DEGREE PROJECT, IN SMART ELECTRICAL NETWORKS AND SYSTEMS , SECOND LEVEL

STOCKHOLM, SWEDEN 2015

Phasor Time-Domain Power System Modeling and Simulation using the Standardized Modelica Language: Conventional and Power Electronic-Based Devices

MOHAMMED AHSAN ADIB MURAD

KTH ROYAL INSTITUTE OF TECHNOLOGY

ELECTRICAL ENGINEERING

Phasor Time-Domain Power System Modeling and Simulation using the Standardized Modelica Language: Conventional and Power Electronic-Based Devices

Mohammed Ahsan Adib Murad

Supervisor and Examiner: Prof.Dr.-Ing. Luigi Vanfretti, KTH.
Supervisor: Francisco Gómez, KTH.

Electric Power Systems Dept.
Royal Institute of Technology (KTH)
December 2014

Abstract

Modern electric power systems are complex networks that ensure continuous supply of electricity. For the planning and operation of these complex networks, modeling and simulations are essential to satisfy operational requirements or planning constraints. Traditionally, dynamic modeling and simulation of power systems is performed by using different and mostly incompatible software packages. This issue is currently being addressed through the Common Information Model (CIM) for power system applications, however, there are still many challenges for Power system dynamic modeling and simulation without any ambiguity.

To overcome these challenges, this thesis develops unambiguous models for consistent model exchange, which are compatible in different simulation environments that support the Modelica language. Modelica, an object-oriented equation-based standardized language, is proposed as possible solution to these challenges as it can represent and exchange dynamic models with a strict mathematical description.

In this thesis, modeling and simulation of controllable power electronic-based components and conventional components for phasor time-domain simulation is carried out using Modelica. The work in this thesis contributes to a Modelica power systems library being developed by KTH SmartTS Lab under the FP7 iTesla project and other projects supported by Statnett SF. Both software implementation in Modelica of each component, and software-to-software validation against PSAT is carried out.

Sammanfattning

Moderna elkraftsystem r komplexa ntverk som skerstller en kontinuerlig tillfrsel av elkraft. Fr att kunna uppfylla de tekniska krav och begrnsningar som stlls vid planering och drift r modellering och simulering av dessa komplexa ntverk av avgrande betydelse. Vanligtvis sker modellering och simulering av kraftsystemets dynamik i olika och oftast inkompatibla programvaror. Detta problem sker Common Information Model (CIM) fr kraftsystemkomponenter att lsa, men mnga hinder kvarstr innan dynamisk modellering och simulering av kraftsystemet kan ske utan tvetydigheter.

Fr att lsa dessa utmaningar s utvecklar detta examensarbete entydiga modeller fr ett konsekvent utbyte av modeller mellan simuleringsprogramvaror som stdjer modelleringssprket Modelica. Modelica, ett standardiserat, objektorienterat och ekvationsbaserat sprk, fresls som en mjlig lsning fr dessa utmaningar d det kan representera och utbyta modeller genom en strikt matematisk beskrivning.

I detta examensarbete utfrs modellering och simulering i Modelica av bde styrbara kraftelektronik och konventionella kraftsystemkomponenter fr tidssimulering med fasvektorer. Detta examensarbete bidrar till det Modelica bibliotek fr elkraftsystem som utvecklas av KTH SmartTS Lab i FP7 projektet iTesla och andra projekt som stttas av Statnett SF. Modellerna som utvecklas i Modelica valideras med hjlp av mjukvaru-validering med PSAT som referens.

Acknowledgement

I am grateful to **Assoc. Prof. Dr.-Ing. Luigi Vanfretti** for giving me the opportunity as his master thesis student, he has been a great support during the whole work. In addition, special gratitude and thank to my supervisor **Francisco Gómez** for his generous help, warm encouragement and support.

I would like to thank my program coordinator, Assoc. Prof. Hans Edin, for his guidance for the past two years of my master's program. In addition, special thank to EIT/KIC-InnoEnergy for funding my studies in Smart Electrical Network and Systems (SENSE) program.

I would like to thank all the people who helped me and supported me during this project and special thank to Apostolis Kristal, Md Jahidul Islam Razan, Le Qi, Yuwa Choompoobutrgool, Wei Li, Jan Lavenius and all the members of the SmarTS Lab. Without the help from the group members, the completion of this thesis would not have been possible. Finally, I would like to thank my beloved family for their support, as well as all of my friends.

Contents

List of Figures

List of Tables

Notations

PSAT	Power System Analysis Toolbox
EMT	Electro-Magnetic Transient
DAE	Differential and Algebraic Equation
GUI	Graphical User Interface
ODE	Ordinary Differential Equation
TCSC	Thyristor Controlled Series Compensator
STATCOM	Static Synchronous Compensator
SSSC	Static Synchronous source Series Compensator
UPFC	Unified Power Flow Controller
ULTC	Under Load Tap Changer
PST	Phase Shifting Transformer
TWT	Three Winding Transformer
AVR	Automatic Voltage Regulator
FACTS	Flexible AC Transmission System
IEEE	Institute of Electrical and Electronics Engineers

Chapter 1

Introduction

1.1 Background

Todays electric power systems are large and complex. These systems consist of many kinds of interconnected components to generate, transmit and distribute electrical energy continuously to a large consumers spread over the vast geographical area. Due to the interconnection of many individual components in these systems there exist a large variety of dynamics which can affect the system as a whole in many ways. These dynamics can be divided into groups distinguished by their cause, consequence, time frame, physical character or the place in the system where these dynamics occur [1]. The different Power System dynamics are:

- Electromagnetic Transients
- Electromechanical Transients
- Quasi-Steady state Dynamics

Power system modeling and simulation is essential to satisfy operational requirements or planning constraints [2]. In power system analysis, to deal with different dynamics, different kinds of models are used to simulate each of the dynamic phenomena shown in figure 1.1.

- **Electro-Magnetic Transient Model:** Electro-magnetic transient phenomena occurs in less than of a microsecond and involves the power systems response to events such as lightning strikes, switching operation etc. To model these phenomena, Electro-magnetic transient models are used. These models are described mathematically by differential and algebraic equations. To use these mathematical models in digital simulation the methods used are: state variable analysis and difference equation's [3], [4]. A well known simulation software used to analyze these models is EMTP-RV [5].

- **Electro-mechanical Transient Model:** Electro-mechanical transients occur in the range of millisecond to seconds. An example of such transient is the oscillation of the rotating masses of the generators and motors that occurs following a disturbance or due to the operation a of protection system. The mathematical models of electro-mechanical transients are simplified or averaged from electro-magnetic transient models. This simplified mathematical models are used in digital modeling. This simulation approach is known as phasor time domain simulation. The simulation software packages used to analyze this kind of models are PSAT [6], EUROSTAG [7], PSS/E [8], PSCAD [9] etc.

- **Quasi-Steady State Model:** Quasi-steady state (QSS) dynamics occur for more than one second. An example of QSS is the thermodynamic change of the boiler control action in steam power plants to meet the demand of automatic generation controls. Mathematically, in these kind of models electro-magnetic transients are neglected and the system is modeled by algebraic equations [10]. The only tool used in practice for this kind of simulations is astre [11].

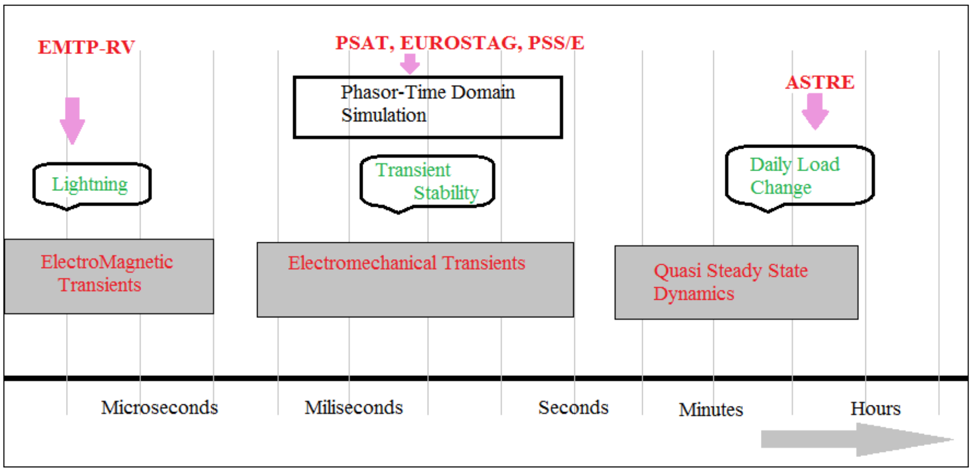


Figure 1.1: Time frame of different power system dynamics.

Though there exists a lot of simulation tools, the rapid change of grid and penetration of intermittent renewable sources making the simulation of power system challenging.

1.2 Problem Definition

Existing tools for power system phasor time domain simulation are exposed to certain limitations such as limited simulation features [2], limited abilities for consistent model exchange [12] or handling of penetration of new devices (FACTS).

Power system courses in the educational institutes deal with complex physical phenomena and detailed mathematical models of power systems. It is difficult to visualize the main concept of the cumbersome power system networks. Hence reproducing the complex power system phenomena through computer-based simulations is an efficient solution for educational and research purposes [13]. Commercially available power system simulation software packages are used in these educational institutes. Few problems with these software packages are: they do not allow changing the source code or adding new parameters [14]. In order to increase the flexibility of power system simulation softwares, an Open Source approach is taken by the power system academic community. Some examples of the Open Source software packages are UWPFLOW [15], PSAT [6], PowerWeb [16] and ObjectStab [17].

Modelica is a new promising modeling language. It is an equation based, object oriented, open source language, offering several advantages to the modeling community. It also offers a solution for covering model exchange challenges, easy modification of models, easy development of custom models and providing unambiguous simulation results among different tools [18]. One software that supports the Modelica language is OpenModelica.

At present a Power System library is developed by SmarTS Lab within the FP7 iTesla project [19]. Most of the power system component models have already been implemented in the library. Additional models are needed to perform time domain simulations of small and medium sized power system networks. The problems this thesis focuses are:

- Improving the iTesla power system library with the development of new Modelica models of power electronics based components for phasor time domain simulation.
- Validating the models and making validated test system models.

- Investigating the feasibility of the models to be applied to the small and medium power system networks such as IEEE 9-bus system and IEEE 14-bus system.

1.3 Objectives

Taking into account the problems described in the previous section, the following objectives are identified for this work:

- Literature review on power system simulation methods and the contribution of iTesla project in this field.
- How to use Modelica and PSAT for power system simulation.
- Develop the Modelica models of the Flexible AC Transmission System (FACTS) based devices, conventional power system devices (Transformers) which are used for hybrid electromagnetic and electromechanical power system models.
- Checking the validity of all the models against PSAT. PSAT contains the implementation of the models, which will be used in this work. So, a validation of the simulation outputs from the Modelica models is necessary to check the same behavior according to PSAT simulations.
- Prove the feasibility of the models into small and medium power system networks like IEEE 9-bus and IEEE 14-bus systems.

1.4 Overview of the Report

This report is divided into three sections. In the first section, involving chapter 1 and 2, background of the project with an introduction about Modelica and PSAT is given. In the second section, Power System component modeling in Modelica is discussed. In this section the modeling methodology, mathematical models of the components, implementation of the models, test system for the models and software to software validation results are given. This section covers the chapter 3 and 4. Finally in the third section, a small discussion about the experience with Modelica and future work are summarized. This part covers the chapter 5.

Chapter 2

Modeling Languages

2.1 Modelica

Modelica is an object oriented, acuasal, equation based, open source language for describing complex mathematical behavior. It supports dynamic modeling and simulation, for complex systems and applications from different domains such as Electrical, Mechanical, Hydraulic, Thermal, Control and Electric Power Engineering. Modelica models are described in schematics (see figure 2.1).

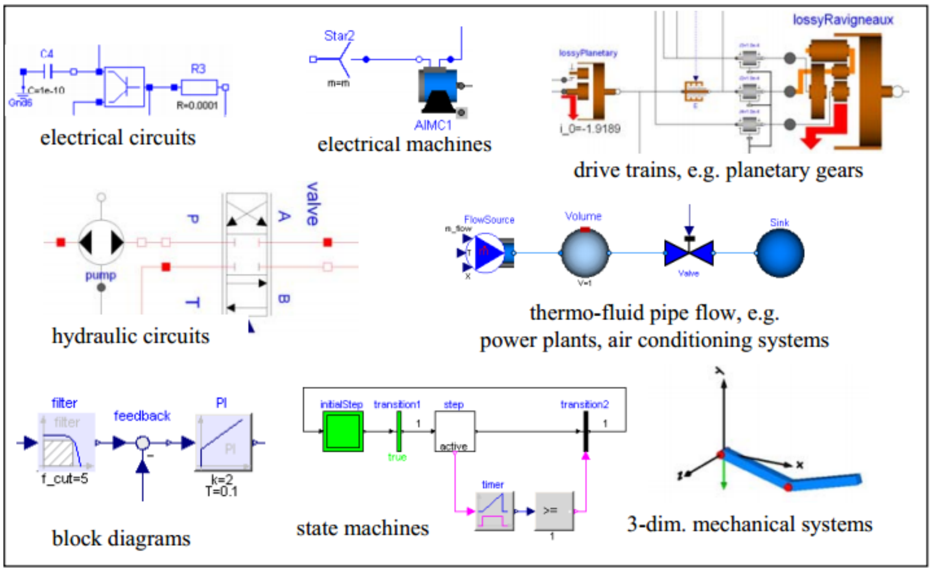


Figure 2.1: Examples of Modelica Models [24].

Modelica Design Group develops the free library for multi-domain modeling known as Modelica Standard Library. All the versions of the Modelica Language are available on-line (https://modelica.org/).

2.2 Modelica Features

Modelica offers several advantages to the modeling community both in academia and industry. Being a standard language, Modelica is supported by different modeling and simulation tools [23]. Most important features of Modelica are [20], [21]:

- Modelica is based on equation statements instead of assignment statements. That is why, it permits acausal modeling.
- Different domains such as Electrical, Mechanical, Thermodynamic, Hydraulic, Biological and Control applications can be connected and used in the same model.
- It provides hierarchical system architecture capabilities and visual component programming.
- Allows the exchange of the models among simulation solvers, which are able to compile Modelica code and provides unambiguous simulation results among different tools.
- Modelica models are solver independent.
- Modelica is an object-oriented language with universal class concept that unifies classes, generics-known as templates in C++ and general subtyping into a single language construct. This helps evolution of models.

2.3 Modelica Simulation Environment

Modelica language is used for modeling complex mathematical problems. Different Modelica simulation environments allow one to make these models using Graphical User Interface (GUI) editors, text-based editors or both. The currently available simulation environments for Modelica are given in the following table 2.1. In this work Dymola [22] and Open Modelica simulation environment is used.

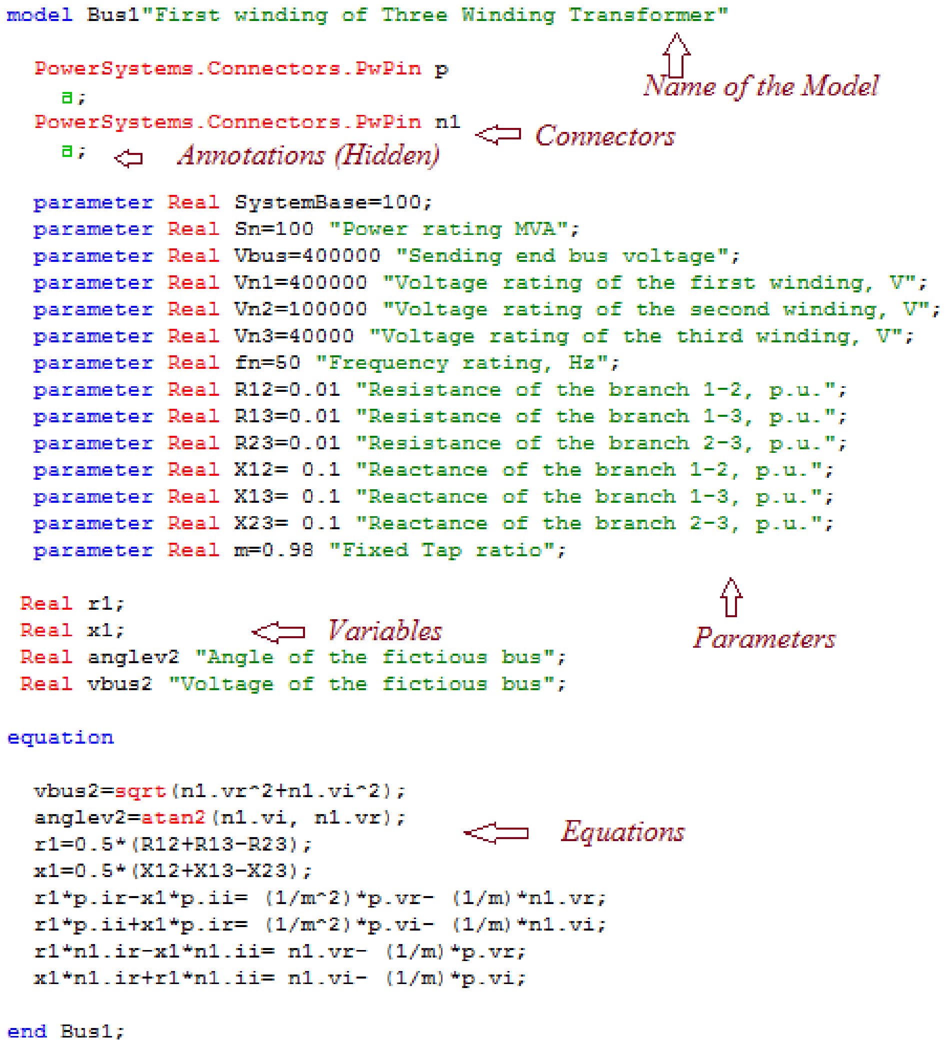
Table 2.1: Modelica based software environments.

Commercial Modelica Simulation Environment		**Free Modelica Simulation Environment**	
Created By	***Name***	***Created By***	***Name***
CyDesign Labs	CyModelica, Vertex, Converge	Modelon AB	JModelica.org
Dassault Systèmes	Dymola	Linköping University	OpenModelica
ITI GmbH (Germany)	Simulation X	INRIA (France)	SCICOS
LMS	LMS Imagine.Lab AMESim		
Modelon AB	OPTIMICA Studio		
Wolfram	Wolfram SystemModeler		
MapleSoft (Canada)	MapleSim		

2.4 Modelica Programming

Modelica programming can be done by using Graphical editor or writing the code in Textual editor. In graphical editor a new model can be created by dragging and dropping the models from the available models. Textual editor is used for writing the executable code. That means the programming is done in the textual editor to create the models. In the text editor model name is declared first, then the variables are declared. Finally, the equations are added with initialization. Detail programming in Modelica text editor can be found in [20], [21], [24].

The simulation editor is used to translate and simulate the models. Simulation editor also used to visualize the results. One of the simple example of Modelica code is given in figure 2.2.



```
model Bus1"First winding of Three Winding Transformer"

  PowerSystems.Connectors.PwPin p
    a;
  PowerSystems.Connectors.PwPin n1
    a;

  parameter Real SystemBase=100;
  parameter Real Sn=100 "Power rating MVA";
  parameter Real Vbus=400000 "Sending end bus voltage";
  parameter Real Vn1=400000 "Voltage rating of the first winding, V";
  parameter Real Vn2=100000 "Voltage rating of the second winding, V";
  parameter Real Vn3=40000 "Voltage rating of the third winding, V";
  parameter Real fn=50 "Frequency rating, Hz";
  parameter Real R12=0.01 "Resistance of the branch 1-2, p.u.";
  parameter Real R13=0.01 "Resistance of the branch 1-3, p.u.";
  parameter Real R23=0.01 "Resistance of the branch 2-3, p.u.";
  parameter Real X12= 0.1 "Reactance of the branch 1-2, p.u.";
  parameter Real X13= 0.1 "Reactance of the branch 1-3, p.u.";
  parameter Real X23= 0.1 "Reactance of the branch 2-3, p.u.";
  parameter Real m=0.98 "Fixed Tap ratio";

 Real r1;
 Real x1;
 Real anglev2 "Angle of the fictious bus";
 Real vbus2 "Voltage of the fictious bus";

equation

  vbus2=sqrt(n1.vr^2+n1.vi^2);
  anglev2=atan2(n1.vi, n1.vr);
  r1=0.5*(R12+R13-R23);
  x1=0.5*(X12+X13-X23);
  r1*p.ir-x1*p.ii= (1/m^2)*p.vr- (1/m)*n1.vr;
  r1*p.ii+x1*p.ir= (1/m^2)*p.vi- (1/m)*n1.vi;
  r1*n1.ir-x1*n1.ii= n1.vr- (1/m)*p.vr;
  x1*n1.ir+r1*n1.ii= n1.vi- (1/m)*p.vi;

end Bus1;
```

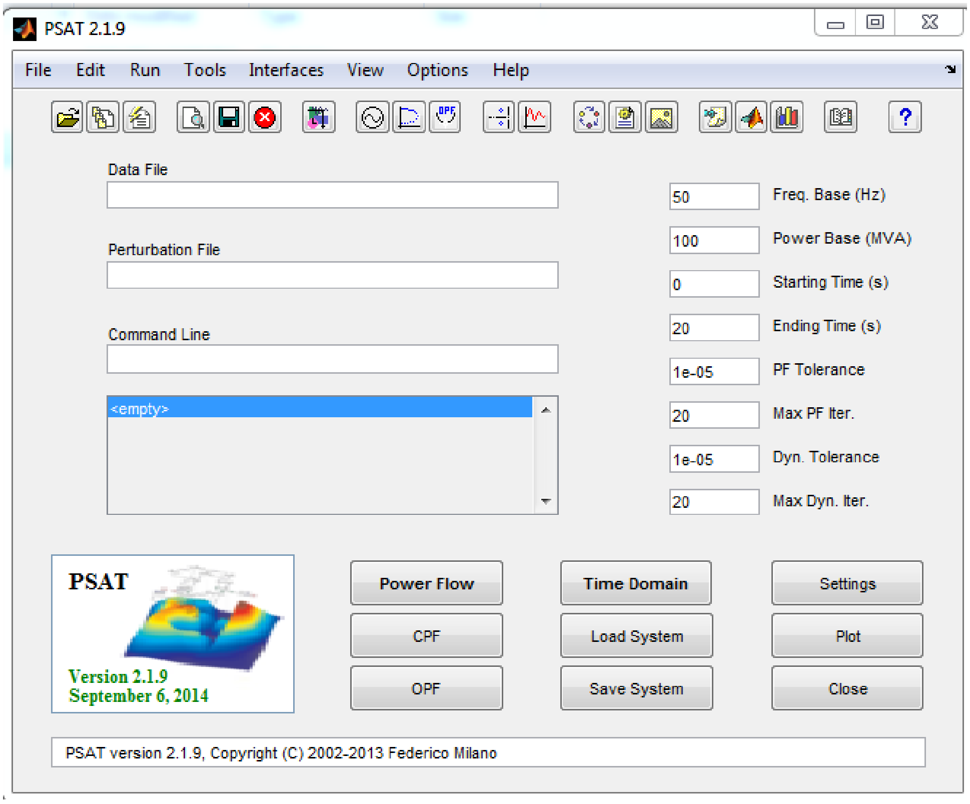
Figure 2.2: Textual programming in Modelica.

