DQ_dq);
35 parameter Real PSippq0 = real(fpp_dq);
36 parameter Real PSippd0 = imag(fpp_dq);
37 //Initialize current and voltage components of dq axes.
38 parameter Real iq0 = real(I_dq);
39 parameter Real id0 = imag(I_dq);
40 parameter Real ud0 = -(PSippq0 - Xppq*iq0)- Ra*id0;
41 parameter Real uq0 = PSippd0 - Xppd*id0- Ra*iq0;
42 //Initialize mechanical power and field voltage.
43 parameter Real Pm0=p0;
44 parameter Real Efd0 = dsat*PSippq0+PSippq0+(Xpd-Xpp)*id0+(Xd-Xpd)*id0;
```

4.2.2. Initialization

A dynamic model describes how the states evolve with time. Before starting a simulation, initialization is needed for all the states of the dynamic model. During this phase, the derivatives of the continuous variable and the pre-simulation value of discrete variables are also considered to be algebraic unknown. Initial value of each state can be set explicitly. For an ordinary differential equation $\dot{x} = f(x, t)$, it is convenient to start at steady state. Dymola allows to specify $\dot{x} = 0$ ($\mathbf{der}(\mathbf{x})=0$) as an initial condition, and the solver will automatically calculate the initial value of x . For discrete variables, built-in operator **pre** is used to initialize the states. Or the user can set an initial guess value directly follow the definition of the variable or within initialization section with equation or algorithm. The code is shown as 4.2.2.

Listing 4.4: Connector Initialization equations

```

1 parameter Real vr0(start=v0*cos(anglev_rad));
2 parameter Real vi0(start=v0*sin(anglev_rad));
3 parameter Real ir0(start=(p0*vr0 + q0*vi0)/(vr0^2 + vi0^2));
4 parameter Real ii0(start=(p0*vi0 - q0*vr0)/(vr0^2 + vi0^2));

```

Listing 4.5: Genrou Initialization equations

```

1 initial equation
2   delta=delta0;
3   w=0;
4   der(Epd)=0;
5   der(Epq)=0;
6   der(PSIkd)=0;
7   der(PSIkq)=0;

```

Listing 4.6: Tap Changer Initialization equations

```

1 initial equation
2 pre(up)=false;
3 pre(down)=false;
4 pre(not_up)=false;
5 pre(not_down)=false;
6 pre(Timer1)=0;
7 pre(Timer2)=0;
8 pre(Timer3)=0;
9 pre(Timer4)=0;
10 pre(x1)=0;
11 pre(x2)=false;
12 pre(x3)=0;
13 pre(x4)=false;
14 pre(count1)=0;
15 pre(count2)=0;

```

Listing 4.7: IEEE T2 Initialization equations

```

1 initial algorithm
2 VT0:=Ec0;

```

4. Software implementation

```
3 Efd0:=EFD0;
4 SE_Efd0:=SE(Efd0,SE1,SE2,E1,E2);
5 ini_0:=ini0(VRMAX,KE,E2,SE2,Efd0,SE_Efd0);
6 KE0:=ini_0[1,1];
7 VRMAX0:=ini_0[2,1];
8 VRMIN0:=-ini_0[2,1];
9 VR0:= Efd0*(KE0 + SE_Efd0);
10 VREF:= VR0/KA+VT0+Vs.a0;
```

4.2.3. Equations

The essential feature of Modelica language that the modeler can use physical equations instead of assignments to define the behavior of the class. Thus the order of equations and which of the variables are located on the left or right hand side of the equality sign is irrelevant. The equations is defined under the **equation** keyword. It contains the original equations or connection between connectors. But if needed, a sequence of assignments can also be defined under the **algorithms** keyword.

4.2.3.1. Equality Equations

Since all of the necessary equations which rules the electro-magnetic behavior of the generator has been established in last chapter, the tasks remained are just to type those equations in Modelica code which basically is in the same form as the equations set itself shown as below. Additionally, the equation which describe the sending and receiving end voltage relation of two winding transformer with a complex tap ratio is also shown on the bottom of the code below.

Listing 4.8: Differential equations of Genrou

```
1 equation
2 der(Epq) = (1/Tpd0)*(E_fd -XadIfd);
3 der(Epd) = (1/Tpq0)*(-1)*(XaqIlq);
4 der(PSIkd) =(1/Tppd0)*(Epq-PSIkd-(Xpd-Xl)*id);
5 der(PSIkq) =(1/Tppq0)*(Epd-PSIkq+(Xpq-Xl)*iq);
6 der(w) =(((Pm-D*w)/(w+1)-Te))/(2*H);
7 der(delta)=wbase*(w);
```

Listing 4.9: Algebraic equations of Genrou

```
1 Te = PSId*iq - PSIkq*id;
2 PSId = PSIppd - Xppd*id;
3 PSIkq = PSIppq - Xppq*iq;
4
5 PSIppd=Epq*K3d+PSIkd*K4d;
6 PSIppq=-Epd*K3q-PSIkq*K4q;
7 PSIpp=sqrt(PSIppd*PSIppd+PSIppq*PSIppq);
8
9 ud = -PSIkq*(w+1)- Ra*id;
10 uq = PSId*(w+1)- Ra*iq;
```

```

11  anglev =atan2(p.vi, p.vr);
12  Vt = sqrt(p.vr^2 + p.vi^2);
13  anglei =atan2(p.ii, p.ir);
14  I = sqrt(p.ii^2 + p.ir^2);
15
16  Delta_XadIfd = Se(PSIpp,s10,s12)*PSIppd;
17  Delta_XaqIlq= Se(PSIpp,s10,s12)*PSIppq*(Xq-Xl)/(Xd-Xl);
18  XadIfd=K1d*(Epq-PSIkq-(Xpd-Xl)*id)+Epq+id*(Xd-Xpd)+Delta_XadIfd;
19  XaqIlq=K1q*(Epd-PSIkq+(Xpq-Xl)*iq)+Epd-iq*(Xq-Xpq)-Delta_XaqIlq;
20
21          //Equations of the connector variables
22          //Interface with other electrical elements
23
24  [p.ir;p.ii]=-[sin(delta),cos(delta);-cos(delta),sin(delta)]*[id;iq];
25  [p.vr;p.vi]=[sin(delta),cos(delta);-cos(delta),sin(delta)]*[ud;uq];
26  -P=p.vr*p.ir + p.vi*p.ii;
27  -Q=p.vi*p.ir - p.vr*p.ii;
28
29          //Equations of the connector variables
30          //Interface with nelectrical elements (regulators)
31
32  dwr=w;
33  Ec=Vt;
34  Pe=P;

```

Listing 4.10: Two winding transformer equation

```

1  equation
2  connect(phaseship1.n, P2);
3  connect(pwLine1.n, phaseship1.p);
4  connect(pwLine1.p, P1)
5          // equation of pwLine
6  equation
7  R*(n.ir - G*n.vr + B*n.vi) - X*(n.ii - B*n.vr - G*n.vi)=n.vr - p.vr;
8  R*(n.ii - B*n.vr - G*n.vi) + X*(n.ir - G*n.vr + B*n.vi)=n.vi - p.vi;
9  R*(p.ir - G*p.vr + B*p.vi) - X*(p.ii - B*p.vr - G*p.vi)=p.vr - n.vr;
10 R*(p.ii - B*p.vr - G*p.vi) + X*(p.ir - G*p.vr + B*p.vi)=p.vi
11          // equation for phaseship
12 equation
13  [p.ir;p.ii]=-[sin(A),cos(A);-cos(A),sin(A)]* [n.ir;n.ii];
14  [p.vr;p.vi]=[sin(A),cos(A);-cos(A),sin(A)]*[n.vr;n.vi];

```

4.2.3.2. Conditional Equations

The conditional equations could when if-equations or when-equation. If-equation state that whenever the condition is satisfied the equation is active otherwise is inactive. On the other hand, the equation under when condition is active only if the value of the condition goes from false to true.

When-equation is crucial expression when describe the behavior relative with discrete time variables whose value will only been changed when there is an event. There are three types of variables, *Boolean*, *Integer*, *String*, are discrete variable by natural. The type *Real* could be also specified as discrete variable with prefix **discrete**. They are used wildly when model

4. Software implementation

tap changer of the three winding transformer.

The logic for tap changer action has already been established in Section???. But when adopting those equations, some modification should be made. Because the measured signal from controlled bus is an condition who trigger the up or down movement, while the movement will then effect the voltage of the controlled bus. However, in Modelica the solver will evaluate equations using "candidate solutions" as part of the simulation process. Then the tool will detect an inconsistency in the system of equations, because the condition is effected by the equation inside the when condition. Dymola will complain about it and the simulation fails. To solve this problem, one can add a **pre** operator around the condition in the when clause. The code which describe the logic for tap changer is shown below:

Listing 4.11: logic for decision move up including the regulator delay

```
1 T1=tau;//regulator delay time
2 //if input is less then the lower limit by dV then up is true
3 up = (Vmin-u)>dV;
4 not_up = not up;
5 //detect rising edge of up
6 when edge(up) then
7 //record the time when decision is up
8 Timer1= time;
9 end when;
10 //detect falling edge of up
11 when edge(not_up) then
12 //record the time when voltage return
13 //to the control bank from lower side
14 Timer2=time;
15 end when;
16 //if the voltage only stay within the band less than tau
17 when (Timer2-Timer1)<tau and Timer2>Timer1 then
18 //then it should only wait for another tau-(Timer2-Timer1)
19   reinit(T1, tau-(Timer2-Timer1));
20 end when;
21 //if time elapse longer than tau, the motor will act every TC
22 x1=integer((time - Timer1-T1)/TC);
23 x2=x1>pre(x1) and x1>0;
24 // Add operator "pre" if necessary
25 when pre(x2) and pre(up) then
26 count1 = pre(count1)+1;
27 end when;
```

Listing 4.12: Logic for tap motor

```
1 //following logic detect the direction of each action
2 when change(count1) or change(count2) then
3   if change(count1) then
4     s=1;
5   elseif change(count2) then
6     s=-1;
7   else
8     s=0;
9   end if;
```

```

10 m=pre(m)+s*dtap;//change by tap step size
11 end when;
12 //Keep tap ratio stay within the limit
13 if m>Rmax then
14     y=Rmax;
15 elseif m<Rmin then
16     y=Rmin;
17 else
18     y=m;
19 end if;

```

4.2.3.3. Wind up and Nonwind up limit

If-condition is very useful when describe the saturation behavior of the integrator which is wildly used within the regulators diagram. When describing Non-Windup limit, not only the output of the integrator should be set to the saturated value but also the derivative of the output should be set to be zero. The code is attached here.

Listing 4.13: Single time constant block- Non Windup Limiter

```

1 class NonWindupLag
2   input PowerSystems.Connectors.ImPin p1
3   output PowerSystems.Connectors.ImPin n1
4   parameter Real Ymin "Min limit";
5   parameter Real Ymax "Max limit";
6   parameter Real K "Gain";
7   parameter Real T "Lag Time Constant";
8   parameter Real nStartValue "Output start value";
9
10  initial equation
11    n1=nStartValue;
12  equation
13
14    if n1<Ymin and der(n1) < 0 then
15      der(n1)=0;
16
17    elseif n1 >Ymax and der(n1) > 0 then
18      der(n1)=0;
19
20    else
21      der(n1)=-1*n1/T + K*p1/T;
22    end if;
23
24 end NonWindupLag;

```

On the other hand, one just need to add a limiter (from Dymola PowerSystems library) after the integrator to model a Wind-up limiter. The code is also attached here:

Listing 4.14: Simple Limiter

```

1 class ImLimiter "Limit the range of a signal"
2   input PowerSystems.Connectors.ImPin p1
3   output PowerSystems.Connectors.ImPin n1

```

4. Software implementation

```
4 parameter Real Ymin "Min limit";
5 parameter Real Ymax "Max limit";
6 equation
7   n1=if p1 > Ymax then Ymax elseif p1 < Ymin then Ymin else p1;
8 end ImLimiter;
```

4.2.3.4. User Defined Function

Sometimes, equations are very complex, e.g., the saturation behavior of generator, thus a saturation function is built explicitly with flux or field voltage as input, and the saturation factor as output. The function is called when the value of the factor is needed. The idea of the calculation of saturation factor has been explained in Section 3.1.3.4. Since the parameters used for generator and exciter saturation are different, the calculation method is different although both of them base on calculate the undetermined coefficients of function with two points lies on the saturation curve. The code for the function is shown as:

Listing 4.15: Generator Saturation Function

```
1 function Se
2 extends Modelica.Icons.Function;
3 input Real u "Psipp";
4 input Real s10;
5 input Real s12;
6 output Real sys;
7 protected
8   parameter Real a= sqrt(s10/(s12*1.2));
9   parameter Real A= 1.2-(1-1.2)/(a-1);
10  parameter Real B=1.2*s12*(a-1)^2/(1-1.2)^2;
11 algorithm
12 if (u == 0.0 or s10 <= 0.0) then
13   sys:= 0;
14 else
15   if u <= A then
16     sys:= 0;
17   else
18     sys:= (B*(u - A) ^ 2/u);
19   end if;
20 end if;
21 end Se;
```

Listing 4.16: Exciter Saturation Function

```
1 function SE
2 extends Modelica.Icons.Function;
3 input Real u "Psipp";
4 input Real SE1;
5 input Real SE2;
6 input Real E1;
7 input Real E2;
8 output Real sys;
9 protected
10  parameter Real a=sqrt(SE1*E1/(SE2*E2));
11  parameter Real A=E2-(E1-E2)/(a-1);
```

```
12   parameter Real B=SE2*E2*(a-1)^2/(E1-E2)^2;
13 algorithm
14 if (SE1 == 0.0 or u <= 0.0) then
15   sys:= 0.0;
16 else
17   if u <= A then
18     sys:= 0.0;
19   else
20     sys:= B*(u - A) ^ 2/u;
21   end if;
22 end if;
23 end SE;
```


5

Validation Results

The validation procedure and the analysis of the results are described in the chapter. The performance of the implemented models are firstly validated in small scale power system models. Validation starts with the main dynamic device generator. After that, the regulators of synchronous machine are then added. After the performance of all the components have been proved, further investigation are performed with two Norwegian real subsystem models.

The transient behavior of the dynamic components are of prime interest. The frequency and the system base used in all test system are the same which are 50HZ and 100MVA respectively. While the simulation time scale and the applied perturbations are different for different studies. The setup of the test systems and the applied disturbance are explained in turns. The steady state and dynamic parameters used in the test systems are provided in Appledix. Note that the validation the performance of under load tap changer (ULTC) is not carried out in this thesis due to the limitation of PSS/E capacity. Currently, they has canceled the function of extended simulation in PSS/E which supposed to be used for performing the simulation of system which contains a ULTC. As a result, the validation of ULTC will be carried out in the future against other software. Since the modeling method is general and can be modified easily.

5.1. Small Scale Power System Tests

The basic scenario used for study the transient performance of generator and its regulator is shown in the diagram 5.1 where the generator is delivery power through one transmission line to the load and then through two parallel lines to a large system in remote. The power flow results are shown in Table5.1. The unit of the angle is degree, all of the other value are expressed in per unit on system base. Since each generator and its regulators are on machine base, there is needed for changing base. This is achieved by explicitly introducing the bus for changing base at the terminal of generators. Note that positive sign is used when the power is injected into the bus node.

The power flow results are picked up from PSS/E. As it has been described in section 3.1.4, the generator terminal quantities such as the phasor of voltage and apparent power served as input variables for initialization calculation. The accuracy of the power flow results have a direct affect on the correctness of the initialization. The more the complexity of the system, the higher the tolerance of power flow input values are required.

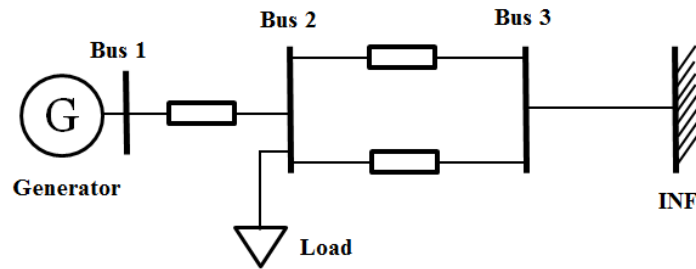


Figure 5.1.: Test system diagram

Table 5.1.: Power Flow Result

Bus	Voltage	Angle	Active power	Reactive power
1	0.999999	4.0463	0.399989	0.054165
3	1	-3.00563e-7	0.100184	0.080064

Validation results of the components should demonstrate both the correctness of the initialization and the transient behavior of the dynamic components under disturbance. In order to check the initialization results, all of the system are running under stationary state for two seconds. After that, different perturbations are applied to force the systems into transient. The transient disturbances are illustrated in Figure 5.2(a) to 5.2(d), they could be a three phase to ground fault in the middle of the transmission lines or at the terminal bus of generator. One can also apply load variational or a step change of regulator's reference value to the system. All of the perturbation were modeled in the same way as they are in PSS/E.

Within in a period of a few seconds after the disturbance, the system might suffer from large excursions of generator rotor angles, power flows, bus voltages, and other system variables. The dynamic behavior of the power system during this time range related to the dynamic introduced by the rotor inertia storage energy, the field electromagnetic transients, and the controls of generators. Thus the signal such as generator speed deviation $\Delta\omega$, the bus voltage magnitude, and the injected active and reactive power P and Q of generator are recored and compared.

When there exist regulators in the system, the output quantities of those equipments are compared and analyzed. Specially, the limit of regulation should be broken if there exist one, so that one can make sure that the system react in the same way when regulators hits their limits. The simulations were continuing until the transient period of the system dies out completely in order to make sure that when transient has died out, the test systems built in Dymola and PSS/E can return to the same equivalent operation point. But in most of the cases, the new steady state is achieved around 20s, thus the simulation time length could be set to 20s. The figures attached in this chapter were shown with zoom in action where the time length in the figures might be less than 20s.

Since in PSS/E, except generator, all the dynamic devices can not be tested individually, thus different components are combined and tested in groups. Certain perturbation from Figure 5.2 are applied to active the specific functions of the components such as the saturation function