2.5 Power System Analysis Toolbox (PSAT)

PSAT is a Matlab based toolbox for static and dynamic analysis of power system [6]. As a software tool for power systems analysis, PSAT offers various routines:

- Continuation Power Flow
- Optimal Power Flow
- Small Signal Stability Analysis
- Time Domain Simulations
- Phasor Measurement Unit (PMU) placement

The main graphical user interface of PSAT is shown in figure 2.3. To make power system networks in PSAT, one can use data files or single-line diagrams via the GUI (using PSAT-simulink library). Then single line network or simulink diagram, is loaded via the data file field in the main GUI. The diagram must be saved before loading. After that this diagram is translated into PSAT readable data file and then any of available routines can be simulated.



Data File

Perturbation File

Command Line

<empty>

Freq. Base (Hz): 50
Power Base (MVA): 100
Starting Time (s): 0
Ending Time (s): 20
PF Tolerance: 1e-05
Max PF Iter.: 20
Dyn. Tolerance: 1e-05
Max Dyn. Iter.: 20

PSAT version 2.1.9, Copyright (C) 2002-2013 Federico Milano

Figure 2.3: Graphical User Interface of PSAT.

PSAT contains different power system components models that have not yet been modeled using Modelica. Thus, PSAT models are taken as the reference models, in order to compare their performance against the equivalent Modelica models, developed in this work.

Chapter 3

Power System Component Modeling

3.1 Introduction

In the context of the iTesla project a Modelica library is developed with different power system components. Each of these components have been validated against their implementation in a most reliable proprietary software such as: Eurostag, PSS/E and PSAT. The steps involved in the developing of the components successfully are:

- Read the documentation and understand the mathematical and conceptual background of the component.
- Identify all the equations, explaining the dynamic behavior of the component.
- Make a list of the parameters and variables used in these equations.
- Write the Modelica code in the text layer and use the Modelica Standard Library when necessary.
- Initialize the model by identifying the initial conditions for both parameters and variables.
- Carry out a software to software validation of the Modelica models against the reference models.

This work is intended to develop more components to add in the library, having PSAT models as reference models. In this chapter, a detailed description of the developed models and the implementation procedure in Modelica is given.

3.2 Power System Modeling

In order to develop a new power system model in Modelica three of the following items are required.

- Components
- A connection mechanism
- A component framework

Each component is an instance of a Modelica class. Components are connected with the connection mechanism known as connection diagram. One connection diagram is shown in figure 3.1. The connection of components is done using connectors. Connectors are used to create the communication link between the components. The component framework in turn ensures that the communication between components function properly. For complex systems there exists a large number of connected components which can be hierarchically decomposed [20].

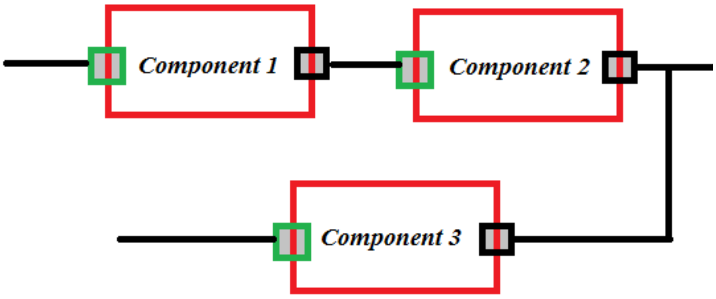


Figure 3.1: Connection diagram for components.

3.2.1 Connectors and Connection

Modelica provides the notions of classes and objects like other object oriented programming languages. In Modelica, every object is a class which defines the object behaviour and data. One of the special Modelica class is the connector class, which used to interconnect different components [25].

The PwPin (Power Pin) connector [23] class is the base connector used to connect the power system components. PwPin connector is defined with four variables: Real voltage and current (vr and ir) and Imaginary voltage and current (vi and ii). The connector is shown in figure 3.2.

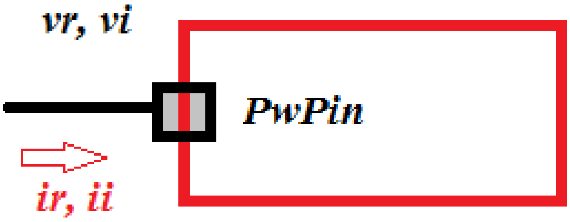


Figure 3.2: Schematic of electrical PwPin connector.

Components with the same type of connectors (PwPin) can be connected together. This procedure is represented as an equation in Modelica. A connection between two components is shown in figure 3.3.

Two types of couplings based on the variables are created after the connection (with flow prefix or without): **a. Equality coupling** : This is for non-flow variables based on Kirchhoffs first law. **b. Sum-to-zero coupling** : This is for flow variables based on Kirchhoffs current law.

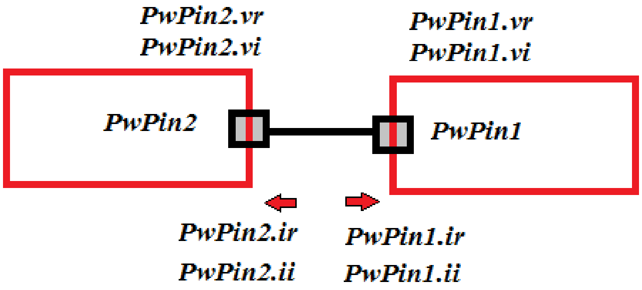


Figure 3.3: Connecting two components in Modelica.

3.2.2 Modeling Methodology

To model a power system component in Modelica some basic methodologies are followed:

- The developer can use the Modelica Standard Library to make a new model. The input and output connectors should be labeled properly, so that users can easily identify.
- The models should be organized in specific packages, e.g.: generator models in the Machine package.
- Identify the parameters and use the attribute public or protected.
- All the parameters that can be modified by the user, should propagate to the top-layer of the model, so that the user does not need to look at the bottom layer to change any value. The developer should also define default values for the parameters. The parameters should be well documented as well.
- While developing the components import all the constants from Modelica.Constants.
- The final model of the component should be a single block only. In some cases the model is composed by own developed models or other components from the library. In that case the developer must combine them in one final block with one icon.

3.3 FACTS

To facilitate Flexible AC Transmission System (FACTS) several devices are used. Such FACTS devices are based on power electronics converters. They can be classified into shunt or series devices or a combination of both [28]. The FACTS devices covered in this thesis are Static Synchronous Compensator (STATCOM), Thyristor Controlled Series Compensator (TCSC), Static Synchronous Source Series Compensator (SSSC) and Unified Power Flow Controller (UPFC). These models are averaged valued models, as in PSAT. This is because in phasor time-domain simulation the aim is to observe the phenomena of power system network with FACTS devices without describing in detail of the underlying switching of the power electronics in these devices. Next subsection covers the mathematical description of the models and the corresponding Modelica implementation.

3.3.1 Thyristor Controlled Series Compensator(TCSC)

TCSC is a kind of series regulator used to control the active and reactive power that flow through the line. The regulation is based on the insertion of a series capacitive reactance in the same line. A TCSC series controller is shown in figure 3.4.

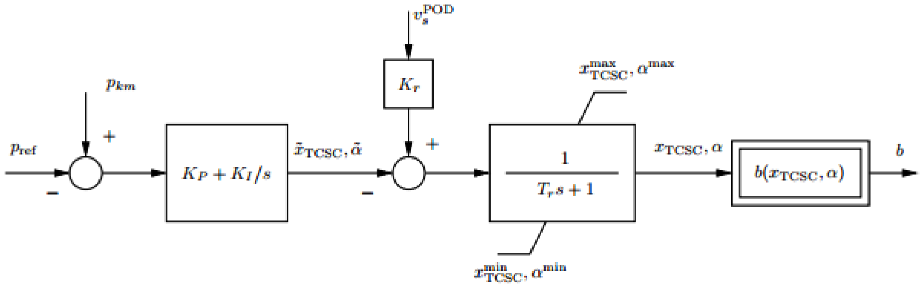


Figure 3.4: Control block diagram of TCSC [27].

The two input signals of the regulator are line power(P_{km}), reference power (P_{ref}) and the output signal is the series susceptance (b).

The differential equations of TCSC are:

$$\dot{x}_1 = ([\breve{x}_{TCSC}, \breve{\alpha}] + K_r v_s^{POD} - x_1)/T_r \tag{3.1}$$

$$\dot{x}_2 = -K_I(P_{km} - P_{ref}) \tag{3.2}$$

and

$$[\breve{x}_{TCSC}, \breve{\alpha}] = K_P(P_{km} - P_{ref}) + x_2 \tag{3.3}$$

where $\dot{x}_1$and $\dot{x}_2$ are the state variables. The series reactance $\breve{x}_{TCSC}$ and the firing angle $\breve{\alpha}$ refer to the state variable $\dot{x}_1$.

The series susceptance b is the output signal of the regulator, given by

$$b(x_{TCSC}) = -\frac{\frac{x_{TCSC}}{x_{km}}}{x_{km}(1 - \frac{x_{TCSC}}{xkm})} \tag{3.4}$$

and for firing angle regulator

$$b(\alpha) = \pi(k_x^4 - 2k_x^2 + 1)\cos k_x(\pi - \alpha)/[x_C(\pi k_x^4 \cos(\pi - \alpha) - \pi \cos k_x(\pi - \alpha)$$

$$+2k_x^4\alpha\cos k_x(\pi-\alpha)+2\alpha k_x^2\alpha\cos k_x(\pi-\alpha)-k_x^4\sin 2\alpha\cos k_x(\pi-\alpha)$$

$$+k_x^2\sin 2\alpha\cos k_x(\pi-\alpha)-4k_x^3\cos^2\alpha\sin k_x(\pi-\alpha)$$

$$-4k_x^2\cos\alpha sin\alpha\cos k_x(\pi-\alpha))] \tag{3.5}$$

here the term k_x is defined by

$$k_x=\sqrt{\frac{x_C}{x_L}}$$

3.3.2 TCSC in Modelica

Based on $b(x_{TCSC})$ and $b(\alpha)$ two models are implemented in Modelica, named TCSCReactance and another TCSCAlpha. The PwPin connector class is used in both the models. The equations are written in the textual editor.

Reactance Model: First step in the implementation of this model in Modelica, is to declare a set of parameters (see the code below):

```
model TCSCReactance

  PowerSystems.Connectors.PwPin p
    annotation (Placement(transformation(extent={{-119,-8},{-99,12}})));
  PowerSystems.Connectors.PwPin n
    annotation (Placement(transformation(extent={{100,-10},{120,10}})));

  constant Real pi=Modelica.Constants.pi;
  parameter Real SystemBase=100;
  parameter Real Vbus=400000 "Bus nominal voltage, change of base";
  parameter Real Sn=100 "Power rating, MVA";
  parameter Real Vn=400000 "Voltage rating, KV";
  parameter Real f=50 "Frequency rating, Hz";
  parameter Real Cp=0.10 "Percentage of series compensation %";
  parameter Real Tr=0.5 "Regulator time constant,s";
  parameter Real xTCSCmax=0.05 "Maximum Reactance";
  parameter Real xTCSCmin= -0.05 "Minimum Reactance";
  parameter Real Kp=5 "Proportional gain of PI controller p.u./p.u.";
  parameter Real Ki=1 "Integral gain of PI controller p.u./p.u.";
  parameter Real Kr=10 "Gain of the stabilizing signal p.u./p.u.";
  parameter Real Vs_POD=0 "Power oscillation damper signal";
  parameter Real x_L=0.1 "Line reactance p.u.";
```

```
  parameter Real pref= 0.080101913348342 "Reference Power flow, p.u.";
  parameter Real rL=0.01 "Line resistance, p.u.";
  parameter Real x_TCSC0=0.01 "Initial series reactance TCSC p.u.";
  parameter Real x20=0.01 "Initial valu of the state varible x2";
  Real vk " Nominal bus voltage of bus k ";
  Real vm " Nominal bus voltage of bus m";
  Real pkm "Active power flow from bus k to m";
  Real b;
  Real x_TCSC;
  Real x2(start=x20);
  Real x0;
```

To change the value of the parameters with the system base, some parameters are defined as protected. These are given below:

```
protected
  parameter Real Vb2new=Vbus*Vbus;
  parameter Real Vb2old=Vn*Vn;
  parameter Real R=rL*(Vb2old*SystemBase)/(Vb2new*Sn);
  parameter Real X=x_L*(Vb2old*SystemBase)/(Vb2new*Sn);
  parameter Real xTCSC_max=xTCSCmax*(Vb2old*SystemBase)/(Vb2new*Sn);
  parameter Real xTCSC_min=xTCSCmin*(Vb2old*SystemBase)/(Vb2new*Sn);
  parameter Real y=1/X;
```

After declaring all the parameters needed, the next step is to declare the equations of the model. So, the equations for the state variables and series susceptance, (equation 3.1, 3.2, 3.3 and 3.4) are declared in the equation section of the Modelica code. The limiters of the regulator are implemented using a corresponding if statement. Initialization of the state variables and reference power are taken directly from the power flow solution, calculated in PSAT. Finally, the series compensation is declared by adding the susceptance value within the line admittance. The equation section is added below:

```
initial equation
x_TCSC=x_TCSC0;

equation
    vk=sqrt(p.vr^2+p.vi^2);
    vm=sqrt(n.vr^2+n.vi^2);
    if (x_TCSC>xTCSC_max) and (der(x_TCSC)>0) and (der(x2)>0) then

     der(x_TCSC)=0;
```

```
     der(x2)=-Ki*( pkm-pref);
     b= -(xTCSC_max/X)/ (X*(1-(xTCSC_max/X)));

     elseif (x_TCSC<xTCSC_min) and (der(x_TCSC)<0) and (der(x2)<0) then

     der(x_TCSC)=0;
     der(x2)=-Ki*( pkm-pref);
     b=  -(xTCSC_min/X)/ (X*(1-(xTCSC_min/X)));

     else

    der(x_TCSC)= (Kr*Vs_POD-Kp*(pkm-pref)+x2-x_TCSC)/Tr;
    der(x2)=-Ki*( pkm-pref);
    b= - (x_TCSC/X)/ (X*(1-(x_TCSC/X)));

     end if;

    pkm=(p.vr*p.ir + p.vi*p.ii);
    x0=-(Kp*(pkm-pref)-x2);
    p.ii=(y-b)*(n.vr-p.vr);
    p.ir=(y-b)*(p.vi-n.vi);
    n.ii=(y-b)*(p.vr-n.vr);
    n.ir=(y-b)*(n.vi-p.vi);
```

Alpha Model: As this model is implemented in the same way as the reactance model, the code is not included here.

3.3.3 Static Synchronous Compensator(STATCOM)

STATCOM is a kind of shunt connected FACTS device used to regulate reactive power by using a Voltage Source Converter (VSC). Here the STATCOM model is a simplified current injection model shown in figure 3.5 [27].

The differential equation of the STATCOM is:

$$\dot{i}_{SH} = (K_r(v_{ref} + v_s^{POD} - v) - i_{SH})/T_r \tag{3.6}$$

The reactive power injection:

$$q = i_{SH}v \tag{3.7}$$

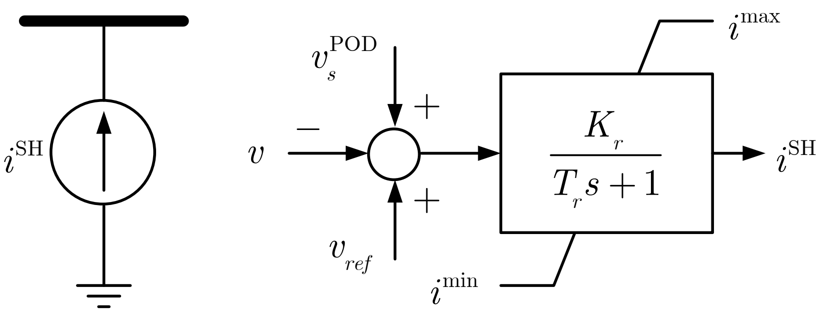


Figure 3.5: Control block diagram of STATCOM [27].

This model has three input signals: bus voltage (v), reference voltage (v_{ref}) and the power oscillation damper signal (v_s^{POD}). The output signal is the shunt current (i_{SH}) which is injected to the connected bus.

3.3.4 STATCOM in Modelica

Based on the control diagram of the STATCOM two models are created in Modelica. To show different ways of modeling with Modelica, one of the models is developed using the graphical editor (based on block diagram) and the other is developed using the textual editor (based on equations). The graphical editor model is shown in figure 3.6.

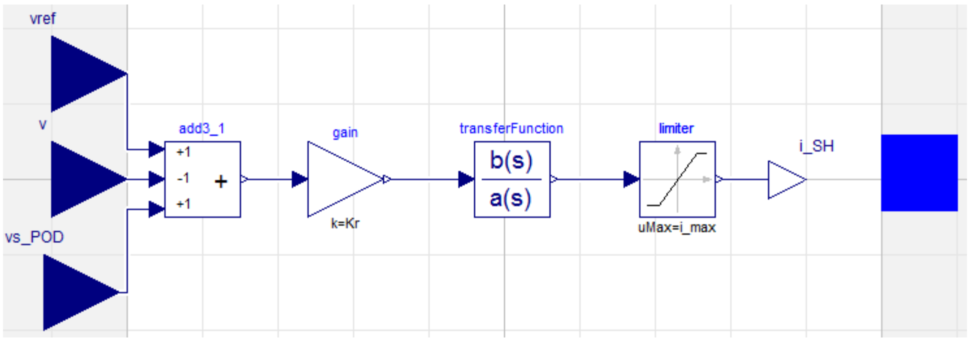


Figure 3.6: STATCOM GUI based model in Modelica.

In the text based model the state variable and reactive power injection are calculated using the equation 3.6 and 3.7. Then reactive power injection is related to the PwPin variables. The limiters of the regulator are implemented using the if statement. The Modelica code is given below:

```
model STATCOM "Static Synchronous Compensator model with equation"

  PowerSystems.Connectors.PwPin p(vr(start=vr0), vi(start=vi0))

  constant Real pi=Modelica.Constants.pi;
  parameter Real SystemBase=100;
  parameter Real Vbus=400000 "Bus nominal voltage, V";
  parameter Real Sn=100 "Power rating, MVA";
  parameter Real Vn=400000 "Voltage rating, V";
  parameter Real fn=50 "Frequency rating, Hz";
  parameter Real Kr=50 "Regulator gain, p.u./p.u.";
  parameter Real Tr=0.01 "Regulator time constant, s";
  parameter Real i_Max= 0.7 "Maximum current, p.u.";
  parameter Real i_Min= -0.1 "Minimum current, p.u.";
  parameter Real v_ref=1.002791151905167
    "Reference voltage of the STATCOM regulator, from power flow";
  parameter Real v_POD=0 "Power oscillation damper signal";
  parameter Real v0=1 "Initial value of the bus voltage connected to STATCOM";
  parameter Real anglev0=-0.000213067852480
    "Initial angle of the Bus connected to STATCOM, from power flow";

  Real i_SH;
  Real v(start=v0);
  Real Q;

protected
  parameter Real Iold=Sn/Vn;
  parameter Real Inew=SystemBase/Vbus;
  parameter Real i_max=i_Max*Iold/Inew;
  parameter Real i_min=i_Min*Iold/Inew;
  parameter Real vr0=v0*cos(anglev0) "Initialitation";
  parameter Real vi0=v0*sin(anglev0) "Initialitation";
  parameter Real io= Kr*(v_ref+v_POD-v0) "Initialization";

initial equation

 i_SH=io;

equation

 v=sqrt(p.vr^2+p.vi^2);
 0=p.vr*p.ir + p.vi*p.ii;
-Q=p.vi*p.ir - p.vr*p.ii;

if (i_SH>i_max) and (der(i_SH)>0) then
```

```
  der(i_SH)=0;
  Q=i_max*v;

 elseif (i_SH<i_min) and (der(i_SH)<0) then

   der(i_SH)=0;
   Q=i_min*v;

  else

  der(i_SH)= (Kr*(v_ref+v_POD-v)-i_SH)/Tr;
  Q=i_SH*v;

  end if;

end STATCOM;
```

3.3.5 Static Synchronous Source Series Compensator(SSSC)

The SSSC is a kind of the series connected FACTS device, which is represented by a series voltage source (see figure 3.7).

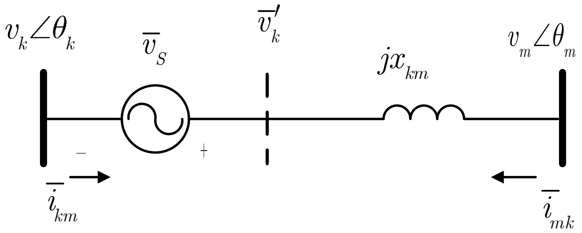


Figure 3.7: Circuit diagram of SSSC [27].

The controllable parameter of SSSC is the magnitude of $\bar{v}_S$, which is kept always in quadrature relation with the line current. The dynamic control block is shown in figure 3.8.

The input signals are line power(P_{km}) and reference power (P_{ref}), and the output signal is the series voltage ($\bar{v}_S$) which is injected in the line. The dynamic behavior of

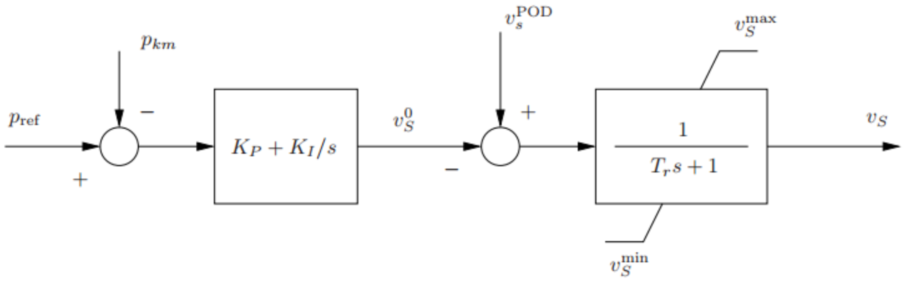


Figure 3.8: Control block diagram of SSSC [27].

the SSSC is described by the following differential equation (calculated from the output of the control block).

$$\dot{v}_S = (v_s^o + v_s^{POD} - v_s)/T_r \tag{3.8}$$

The SSSC control depends on the input signal v_s^o. Two different control modes are implemented in Modelica: 1) constant voltage 2) constant power flow.

Constant Voltage: In this control mode, the input v_s^o is constant and it does not depend on the line current.

Constant Power Flow: In this control mode, the input signal v_s^o is the output of PI controller as shown in figure 3.8. The equations describing the output for this control are :

$$v_s^o = K_p(p_{ref} - p_{km}) + v_{pi} \tag{3.9}$$

and

$$\dot{v}_{pi} = K_I(p_{ref} - p_{km}) \tag{3.10}$$

3.3.6 SSSC in Modelica

The SSSC model created in Modelica has been developed considering two different control modes: 1) Constant voltage and 2) Constant power flow. For both control modes, the code implementation of the SSSC model is same. So only constant power flow model implementation is discussed here.

In the Modelica model the parameters are declared first. In the equation section, the equations for the input signals: equations 3.9 and 3.10 are written directly. Using these two inputs, the output of the control block from equation 3.8 is calculated. This output is the voltage, which is injected into the line. This voltage is in quadrature with the current in the line, so the current angle is calculated. Then, this angle is imposed with the voltage. In the end the injected voltage is related to the PwPin connectors. The code of the SSSC for constant power flow mode is given below without the declaration of the parameters:

```
model SSSC_CPF

  PowerSystems.Connectors.PwPin p
  PowerSystems.Connectors.PwPin n

initial equation
 vpi=vpi0;

equation

when sample(0,0.005) then
  ip=pre(p.ir);
  iq=pre(p.ii);
end when;

 itheta=atan2(ip,iq);
 vk=sqrt(p.vr^2+p.vi^2);
 vm=sqrt(n.vr^2+n.vi^2);
 pkm=(p.vr*p.ir + p.vi*p.ii);

   if (vs>vs_max) and (der(vs)>0) then
 der(vs)=0;
 der(vpi)=(p_ref-pkm)*Ki;
 vs0=(p_ref-pkm)*Kp+vpi;
 vp= abs(vs_max)*sin(itheta);
 vq= abs(vs_max)*cos(itheta);

 elseif (vs<vs_min) and (der(vs)<0) then
 der(vs)=0;
 der(vpi)=(p_ref-pkm)*Ki;
 vs0=(p_ref-pkm)*Kp+vpi;
 vp= abs(vs_min)*sin(itheta);
 vq= abs(vs_min)*cos(itheta);

   else
 der(vs)=(vs0+vs_POD-vs)/Tr;
 der(vpi)=(p_ref-pkm)*Ki;
 vs0=(p_ref-pkm)*Kp+vpi;
```

```
vp= abs(vs)*sin(itheta);
vq= abs(vs)*cos(itheta);
end if;

 if vs>=0 then
   vq=p.vr-n.vr;
   vp=n.vi-p.vi;
  else
   vq=n.vr-p.vr;
   vp=p.vi-n.vi;
end if;
p.ir=-n.ir;
p.ii=-n.ii;

 annotation
end SSSC_CPF;
```

3.3.7 Unified Power Flow Controller(UPFC)

The UPFC is a shunt and series connected device which is modeled as a combination of STATCOM and SSSC shown in figure 3.9. Here $\bar{v}_S$ is the series voltage source and $\bar{i}_{SH}$ is the shunt current source.

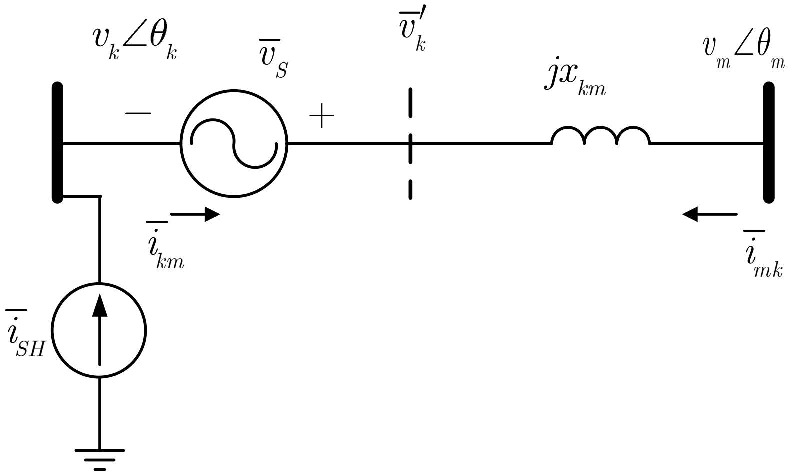


Figure 3.9: Circuit diagram of UPFC [27].

The equation of the series and shunt sources are

$$\bar{v}_S = (v_p + v_q)e^{j\phi} \tag{3.11}$$

and

$$\bar{i}_{SH} = (i_p + i_q)e^{j\theta_k} \tag{3.12}$$

The differential equations of the dynamic UPFC model are

$$\dot{v}_p = (v_{p0} + u_1 v_s^{POD} - v_p)/T_r \tag{3.13}$$

$$\dot{v}_q = (v_{q0} + u_2 v_s^{POD} - v_q)/T_r \tag{3.14}$$

$$\dot{i}_q = [K_r(v_{ref} + u_3 v_s^{POD} - v_k) - i_q]/T_r \tag{3.15}$$

If the power oscillation damping signal is given then u_1, u_2 and u_3 are 1 otherwise 0. The dynamic model is described by the control blocks shown in figure 3.10.

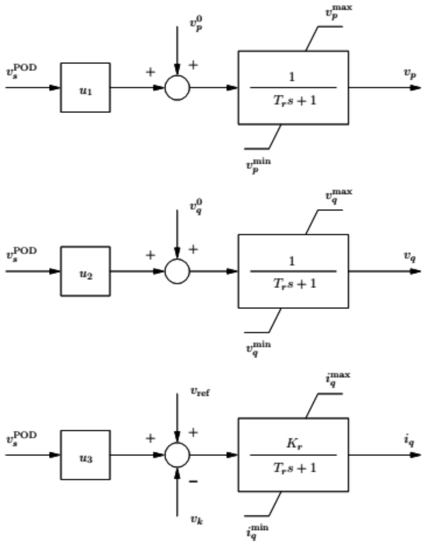


Figure 3.10: Control block diagram of UPFC [27].

3.3.8 UPFC in Modelica

To implement the UPFC model in Modelica, the series and shunt components are made in two different models. Then two models are combined together to form the complete UPFC model. The implementation methodology of the series component is same as the

SSSC implementation and the shunt part is same as the STATCOM implementation, with minor differences, the codes are not added here.

3.4 Transformer

Two regulating transformers (ULTC and PST) and one static transformer (TWT) are implemented in Modelica. The detail mathematical description is given in following sections.

3.4.1 Three-Winding Transformer(TWT)

The Three-winding Transformer is basically described as three two winding transformers connected in star (see figure 3.11 [2]).

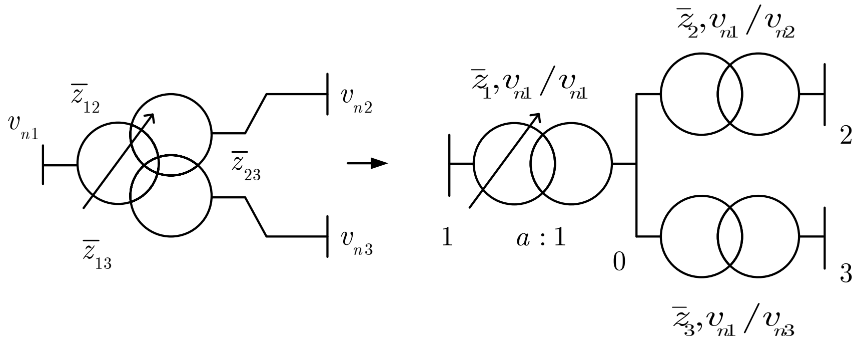


Figure 3.11: Equivalent circuit: Three-Winding Transformer [27].

The star impedances of the triangle branches are:

$$\begin{aligned} \bar{z}_{12} &= \bar{z}_1 + \bar{z}_2 \\ \bar{z}_{13} &= \bar{z}_1 + \bar{z}_3 \\ \bar{z}_{23} &= \bar{z}_2 + \bar{z}_3 \end{aligned} \tag{3.16}$$

Thus

$$\begin{aligned}\bar{z}_1 &= (\bar{z}_{12} + \bar{z}_{13} - \bar{z}_{23})/2 \\ \bar{z}_2 &= (\bar{z}_{12} + \bar{z}_{23} - \bar{z}_{13})/2 \\ \bar{z}_3 &= (\bar{z}_{13} + \bar{z}_{23} - \bar{z}_{12})/2\end{aligned} \tag{3.17}$$

3.4.2 TWT in Modelica

The two winding transformer in Modelica is modeled as a transmission line with only series impedance without iron losses [23]. Three Winding Transformer is implemented by using the method of equivalent three two-winding transformers (see the three branches of transformer in figure 3.11), but in the case of Three Winding Transformer the impedances are taken as a resulting star impedance (equation 3.17); in the first branch a fixed tap ratio is taken into account. The code for first branch is given in following:

```
model Branch1 "First winding of Three Winding Transformer"

  PowerSystems.Connectors.PwPin p
  PowerSystems.Connectors.PwPin n1

  parameter Real SystemBase=100;
  parameter Real Sn=100 "Power rating MVA";
  parameter Real Vbus=400000 "Sending end bus voltage, V";
  parameter Real Vn1=400000 "Voltage rating of the first winding, V";
  parameter Real Vn2=100000 "Voltage rating of the second winding, V";
  parameter Real Vn3=40000 "Voltage rating of the third winding, V";
  parameter Real fn=50 "Frequency rating, Hz";
  parameter Real R12=0.01 "Resistance of the branch 1-2, p.u.";
  parameter Real R13=0.01 "Resistance of the branch 1-3, p.u.";
  parameter Real R23=0.01 "Resistance of the branch 2-3, p.u.";
  parameter Real X12= 0.1 "Reactance of the branch 1-2, p.u.";
  parameter Real X13= 0.1 "Reactance of the branch 1-3, p.u.";
  parameter Real X23= 0.1 "Reactance of the branch 2-3, p.u.";
  parameter Real m=0.98 "Fixed Tap ratio";

 Real r1;
 Real x1;
 Real anglev2 "Angle of the fictious bus";
 Real vbus2 "Voltage of the fictious bus";
```

```
equation
  vbus2=sqrt(n1.vr^2+n1.vi^2);
  anglev2=atan2(n1.vi, n1.vr);
  r1=0.5*(R12+R13-R23);
  x1=0.5*(X12+X13-X23);
  r1*p.ir-x1*p.ii= (1/m^2)*p.vr- (1/m)*n1.vr;
  r1*p.ii+x1*p.ir= (1/m^2)*p.vi- (1/m)*n1.vi;
  r1*n1.ir-x1*n1.ii= n1.vr- (1/m)*p.vr;
  x1*n1.ir+r1*n1.ii= n1.vi- (1/m)*p.vi;

end Branch1;
```

After making three branches in the same way, all three branches are connected (see figure 3.12). Finally, all the parameter of these three branches are declared in upper label (by using the component reference) to finalize the model.

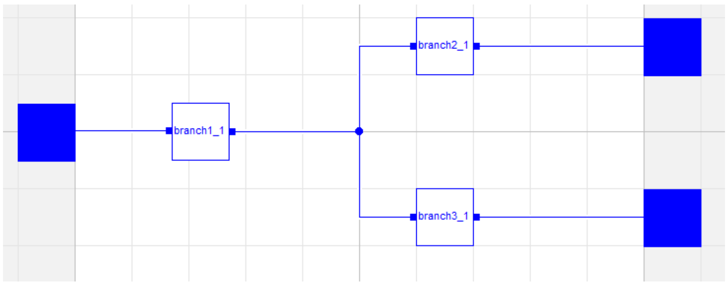


Figure 3.12: Three Winding Tansformer in Modelica.

3.4.3 Under Load Tap Changer

The Under Load Tap Changer is a kind of regulating transformer. The equivalent pi circuit and the secondary voltage control scheme control of ULTC is shown in figure 3.13 and 3.14. If the tap ratio step $\Delta m = 0$ then:

$$\tilde{m} = m \tag{3.18}$$

To control the voltage and reactive power:

$$\begin{aligned} \dot{m} &= -Hm + K(v_m - v_{ref}) \\ \dot{m} &= -Hm + K(q_{ref} + q_m) \end{aligned} \tag{3.19}$$

where, H is the integral deviation, K is the inverse time constant, v_m is secondary bus voltage and v_{ref} is the reference voltage.

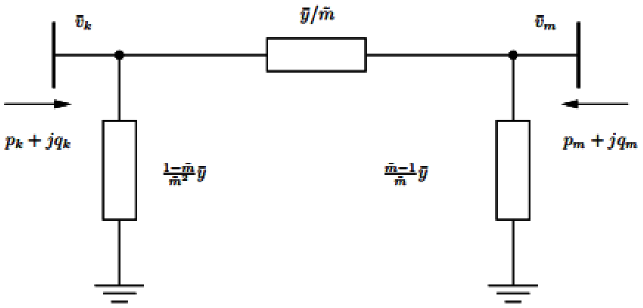


Figure 3.13: Equivalent π circuit: Under Load Tap Changer [27].

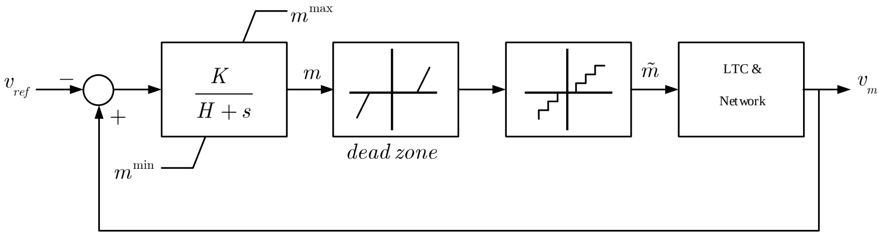


Figure 3.14: Secondary voltage control scheme of ULTC [27].

3.4.4 ULTC in Modelica

In Modelica, the implementation of the ULTC is a continuous model so the tap ratio step taken is $\Delta m = 0$. The model is implemented using the PwPin connector for inputs and outputs. To calculate the current variable of the connector, equation 3.20 is used (using equivalent π circuit [2]). To calculate the dynamic tap ratio m, the equation 3.19 is used. The limits of the controller are implemented using the if statement. The Modelica code is given below:

$$\begin{bmatrix} \bar{i_k} \\ \bar{i_m} \end{bmatrix} = \bar{y} \begin{bmatrix} \frac{1}{m^2} & -\frac{1}{m} \\ -\frac{1}{m} & 1 \end{bmatrix} \begin{bmatrix} \bar{v_k} \\ \bar{v_m} \end{bmatrix} \tag{3.20}$$

```
model ULTC_VoltageControl
  "Under Load Tap Changer, continous model, secondary voltage control"
```

```
  PowerSystems.Connectors.PwPin p

  PowerSystems.Connectors.PwPin n

  parameter Real SystemBase=100;
  parameter Real Vbus1=400000 "Sending end Bus nominal voltage, change of base";
  parameter Real Vbus2=100000 "Receiving end Bus voltage, change of base";
  parameter Real Sn=100 "Power rating MVA";
  parameter Real Vn=400000 "Voltage rating, primary side KV";
  parameter Real fn=50 "Frequency rating Hz";
  parameter Real kT=4 "Nominal Tap ratio (V1/V2), kV/kV";
  parameter Real H=0.001 "Integral deviation, p.u.";
  parameter Real K= 0.10 "Inverse time constant, 1/s";
  parameter Real m_max=0.98 "Max tap ratio, p.u./p.u.";
  parameter Real m_min=0.9785 "Min tap ratio, p.u./p.u.";
  parameter Real deltam=0 "Tap ratio step, p.u./p.u.";
  parameter Real v_ref=1.0 "Reference voltage, p.u.";
  parameter Real xT= 0.001 "Transformer Reactance, p.u.";
  parameter Real rT=0.1 "Transformer Resistance, p.u.";
  parameter Real d= 0.05 "Dead zone percentage, %";
  parameter Real vm0=1.008959700699460
    "Initial value of the voltage of the Controlled Bus";
  parameter Real m0=0.98;

  Real m "Tap ratio";
  Real vk "Voltage at primary, p.u.";
  Real vm "Voltage at secondary p.u.";
  Real anglevk "Angle at primary";
  Real anglevm "Angle at secondary ";

protected
  parameter Real V2= Vn/kT "Secondary voltage";
  parameter Real Vb2new=Vbus1*Vbus1;
  parameter Real Vb2old=Vn*Vn;
  parameter Real R=rT*(Vb2old*SystemBase)/(Vb2new*Sn)
    "Transformer Resistance, p.u.";
  parameter Real X=xT*(Vb2old*SystemBase)/(Vb2new*Sn)
    "Transformer Reactance, p.u.";
  parameter Real vref=v_ref*(V2/Vbus2);

initial equation
  m=m0;

equation
```

```
   vk=sqrt(p.vr^2+p.vi^2);
   vm=sqrt(n.vr^2+n.vi^2);
   anglevk=atan2(p.vi, p.vr);
   anglevm=atan2(n.vi, n.vr);
if (m>m_max) and  (der(m)>0) then

  R*p.ir-X*p.ii= (1/m_max^2)*p.vr- (1/m_max)*n.vr;
  R*p.ii+X*p.ir= (1/m_max^2)*p.vi- (1/m_max)*n.vi;
  R*n.ir-X*n.ii= n.vr- (1/m_max)*p.vr;
  X*n.ir+R*n.ii= n.vi- (1/m_max)*p.vi;
  der(m)=0;

elseif (m<m_min) and (der(m)<0) then

  R*p.ir-X*p.ii= (1/m_min^2)*p.vr- (1/m_min)*n.vr;
  R*p.ii+X*p.ir= (1/m_min^2)*p.vi- (1/m_min)*n.vi;
  R*n.ir-X*n.ii= n.vr- (1/m_min)*p.vr;
  X*n.ir+R*n.ii= n.vi- (1/m_min)*p.vi;
  der(m)=0;

else

  R*p.ir-X*p.ii= (1/m^2)*p.vr- (1/m)*n.vr;
  R*p.ii+X*p.ir= (1/m^2)*p.vi- (1/m)*n.vi;
  R*n.ir-X*n.ii= n.vr- (1/m)*p.vr;
  X*n.ir+R*n.ii= n.vi- (1/m)*p.vi;
  der(m)= -(H*m)+K*(vm-vref);

end if;
end ULTC_VoltageControl;
```

3.4.5 Phase Shifting Transformer(PST)

Equivalent circuit and control block of phase shifting transformer are shown in figure 3.15 and 3.16.