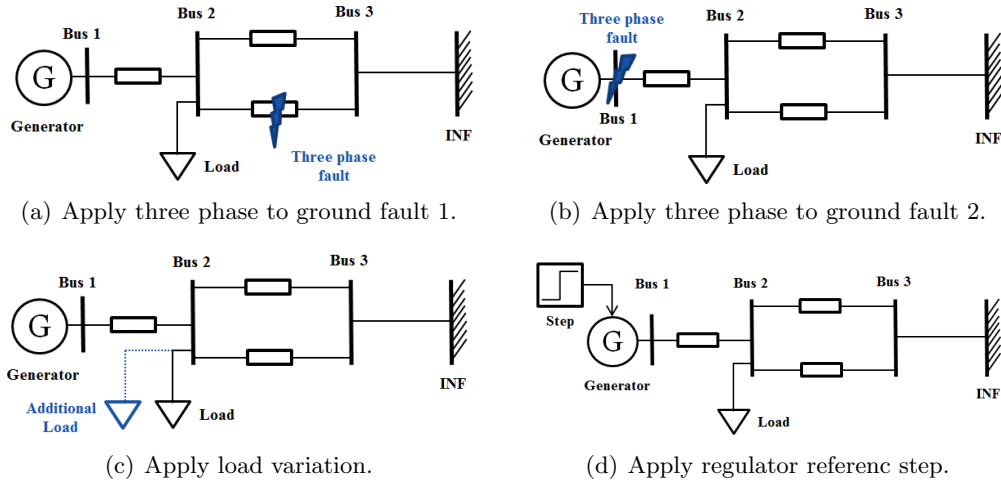


Figure 5.2.: Different perturbation cases.

Table 5.2.: Combination of Dynamic Devices

No.	Dynamic Device
1	Genrou
2	Gensal
3	Genrou+IEEEET2+IEESGO
4	Gensal+SCRX+HYGOV
5	Gensal+IEEEEST+SEXS
6	Genrou+IEEEET1+STAB2A+LFDRL

of detail exciter model (IEEEET1 and IEEEET2), the negative current logic of SCRX, and the no-windup and windup limits within HYGOV. The combinations are summarized as in Table5.2. For the sake of controlling variables, the setting of all simulations performed in PSS/E and Dymola are chosen to be as similar as possible. The settings are listed as below, beside that all the variables which were picked up to be recorded and compared are also listed out. The error in percentage the normal distribution of the absolute difference are also plotted. But due to the limit of the space, only the plots of error for 3_{rd} case are attached in this chapter.

Dynamic setting in PSS/E version 33

Acceleration: 0.5

Network solution iterations: 25

Tolerance: 0.0001

Island frequency acceleration: 1.000

Island frequency tolerance: 0.0005

Time step: 5ms

Frequency filter: 0.0333

Delta threshold intermediate: 0.04

Delta threshold Island frequency: 0.016667

5. Validation Results

Network frequency dependence: off
Plot every time step

Dynamic setting in Dymola (1):

Algorithm: *Rkfix2*
Tolerance: 0.0001
Time step: *5ms*
Output interval length: 0.005
Number of output interval: 5000

Dynamic setting in Dymola (2):

Algorithm: *Rkfix4*
Tolerance: 0.0001
Time step: *5ms*
Number of output interval: 6000

Variables to be recorded:

Quantities at generator terminal:

speed deviation ω
terminal voltage V_t
generated active P_{gen}
generated reactive power Q_{gen} .

Quantities at load bus:

the magnitude of bus voltage θ_{load}
the angle of bus voltage V
consumed active power P_{load}
consumed reactive power Q_{load} .

output of regulators.

governor output, the mechanical power of generator P_m .
exciter output, the field voltage of generator E_{fd} .
stabilizer output $V_{oth.sg}$.

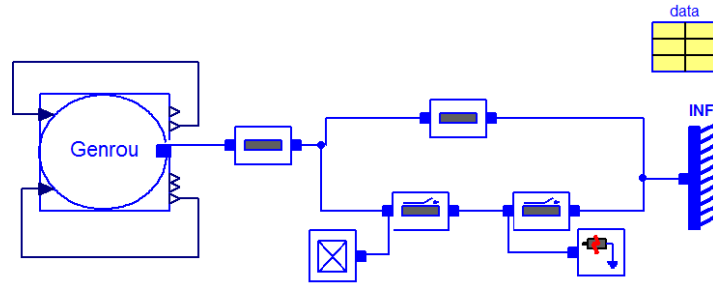


Figure 5.3.: Diagram in Dymola of the 1_{st} test system.

Table 5.3.: Initialization

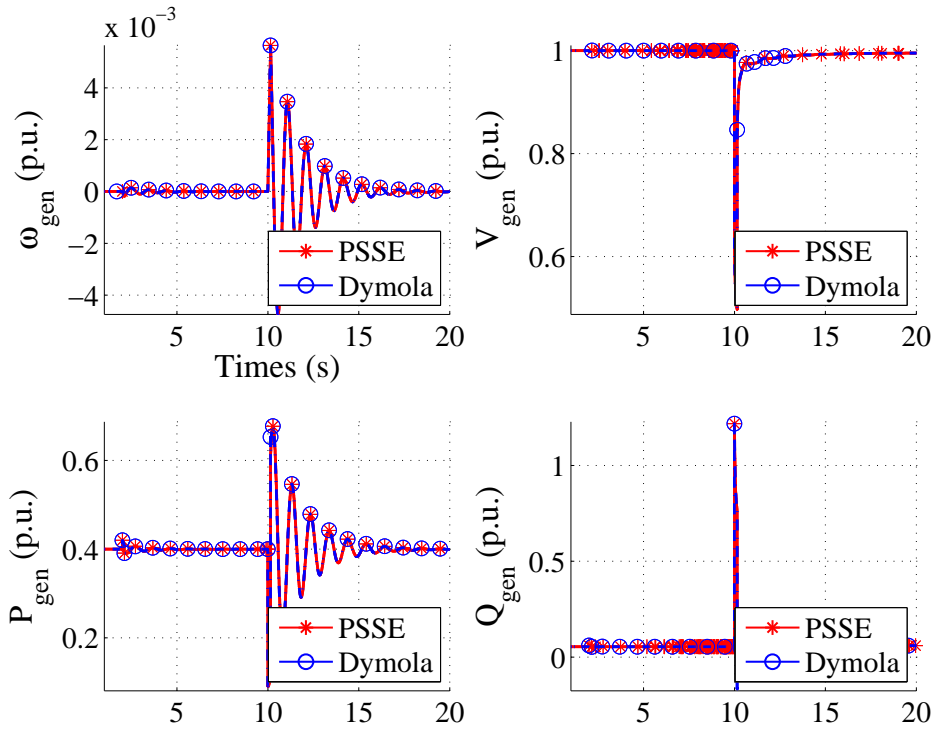
No.	Quantities	PSS/E	Dymola
1	EFD0	1.327474832534790	1.327474594116211
2	Pm0	0.4	0.4
3	ID0	0.2169	0.216949
4	IQ0	0.3404	0.340392
5	Angle	28.85	28.846

5.1.1. Validation of Genrou

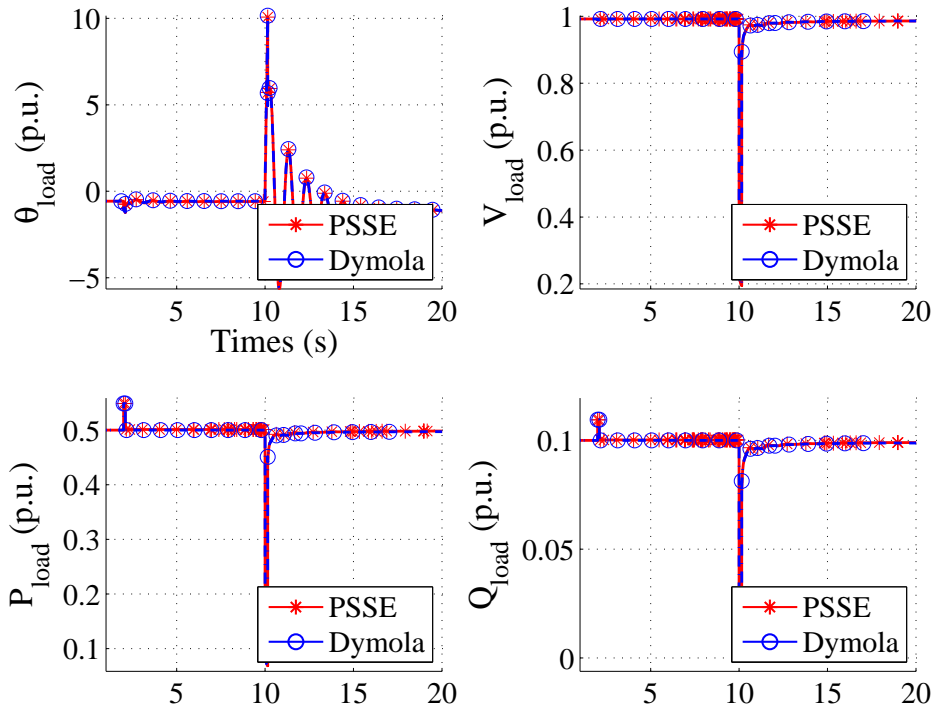
In this section, validation results of round role generator model were analysis. Below are the diagram for the test used in Dymola and a list of the simulations performed during testing. Since the initialization of generator model is the start point of dynamic simulation, the accuracy of the initial values should be check carefully. The results of initialization were recorded in Table5.3.

List of simulations:

1. running under steady state for 2 seconds.
2. Vary the system load with constant P/Q ratio Figure5.2(c).
3. After 0.1s later, the load was restored to its original value .
4. Run simulation to 10s.
5. Apply three phase to ground fault as shown in Figure5.2(a).
6. After 0.15s later, the fault was cleared by tripping the line.
7. Run simulation until 20 seconds.



(a) Generator terminal quantities.



(b) Load bus quantities.

Figure 5.4.: Validation result of 1_{st} test system (1).

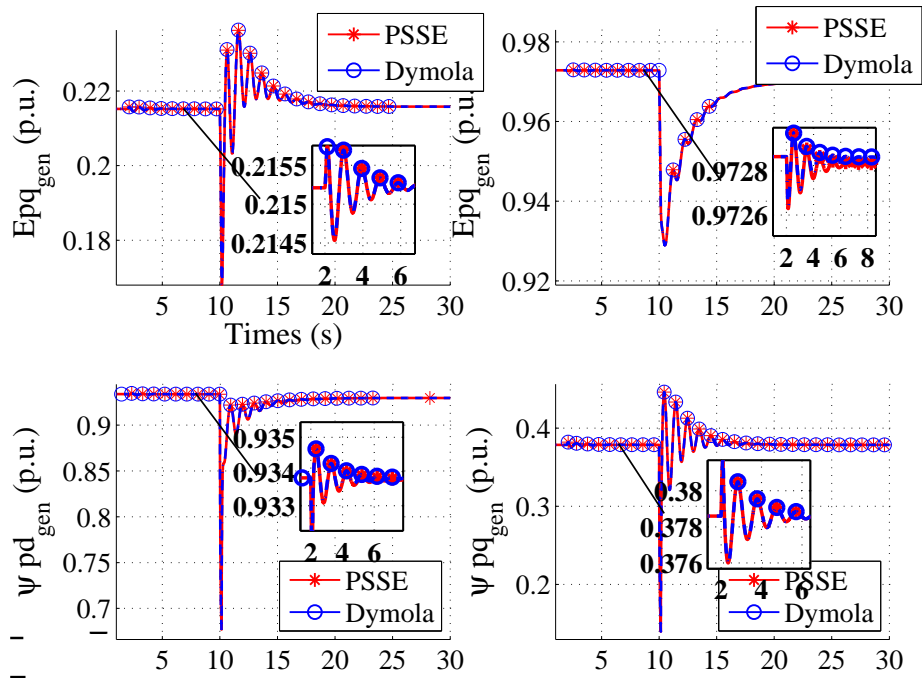
From the results we can see that, two software agreed with each other during initialization state. The error were less than the set tolerance. After that, the dynamic simulation was performed, and the results were shown as in Figure 5.4 to 5.5(a). Beside the quantities at generator and the load bus, the values of the sub-transient flux linkages and transient voltages were recored and compared. It is significantly that the performances of the Modelica models agreed with the PSS/E reference models in high degree.

From the figures we can see that, during steady state, the load consumption are supplied partly by the generator and partly by the infinite bus. When the additional load was added into the system, there occurred a negative mismatch between the mechanical and consumed electrical power which forced the generator to slow down. But since the equivalent impedance at the load bus was reduced, thus the field current of the generator merged instantaneously.

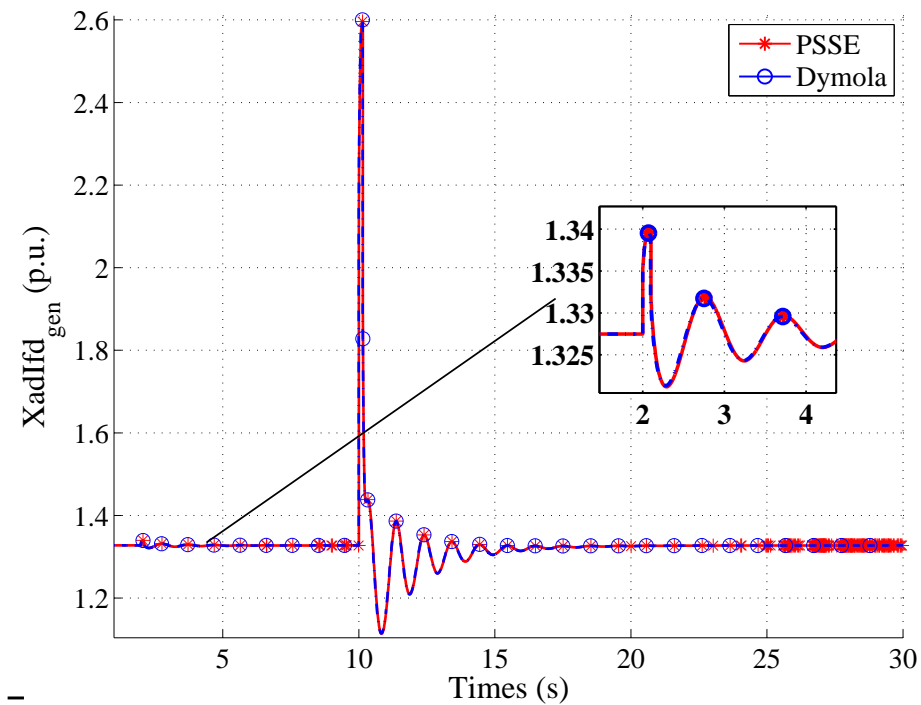
As there were neither exciter nor governor equipped in the generator unit, the field voltage and mechanical power remained their initial value during the whole simulation. Due to the infinite inertia of the remote system, the transient of the generator would die out and return to its steady state. Follow the first perturbation, at 10s the three phase fault was applied, due to the low impedance of the fault, other electrical components were short circuited which are reflected as magnitude of all bus voltage reduced abruptly. The calculated generator reactive power was higher which attempted to maintain the voltage level of the system.

Since the magnetic fluxes in the air-gap can not change their value instantaneously, the resultant equivalent electrical force remained the same value at the instance when the fault was applied. The short circuit current from the generator flowed through line impedance to the fault. Because of the low impedance of the transmission system, the value of short circuit current was very high compared to normal operation. This is shown in Figure 5.11(a), where there existed a surge of generator current directly followed the fault.