The differential equations describing the PST transformer are given by:

$$\begin{aligned} \dot{\alpha} &= K_p(p_k - p_{mes})/T_m + K_i(p_{mes} - p_{ref}) \\ \dot{p}_{mes} &= (p_k - p_{mes})/T_m \end{aligned} \tag{3.21}$$

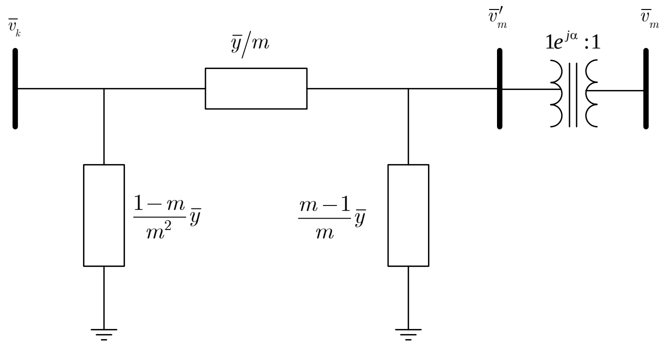


Figure 3.15: Equivalent circuit of Phase Shifting Transformer [27].

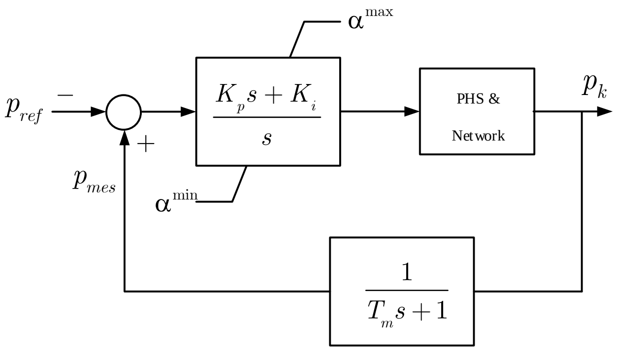


Figure 3.16: Control scheme of Phase Shifting Transformer [27].

where, α is the phase angle, $pmes$ is the measured power flow, p_k is the real power flow. K_i, K_p, T_m are integral gain, proportional gain and measurement time constant, respectively.

3.4.6 PST in Modelica

In Modelica the implementation of the PST model is similar to the ULTC model but, in this case, the tap ratio is fixed and the angle is changed. The final model is implemented by using two sub models. First, the fixed tap ratio and line impedance (see figure 3.15) are declared, using the following code structure:

```
equation
vk=sqrt(p.vr^2+p.vi^2);
vm=sqrt(n.vr^2+n.vi^2);
```

```
  anglevk=atan2(p.vi, p.vr);
  anglevm=atan2(n.vi, n.vr);
  R*p.ir-X*p.ii= (1/m^2)*p.vr- (1/m)*n.vr;
  R*p.ii+X*p.ir= (1/m^2)*p.vi- (1/m)*n.vi;
  R*n.ir-X*n.ii= n.vr- (1/m)*p.vr;
  X*n.ir+R*n.ii= n.vi- (1/m)*p.vi;
```

Next the controller (figure 3.16) equations (equation 3.21) are declared in another part. The relation between PST angle *alpha* with PwPin connector variables are calculated by using the relation $\bar{v}'_m : v_m = 1e^{j\alpha} : 1$ (see figure 3.15). Finally, both parts are added together to make the final model. The implementation of the controller part is given below (only the equation section):

```
initial equation
  alpha=alpha0;
  pmes=pmes0;
equation
  vk=sqrt(p.vr^2+p.vi^2);
  vm=sqrt(n.vr^2+n.vi^2);
  anglevk=atan2(p.vi, p.vr);
  anglevm=atan2(n.vi, n.vr);

  if (alpha>alpha_max) and (der(alpha)>0) and (der(pmes)>0) then

  der(alpha)=0;
  der(pmes)=(pk-pmes)/Tm;
  p.vr=n.vr*cos(alpha_max)-n.vi*sin(alpha_max);
  p.vi=n.vr*sin(alpha_max)+n.vi*cos(alpha_max);
  p.ir+n.ir=0;
  p.ii+n.ii=0;

  elseif (alpha<alpha_min) and (der(alpha)<0) and (der(pmes)<0) then

  der(alpha)=0;
  der(pmes)=(pk-pmes)/Tm;
  p.vr=n.vr*cos(alpha_min)-n.vi*sin(alpha_min);
  p.vi=n.vr*sin(alpha_min)+n.vi*cos(alpha_min);
  p.ir+n.ir=0;
  p.ii+n.ii=0;

  else
```

```
  der(alpha)= (Kp*(pk-pmes)/Tm)+Ki*(pmes-pref);
  der(pmes)=(pk-pmes)/Tm;
  p.vr=n.vr*cos(alpha)-n.vi*sin(alpha);
  p.vi=n.vr*sin(alpha)+n.vi*cos(alpha);
  p.ir+n.ir=0;
  p.ii+n.ii=0;

  end if;
end pst2;
```

Chapter 4

Power System Model Validation

At this stage, Modelica models have to be validated against their reference model. A software to software validation has to be performed. For this task, simple test scenario is modeled in both Modelica and PSAT. In both scenarios, the same perturbation is applied. This will allow the comparison of the simulation results generated by both modeling tools. More details on the test models are described in the following sections.

4.1 Simulation Setup

The simulation set up used for the validation is given in table 4.1.

Table 4.1: Simulation set up for the validation.

Set up	**Modelica**	**PSAT**
Simulation Environment	Dymola	Matlab
Integration Algorithm	Dassl [22]	Trapezoidal Rule
Time step	.01s	.01s
Tolerance	.00001	.00001
Power flow	Not implemented yet	Newton Raphson method

4.2 Initialization

Prior to a power system time domain simulation, the power flow calculation needs to be done for computing the steady-state values of the network, which are used to initialize the models for dynamic simulations. Due to multi-domain behavior of Modelica, it lacks a power flow computation method. So the power flow calculation is done with another power system analysis software, in this case PSAT, and the values are used to initialize the corresponding Modelica model. The initialization done in two ways [26]:

- Start values for variables.
- Initial equations and initial algorithms.

One of the problems when modeling with mathematical equations is to determine how many initial conditions are required to have a well-posed model. The answer is that there should be the same number of initial equations as states of the system [21]. In any model, the variables being differentiated are known as the state variables. To initialize these variables, the derivative of the state variables is taken as zero to form the equation. Then, the initial equations can be solved with the help of power flow calculations. All the initial equations used are given below, and in the case of the TCSC, the initialization is carried out by setting the start values directly from power flow solution.

- STATCOM:
$$i_{SH0} = K_r(v_{ref} + v_s^{POD} - v0)$$
- SSSC:
$$v_s0 = v_s^o + v_s^{POD}$$
- UPFC:
$$i_q0 = K_r(v_{ref} + u_3 v_s^{POD} - v_k0)$$
$$v_p0 = v_{p0} + u_1 v_s^{POD}$$
$$v_q0 = v_{q0} + u_2 v_s^{POD}$$

4.3 Validation of FACTS Devices

All the FACTS devices developed in this work are, in simple terms: regulators. Every regulator output undergoes a limiter. In order to check the upper and lower limit of these limiters a hit limit test has been implemented. In the same hit limit test, the loads are also changed for a certain time. The validation scenarios with the perturbation are discussed in the following sections.

4.3.1 Validation of TCSC model

For the two TCSC models, the same model has been created in both Modelica and PSAT, for validating the outputs. The test model consists of a simple network with three buses, one synchronous generator (Second Order) in the bus [1] and a TCSC placed between bus [1] and bus [2] to control the power flow. In bus [3], a PQ load is placed. In Modelica as transmission line parameters are included in the TCSC, there is no need to put an additional transmission line. The test models are shown in the figures 4.1 and 4.2. To compare the dynamic response of both models, the same perturbation is applied:

Synchronous Machine:

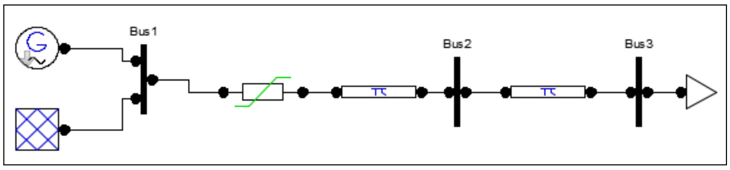
- V_f, square oscillation $-0.045 * square(2\pi * 0.1 * t)$ from time 0s to 20s.

PQ Load

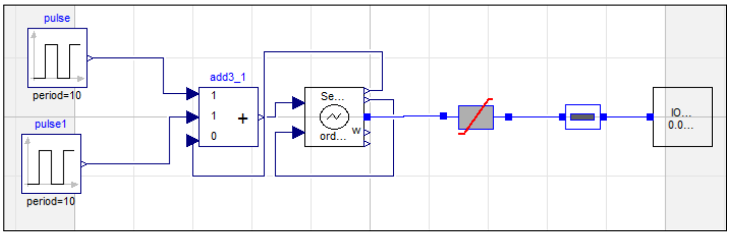
- P, an active load is added to hit the limit of the limiter of the regulator $+.01$ from time 2s to 10s and $-.01$ from time 12s to 20s.
- Q, a reactive load is added to hit the limit of the limiter of the regulator $+.01$ from time 2s to 10s and $-.01$ from time 12s to 20s.

4.3.2 Simulation Result of TCSC validation

Simulation outputs from Modelica and PSAT simulations are shown in figure 4.3 and 4.4. The validation procedure compares the output signals by changing the generator

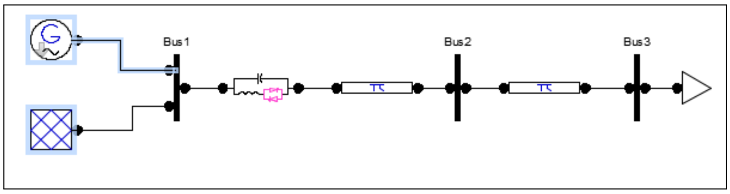


(a) In PSAT.

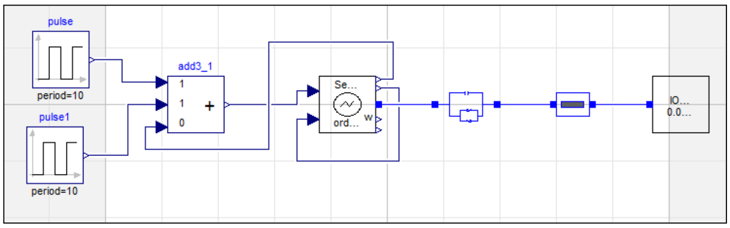


(b) In Modelica.

Figure 4.1: Test Model of TCSC type: Reactance.



(a) In PSAT.



(b) In Modelica.

Figure 4.2: Test Model of TCSC type: Alpha.

field voltage [vf] and P, Q values of the loads. The TCSC is used to control the power flow [P_{ij}] between Bus [1] and Bus [2] by controlling the line susceptance [b]. Due to changes in load, the TCSC susceptance changes accordingly keeping the power flow same as reference power [P_{ref}]. Due to limit[$x^{max}_{TCSC}, x^{min}_{TCSC}$ and $\alpha_{max}, \alpha_{min}$] provided to the state variables [$x1$ and $x2$], the susceptance and states variables reach the upper and lower limit (see in figure 4.3 and 4.4 the state variable x1 in the upper right plot).

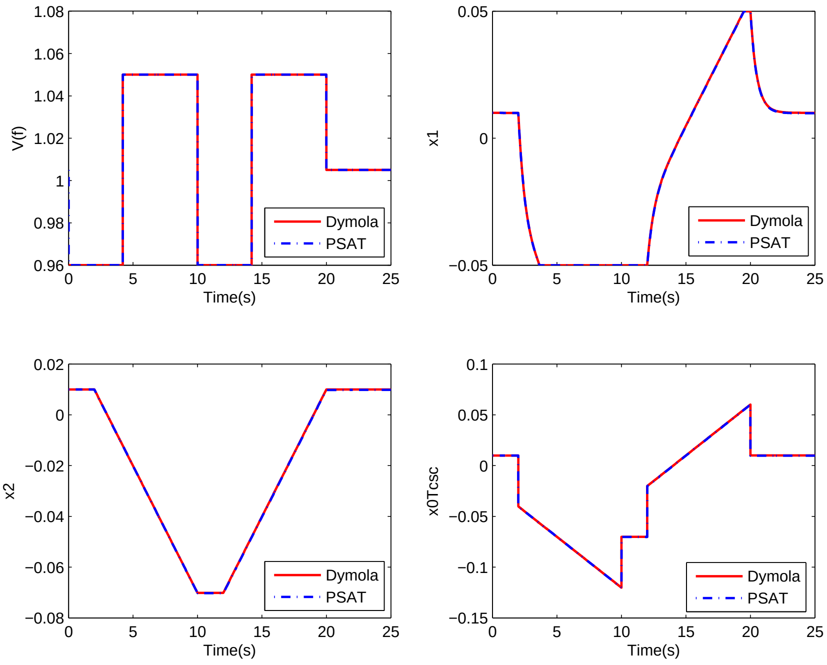


Figure 4.3: Software to software validation of TCSC Reactance Model.

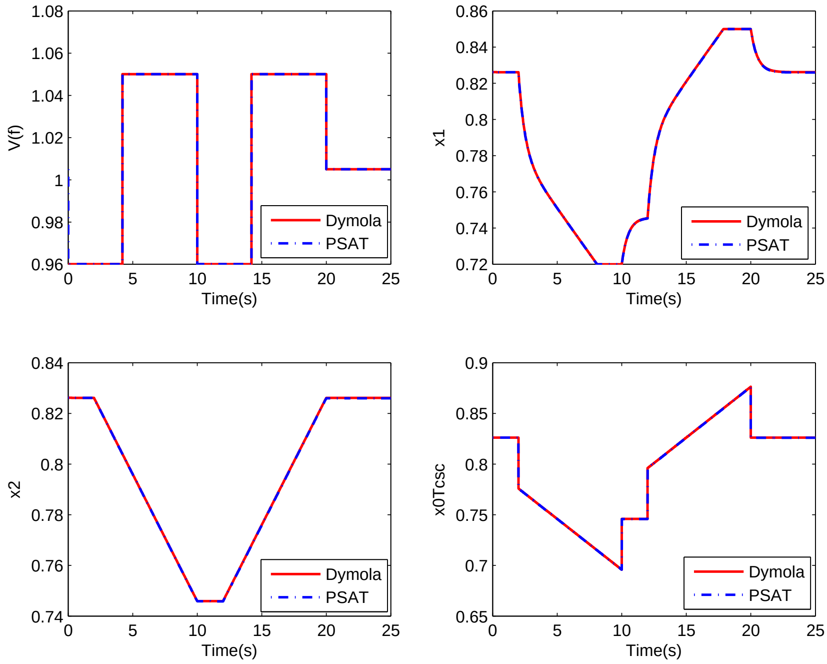


Figure 4.4: Software to software validation of TCSC Alpha Model.

These results state that the simulation of the TCSC model yields the same result in both software tools.

4.3.3 Validation of STATCOM model

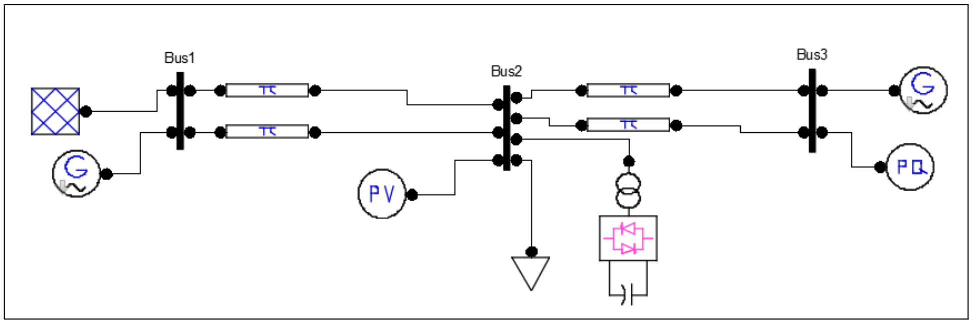
To validate the two STATCOM models described in this work, two test models are created in Modelica and one in PSAT (see figure 4.5). These simple test systems are composed of three buses, with the STATCOM model connected at bus [2], to control the voltage at bus [2]. Two perturbations are applied in the test system, affecting the generator and load:

Synchronous Machine:

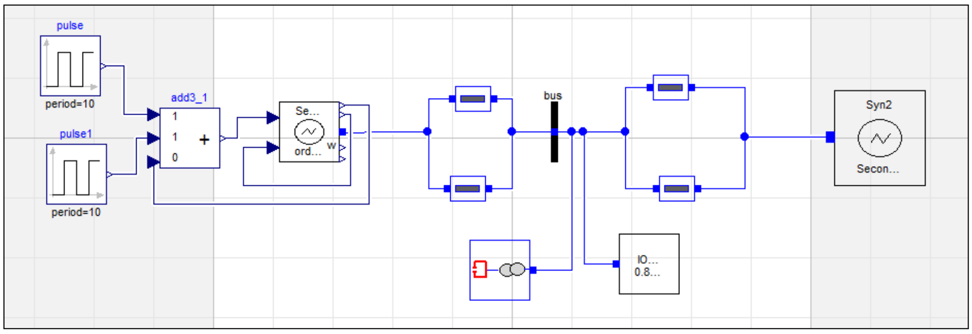
- V_f, square oscillation $-0.045 * square(2\pi * 0.1 * t)$ from 0s.

PQ Load

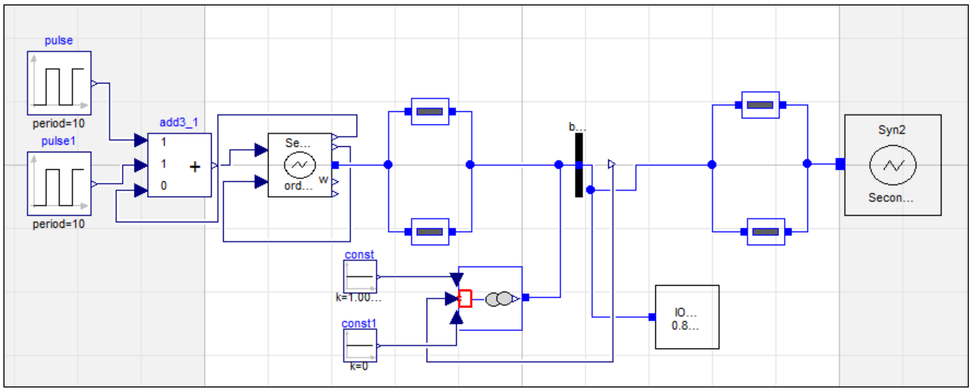
- Q, reactive load is added to hit the limit of the limiter of the regulator $+.42$ from 8s



(a) In PSAT.



(b) In Modelica [Text Based Model].



(c) In Modelica [GUI Based Model].

Figure 4.5: Test Model of STATCOM.

4.3.4 Simulation Result of STATCOM

Simulation outputs are shown in figure 4.6 and A.4. The perturbation applied in the generator field voltage [vf] and in the reactive load [Q] in Bus [2], affects the voltage of Bus [2]. STATCOM keeps the Bus [2] voltage within the limit by injecting current [i_{SH}] (see figure 4.6 and A.4). The current injection also reaches its limit [i_{max}, i_{min}] in oder to keep the bus voltage stable. The results of both the softwares are similar.

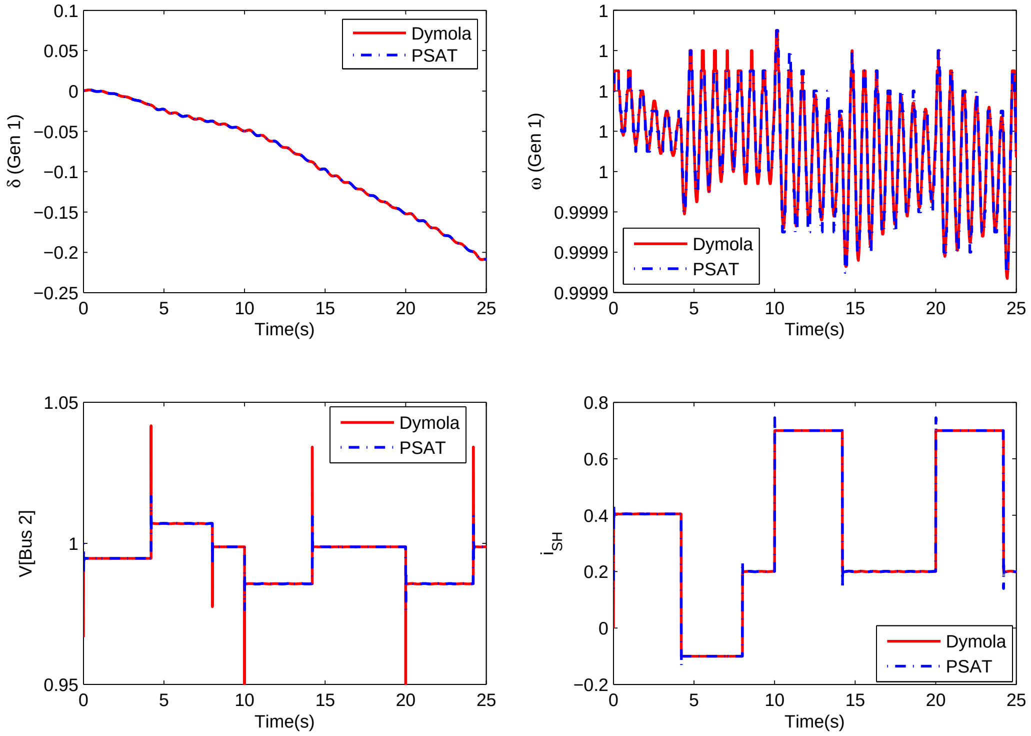


Figure 4.6: Software to software validation of STATCOM GUI based Model.

4.3.5 Validation of SSSC model

To validate SSSC model for both of the modes, a test system is created composed of three buses. The SSSC is placed between Bus [1] and Bus [2]. The perturbation applied to the test system is as described in the validation of the TCSC model. The test models are shown in figure 4.7 and 4.8. The simulation the result of constant power flow SSSC model is shown in figure 4.9 and result of constant voltage model is added in figure A.5.

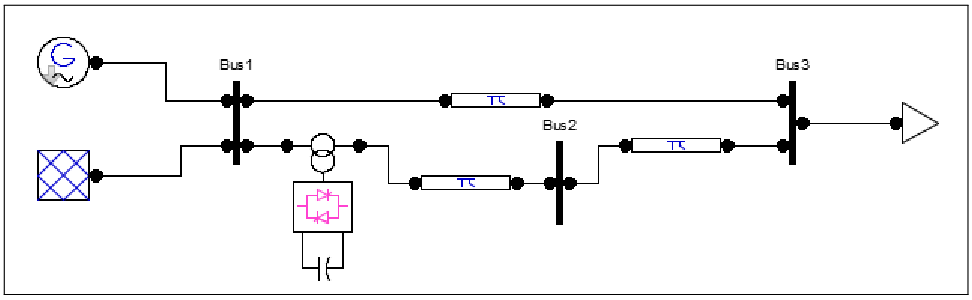


Figure 4.7: Test Model of SSSC in PSAT.

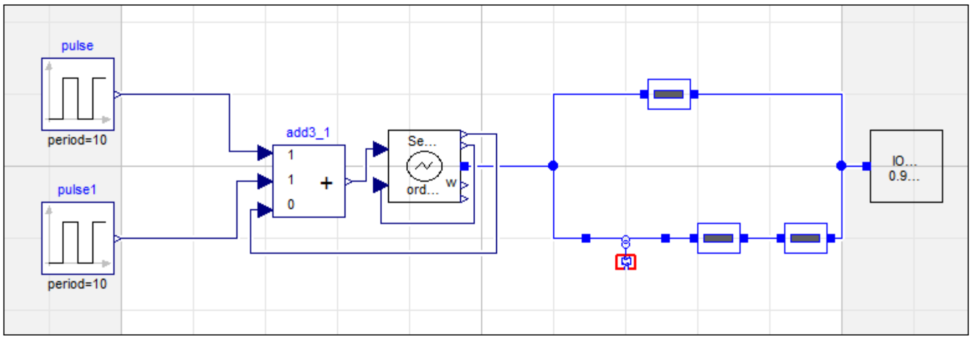


Figure 4.8: Test Model of SSSC in Modelica.

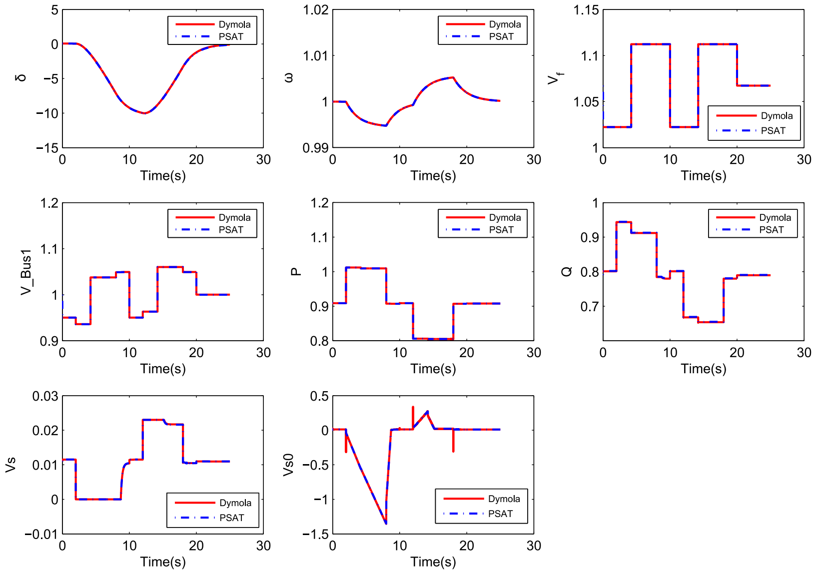
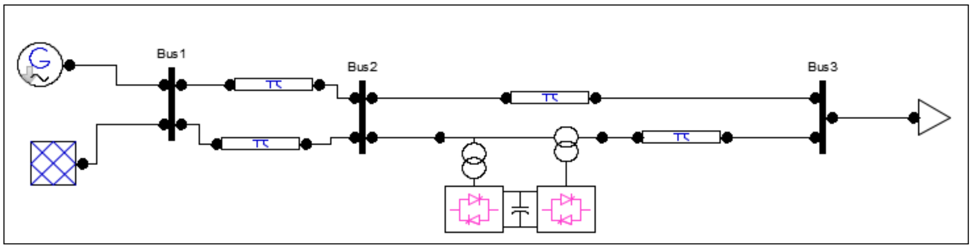


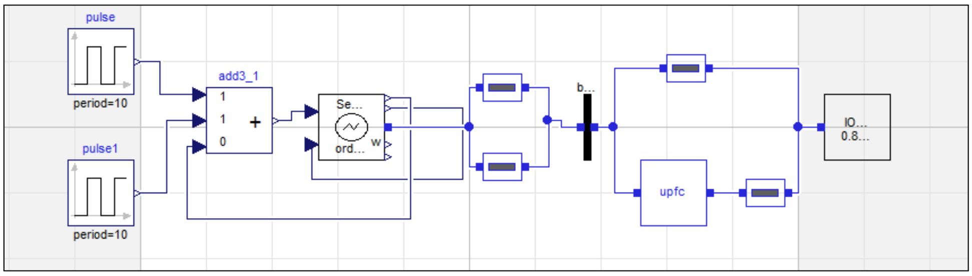
Figure 4.9: Software to software validation of SSSC in constant power mode.

4.3.6 Validation of UPFC model

To validate the UPFC model, a three bus test system is created and the UPFC is placed between bus [2] and [3]. The perturbation applied is same as the TCSC but, in this case, the PQ loads are varied by .02 p.u. The test systems are shown in figure 4.10. The simulation result of UPFC are shown in figure 4.11.



(a) In PSAT.



(b) In Modelica.

Figure 4.10: Test Model of UPFC.

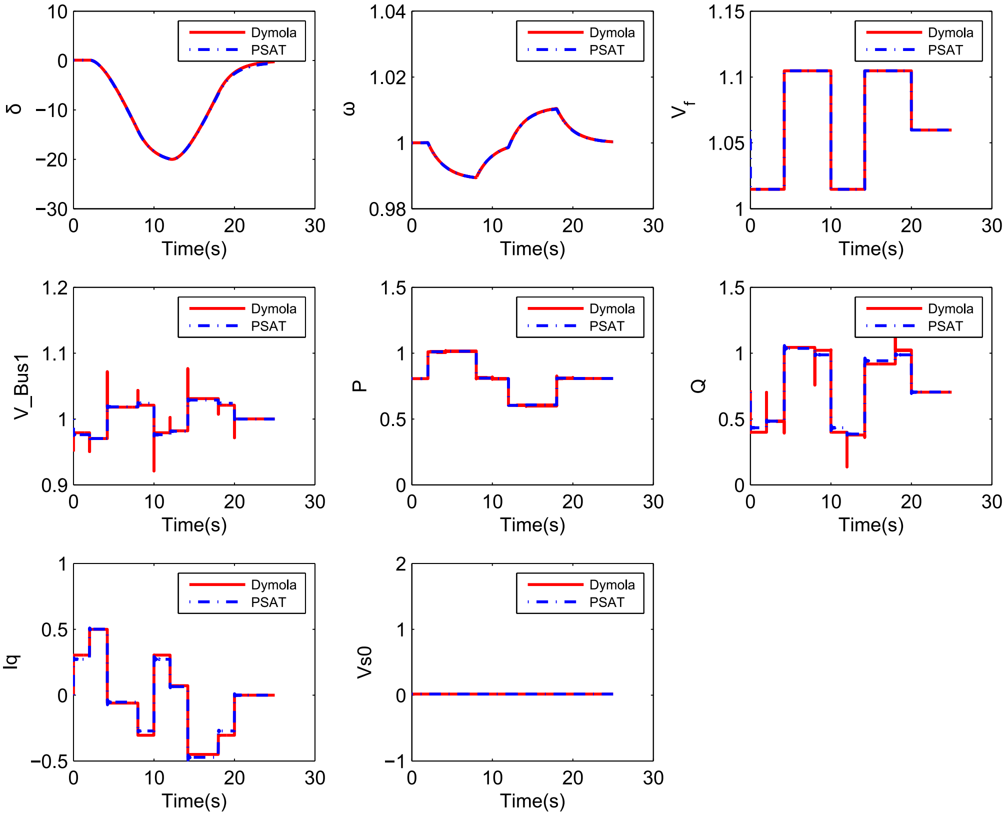


Figure 4.11: Software to software validation of UPFC in constant power mode.

4.4 Transformer

Three different test systems have been implemented for the validation of the TWT, ULTC and PST transformer models (see figures 4.12, 4.13 and 4.14). Also, different perturbations have been applied in the test systems models:

Synchronous Machine for all cases:

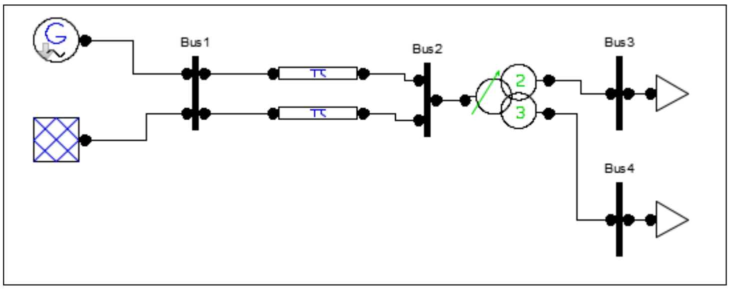
- V_f, sinusoidal oscillation $-0.001 * sin(2\pi * 0.2 * t)$ from time 0s to 5s.

PQ Load for TWT and ULTC

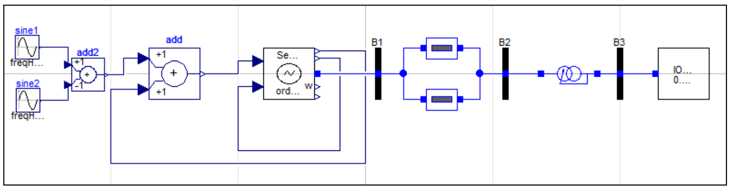
- Q, a reactive load is added +.01 (for ULTC +.05) from time 5s to 8s and −.01 (for ULTC −.05) from time 8s to 12s.

PQ Load for PST

- P, an active load is added +.02 from time 5s to 8s and −.02 from time 8s to 12s.
- Q, a reactive load is added +.01 from time 5s to 8s and −.01 from time 8s to 12s.

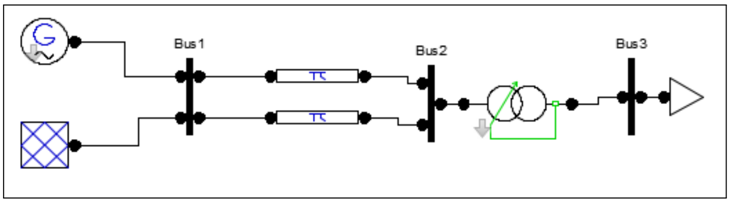


(a) In PSAT.

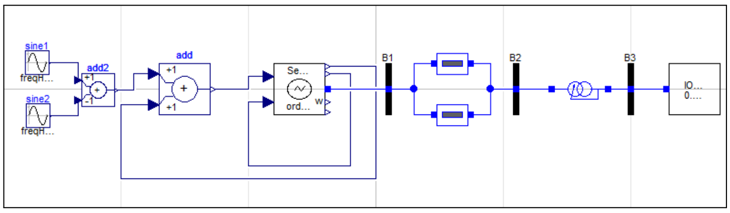


(b) In Modelica.

Figure 4.12: Test Model of TWT.

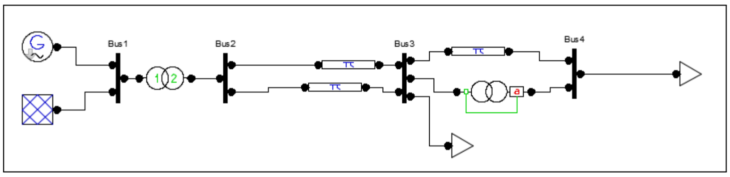


(a) In PSAT.

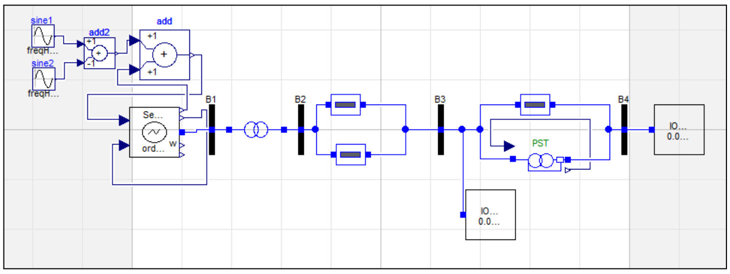


(b) In Modelica.

Figure 4.13: Test Model of ULTC.



(a) In PSAT.



(b) In Modelica.

Figure 4.14: Test Model of PST.

4.5 Transformer Simulation Result

Modelica simulation outputs are, again, compared against the reference model developed in PSAT. The following results show the comparison between Modelica and PSAT simulations for the ULTC model (The simulation results of TWT and PST simulations are included in the figure A.6 and A.7). The dynamic tap ratio m of ULTC is used to control the Bus[3] voltage. Due to the changes of load at Bus [3] (see the perturbation applied) the tap ration m varies (see figure 4.15). The variation of tap ratio is consistent in both the software packages.

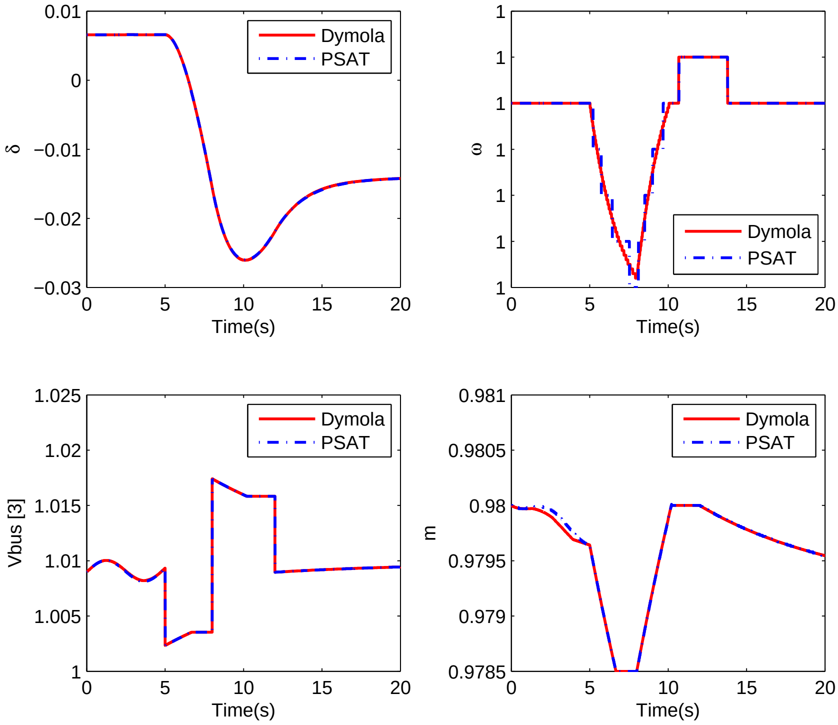


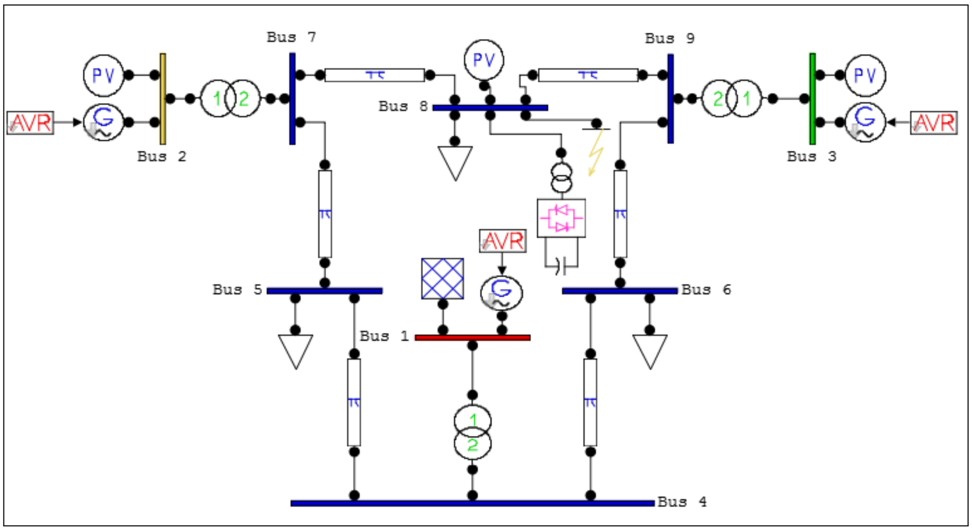
Figure 4.15: Software to software validation of ULTC.

4.6 IEEE Standard Test System

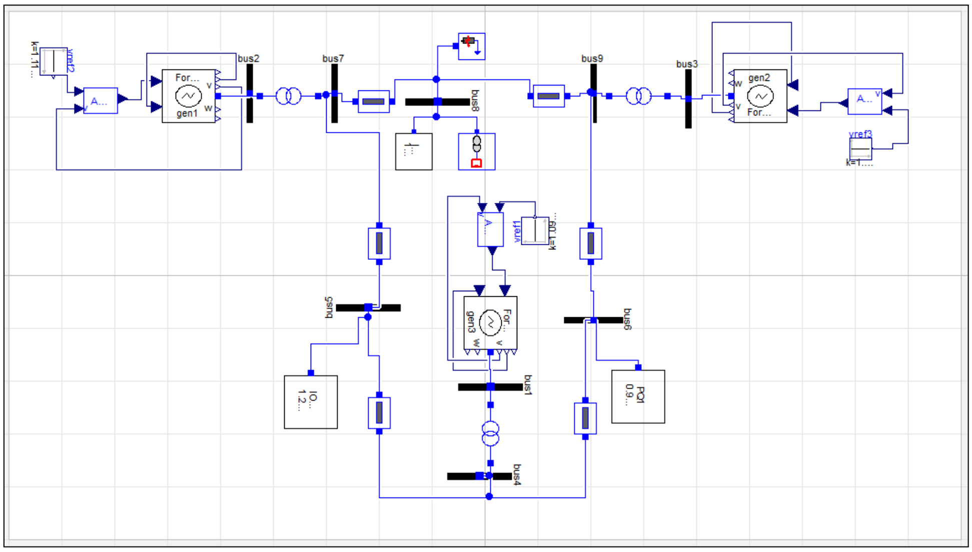
Having the Modelica implementation and validation of all the components described above, IEEE 9-Bus and 14-Bus test system is implemented and tested with the components developed in this work.

4.6.1 IEEE 9-Bus Test System

The IEEE 9-Bus Test System represents a portion of the Western System Coordinating Council (WSCC) [29] (The single line diagram of the IEEE 9-Bus system with the complete data is given in Appendix A.3.1). Four different versions of the 9-Bus test system were implemented in Modelica: (a) STATCOM is connected in bus 8 and the (b) TCSC, (c) SSSC and (d) UPFC are connected in between bus 8 and bus 9. The system with STATCOM is shown in figure 4.16 and the system with TCSC is shown in figure A.1.



(a) In PSAT.



(b) In Modelica.

Figure 4.16: IEEE 9-Bus Test system.

4.6.2 IEEE 14-Bus Test System

The IEEE 14-Bus Test case represents a portion of the American Electric Power System which is located in the Midwestern US [30].(The single line diagram of the IEEE-14 Bus system with the complete data is given in Appendix A.3.2). An available model of this system is already implemented in PSAT. For the Modelica implementation of this system, the Power System Library and developed components in this work have been used. A scenario with the presence of small disturbances is modeled in both simulation tools, to observe and compare its dynamic behavior. The ULTC and PST were placed in between

Bus 4 and Bus 9 (see figures 4.17 and 4.18) in two different IEEE 14-Bus test systems. In the case of the TWT, the TWT is placed between Bus 4, Bus 8 and Bus 9 in another IEEE 14-Bus system (shown in figure A.2 and A.3). In this case Bus 7 is not used as it becomes a fictitious bus inside the TWT. All these test systems were also implemented in PSAT.