The electrical torque kept its pre-disturbance value as well as the power. Due to the low resistance of the post fault system, the delivered active power dropped significantly, hence there exist a unbalance between the generation and consumption. The result was increasing of the generator speed shown in Figure 5.4(a).



(a) Generator transient and subtransient electric voltage and flux.



(b) Generator terminal current.

Figure 5.5.: Validation result of 1_{st} test system (2) .

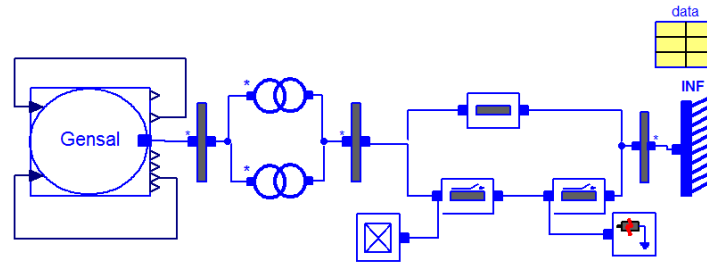


Figure 5.6.: Diagram in Dymola of the 2_{nd} test system.

5.1.2. Validation of Gensal

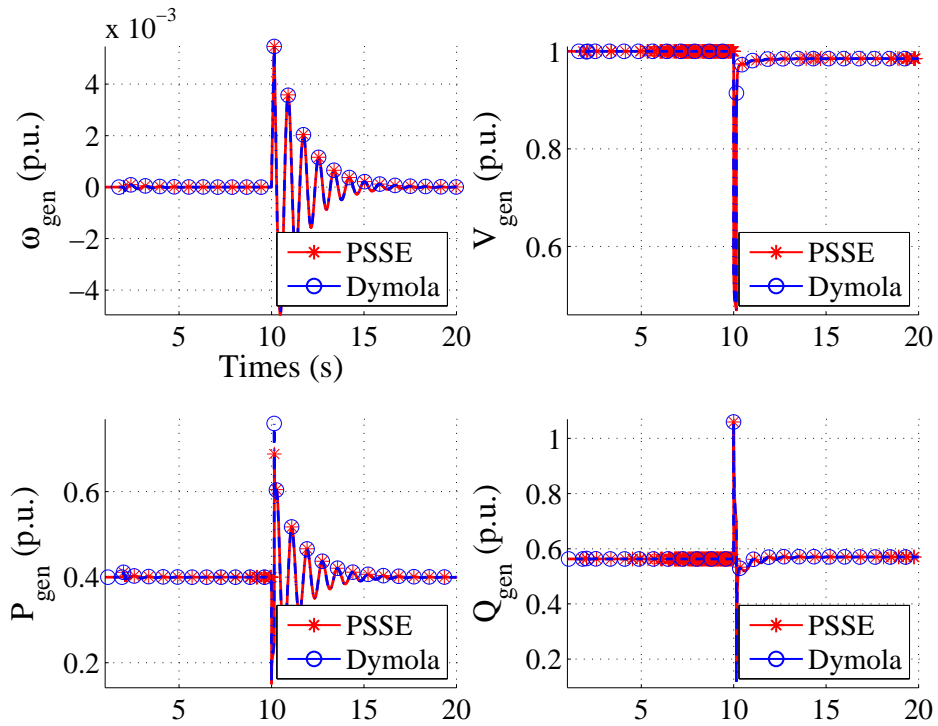
The validation of Gensal model has been proved by the same test system used when testing Genrou. For the sake of saving space the plots were not shown here. The results shown in this section were those from a test system with some modification introduced to the basic scenario. The diagram used in the test system is the one in Figure 5.6. The purpose for the modification are that on one hand to confirm the function of bus with changing base, and on the other hand the function of two winding transformer with phase shift. The network data were attached in Appendix.

The parameters for generator were entered on machines' base (60 MVA and 1000 MVA) which were different from the power rating of the system (100 MVA). Meanwhile the network were expressed on system base. The changing base action is introduced by connecting the machines and the network through a bus model. The "current" flowed out of the bus were scaled by the factor $\frac{S_{base}}{m_{base}}$, thus the base were changed from machine base to system base. Note that, the machine should be connected to the side of the bus with a "*" sign.

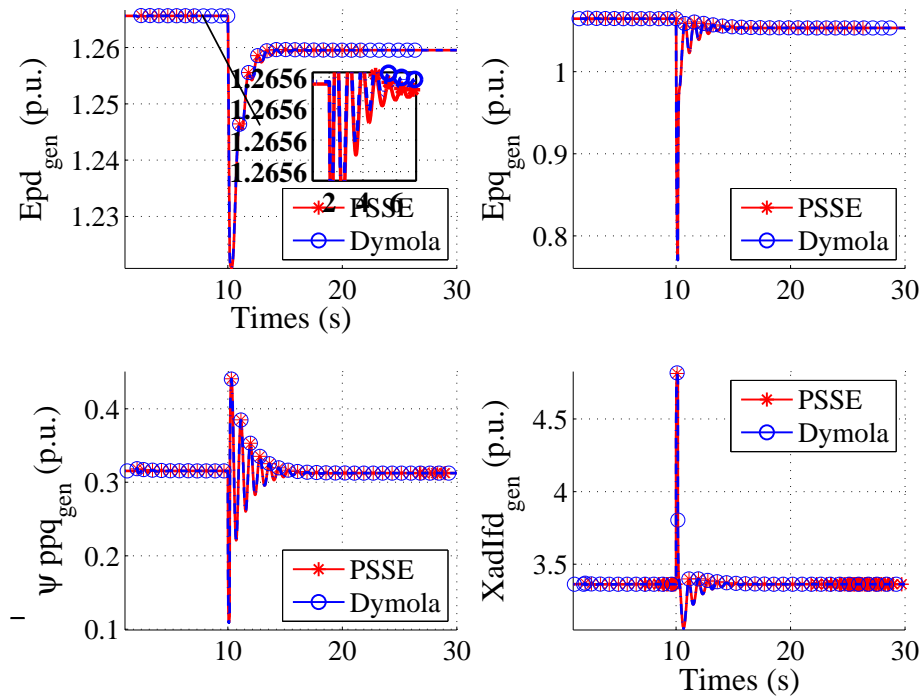
When connecting bus with different nameplate voltage, an transformer should be introduced to adjust the voltage. Different connection of the transformer will result in a phase shift between the sending and receiving end bus voltage phasor. There were two transformers equipped in the test system, the upper one connected in a way which will not introduce phase shift, but lower one can result in 30 degree lead at primary side. The impedance and other parameters for the transformers are provided in the Appendix. In order to validate the phase shift function of the transformers, the angle difference between the sending and receiving ends were plotted and compared in Figure 5.8.

Figure 5.8 shows us that, the angle shift of transformer from two software matches each other very well. The system in Dymola is identical to the reference PSS/E test system both in initialization and dynamic state. The results proved the behavior of model Gensal and the actions of changing power and voltage base. By including these changing base action, it is possible to use the developed library to present larger system which might include several machines with different power rating connected by step-up or set-down transformers with phase shift.

Since the changing base action has been proved, for the sake of convenient, the following test were performed with all parameters on system base.



(a) Generator transient and subtransient electric voltage and flux.



(b) Generator terminal current.

Figure 5.7.: Validation result of 2_{nd} test system (2) .

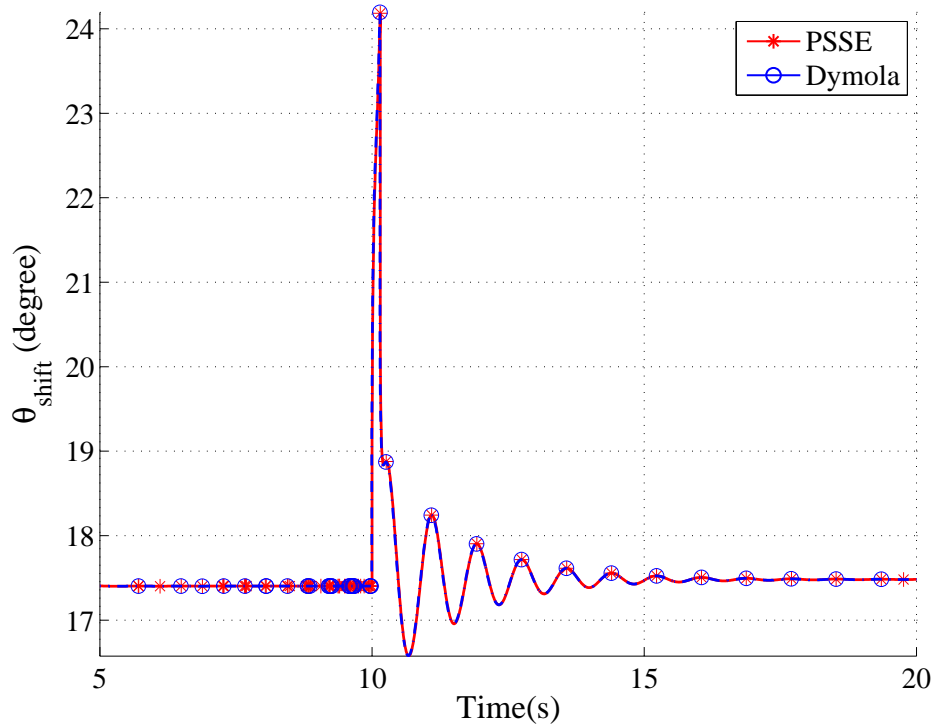


Figure 5.8.: Bus voltage's angle shift between sending and receiving ends of the transformer.

5.1.3. Validation of IEEET2 and IEESGO

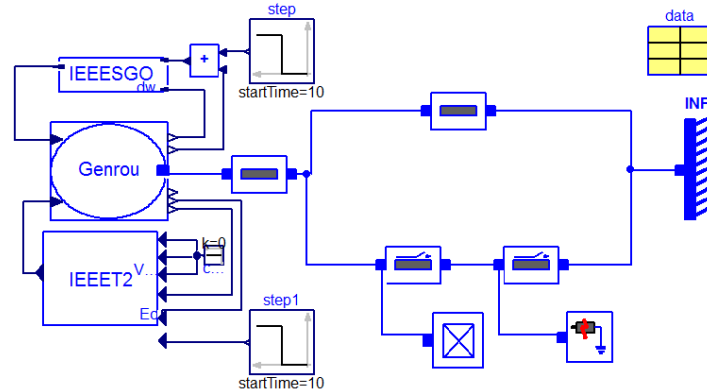


Figure 5.9.: Diagram in Dymola of the 3_{rd} test system.

The purpose of simulations is to validate the transient response of generator and exciter IEEET2 and governor IEESGO. But the focus is on the behavior of the regulators. Below is a list of simulations to be performed and diagram of the test system in Dymola. The results are shown in Figure 5.10 to 5.12. The three phase to ground fault was applied at same location with the same range as the two previous cases.

The regulators contributed new dynamics to the system and they will be discussed later. After the fault was cleared, the reference step changes were applied. The results shown us that

the regulators modeled in Modelica functioned very well since the system behavior in two software confirmed each other. The further comparison are done by plotting the percentage error of the records.

List of simulations:

1. running under steady state for 2 seconds.
2. Apply three phase to ground fault as shown in Figure 5.2(a).
3. After 0.15s later, the fault was cleared by tripping the line.
4. Run simulation to 10s.
5. Apply $-0.002p.u.$ step in governor reference.
6. Apply $0.01p.u.$ step in exciter reference.
7. Run simulation until 20 seconds.

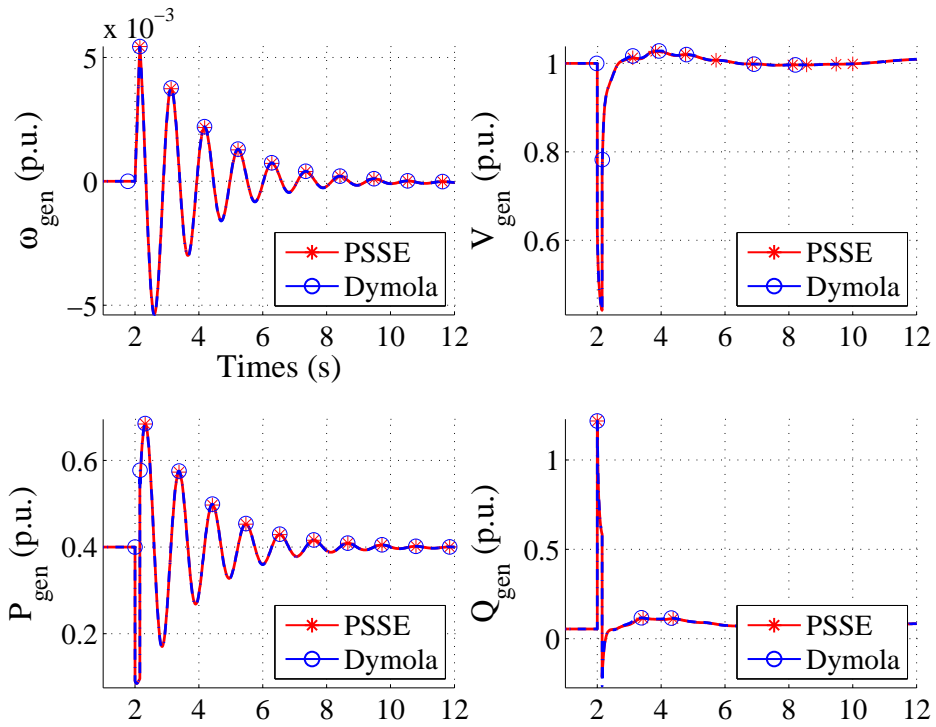
When the three phase to ground fault was applied, regulators of the generator detected the changes caused by the faults. The increasing of the speed activated the governor system, and the mechanical power was then reduced in order to decrease the accelerate power which can be seen in Figure 5.12(b). While the drop in generator terminal voltage alarmed the excitation system to increase the field voltage. By doing this the excitation system attempted to increase the induced voltage in the armature windings in order to compensate the big drop. The reaction of exciter was shown in Figure 5.11(b). However, due to the limit of power supply, the output of the exciter regulator was saturated and remained its upper limit $4p.u.$ during the fault.