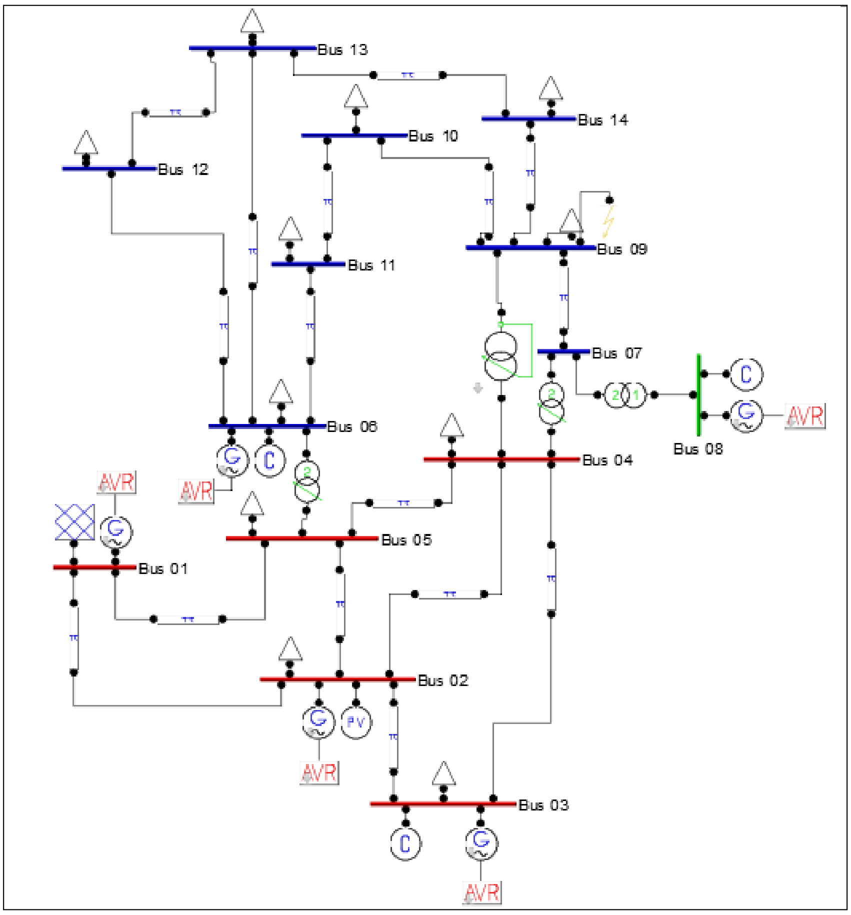


Figure 4.17: IEEE 14-Bus Test system in PSAT with ULTC.

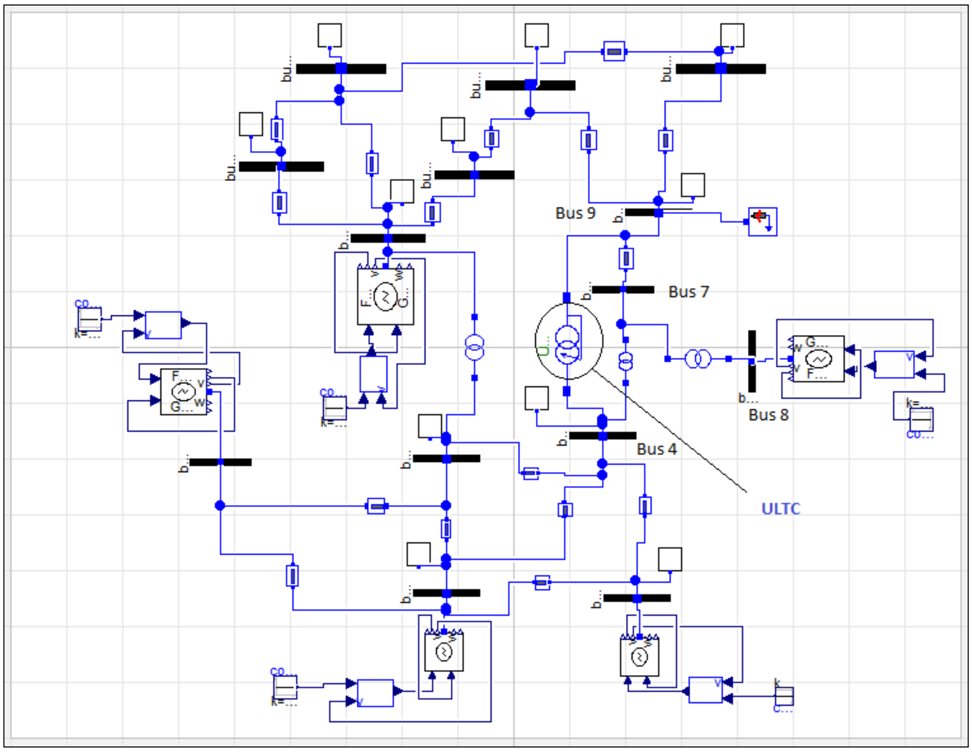


Figure 4.18: IEEE 14-Bus Test system in Modelica with ULTC.

4.6.3 Quantitative Assessment

The qualitative observation only provide an insight of the validity of a model. In contrast, a quantitative assessment allows to measure the validity of a model response against its reference in numerical metrics. For the validation of small test systems only qualitative assessment is done, but for IEEE 9-Bus and 14-Bus systems, results of two different software packages are analyzed both graphically and numerically. The quantitative assessment is carried out using the Root Mean Square Error (RMSE) [31]. The RMS value of the error is calculated using the equation:

$$Z_{RMSE} = \sqrt{1/n[(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2]} \tag{4.1}$$

where, $x_1, x_2, ..., x_n$ are the discrete measurement point at time $t_1, t_2, ..., t_n$ for software package (a) and $y_1, y_2, ..., y_n$ are the discrete measurement points at time $t_1, t_2, ..., t_n$ for software package (b). Z_{RMSE} is the RMS value of the error of Z variable.

4.6.4 Simulation Result of IEEE 9 and 14-Bus Test System

The time domain simulations were executed in OpenModelica (OM) for IEEE 9-Bus test system and in Dymola for IEEE 14-Bus test system. The system is initialized with the power flow solution calculated in PSAT. In the test cases of IEEE 9-bus the same perturbation is applied: three phase fault is applied in bus 8 at 3s with clearing time 100ms. For IEEE 14-bus test system scenarios simulates a fault at bus 9, which occurs at 10s from the start of the simulation, with a clearing time of 100ms. The simulation of is done for 25s.

The solver used in both the softwares is Rungekutta. [1] The simulations were done for 25s, with a time step of 0.001s. The RMSE was calculated using 25000 points from both simulation results and for different state variables and for all the tests using equation 4.1. This calculation was performed using Matlab. The RMS error values are given in Table 4.2.

Table 4.2: Calculations using equation 4.1

Model and Variable	RMSE	Model and Variable	RMSE
STATCOM (i_{SH})	$2.8011 * 10^{-6}$	UPFC (v_q)	$3.5200 * 10^{-6}$
TCSC (x_1)	$2.7316 * 10^{-6}$	ULTC (m)	$3.5439 * 10^{-6}$
TCSC (x_2)	$2.6072 * 10^{-6}$	PST (α)	$6.7955 * 10^{-4}$
SSSC (v_s)	$6.9862 * 10^{-6}$	PST (P_{mes})	$6.795 * 10^{-4}$
SSSC (v_{pi})	$4.5086 * 10^{-6}$	TWT (V_{INTBus})	$7.5164 * 10^{-4}$
UPFC (i_q)	$4.6006 * 10^{-5}$		

Illustration of the software to software validation result of IEEE 9-Bus with STATCOM is given in figure 4.19 and results with TCSC, SSSC and UPFC are given in figures A.8, A.9 and A.10. The simulation result of IEEE 14-Bus test system with ULTC is given in figure 4.20 and results with PST and TWT is given in figures A.11 and A.12.

---

[1] Runge-Kutta, second order, fixed time step method.

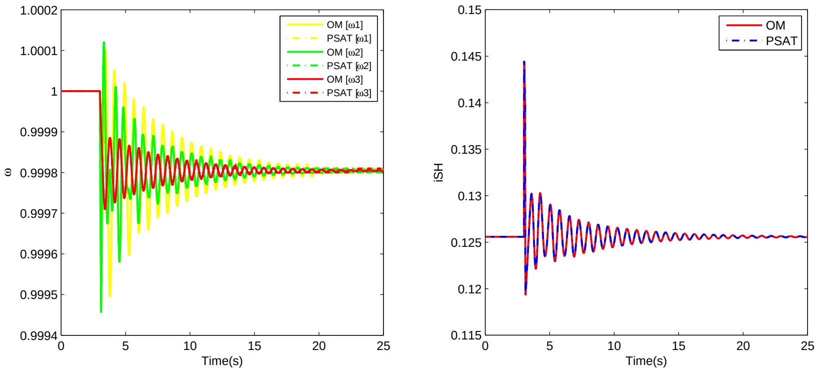


Figure 4.19: Software to software validation results of IEEE 9-Bus with STATCOM.

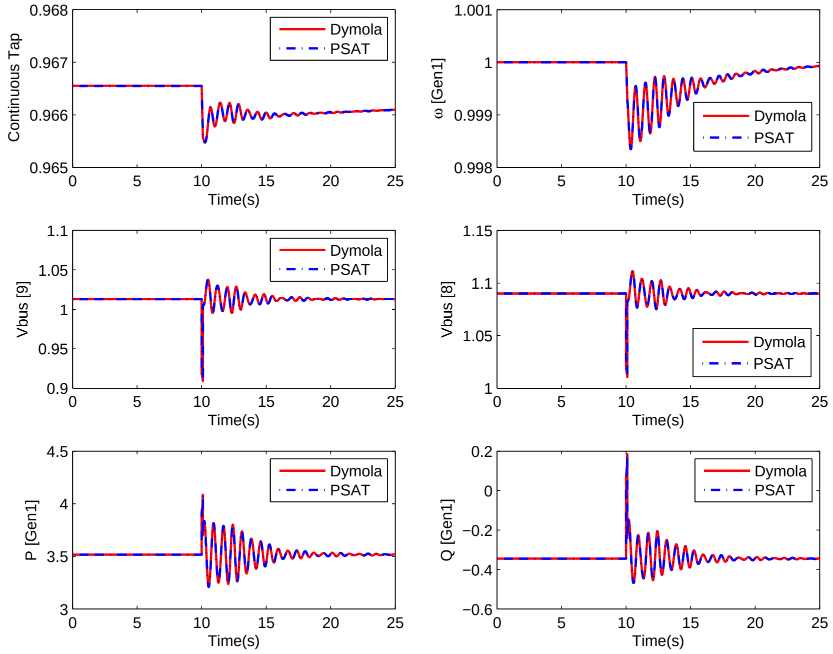


Figure 4.20: Software to software validation results of IEEE 14-Bus with ULTC.

Chapter 5

Discussion and Conclusion

5.1 Modelica

In this thesis, Modelica based power system component modeling, simulation and software to software validation is presented. Modelica is an object-oriented, structured mathematical language, has the ability to exchange the models to different numerical solvers. It is advantageous in a sense that the users do not need to write code to explicitly transport data between objects through assignment statements or message passing code like other object-oriented languages. The code is automatically generated by the Modelica compiler based on the declared equations. Moreover acausal modeling allows reusability of the models as in acausal modeling it is not required to specify a certain data flow direction. In addition Modelica offers different powerful mathematical solvers in the Modelica simulation environment which gives the users much more choice to have the best results. As an example figure 5.1 shows the comparison of three different solvers available in Open Modelica: Runge-Kutta, Dassl (Differential Algebraic System Solver) and Euler with Trapezoidal solver available in PSAT (with TCSC in IEEE 9-Bus). Basically due to efficient simulation and ease of use made Modelica a better choice to the modeling and simulation community. Another advantage of using Modelica is simulation time required less in Modelica. figure 5.2 shows the time required to simulate the IEEE 9-Bus system including TCSC using different solvers whereas the average simulation time took in PSAT with six different tolerances is 608 s.

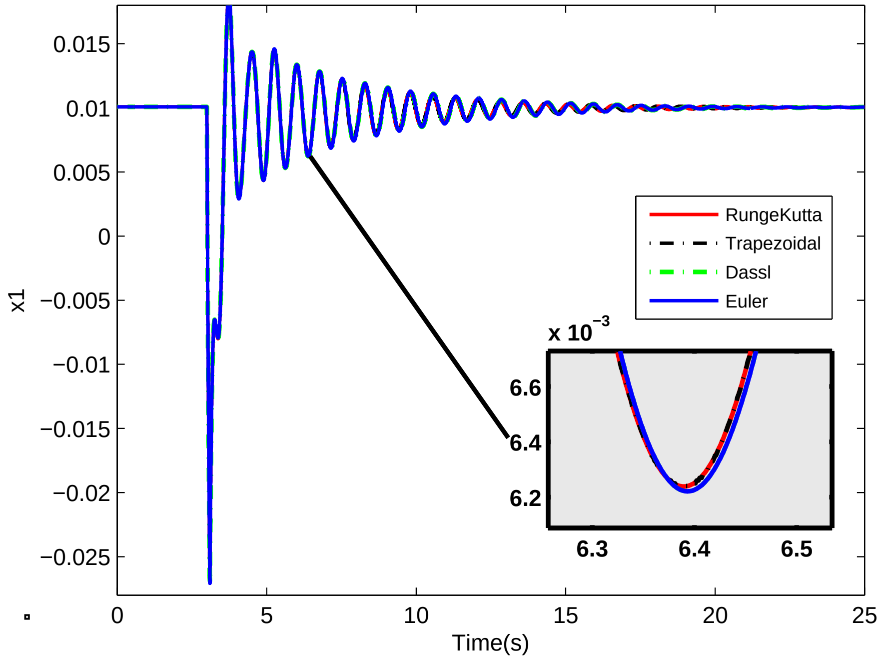


Figure 5.1: Comparison of different solvers of Open Modelica with Trapezoidal rule of PSAT.

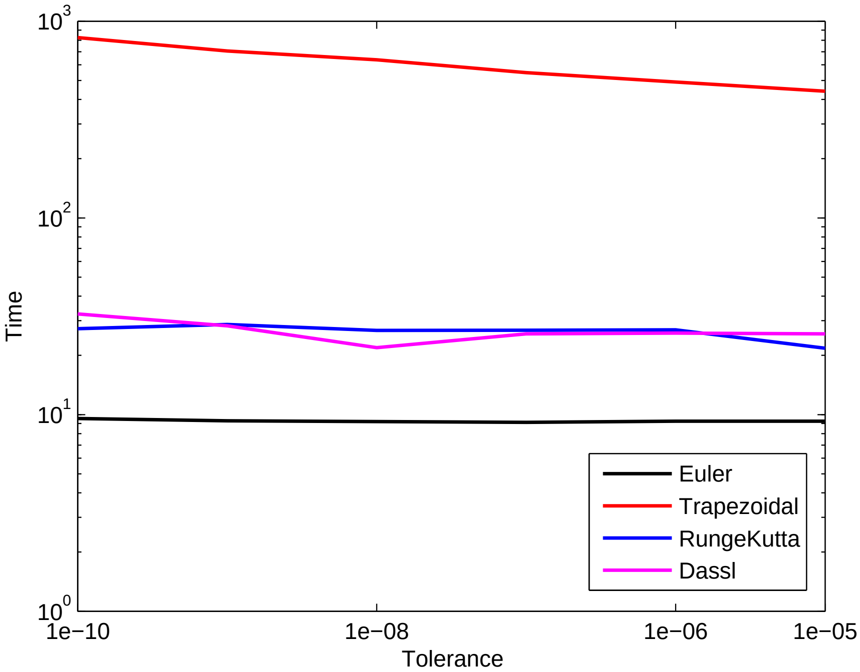


Figure 5.2: Total simulation time spending for the solvers in Open Modelica with different tolerance.

5.2 Modeling and validation

In the chapter 3, a detailed description of the power system component modeling is given. Prior to this work most of the components based on PSAT reference already developed. In this work the components developed also based on PSAT reference. In Modelica one can use the already developed components to make a new model in some cases. In case of one model (STATCOM) existing Modelica components is used. All other models are developed based on the equation declaration method. This proves that Modelica models can be implemented using the existing Modelica tools and/or using previously developed models. Modelica provides more flexibility and better understanding of the models, means one can easily understand what is programmed by seeing the source code. Most of other modeling methods it is difficult to understand only by accessing the source code of the model. The validation results and the dynamic behavior matches (with small difference) with the reference software package which proves that Modelica can be adapted as an universal modeling language.

In chapter 4 and Appendix A all the model validation results are given. After implementing the simple test system, more complex system like IEEE 9-bus and IEEE 14-bus test systems are implemented. The simulation result of the complex systems also consistent with the reference software: PSAT result, but in case of big system the simulation time requires more. This can be improved by further simplifying of the models and also Modelica developers are working on better simulation performance.

5.3 Conclusion

The models simulated in two different software packages accurately predict the same dynamic behavior, so it cannot be said that one is better than other; rather the modeling community is free to choose the modeling language. The results show that the Modelica language is capable of providing similar results of those of typical power system simulation tools for time-domain analysis, and thus, the added values of the Modelica language (model portability and accessibility, and a standardized modeling language) can be fully exploited for power system simulation.

Chapter 6

Future Work

6.1 Future work

This work is applicable for the time domain simulation of cyber-physical power system components. However, the lack of a power flow computation tools make the Modelica models depend on other software tools to compute the steady-state conditions of a power system. An approach to initiate a power flow solver adapted by Modelica tools can be completed.

Regarding the models developed in this work, some improvements have been identified. In case of the ULTC, the model implemented in this work is a continuous model. The ULTC model can be improved by combining the continuous and discontinuous operations of this kind of transformer.

The oscillation damper input has not tested for any of the FACTS models, in future work it can be shown how this damper signal improve the oscillatory stability. This can be carried out using both averaged models and detailed switching models that can be implemented through equation based modeling in future.

Appendix A

Test System and Validation

A.1 Test System

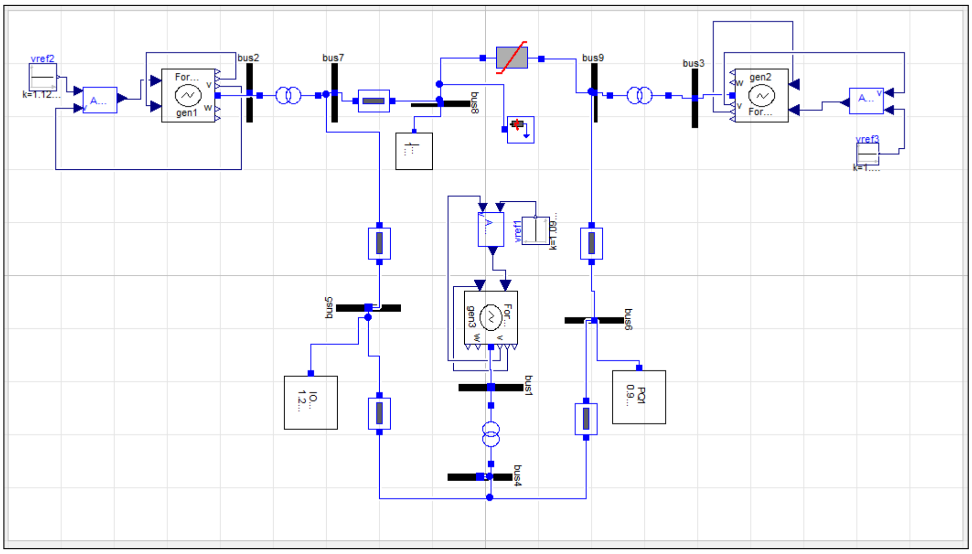


Figure A.1: IEEE 9-Bus Test system with TCSC Reactance Model (between Bus 8 and Bus 9) in OpenModelica.

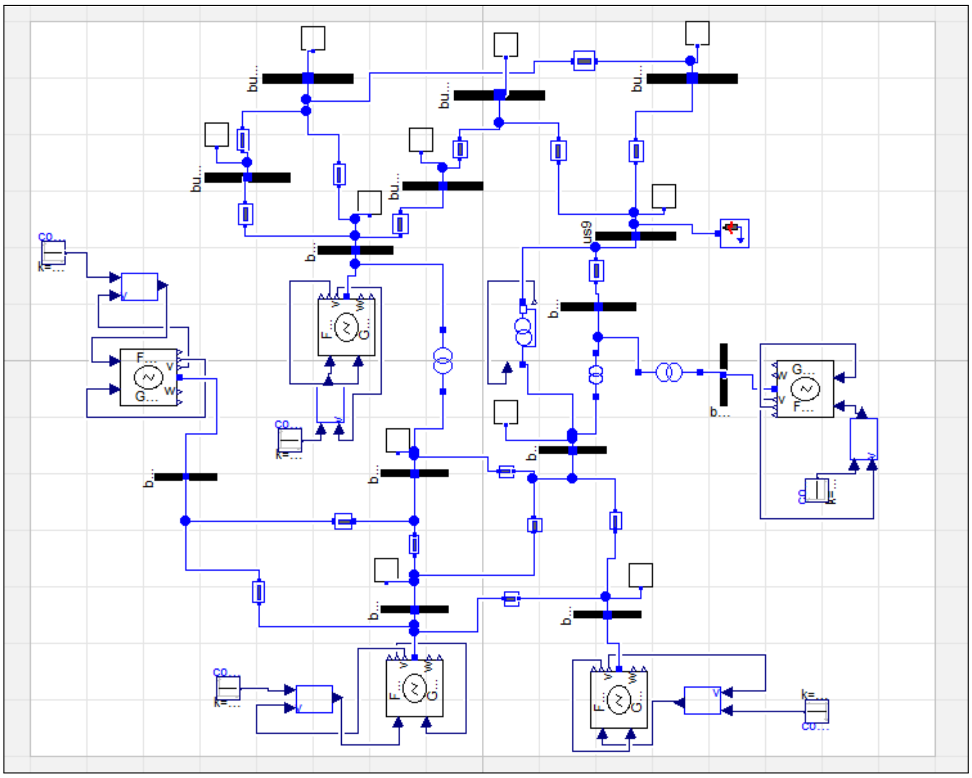


Figure A.2: IEEE 14-Bus Test system with PST (Between Bus 4 and Bus 9) in Modelica.