0.15s later, the fault was clear by tripping the line. The impedance of the network recovered and as a result the current flowed out of the generator drop suddenly. But the value are still higher than its normal operation value because the electric magnetic force were lifted up by the exciter during the fault, and hence still kept its instant value when the fault was cleared. The active power consumption were then lifted up, remind that at this time the mechanical output from the governor system has been decreased which now resulted in a negative power unbalance. The generator was then slowed down. And following the clearing of the fault, the current fed by the generator reduced thus the system bus voltage recovered. But they were a bit higher than the steady state values. The overshoot of the bus voltage was caused by the regulator system.

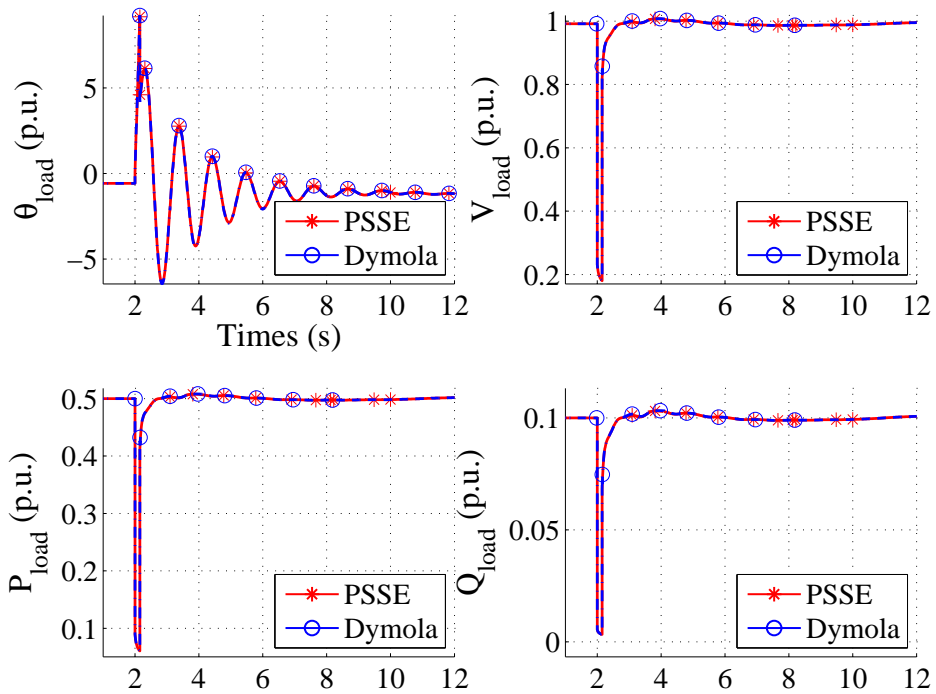
As a result of rising swing of the bus voltage, the generator is delivering more power to the system which can be confirmed by the similar overshoot in the plots of power. The generator was then further slow downed. Once again the governor detected the speed deviation, and then raised the mechanical power. The governor attempted to cancel the speed varying of the generator.

The transient of the system had almost died out at around 10s, the system would return to its steady state if not more perturbation was applied. But at this time, the reference step changes were applied to both governor and exciter. The regulators tried to meet the requirements thus instead of returning to its original equivalent point, the system would finally return to the new stationary state as set by the new reference. The Figures in 5.10 showed that, the final value of generator bus voltage was 0.01 pu higher then the original steady state value.

And the mechanical output of the governor is 0.002 lower than the previous equivalent point. The system reacted in the same way before, during, and after all the perturbation, it is fair to conclude that, the Modelica models, IEEET2 and IEESGO have been proved.

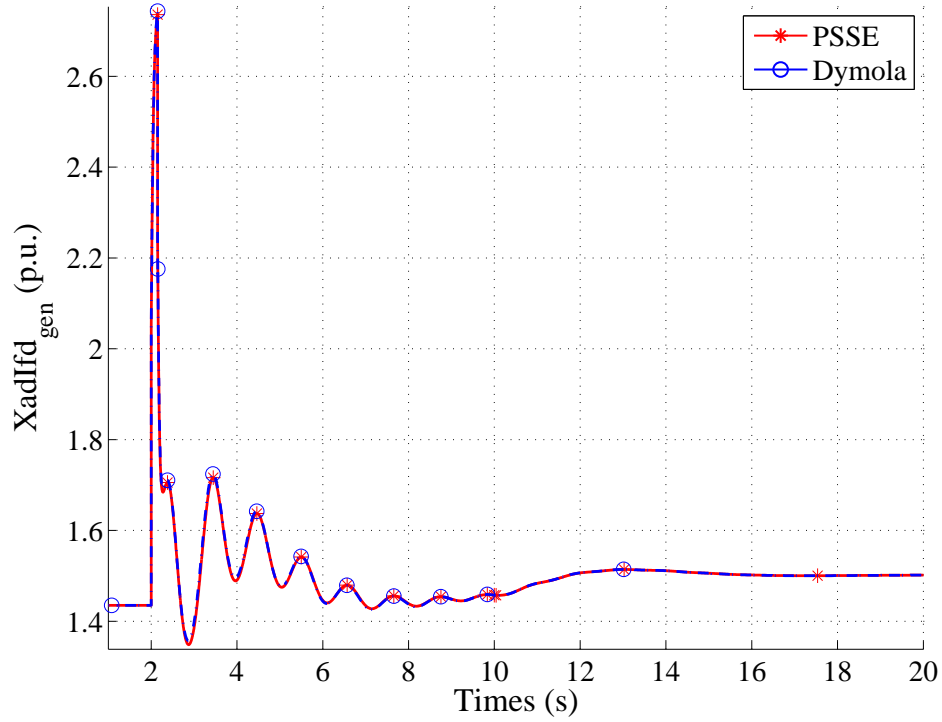


(a) Generator terminal quantities.

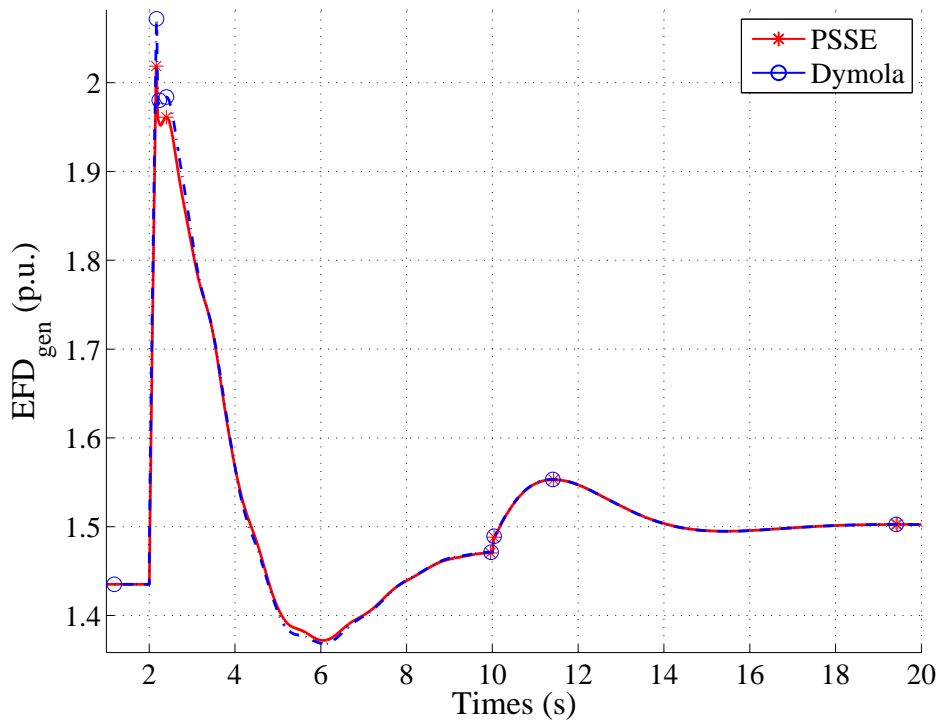


(b) Load bus quantities.

Figure 5.10.: Validation result of 3_{rd} test system (1) .

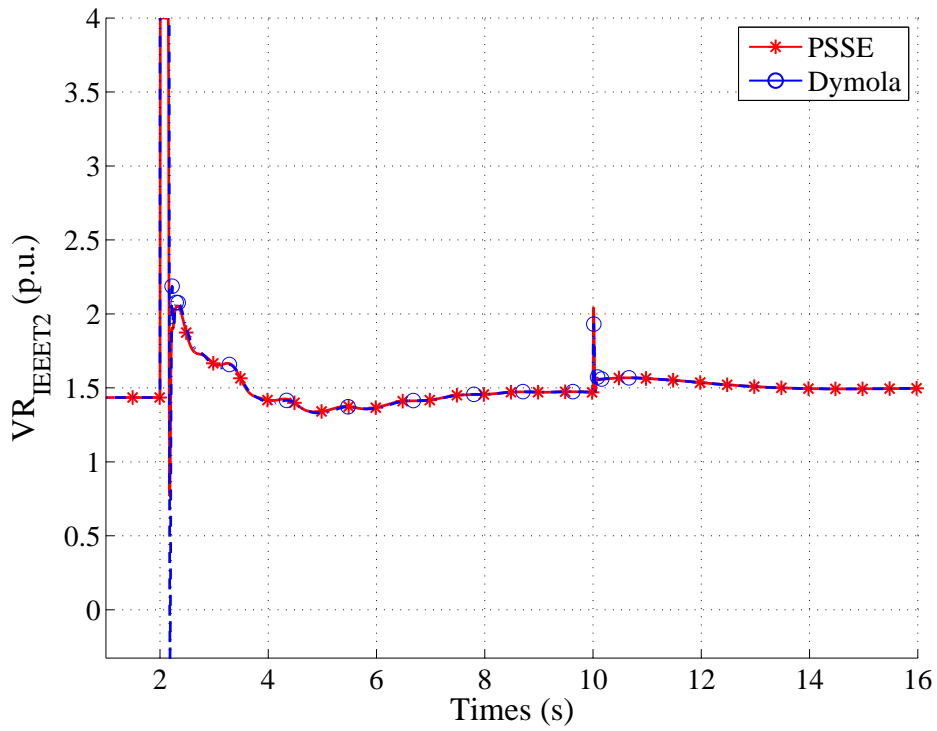


(a) Generator terminal current.

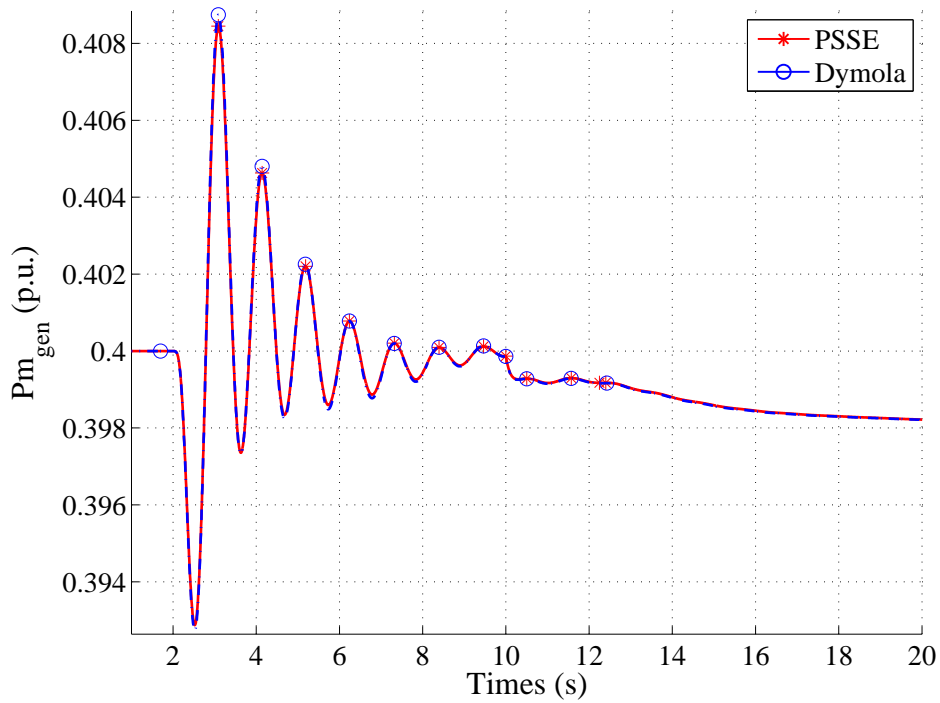


(b) Generator field voltage.

Figure 5.11.: Validation result of 3_{rd} test system (2) .

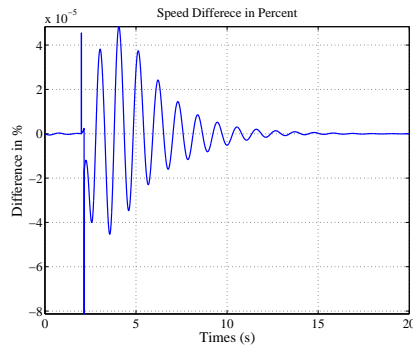


(a) Exciter regulator output.

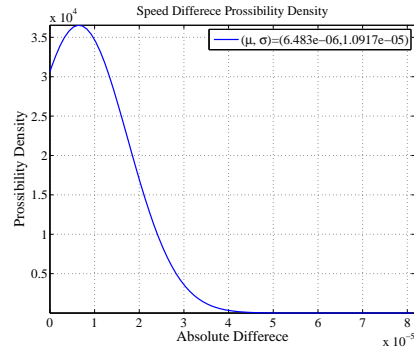


(b) Generator mechanical power.

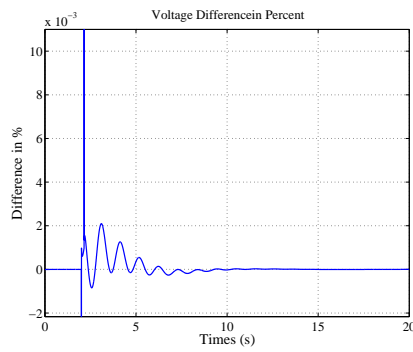
Figure 5.12.: Validation result of 3_{rd} test system (3) .



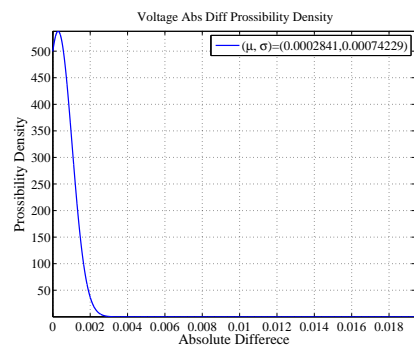
(a) $\frac{\Delta\omega}{\omega_r}$



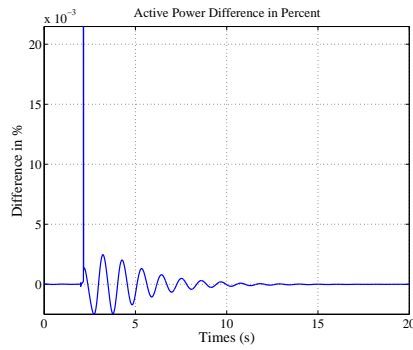
(b) Speed Difference PDF.



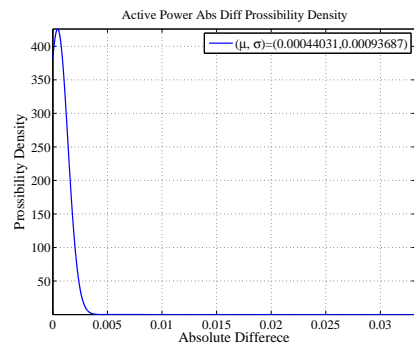
(c) $\frac{\Delta V}{V}$



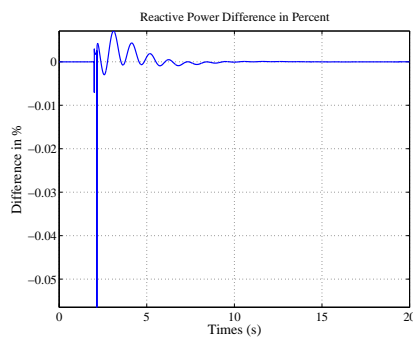
(d) Voltage Difference PDF.



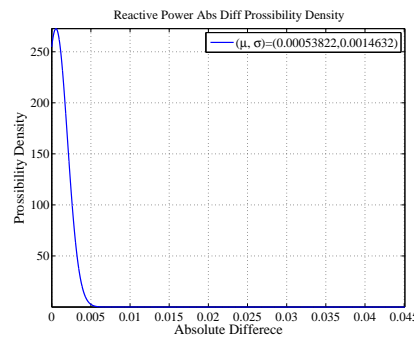
(e) $\frac{\Delta P_{gen}}{P_{gen}}$



(f) Active Power Difference PDF.



(g) $\frac{\Delta Q_{gen}}{Q_{gen}}$



(h) Reactive Power Difference PDF.

Figure 5.13.: Validation result of 3_{rd} test system (4) .

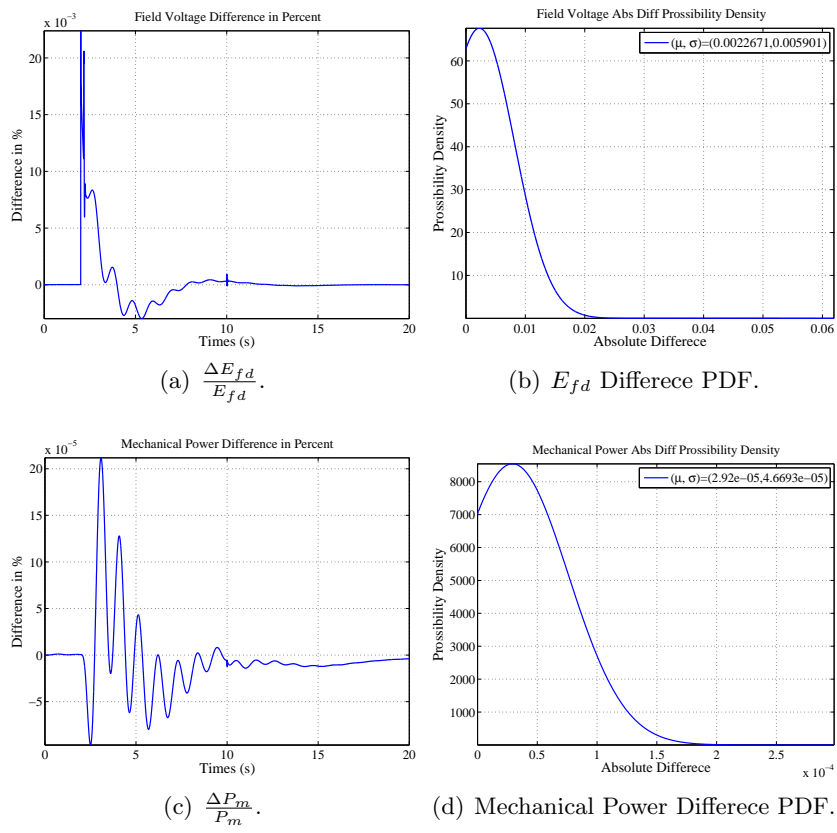


Figure 5.14.: Validation result of 3_{rd} test system (5) .

5.1.4. Validation of SCRXS and HYGOV

The dynamic components in this test are model SCRXS and HYGOV. In order to completely validate the transient behavior of the two models, the simulation scenario was selected as list below. The specific scenario could activate the "Negative current logic" inside model SCRXS and made the governor HYGOV hits its no-windup limit. Thus beside the those quantities shown in previous tests the variables such as the water velocity and desired gate position had been plotted as well. Specific attention should be paid to the validation of model SCRXS, since as it would behavior differently according to a zero or no-zero value of the parameter $\frac{r_c}{r_{fd}}$.

To force the generator current to be negative, which means the generator is forced to consumed reactive power. The perturbation could be a fault which was modeled by connecting an admittance directly to the generator terminal bus. Since the analysis of the fault have been covered in previous example and the focus of this thesis is not the fault analysis, thus from now on for the rest of the tests, only the comparison results will be shown without explanation of the behavior of the variables.

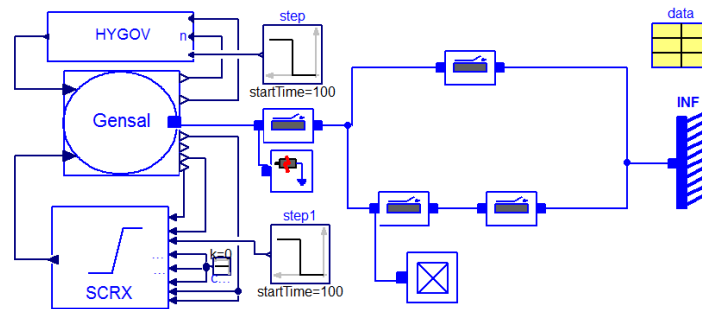


Figure 5.15.: Diagram in Dymola of the 4th test system.

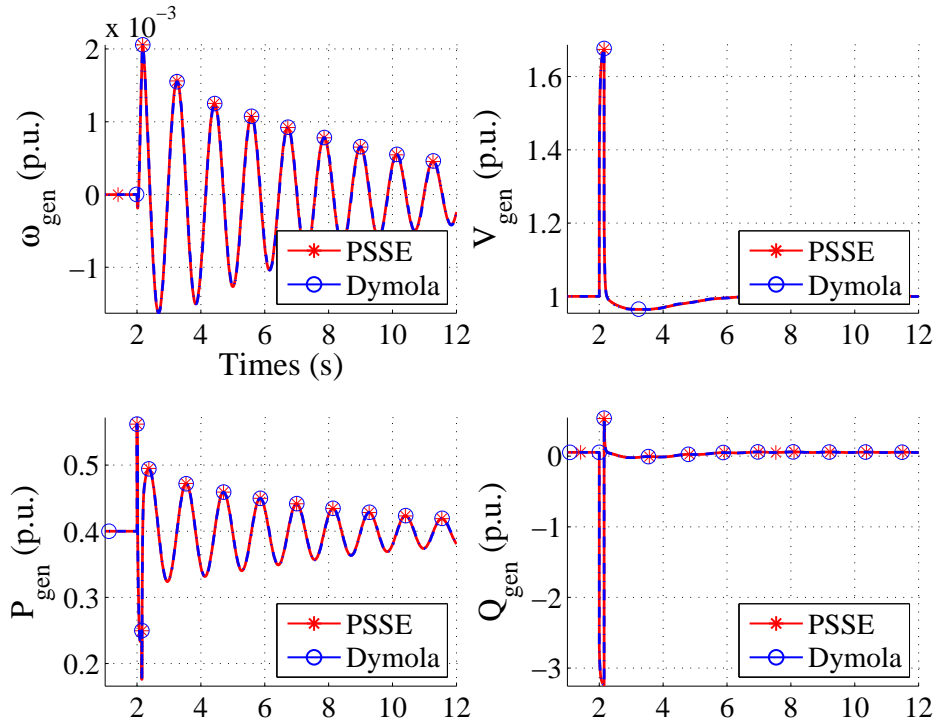
List of simulations:

1. running under steady state for 2 seconds.
2. Apply three phase fault as shown in Figure5.2(b) with conductance $X = -0.5$.
3. After 0.15s later, the fault was self cleared.
7. Run simulation until 20 seconds.

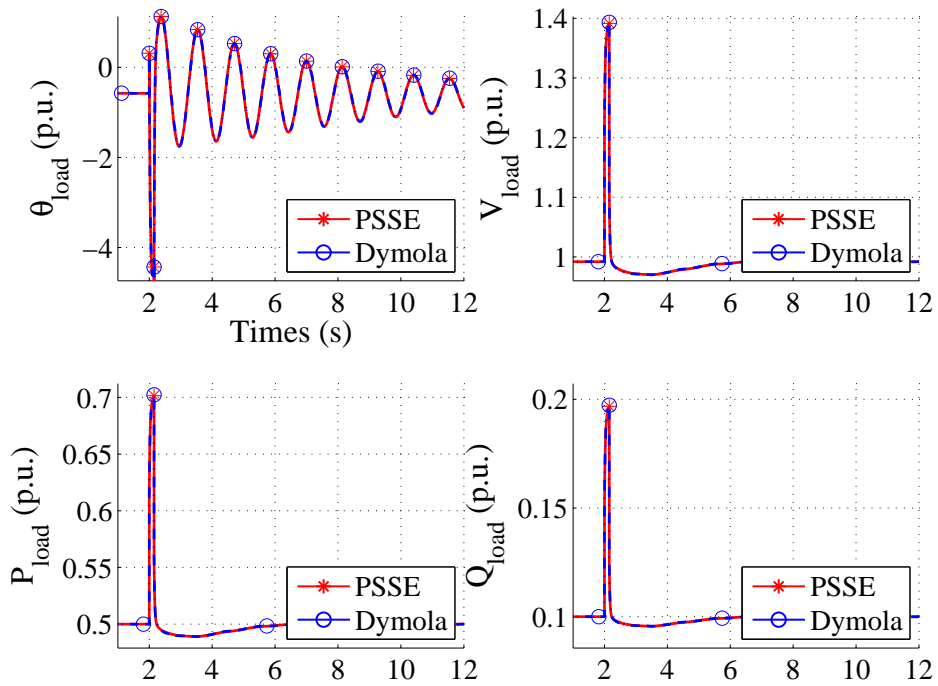
5.1.5. Validation of SEXS and IEEEEST

List of simulations:

1. running under steady state for 2 seconds.
2. Apply three phase to ground fault as shown in Figure5.2(a).
3. After 0.15s later, the fault was self cleared.
4. Run simulation until 20 seconds.

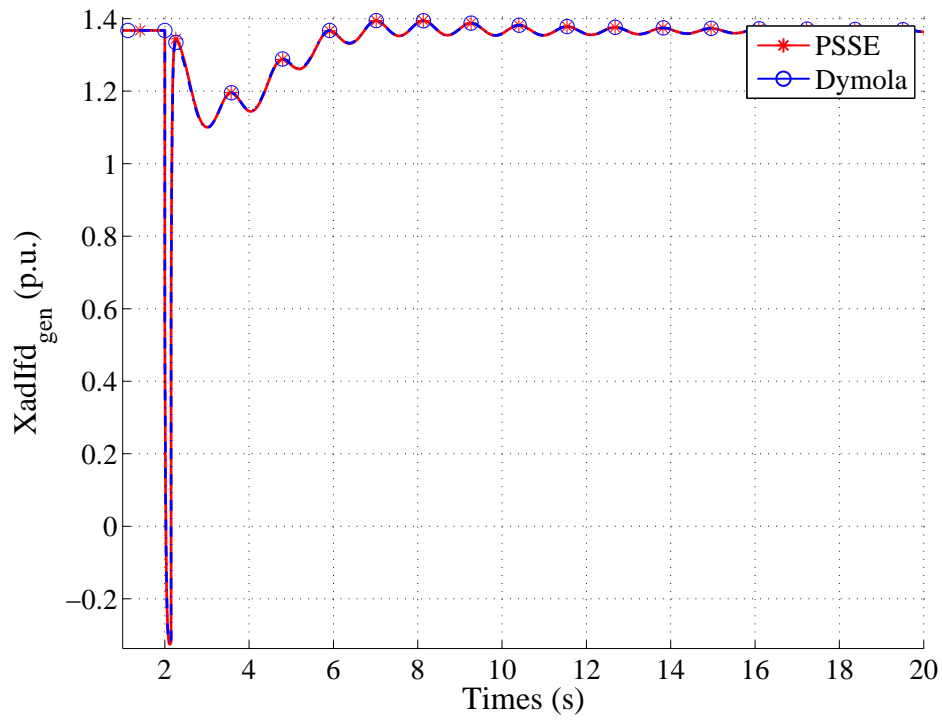


(a) Generator terminal quantities.

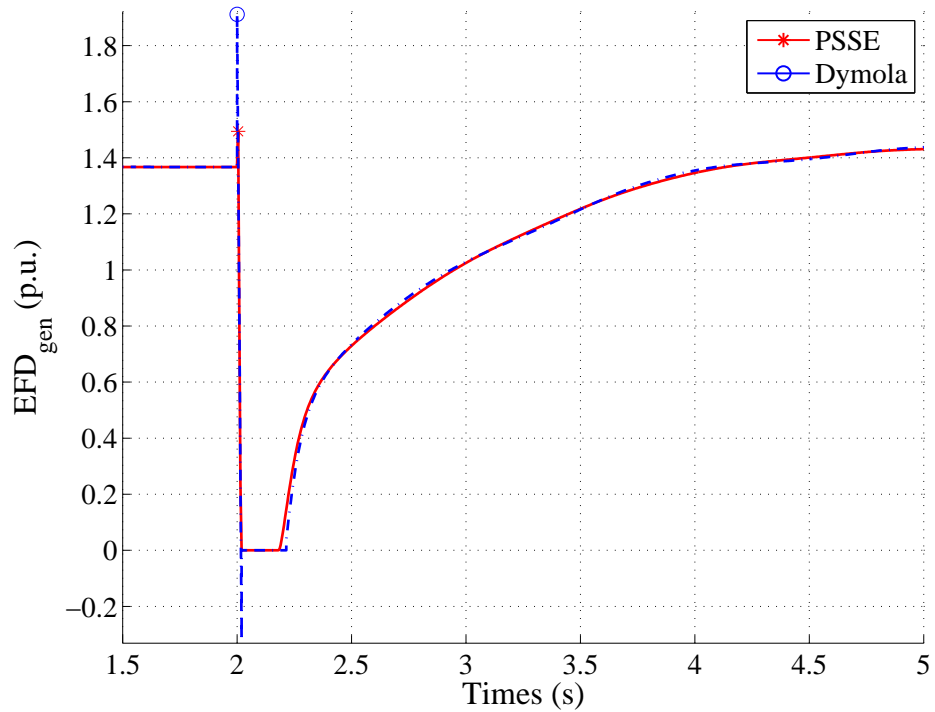


(b) Load bus quantities.

Figure 5.16.: Validation result of 4_{th} test system (1) .

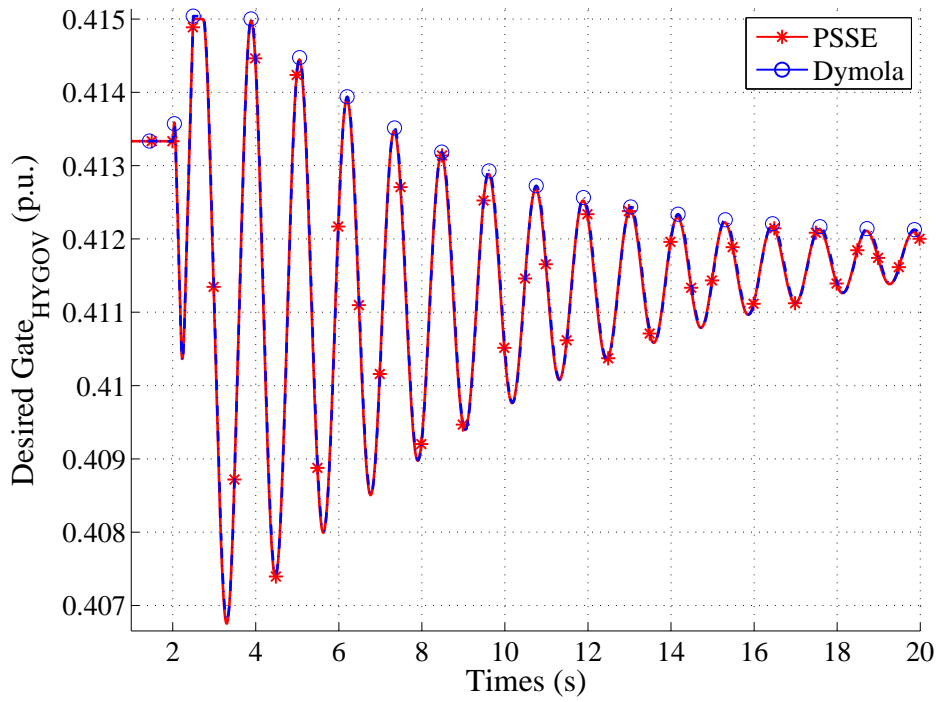


(a) Generator terminal current.

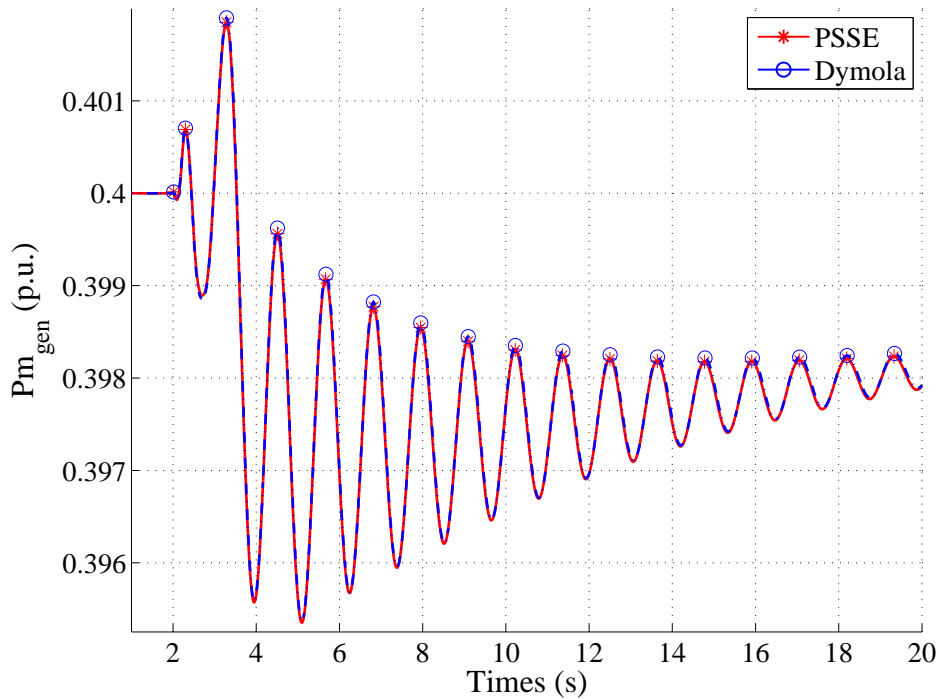


(b) Generator field voltage.

Figure 5.17.: Validation result of 4th test system (2) .

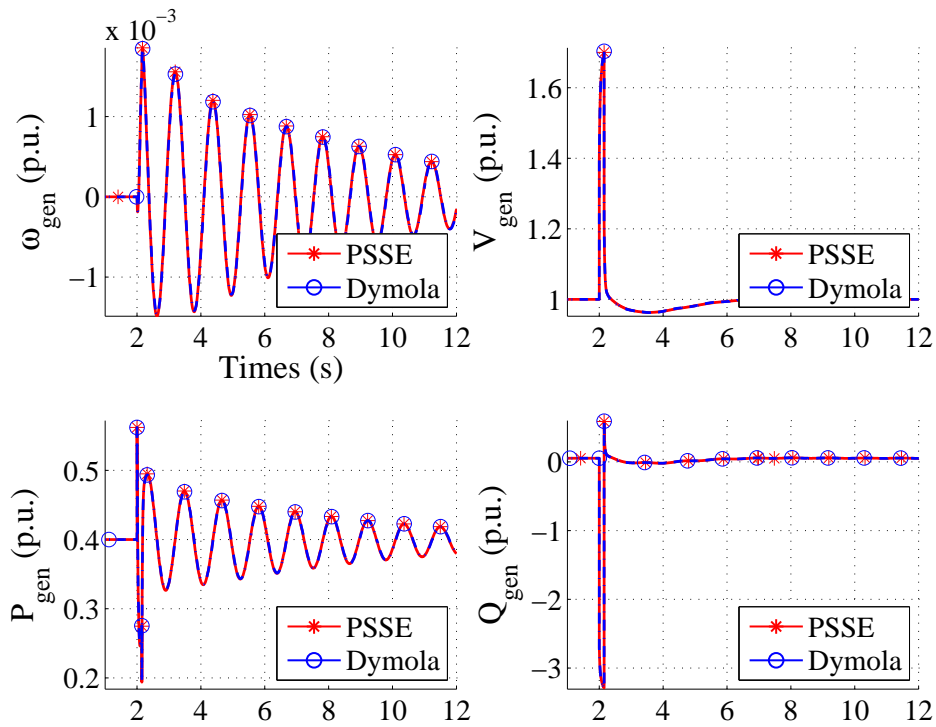


(a) Governor desired gate position.

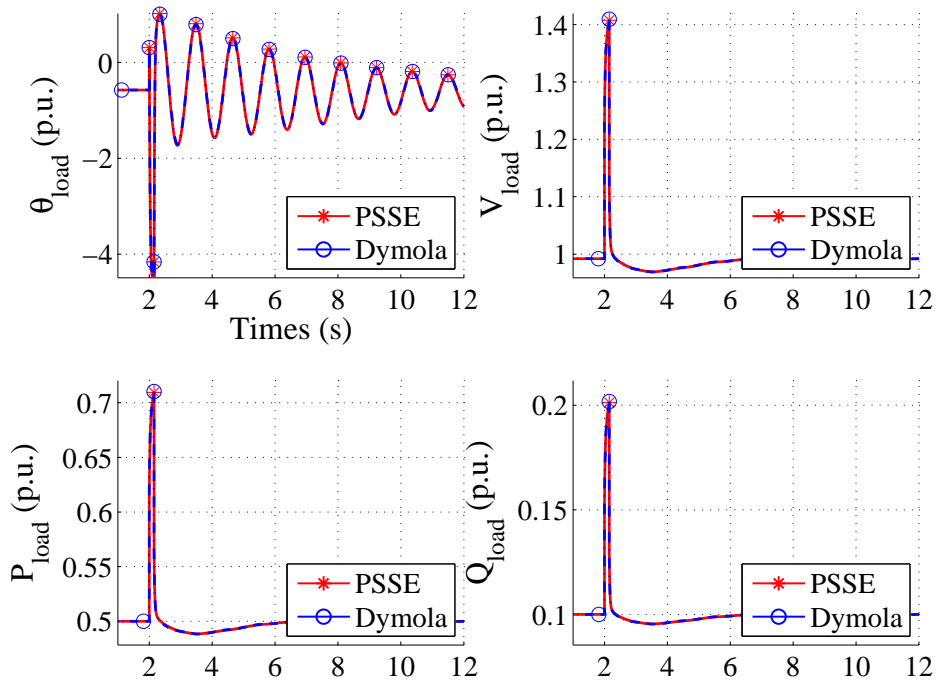


(b) Generator mechanical power.

Figure 5.18.: Validation result of 4th test system (3) .

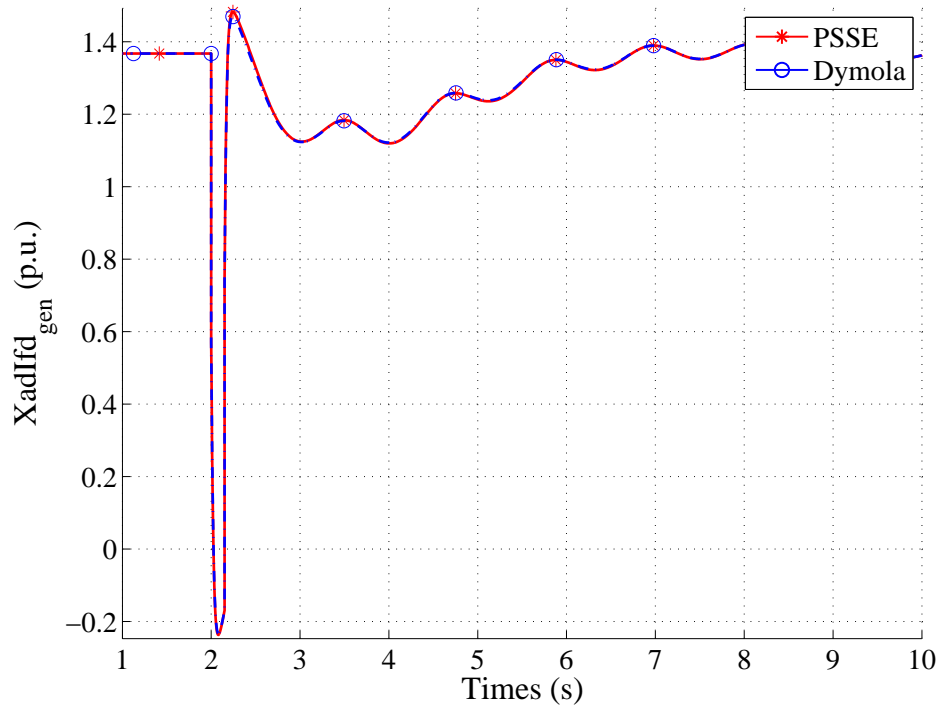


(a) Generator terminal quantities.

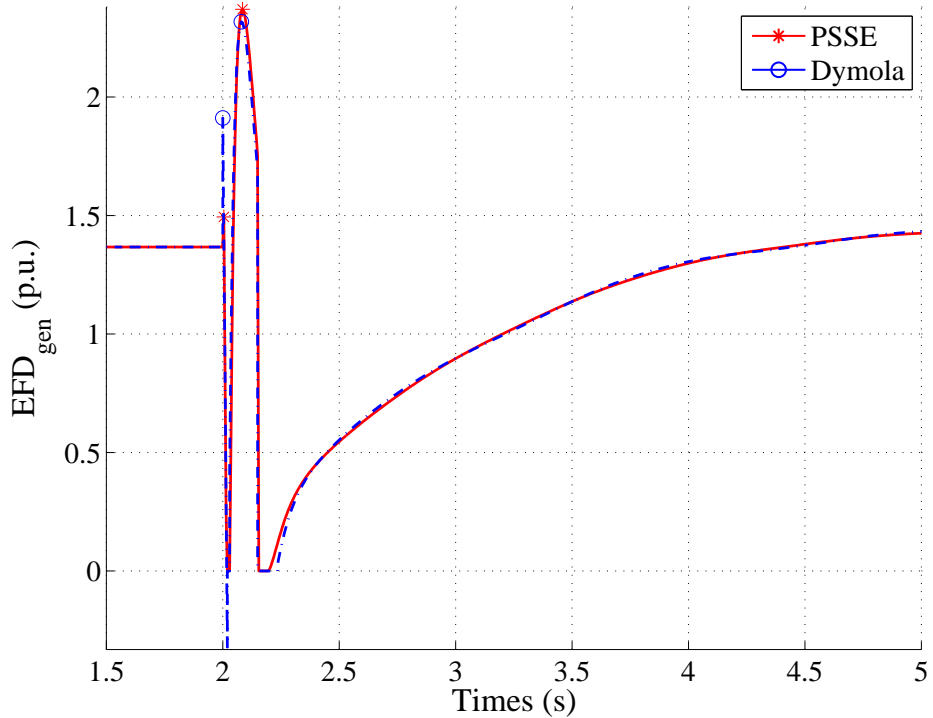


(b) Load bus quantities.

Figure 5.19.: Validation result of 4_{th} test system (4) .

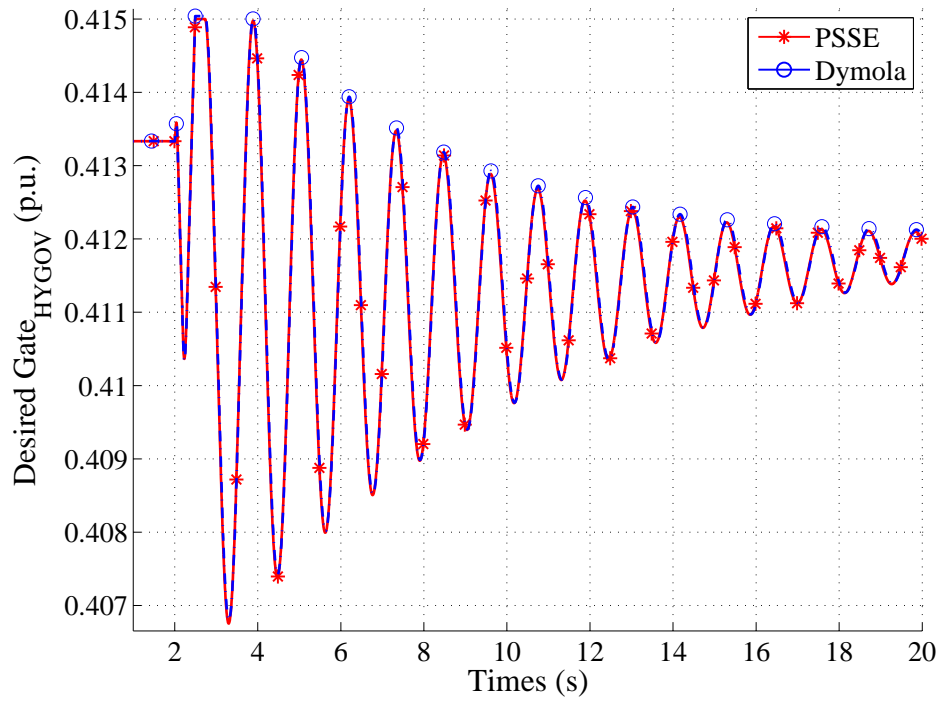


(a) Generator terminal current.

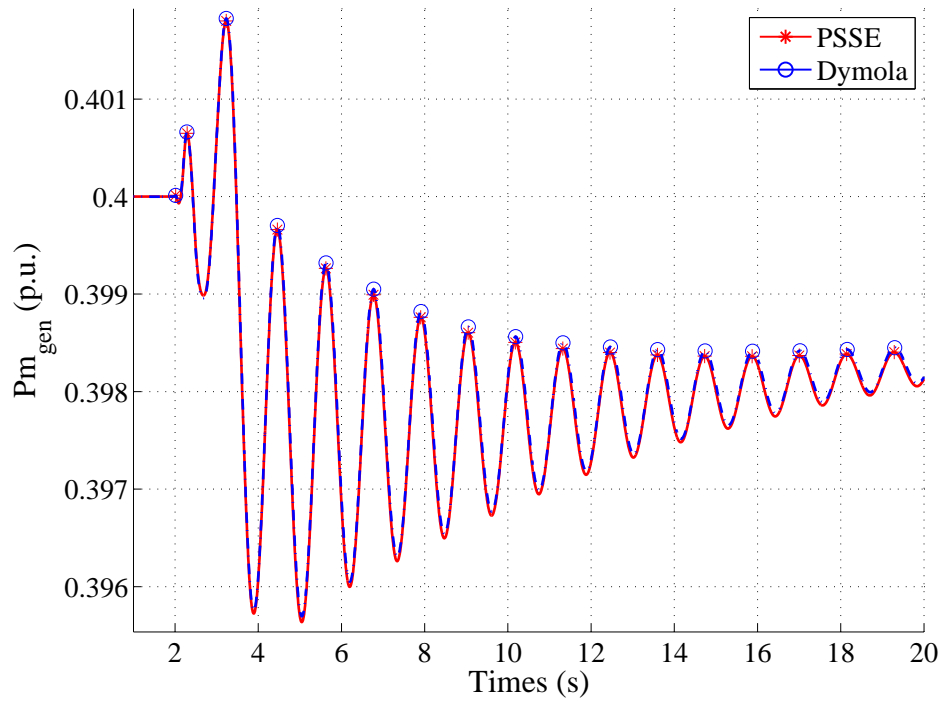


(b) Generator field voltage.

Figure 5.20.: Validation result of 4th test system (5) .



(a) Governor desired gate position.



(b) Generator mechanical power.

Figure 5.21.: Validation result of 4th test system (6) .

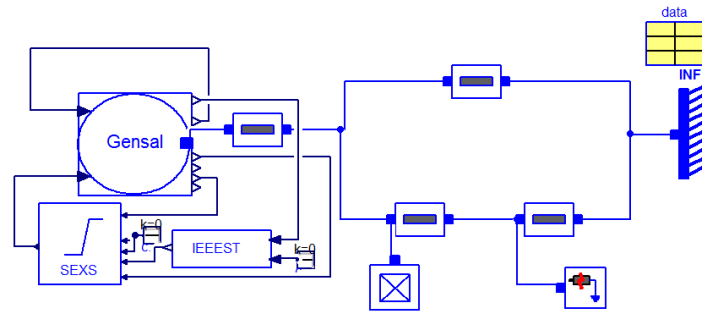


Figure 5.22.: Diagram in Dymola of the 5_{th} test system.

5.1.6. Validation of IEEEET1, STAB2A and LFDRL

List of simulations:

1. running under steady state for 2 seconds.
2. Apply three phase to ground fault as shown in Figure5.2(a).
3. After 0.15s later, the fault was self cleared.
4. Run simulation until 20 seconds.

5.2. Real World Power System

5.3. AKD

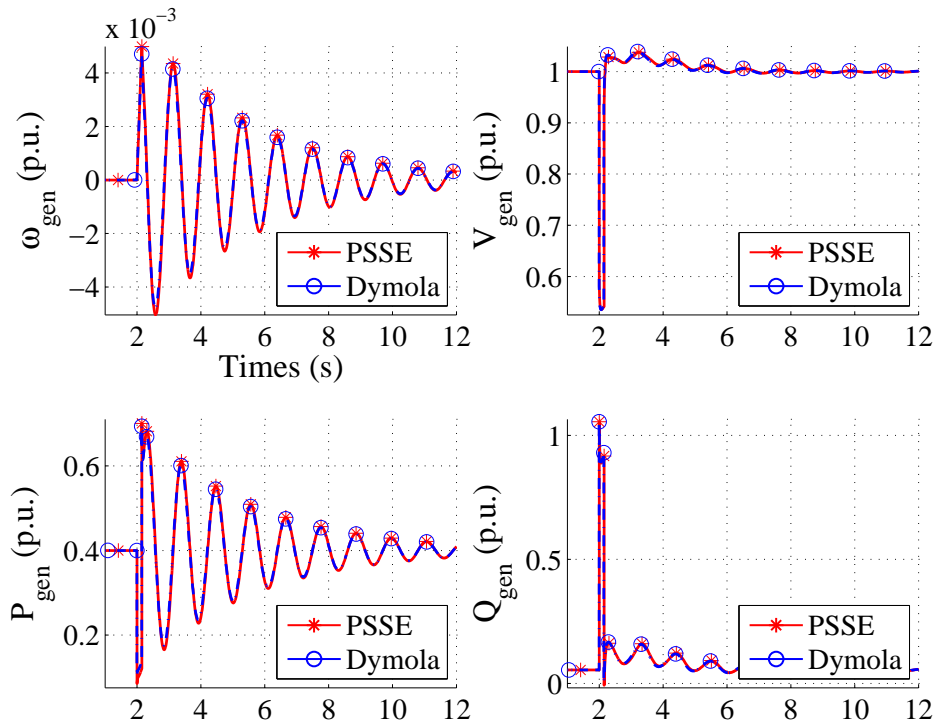
System model AKD is a subset model of Norwegian grid. The system setup is shown in Fig.5.27. It contains five detail generator models All of them are Model Gensal. The three generators on the upper side are the equivalent models of remote grid and the rest two generators which are connected to the network through transformers are the equivalent models of the local grid. These two generators are equipped with regulators. They are governor HYGOV, exciter IEEEET2, and stabilizer IEEEEST.

5.3.1. System Setup

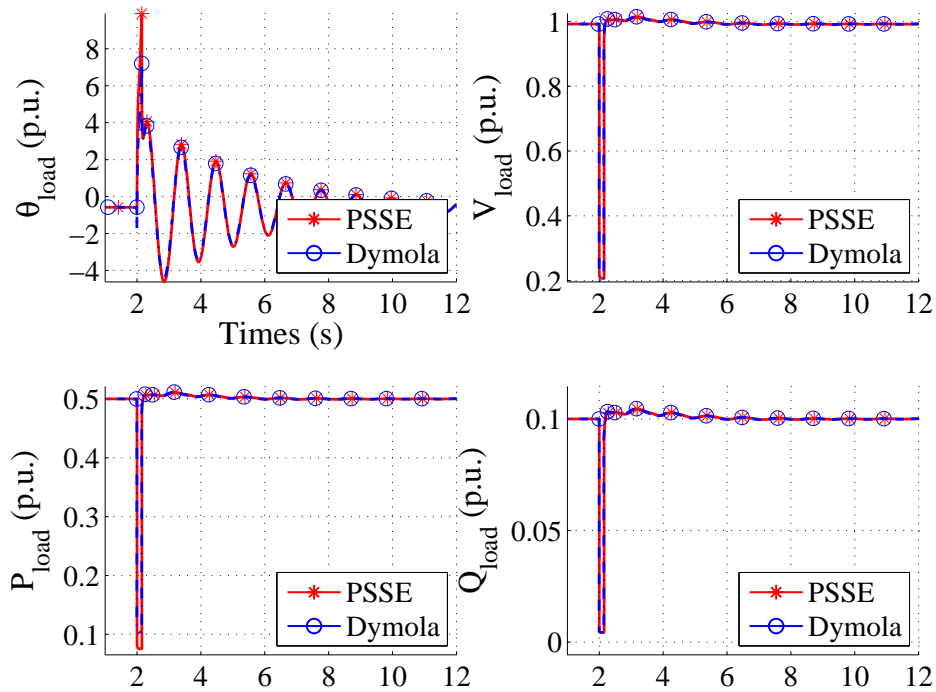
5.3.2. Validation result

List of simulations:

1. Running under steady state for 1s.
1. Open the line for 0.15s.
1. Apply 3phtgf at 7.5s; 0.15s later, clear the fault.
1. Run simulation until steady state.

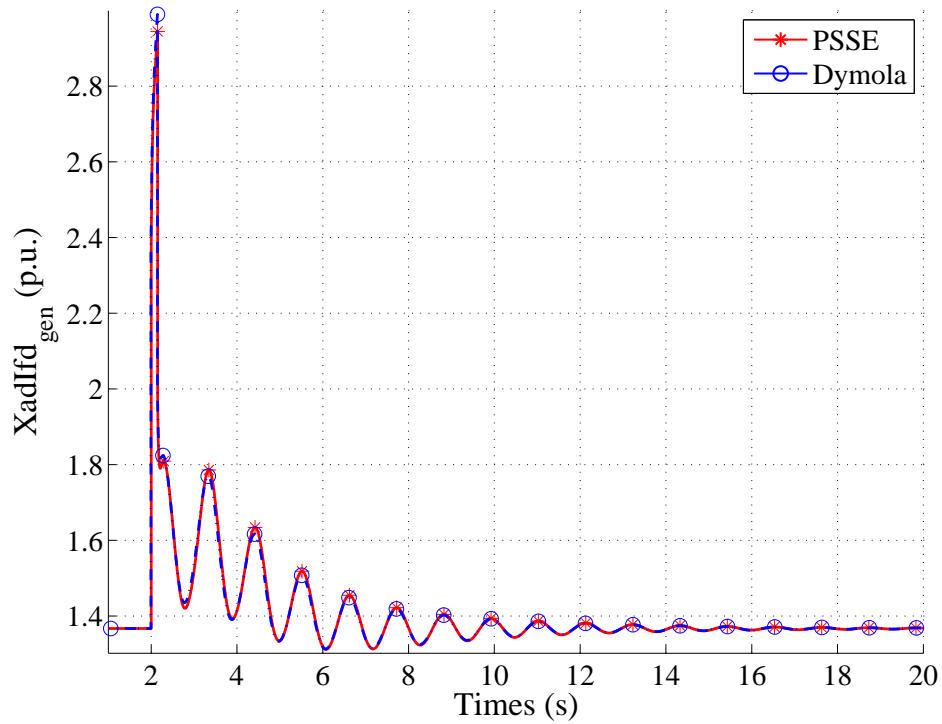


(a) Generator terminal quantities.

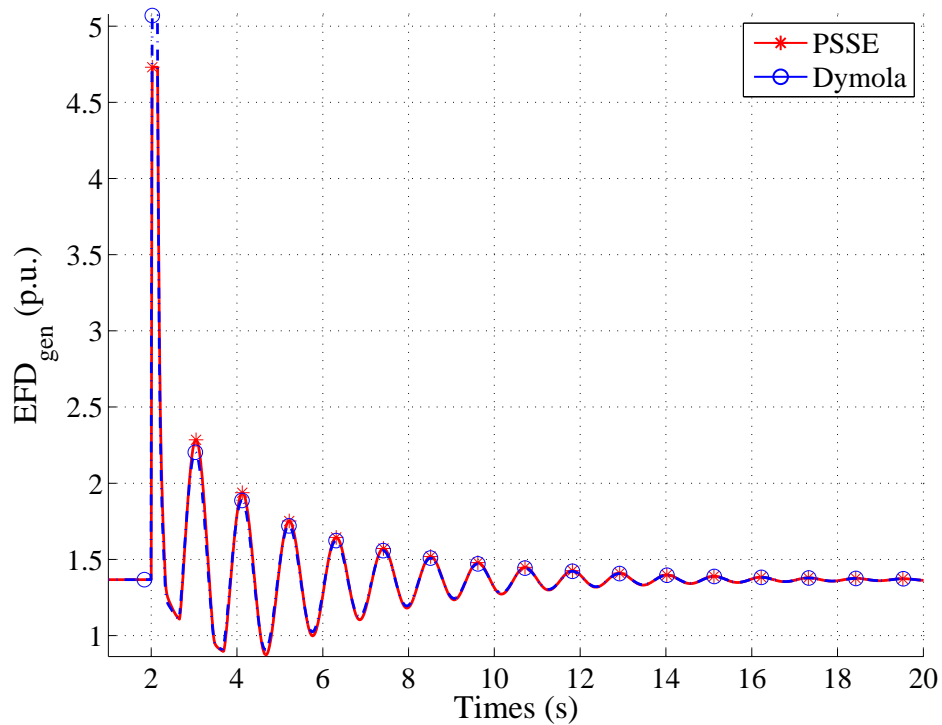


(b) Load bus quantities.

Figure 5.23.: Validation result of 5_{th} test system (1) .



(a) Generator terminal current.



(b) Generator field voltage.

Figure 5.24.: Validation result of 5th test system (2) .

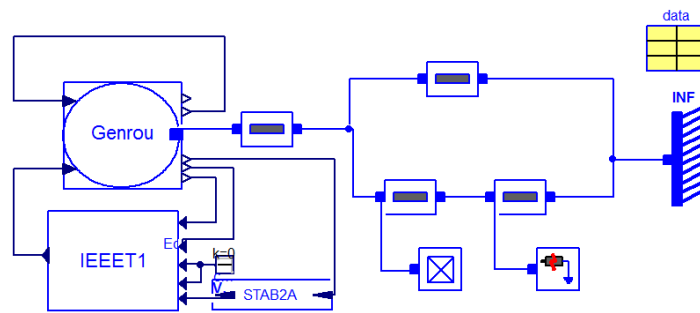
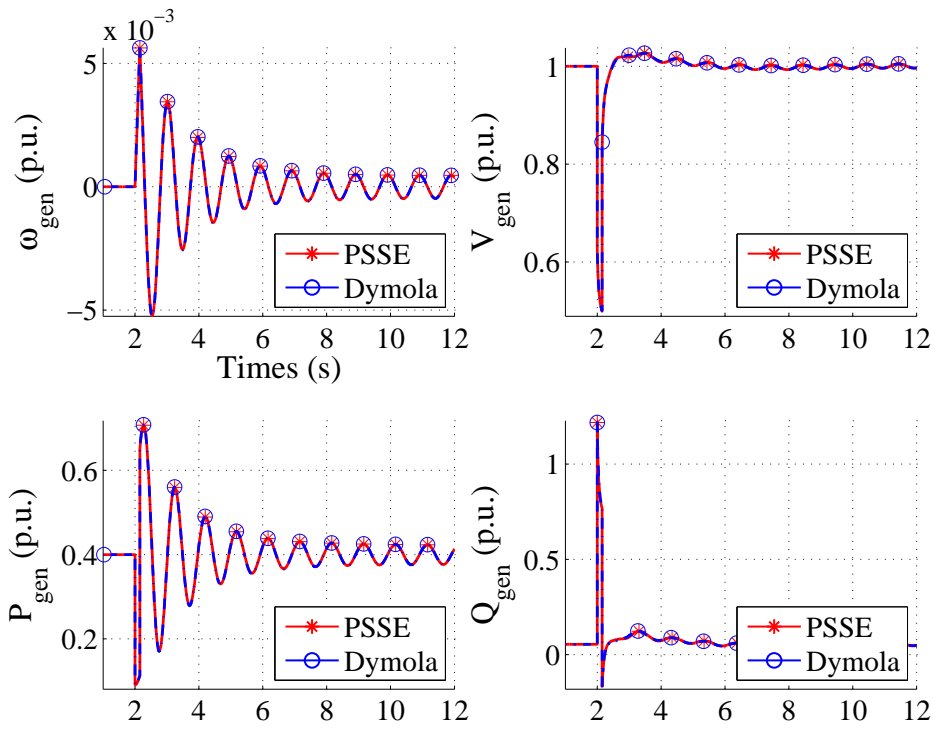
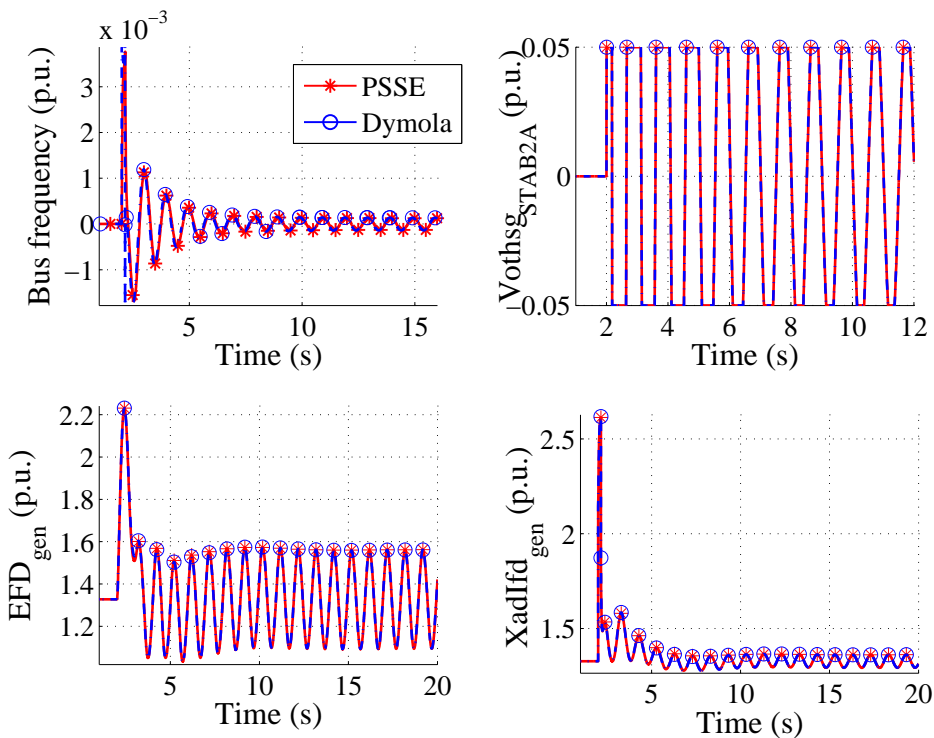


Figure 5.25.: Diagram in Dymola of the 6th test system.



(a) Generator terminal quantities.



(b) Load bus quantities.

Figure 5.26.: Validation result of 6_{th} test system.

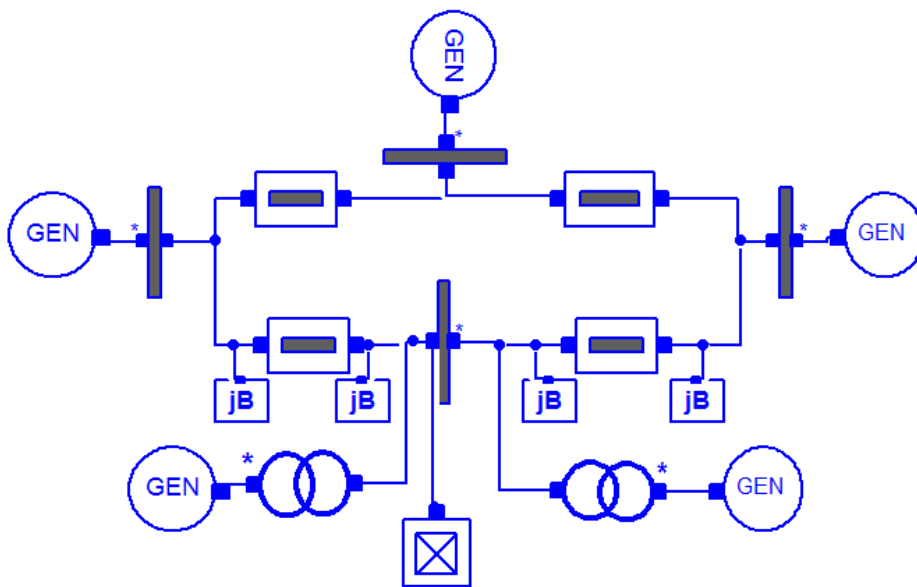
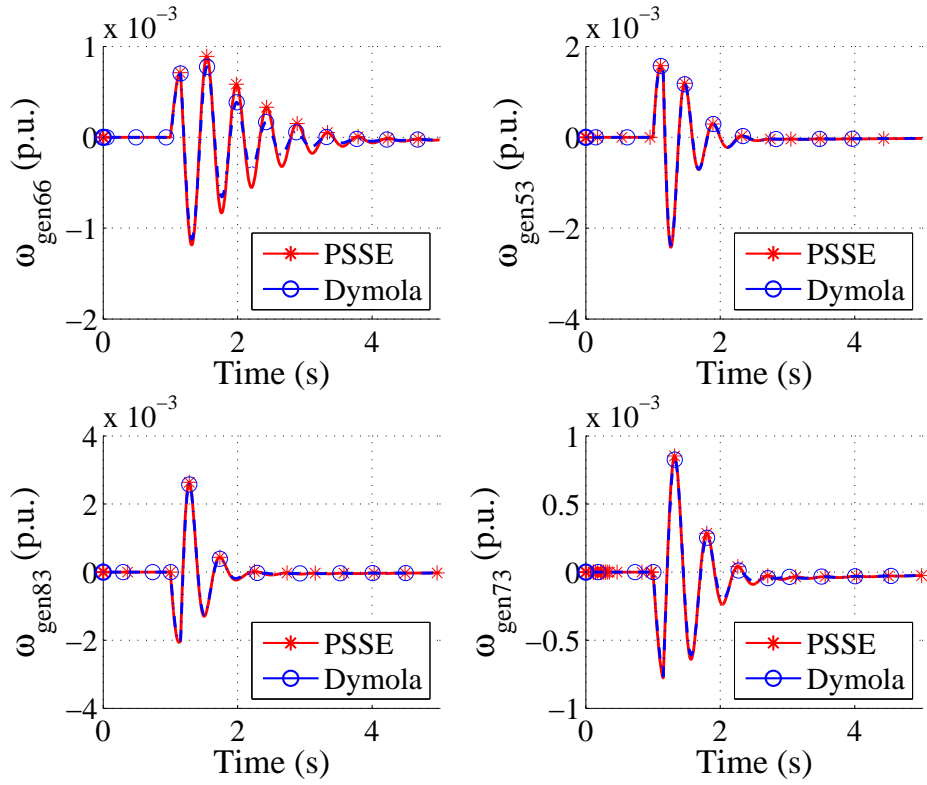
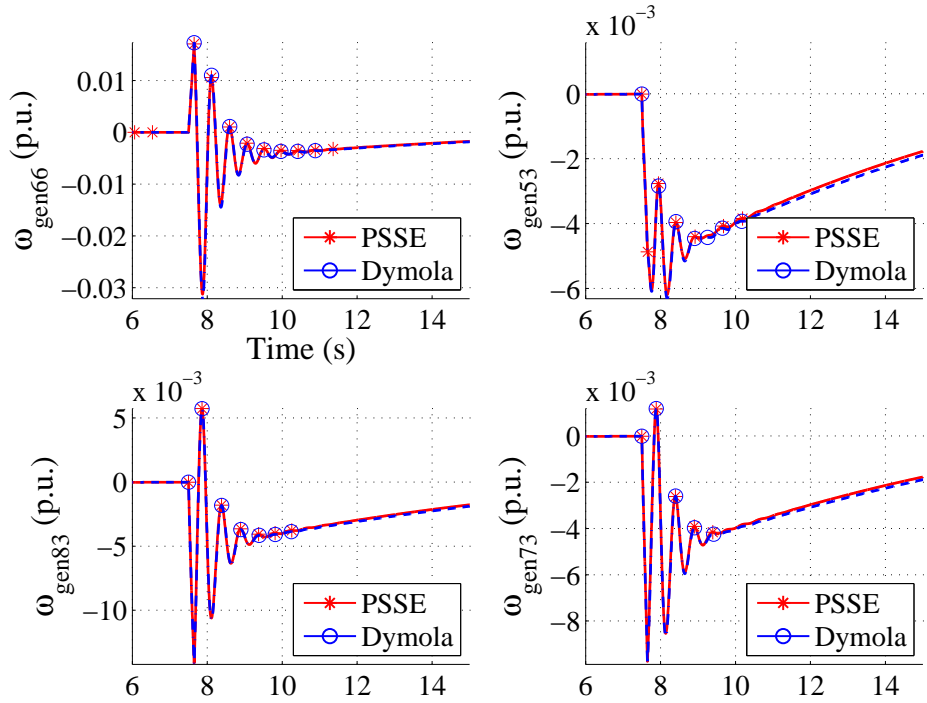


Figure 5.27.: Test system diagram



(a) Generator speed variation(1).



(b) Generator speed variation(2).

Figure 5.28.: Validation result of AKD test system(1).

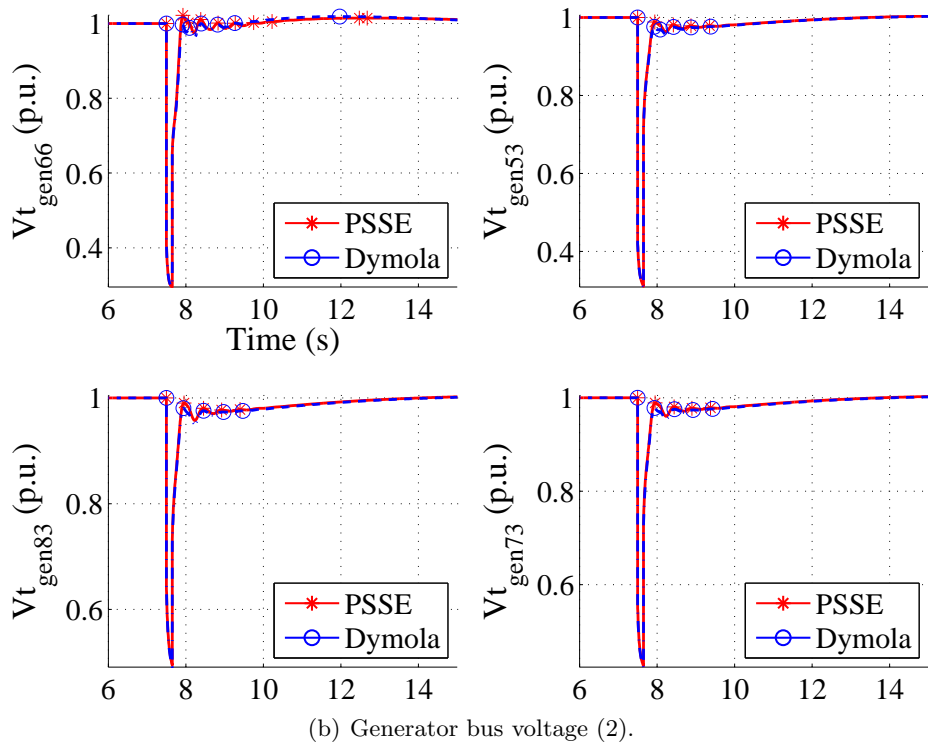
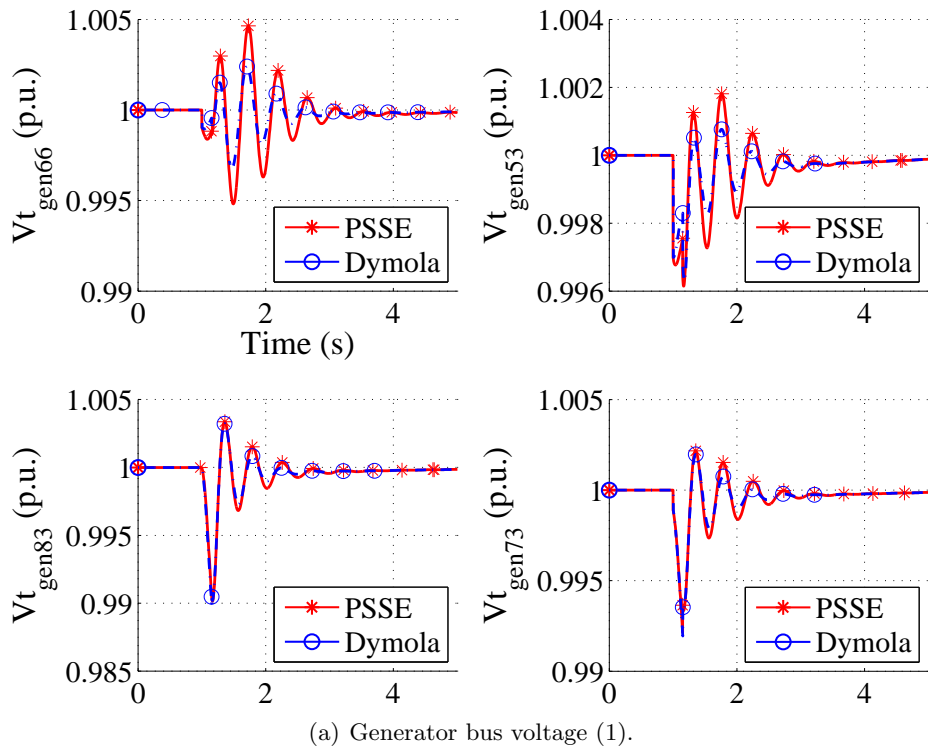
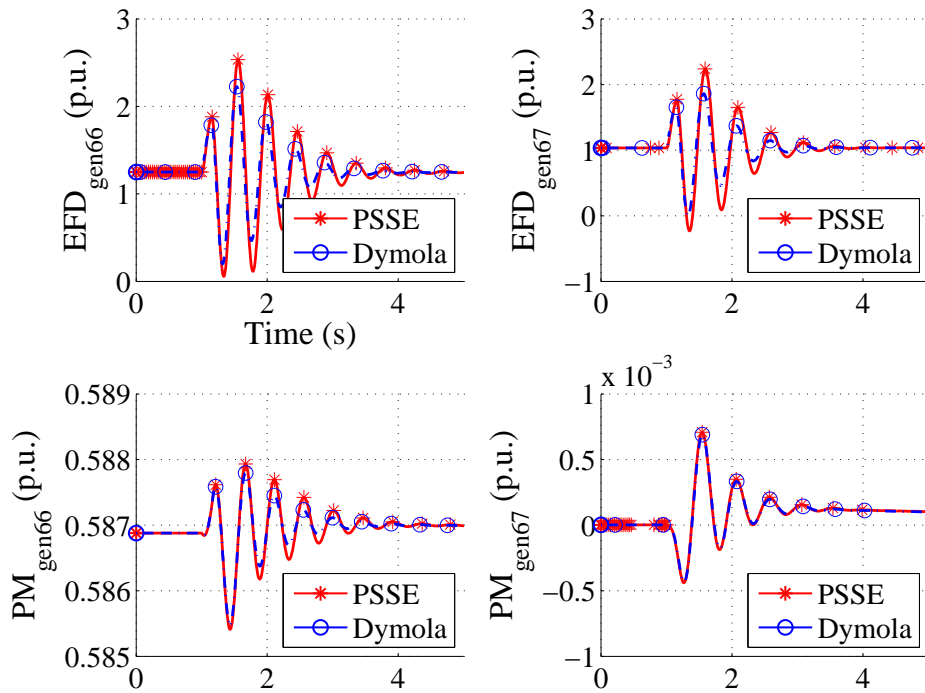
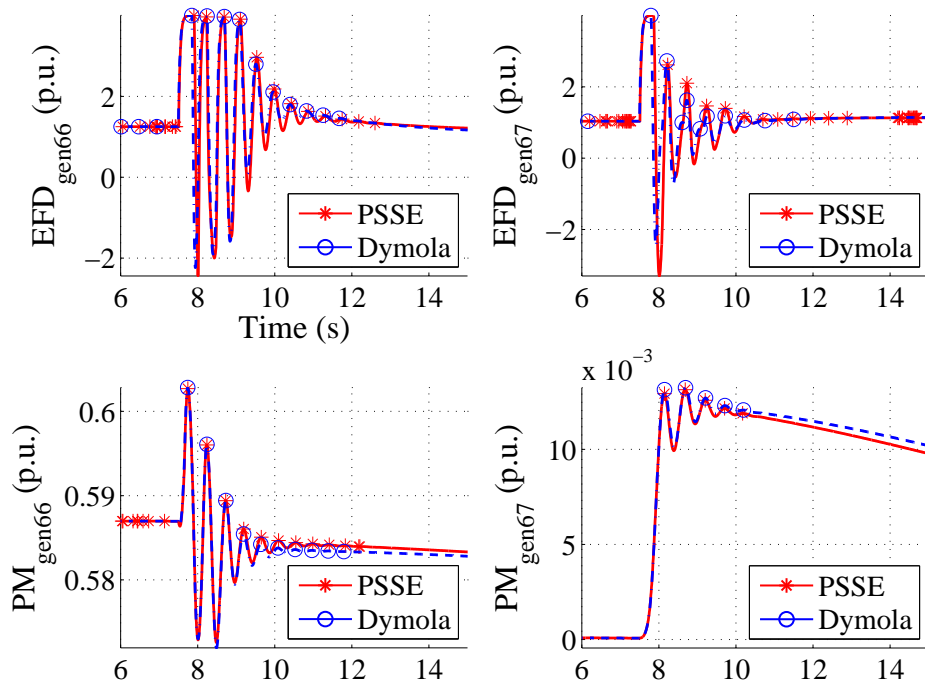


Figure 5.29.: Validation result of AKD test system(1).



(a) Field voltage and mechanical power (1).



(b) Field voltage and mechanical power (2).

Figure 5.30.: Validation result of AKD test system(3).

6

Conclusion

This project has been carried out successfully. For each of the power grid components used in Norwegian grid, a corresponding Modelica class has been built and the validation results show a high degree of similarity between the dynamic performance of Modelica model and that of the reference PSS/E model.

In chapter one, a brief introduction about the background of this project and the motivation to use Modelica in power system analysis has been discussed. In chapter two, the difference between two software Dymola and PSS/E has been studied. It can serve as a reference for one who intend to carry out similar project or to work with these two software.

The modeling method for each components model has been explained in detail in chapter three. For those models which was explained vaguely in PSS/E documents, revisions have been carried out and the performance of them have been proved by validation tests. Thus I can conclude that, the models are now ready to be used in iTesla project. And all future user can have the access to both the source code and modeling methods of the models. This will facilitate the reuse of the models. The models can be utilized in all software which can support Modelica language. Furthermore, the models can be simulated in Matlab through Function Mock Unit (FIM) tool box.