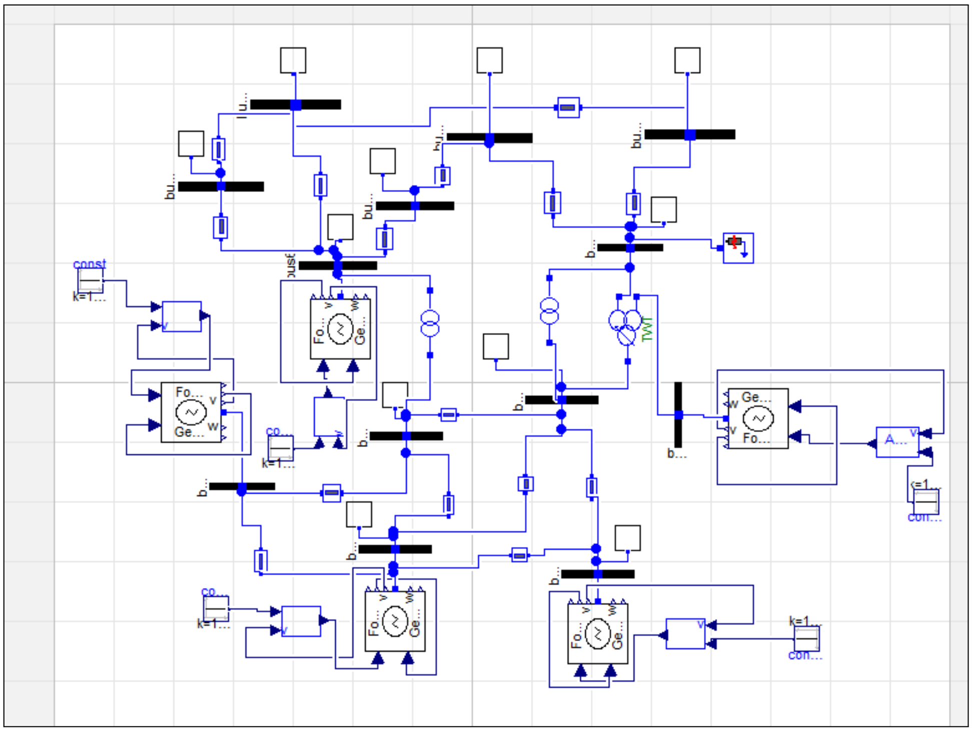


Figure A.3: IEEE 14-Bus Test system with TWT in Modelica.

A.2 Software to Software Validation

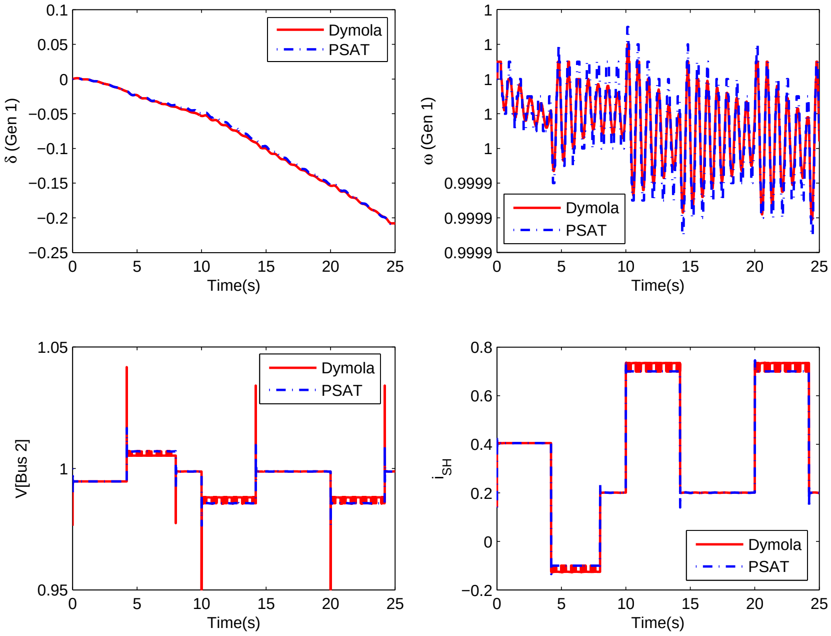


Figure A.4: Software to software validation of STATCOM Text based Model.

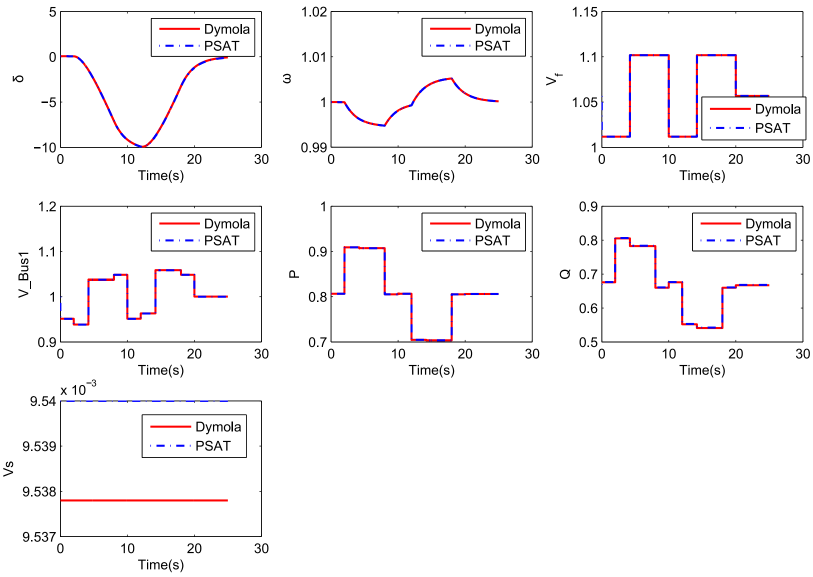


Figure A.5: Software to software validation of SSSC in constant voltage mode.

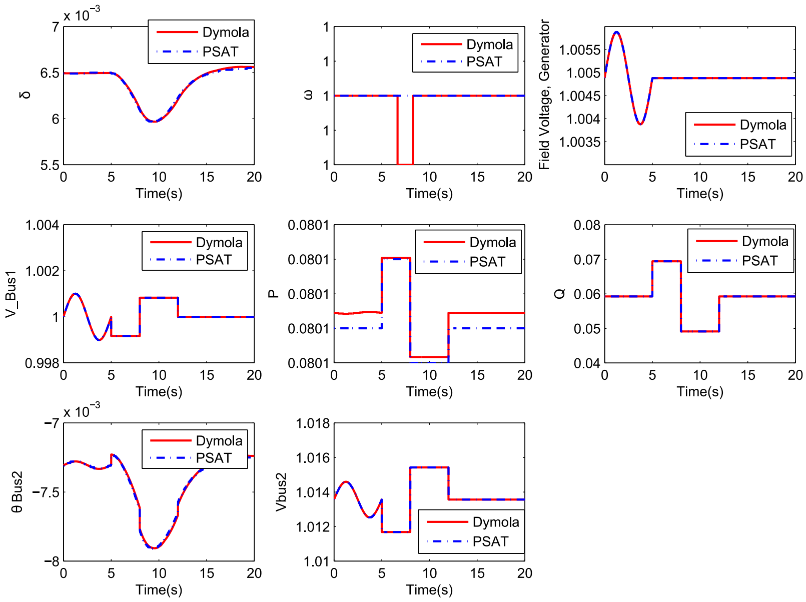


Figure A.6: Software to software validation of TWT.

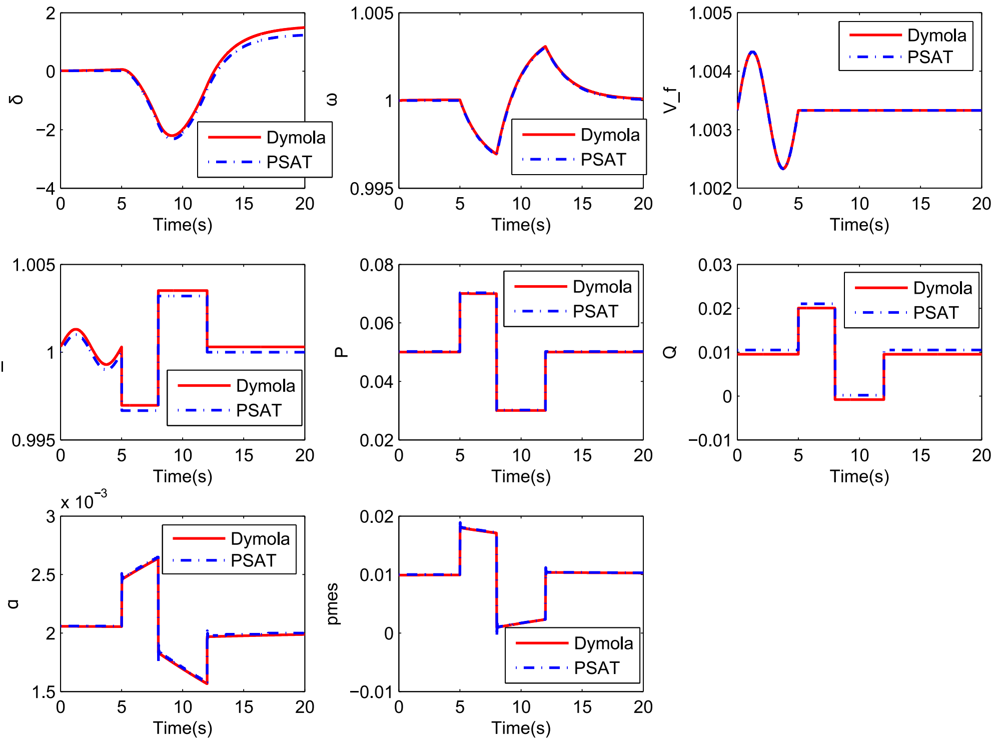


Figure A.7: Software to software validation of PST.

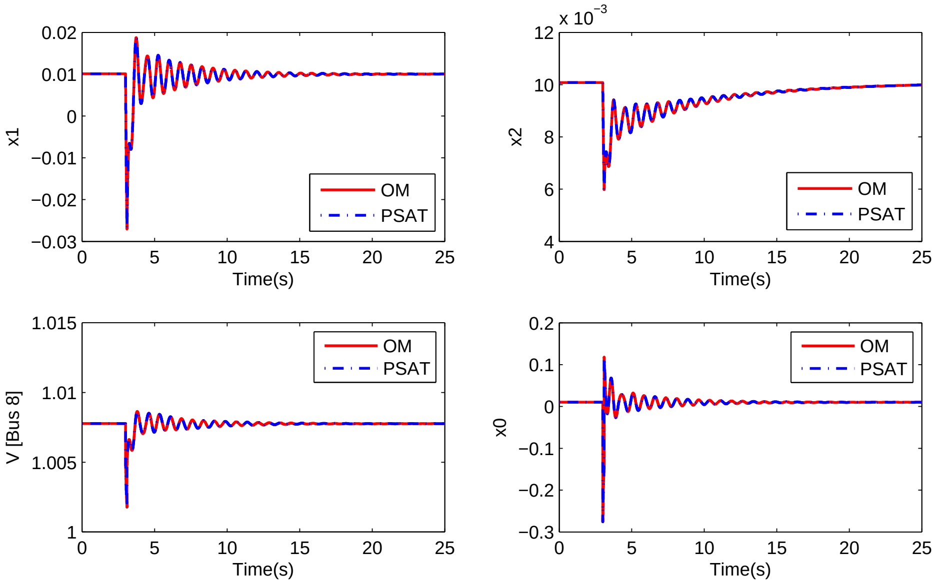


Figure A.8: Software to software validation of IEEE 9-Bus system with TCSC.

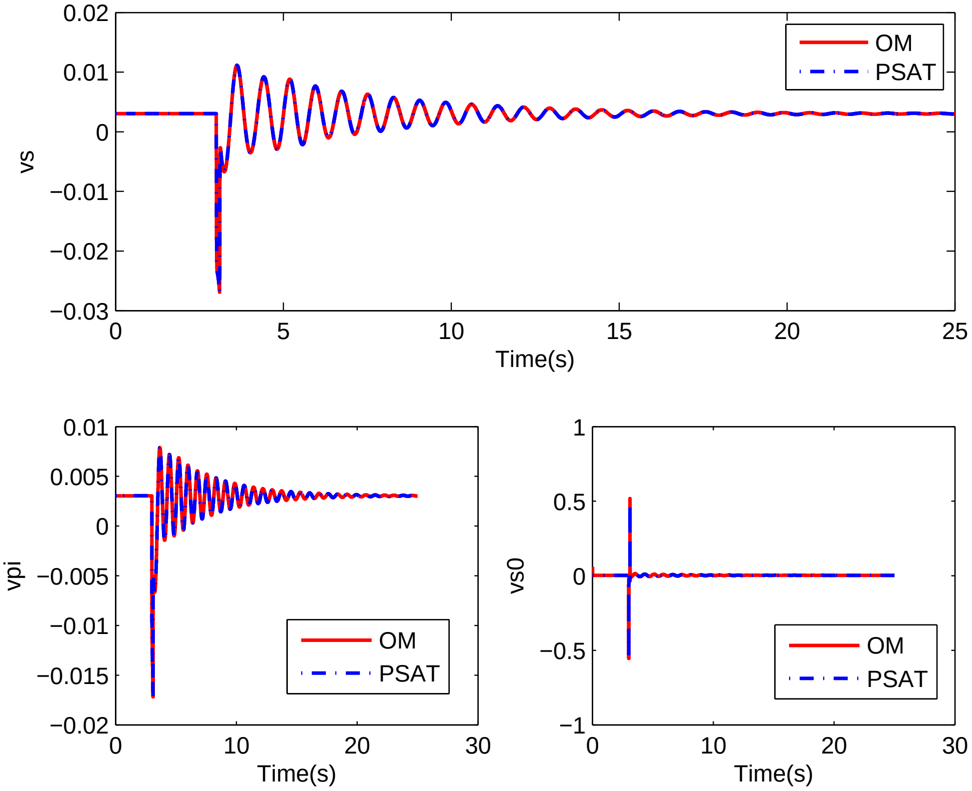


Figure A.9: Software to software validation of IEEE 9-Bus system with SSSC.

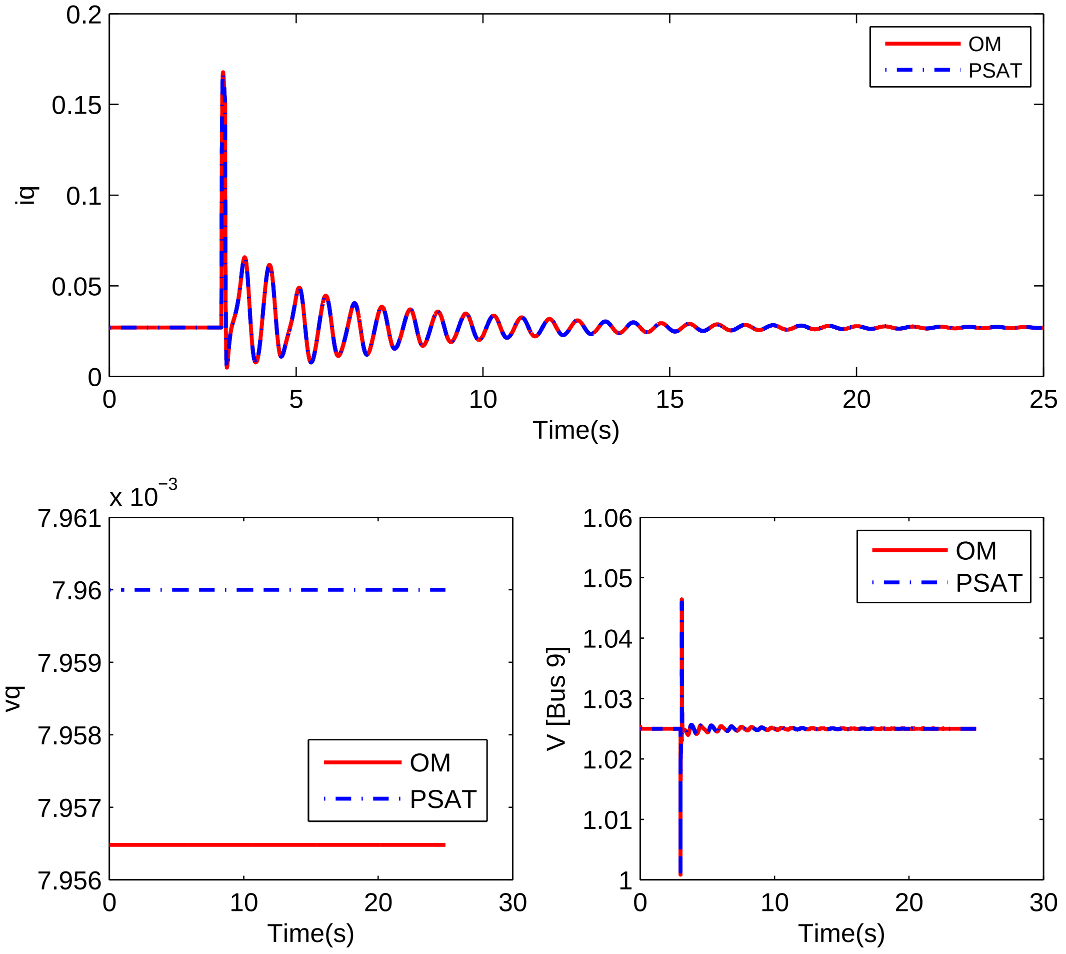


Figure A.10: Software to software validation of IEEE 9-Bus system with UPFC.

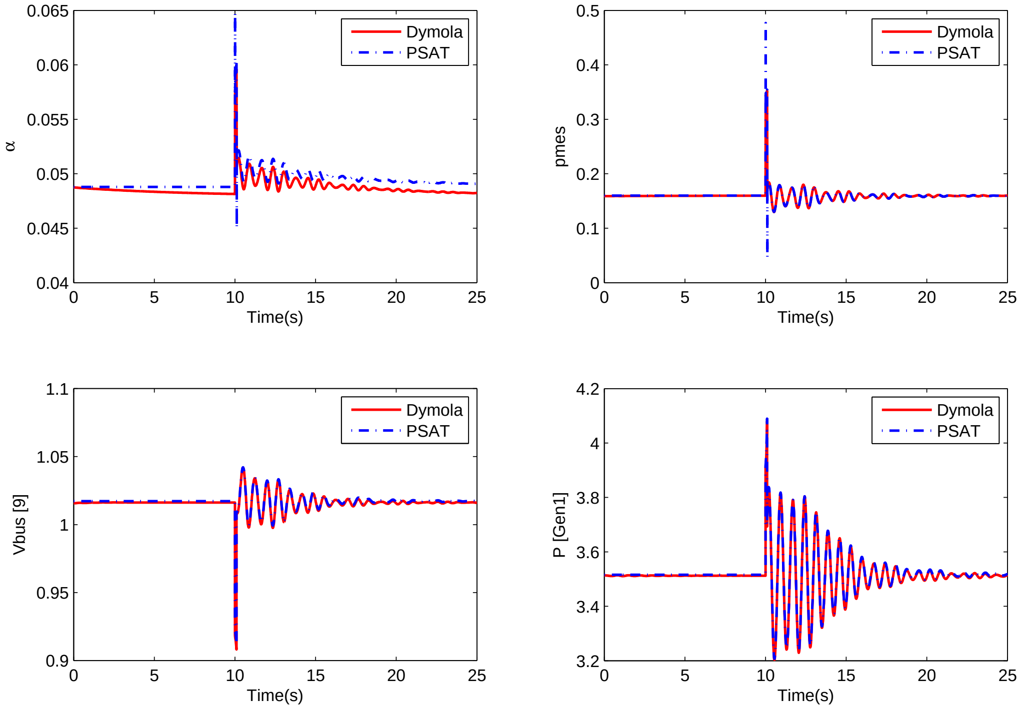


Figure A.11: Software to software validation of IEEE-14 Bus system with PST.