All of the data file concerning the validation tests have been documented properly which can be sued to check the work of the project. The procedures of performing validation tests have been explained step by step in previous chapter. Thus it is easy for one to follow up the work if there is any need to carry out more tests.

Finally, for those who are working with building Modelica models, the Some suggestion towards debugging procedures have been summarized as following. Basically, to ask the following questions can help one debug the models efficiently.

1. Has the model been initialized correctly ? For example, if the records of one variable from PSS/E and Dymola starts from the same points which means the power flow records are entered correctly. Further more, if the results agree with each other during a few second before applying the faults. It means the model is initialized correctly. It is better to make sure that the initialization are correct and then go further to dynamic simulation.
2. Are the applied faults in both systems the same or not ? It is easier to check the

perturbation cases before going through the equations of the models. For example, during load variation, there are several options available in PSS/E, the one with varying the load with constant P Q ratio was applied in the tests. Also, there are some trivial issues might introduce the differences, like the load will be shed down by a factor when the bus voltage goes lower than 0.8 p.u.. To reach good similarity, this particular assumption should also be implemented during tests.

3. Is there anything wrong in the model expression (equations or parameters) ? Typically, the problems can be caused by the parameters or mathematic equations. If the records show the same tendency but difference in magnitude or frequency then most of time it is caused by typing the wrong parameters. On the other hand, if the tendency is not the same, then some of mathematic equations must be wrong. they could either be typed wrongly, or be implemented in a way where Modelica has different interpretation of them. The suggestions are that to study the equations carefully, highlight the suspected ones, and test them separately.
4. Are the set up of two systems are identical or not ? The solution results from different solvers might show different inertia. That's why we said it is better to start with the same solvers. On the other hand, attentions should be paid to the plot interval and the time step if a fix step solver is applied, because the model could behavior wrongly when too large time step is applied.

A

Appendix

A.1. Dynamic parameter

Listing A.1: Dynamica parameters of Genrou test

1	1	'GENROU'	1	5.0000	0.50000E-01	0.70000	0.10000
2		4.0000		0.0000	1.4100	1.3500	0.30000
3		0.60000		0.20000	0.12000	0.10000	0.50000/
4	3	'GENCLS'	1	0.0000	0.0000/		

Listing A.2: Dynamica parameters of GENROU+IEEET2+IEESGO test

1	1	'GENROU'	1	5.0000	0.70000E-01	0.90000	0.90000E-01
2		4.2800		0.0000	1.8400	1.7500	0.41000
3		0.60000		0.2000	0.12000	0.11000	0.39000 /
4	1	'IEEET2'	1	0.0000	100.00	0.14000E-01	4.0000
5		-4.0000		1.0000	0.62000	0.12000	1.0000
6		0.10000		2.3100	0.80000E-01	3.0800	0.30000/
7	1	'IEESGO'	1	0.20000	0.0000	0.50000	0.12000
8		5.0000		0.50000	20.000	0.59000	0.43000
9		0.98000		0.0000	/		
10	3	'GENCLS'	1	0.0000	0.0000	/	

Listing A.3: Dynamica parameters of GENROU+STAB2A+IEEET1+LDFRAL test

1	1	'GENROU'	1	5.0000	0.50000E-01	0.70000	0.10000
2		4.0000		0.0000	1.4100	1.3500	0.30000
3		0.60000		0.20000	0.12000	0.10000	0.50000/
4	1	'STAB2A'	1	1.0000	4.4000	10.000	1.8000
5		1.0000		1.4100	0.10000E-01	0.50000E-01/	
6	1	'IEEET1'	1	0.0000	50.000	0.50000	3.0000
7		-3.0000		0.0000	0.80000	0.78000E-01	0.72600
8		0.0000		2.4000	0.30000E-01	5.0000	0.50000/
9	3	'GENCLS'	1	0.0000	0.0000	/	

Listing A.4: Dynamica parameters of GENSAL+IEEEST+SEXS test

1	1	'GENSAL'	1	6.7000	0.28000E-01	0.35000E-01	4.4100
---	---	----------	---	--------	-------------	-------------	--------

A. Appendix

2		0.0000	1.2200	0.76000	0.29700	0.24000
3		0.12200	0.18600	0.80200	/	
4	1	'IEEEST' 1	1	0	0.0000	0.0000
5		0.0000	0.0000	0.0000	0.0000	0.1535
6		0.05060	0.8971	1	1.5	1.5
7		15	0.1	-0.0500	0.0000	0.00000/
8	1	'SEXS' 1	0.10000	10.000	100.00	0.0200
9		0.0000	4.7300	/		
10	3	'GENCLS' 1	0.0000	0.0000	/	

Listing A.5: Dynamica parameters of GENSAL+SCRX+HYGOV test (rcrfd 0)

1	1	'GENSAL' 1	6.7000	0.28000E-01	0.35000E-01	4.4100
2		0.0000	1.2200	0.76000	0.29700	0.20000
3		0.12000	0.18600	0.80200	/	
4	1	'SCRX' 1	0.10000	10.000	100.00	0.50000E-01
5		0.0000	5.0000	0.0000	0.0000	/
6	1	'HYGOV' 1	0.50000E-01	0.30000	5.0000	0.50000E-01
7		0.50000	0.20000E-01	0.41500	0.0000	1.2500
8		1.2000	0.20000	0.80000E-01/		
9	3	'GENCLS' 1	0.0000	0.0000	/	

Listing A.6: Dynamica parameters of GENSAL+SCRX+HYGOV test (rcrfd 10)

1	1	'GENSAL' 1	6.7000	0.28000E-01	0.35000E-01	4.4100
2		0.0000	1.2200	0.76000	0.29700	0.20000
3		0.12000	0.18600	0.80200	/	
4	1	'SCRX' 1	0.10000	10.000	100.00	0.50000E-01
5		0.0000	5.0000	0.0000	10.000	/
6	1	'HYGOV' 1	0.50000E-01	0.30000	5.0000	0.50000E-01
7		0.50000	0.20000E-01	0.41500	0.0000	1.2500
8		1.2000	0.20000	0.80000E-01/		
9	3	'GENCLS' 1	0.0000	0.0000	/	

Listing A.7: Dynamica parameters of GENSAL test

1	1	'GENSAL' 1	5.0000	0.50000E-01	0.10000	4.0000
2		0.0000	1.4100	1.3500	0.30000	0.20000
3		0.12000	0.10000	0.50000/		
4	3	'GENCLS' 1	0.0000	0.0000/		

Listing A.8: Dynamica parameters of test including three windings transformer

1	3	'GENCLS' 1	0.0000	0.0000/		
2	102	'GENROU' 1	5.0000	0.50000E-01	0.70000	0.10000
3		4.0000	0.0000	1.4100	1.3500	0.30000
4		0.60000	0.20000	0.12000	0.10000	0.50000/
5	102	'STAB2A' 1	1.0000	4.4000	10.000	1.8000
6		1.0000	1.4100	0.10000E-01	0.50000E-01/	
7	102	'IEEET1' 1	0.0000	126.00	0.50000	3.0000
8		-3.0000	0.0000	0.80000	0.78000E-01	0.72600
9		0.0000	2.4000	0.30000E-01	5.0000	0.50000/

Listing A.9: Dynamica parameters of AKD test

1	58553	'GENSAL'	1	7.34	.05	.25	2.31	0
2	1.202	.658	.306	0.19400	.113			
3	.0800	.245/						
4	58573	'GENSAL'	1	7.34	.05	.25	2.31	0
5	1.202	.658	.306	0.19400	.113			
6	.0800	.245/						
7	58583	'GENSAL'	1	7.34	.05	.25	2.31	0
8	1.202	.658	.306	0.19400	.113			
9	.0800	.245/						
10	58566	'GENSAL'	1	7.34	.05	.25	2.31	0
11	1.202	.658	.306	0.18700	.113			
12	.0800	.245/						
13	58566	'IEEET2'	1	.06	150	0	4	
14	-4	1	0.04	.011	0.4	.7		
15	6.5000	.54000E-01	8.0000	.20200	/			
16	58566	'IEEEST'	1	3	0	3	0	0
17	0	0	0	10000	3	1	3	0.001
18	3	-13.5	0.03	-0.03	0	0	/	
19	58566	'HYGOV'	1	0.08	0.4	8.0	0.05	0.2
20	0.2	1.0	0.0	1.2	1.1	0.3	0.1/	
21	58567	'GENSAL'	1	7.90	.05	.25	2.93	0
22	1.13	.632	.310	.19400	.123			
23	.0607	.232/						
24	58567	'IEEET2'	1	.06	150	0	4	
25	-4	1	0.04	.011	0.4	.7		
26	6.5000	.54000E-01	8.0000	.20200	/			
27	58567	'IEEEST'	1	3	0	3	0	0
28	0	0	0	10000	3	1	3	0.001
29	3	-13.5	0.03	-0.03	0	0/		
30	58567	'HYGOV'	1	0.08	0.4	8.0	0.05	0.2
31	0.2	1.0	0.0	1.2	1.1	0.3	0.1/	

B

Appendix

References

- [1] iTesla, url=<http://www.itesla-project.eu/>.
- [2] Modelica, url=<https://www.modelica.org>.
- [3] Dymola, Dynasim, url=www.dynasim.se.
- [4] Dymola Manual Version 5.3a, Dynamic Modeling Laboratory, Dynasim AB, Sweden.
- [5] Panciatici, P.; Chieh, AS., "Equation-based hybrid modeling of power systems for time-domain simulation," *Power and Energy Society General Meeting, 2011 IEEE* , vol., no., pp.1,9, 24-29 July 2011. doi: 10.1109/PES.2011.6039155
- [6] P.Kundur, *Power System Stability and Control*, McGraw-Hill, 1993.
- [7] Federico Milano, *Power System Modelling and Scripting*, Springer, 2010.
- [8] IEEE Standard Board, "IEEE Guide for Synchronous Generator Modeling Practices in Stability Analyses", *IEEE Inc* 1991.
- [9] Siemens PTI, "PSSE Program Application Guide v.30.2", Siemens Power Transmission & Distribution Inc., Power Technologies International, 2005.
- [10] Siemens PTI, "PSSE Program Operation Manual v.30.2", Siemens Power Transmission & Distribution Inc., Power Technologies International, 2005.
- [11] Siemens PTI, "PSSE User Manual v.30.2", Siemens Power Transmission & Distribution Inc., Power Technologies International, 2005.
- [12] Fritzson, Peter. "Principles of object-oriented modeling and simulation with Modelica 2.1," John Wiley & Sons, 2010.
- [13] Modelica Association, "Modelica® - A Unified Object-Oriented Language for Physical Systems Modeling, Version 3.2, March 24, 2010.
- [14] Panciatici, P.; Chieh, A.S., "Equation-based hybrid modeling of power systems for time-domain simulation," *Power and Energy Society General Meeting, IEEE* , vol., no., pp.1,9, 24-29 July 2011 doi: 10.1109/PES.2011.6039155
- [15] Chieh, A.S., Panciatici, P., Picard, J., "Power system modeling in Modelica for time-domain simulation," *PowerTech, 2011 IEEE Trondheim*, vol., no., pp.1,8, 19-23 June 2011.

- [16] Vanfretti, L., Li, W., Bogodorova, T., “Panciatici, P., unambiguous power system dynamic modeling and simulation using modelica tools,” *Power and Energy Society General Meeting (PES), 2013 IEEE*, pp.1,5, 21-25 July 2013.
- [17] Richard P.Schulz, “Synchronous Machine Modeling”, General Electric Company, Schenectady, N.Y. *IEEE* 1975.
- [18] IEEE Standard Board, “IEEE Guide for Synchronous Generator Modeling Practices in Stability Analyses”, *IEEE Inc* 1991.
- [19] IEEE African, “Assessment of Commercially Available Software Tools for Transient Stability : Experience Gained in an Academic Environment” , *IEEE Inc*, 2004.
- [20] “IEEE Recommended Practice for Excitation System Models for Power System Stability Studies,” IEEE Std 421.5-2005(Revision of IEEE Std 421.5-1992) , vol., no., pp.0 1,85, 2006, doi=10.1109/IEEESTD.2006.99499.
- [21] Klempner, G. and Kerszenbaum, I.,“Operation and Maintenance of Large Turbo-Generators” ,*IEEE Press Series on Power Engineering*,Wiley,2004, url=http://books.google.se/books?id=DPdkLp_crLAC
- [22] Carlos Grande-Moran,“Modeling of Three-Winding Voltage Regulating Transformers for Positive Sequence Load Flow Analysis in PSS®E” ,*Siemens Energy, Inc.*,no.106,pp.1-11,July 2010.
- [23] Milano, F., “Hybrid Control Model of Under Load Tap Changers,” *Power Delivery, IEEE Transactions on*, vol.26, no.4, pp.2837,2844, Oct. 2011.doi=10.1109/TPWRD.2011.2167521 .
- [24] Calovic, M.S., “Modeling and Analysis of Under-Load Tap-Changing Transformer Control Systems,” *Power Apparatus and Systems, IEEE Transactions on*, vol.PAS-103, no.7, pp.1909,1915, July 1984, doi=10.1109/TPAS.1984.318657, url=http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=4112762isnumber=4112704
- [25] Turbine & Turbine Governor modeling, Istanbul, May 2011, url=http://64.190.51.133/Programs/Blacksea/Dynamic
- [26] IEEE Committee Report. “Dynamic Models for Steam and Hydro Turbines in Power System Studies”, *IEEE Trans.*, Vol. PAS-92. pp. 1904-1905, November/ December 1973.
- [27] IEEE Working Group Report. “Hydraulic Turbine and Turbine Control Models for System Dynamic Studies ”, *IEEE Trans.*, Vol. PWRS-7, No.1, pp.167-179, February 1992.