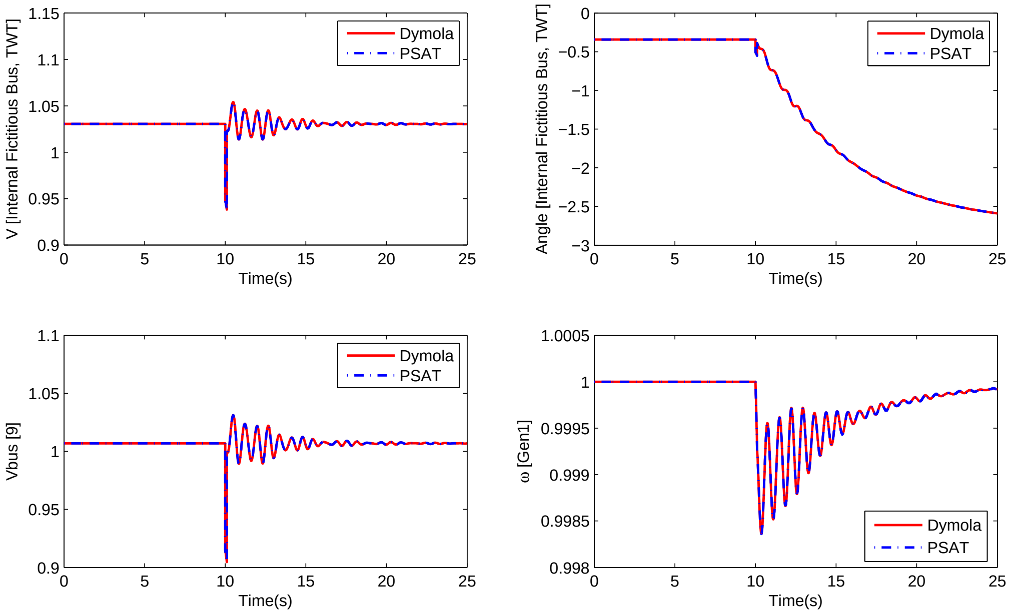


Figure A.12: Software to software validation of IEEE 14-Bus system with TWT.

A.3 Test System Data

A.3.1 9 Bus Test System

The complete data of IEEE 9-Bus test system (shown in figure A.13) is given below:

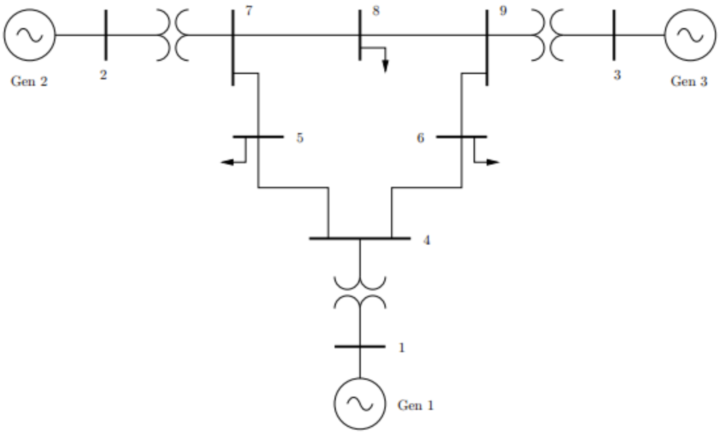


Figure A.13: WSCC 9-Bus test System [27].

Table A.1: Synchronous Generator Data

Parameter	**Symbol**	**Unit**	**Generator 1**	**Generator 2**	**Generator 3**
Bus Number	-	-	1	2	3
Power rating	S_n	MVA	100	100	100
Voltage rating	V_n	kV	16.5	18	13.8
Frequency rating	f_n	Hz	60	60	60
Machine order	-	-	4	4	4
Armature resitance	r_a	p.u.	0	0	0
d-axis synchronous reactance	x_d	p.u.	.1460	.8958	1.3125
d-axis transient reactance	x'_d	p.u.	.0608	.1198	.1813
d-axis sub-transient reactance	x''_d	p.u.	0	0	0
d-axis open circuit transient time constant	T'_{d0}	s	8.96	6	5.89
q-axis synchronous reactance	x_q	p.u.	.0969	.8645	1.2578
q-axis transient reactance	x'_q	p.u.	.0969	.1969	.2500
q-axis sub-transient reactance	x''_q	p.u.	0	0	0
q-axis open circuit transient time constant	T'_{q0}	s	.310	.5350	0.6
Mechanical starting time	$M = 2H$	kWs/kVA	47.28	12.80	6.02
Damping Coefficent	D	-	0	0	0
Speed feedback gain	K_w	gain	0	0	0
Active power feedback gain	K_P	gain	0	0	0

Table A.2: Line Data

Line	**From Bus**	**To Bus**	**Resistance (r, p.u.)**	**Reactance (x, p.u.)**	**Shunt susceptance (b, p.u.)**
Line	9	8	.0119	.1008	.209
Line 1	7	8	.0085	.072	.149
Line 2	9	6	.039	.17	.358
Line 3	7	5	.032	.161	.306
Line 4	5	4	.01	.085	.176
Line 5	6	4	.017	.092	0

Table A.3: Two winding transformer data

Two Winding Transformer	**From Bus**	**To Bus**	**Resistance (r, p.u.)**	**Reactance (x, p.u.)**	Primary and Secondary voltage ratio (KV/KV)
	2	7	0	.0625	18/230
	3	9	0	.0586	13.8/230
	1	4	0	.0576	16.5/230

Table A.4: PQ Load data

Parameter	Symbol	Load 1	Load 2	Load 3
Bus Number	-	6	8	5
Power Rating	S_n	100	100	100
Active Power Rating	P_0	.90	1	1.25
Reactive Power Rating	Q_0	.30	0.35	0.5

Table A.5: AVR data

Parameter	Symbol	AVR 1	AVR 2	AVR 3
Genrator Number	-	1	2	3
Exciter Type	-	2	2	2
Maximum regulator voltage	v_r^{max}	5	5	5
Minimum regulator voltage	v_r^{min}	-5	-5	-5
Amplifier gain	K_a	20	20	20
Amplifier time constant	T_a	0.2	0.2	0.2
Stabilizer gain	K_f	.063	.063	.063
Stabilizer time constant	T_f	0.35	0.35	0.35
Field circuit integral deviation	K_e	1	1	1
Field circuit time constant	T_e	0.314	0.314	0.314
Measurement time constant	T_r	.001	.001	.001
1st ceiling coefficient	A_e	.0039	.0039	.0039
2nd ceiling coefficient	B_e	1.555	1.555	1.555

A.3.2 14-Bus Test System

The single line diagram of IEEE 14-Bus system with the complete data is given in following.

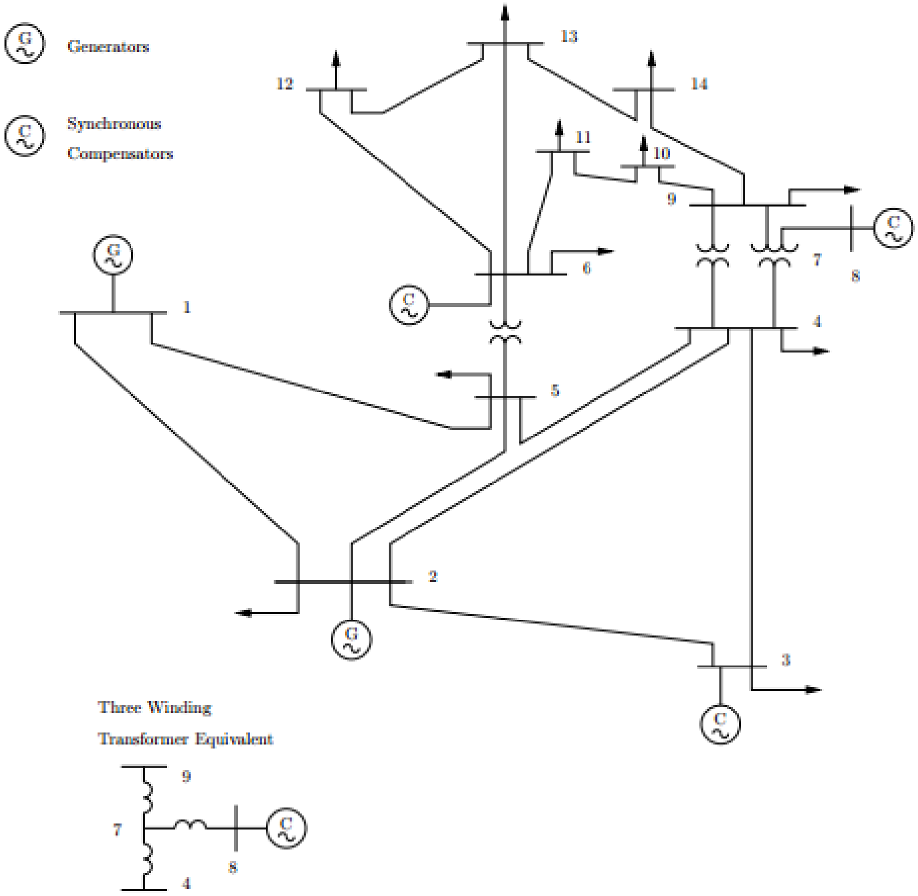


Figure A.14: IEEE 14-Bus test System [27].

Table A.6: Synchronous Generator Data of IEEE 14-Bus system

Parameter	**Symbol**	**Unit**	**Gen 1**	**Gen 2**	**Gen 3**	**Gen 4**	**Gen 5**
Bus Number	-	-	1	3	2	8	6
Power rating	S_n	MVA	100	100	100	100	100
Voltage rating	V_n	kV	615	69	69	25	25
Frequency rating	f_n	Hz	60	60	60	60	60
Machine order	-	-	4	4	4	4	4
Armature resitance	r_a	p.u.	0	.0031	.0031	0.0014	0.0014
d-axis synchronous reactance	x_d	p.u.	.8979	1.05	1.05	1.25	1.25
d-axis transient reactance	x'_d	p.u.	.232	.1850	.1850	.232	.232
d-axis sub-transient reactance	x''_d	p.u.	0.4	0.13	0.13	0.12	0.12
d-axis open circuit transient time constant	T'_{d0}	s	7.4	6.1	6.1	4.75	4.75
q-axis synchronous reactance	x_q	p.u.	.0969	.8645	1.2578		
q-axis transient reactance	x'_q	p.u.	.646	.98	.98	1.22	1.22
q-axis sub-transient reactance	x''_q	p.u.	.646	.36	.36	.715	.715
q-axis open circuit transient time constant	T'_{q0}	s	.001	.3	0.3	1.5	1.5
Mechanical starting time	$M = 2H$	kWs/kVA	10.296	13.08	13.08	10.12	10.12
Damping Coefficent	D	-	2	2	2	2	2
Speed feedback gain	K_w	gain	0	0	0	0	0
Active power feedback gain	K_P	gain	0	0	0	0	0

Table A.7: Line Data of IEEE 14-Bus System

From Bus	**To Bus**	**Resistance (r, p.u.)**	**Reactance (x, p.u.)**	**Shunt susceptance (b, p.u.)**
2	5	.05695	.17388	.304
6	12	.12291	.25581	0
12	13	.22092	.19988	0
6	11	.09498	.1989	.306
11	10	.08205	.19207	0
9	10	.03181	.0845	0
9	14	.12711	.27038	0
14	13	.17093	.34802	0
7	9	0	.11001	0
1	2	.01938	.05917	.0528
3	2	.04699	.19797	.0438
3	4	.06701	.17103	.0346
1	5	.05403	.22304	.0492
5	4	.01335	.043211	.0128
2	4	.05811	.17632	.0374

Table A.8: Two winding transformer data of IEEE-14 Bus system

Two Winding Transformer	**From Bus**	**To Bus**	**Resistance (r, p.u.)**	**Reactance (x, p.u.)**	Primary and Secondary voltage ratio (KV/KV)
	5	6	0	.25202	69/13.8
	4	7	0	.20912	69/13.8
	8	7	0	.17615	18/13.8

Table A.9: PQ Load data of IEEE-14 Bus system

Bus Number	Power Rating (S_n)	Active Power Rating (P_0)	Reactive Power Rating (Q_0)
11	100	.049	.0252
13	100	.189	.0812
3	100	1.3188	.266
5	100	.1064	.0224
2	100	.3038	.1778
6	100	.1568	.105
4	100	.6692	.056
14	100	.2086	.07
12	100	.0854	.0224
10	100	.126	.0812
9	100	.413	.2324

Table A.10: AVR data

Parameter	Symbol	AVR 1	AVR 2	AVR 3	AVR 4	AVR 5
Genrator Number	-	1	3	2	4	5
Exciter Type	-	2	2	2	2	2
Maximum regulator voltage	v_r^{max}	7.32	4.38	4.38	6.81	6.81
Minimum regulator voltage	v_r^{min}	0	0	0	1.395	1.395
Amplifier gain	K_a	200	20	20	20	20
Amplifier time constant	T_a	0.02	0.02	0.02	0.02	0.02
Stabilizer gain	K_f	.002	.001	.001	.001	.001
Stabilizer time constant	T_f	1	1	1	1	1
Field circuit integral deviation	K_e	1	1	1	1	1
Field circuit time constant	T_e	0.2	1.98	1.98	0.7	0.7
Measurement time constant	T_r	.001	.001	.001	.001	.001
1st ceiling coefficient	A_e	.0006	.0006	.0006	.0006	.0006
2nd ceiling coefficient	B_e	0.9	0.9	0.9	0.9	0.9

Table A.11: ULTC (in IEEE-14 Bus) data

Parameter	Symbol	Unit	Value	Parameter	Symbol	Unit	Value
Sending end bus nominal voltage	$Vbus1$	V	69000	Inverse time constant	K	1/s	0.10
Receiving end bus voltage	$Vbus2$	V	13800	Max tap ratio	m_{max}	p.u./p.u.	1.20
Power Rating	Sn	MVA	100	Min tap ratio	m_{min}	p.u./p.u.	0.8
Voltage rating	Vn	V	69000	Tap ratio step	$deltam$	p.u./p.u.	0
Frequency rating	fn	Hz	60	Reference voltage	v_ref	p.u.	1.0129
Nominal tap ratio	kT	KV/KV	5	Transformer Reactance	xT	p.u.	0.55618
Integral deviation	H	p.u.	0.10	Transformer Resistance	rT	p.u.	0

Table A.12: PST (in IEEE-14 Bus) data

Parameter	Symbol	Unit	Value	Parameter	Symbol	Unit	Value
Sending end bus nominal voltage	$Vbus1$	V	69000	Proportional gain	K_p	-	0.05
Receiving end bus voltage	$Vbus2$	V	13800	Integral gain	K_i	-	0.1
Power Rating	Sn	MVA	100	Reference Power	p_{ref}	p.u.	0.16
Primary Voltage rating	$Vn1$	V	69000	Maximum Phase angle	$alpha_{max}$	rad	pi/2
Secondary voltage rating	$Vn2$	V	13800	Maximum Phase angle	$alpha_{min}$	rad	-pi/2
Frequency rating	fn	Hz	60	Transformer Reactance	xT	p.u.	0.55618
Measurement time constant	T_m	s	0.001	Transformer Resistance	rT	p.u.	0
Fixed Tap ratio	m	p.u./p.u.	0.939				

Table A.13: TWT (in IEEE-14 Bus) data

Parameter	Symbol	Unit	Value	Parameter	Symbol	Unit	Value
Power Rating	Sn	MVA	100	Resistance of branch 1-3	$R13$	p.u.	0
Sending end bus voltage	$Vbus$	V	69000	Resistance of branch 2-3	$R23$	p.u.	0
Voltage rating of the first winding	$Vn1$	V	69000	Reactance of branch 1-2	$X12$	p.u.	.31193
Voltage rating of the second winding	$Vn2$	V	13800	Reactance of branch 1-3	$X13$	p.u.	.385127
Voltage rating of the third winding	$Vn3$	V	18000	Reactance of branch 2-3	$X23$	p.u.	.28616
Frequency rating	fn	Hz	60	Fixed tap ratio	m	p.u./p.u.	1
Resistance of branch 1-2	$R12$	p.u.	0				

References

[1] Jan Machowski, Janusz W.Bialek and James R. Bumby, *POWER SYSTEM DYNAMICS: Stability and Control*, John Wiley & Sons Ltd, 2008.

[2] Federico Milano. *Power system modelling and scripting.* Springer, 2010.

[3] Neville Watson and Jos Arrillaga, Power Systems Electromagnetic Transients Simulation, IET Power and Energy Series 39, 2003.

[4] Mahseredjian, J.; Dinavahi, V.; Martinez, J.A., *"Simulation Tools for Electromagnetic Transients in Power Systems: Overview and Challenges,"* Power Delivery, IEEE Transactions on , vol.24, no.3, pp.1657,1669, July 2009.

[5] Online Web: http://emtp.com/

[6] Online: http://faraday1.ucd.ie/psat.html

[7] Online: http://www.eurostag.be/

[8] Mohamad, A.M.; Hashim, N.; Hamzah, N.; Ismail, N.F.N.; Latip, M.F.A., "Transient stability analysis on Sarawak's Grid using Power System Simulator for Engineering (PSS/E)," Industrial Electronics and Applications (ISIEA), 2011 IEEE Symposium on , vol., no., pp.521,526, 25-28 Sept. 2011.

[9] Available [Online] https://hvdc.ca/pscad/

[10] T.Van Cutsem, M.-E. Grenier, D. Lefebvre, Combined detailed and quasi steady-state time simulations for large-disturbance analysis, International Journal of Electrical Power & Energy Systems, Volume 28, Issue 9, November 2006, Pages 634-642, ISSN 0142-0615.

[11] ASTRE : aAdynamic voltage security analysis of transmission networks, [Available online http://www.rte-

france.com/uploads/Mediatheque_docs/international_prestations... /logiciels/an/ASTRE_AN.pdf]

[12] E.Lambert, X.Yang, and X.Legrand,*Is CIM suitable for deriving a portable data format for simulation tools?* IEEE PES General Meeting, pp.19, July 2011.

[13] C. A. Caizares and Z. T. Faur,*Advantages and disadvantages of using various computer tools in electrical engineering courses,*IEEE Trans. Educ., vol. 40, no. 3, pp. 166171, Aug. 1997.

[14] Milano, F.; Vanfretti, L.; Morataya, J.C., *"An Open Source Power System Virtual Laboratory: The PSAT Case and Experience,"* IEEE Transactions on, vol.51, no.1, pp.17, 23 Feb. 2008.

[15] C. A. Caizares and F. Alvarado, UWPFLOW: Continuation and Direct Methods to Locate Fold Bifurcations in AC/DC/FACTS Power Systems, Apr. 2006 [Online]. Available: http://thunderbox.uwaterloo.ca/ claudio/software/pflow.html

[16] R. D. Zimmerman and R. J. Thomas, *PowerWeb: A tool for evaluating economic and reliability impacts of electric power market designs,* in Proc. IEEE Power Systems Conf. and Exposition, New York, Oct. 2004, pp. 15621567.

[17] M. Larsson,*ObjectStab: An educational tool for power system stability studies,*IEEE Trans. Power Syst., vol. 19, no. 1, pp. 5663, Feb. 2004.

[18] 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 , vol., no., pp.1,5, 21-25 July 2013.

[19] iTesla: Innovative Tools for Electrical System Security within Large Areas. [online] http://www.itesla-project.eu.

[20] Peter Fritzson, *Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica*, Wiley, 2011.

[21] Michael M. Tiller, Modelica by Example, 2014, Early access available : [http://book.xogeny.com/]

[22] Dynamic Modeling Laboratory, Users Manual, Version 5.3a.

[23] Bogodorova, T.; Sabate, M.; Leon, G.; Vanfretti, L.; Halat, M.; Heyberger, J.B.; Panciatici, P.,*"A modelica power system library for phasor time-domain simulation," Innovative Smart Grid Technologies Europe (ISGT EUROPE)* , 2013 4th IEEE/PES , vol., no., pp.1,5, 6-9 Oct, 2013.

[24] Modelica®- A Unified Object-Oriented Language for Physical Systems Modeling , Language Specification, Version 3.2.

[25] Angela S Chieh, Patrick Panciatici, and Jerome Picard.*Power system modeling in Modelica for time-domain simulation. In PowerTech*, 2011 IEEE Trondheim, pages 18. IEEE, 2011.

[26] Sven Erik Mattsson, Hilding Elmqvist, Martin Otter, and Hans Olsson.*Initialization of hy-brid differential-algebraic equations in modelica 2.0.*,In 2nd Inter. Modelica Conference 2002, pages 915, 2002

[27] F. Milano, *Power System Analysis Toolbox Documentation for PSAT.* version 2.1.8, 2013.

[28] Mehrdad Ghandhari, Hector Latorre, Lennart Angquist, Hans-Peter Nee and Dirk Van Hertem, *The Impact of FACTS and HVDC Systems on Transient Stability and Power Oscillation Damping*, Royal Institute of Technology, 2012.

[29] Peter W. Sauer and M.A. Pai, *Power System Dynamics and Stability*, Prentice Hall, 1998, ISBN: 0-13-678830-0.

[30] Kodsi, S. K. M., Caizares, C. A. "Modelling and Simulation of IEEE 14 bus System with FACTS Controllers". Technical report,2003. University of Waterloo.

[31] Rogersten, R.; Vanfretti, L.; Wei Li; Lidong Zhang; Mitra, P., *"A quantitative method for the assessment of VSC-HVdc controller simulations in EMT tools,"* Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2014 IEEE PES , vol., no., pp.1,5, 12-15 Oct. 2014