

Object-Oriented Modeling and Simulation of Power Electronic Circuits and Systems for Power Quality Investigation

J. Agashe S. Krishanmurty V. V. Sastry V. Ajjarapu S. S. Venkata

Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011-3060
{janhavi, shashank, sastry, vajjarap, venkata}@iastate.edu

Abstract:

The object-oriented modeling language MODELICA is designed to allow convenient, component-oriented modeling of complex physical systems. The free MODELICA language, free libraries and simulation tools are readily available and ready-to-use. In this paper, an approach is presented that enables efficient modeling, design and simulation of various power electronic systems using a new library developed in MODELICA language. The measurement templates that are developed along with the built-in power system library (ObjectStab) are effective tools for studying various power quality related issues.

1. Introduction:

Power electronics is a complex field, which needs inputs from various fields of electrical engineering, system theory etc. The interdisciplinary nature of this topic is illustrated in the Figure 1.

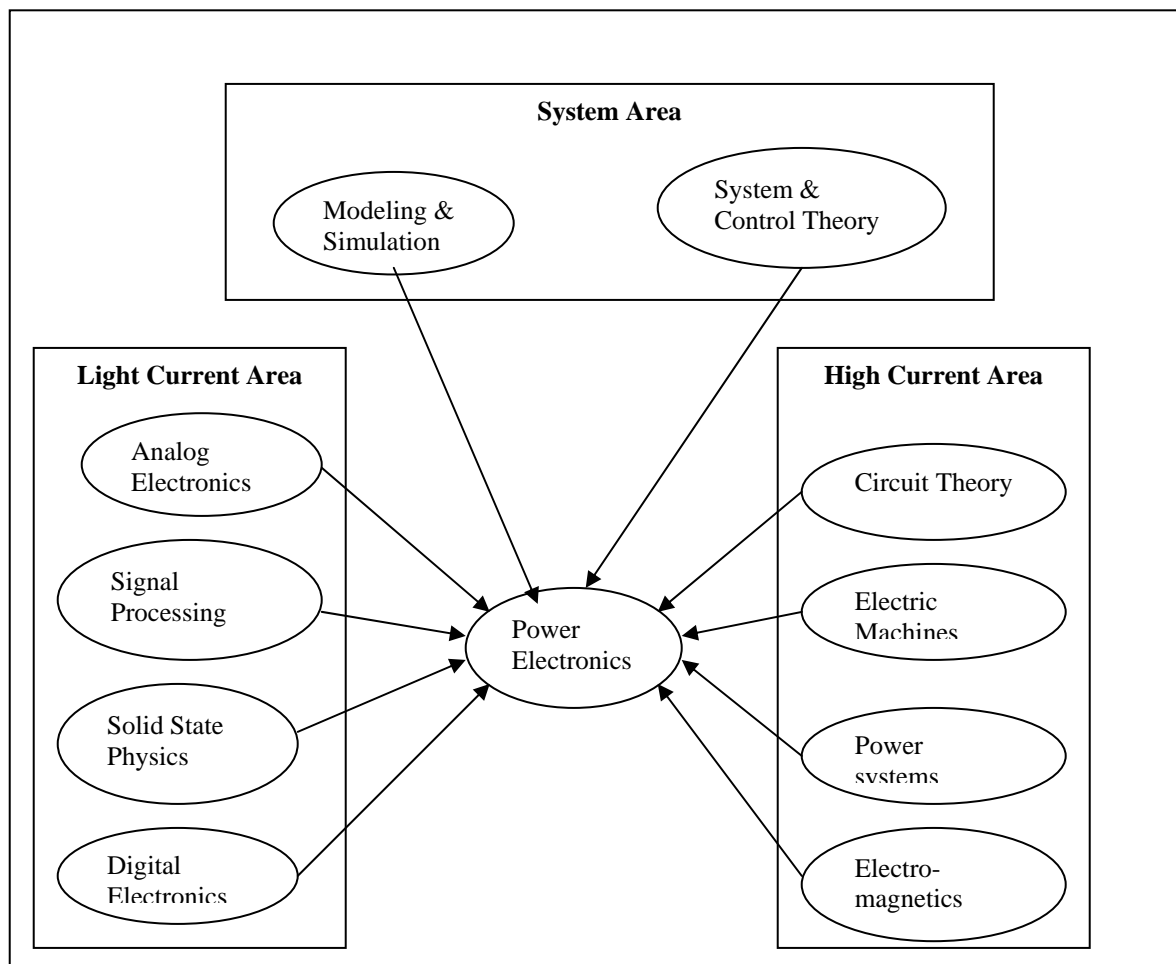


Figure 1:Power Electronics

To simplify the understanding of such a complex inter-disciplinary subject, computer-aided analysis and design tools are necessary. In addition to being multi-domain, power electronic systems are event-driven and usually involve both analog and digital components. Hence a simulator used for studying these systems needs to have capabilities to handle events, deal with hybrid, mixed-mode systems and support multi-domain modeling. Moreover such systems being very complex, it is desirable to have modularity on the modeling level, enabling reusability of models. This is possible if object-oriented modeling environments are used. SABER [1], DYMOLA [2] are few such tools available today. DYMOLA being available on PC platform is more user-friendly.

The object-oriented nature allows re-usability of models, with ease in maintenance of large software necessary for such complex mixed- mode systems. Modeling large systems using smaller and re-usable modules has since become a reality. This methodology has similarity to the Generic Object Models for Substation Field Equipment (GOMSFE) being advocated by EPRI centered on Utilities Communication Architecture (UCA) for substation automation [3].

2. Architecture of DYMOLA:

DYMOLA is an object-oriented modeling and simulation environment developed by DYNASIM. It can handle large multi-domain systems. It has a graphical front-end viz. DYMODRAW which helps in faster visualization, modeling and simulation with symbolic pre-processing. The simulation engine used is DYMOSIM, which is very powerful simulation program for solving ordinary differential equations (ODE) and differential-algebraic equations (DAE). It converts the differential-algebraic system of equations to the state-space form. Graph theory is used to identify the variables to be solved in each equation and to find minimal system of equations. Then the equations are solved symbolically or efficient code for numerical solution is generated. DYMOVIEW is used to visualize dynamic behavior. DYMOVIEW supports a 2- or 3-dimensional model-view built up by graphical objects to provide realistic graphical view and animation.

Decoupling of the modeling and simulation environments using PC windows environment is ideally suited for the study of power quality related issues effectively. Secondly, the graphical capabilities in terms of using DYMODRAW as pre-processor and DYMOVIEW as post-processor of the simulator helped in visualizing the system behavior at various levels resulting in better understanding of the concepts. It permits testing of new ideas such as on-line evaluation through simulation of “Total Harmonic Distortion”, as well as harmonic spectrum evaluation. In addition, average and RMS values of current /voltage can be evaluated. It has thus become an effective tool for research and extension of these ideas to real-time harmonic and VAR compensation. Moreover, these tools are cost-effective as compared to other simulators having the same features. The basic architecture of DYMOLA is shown in Figure 2.

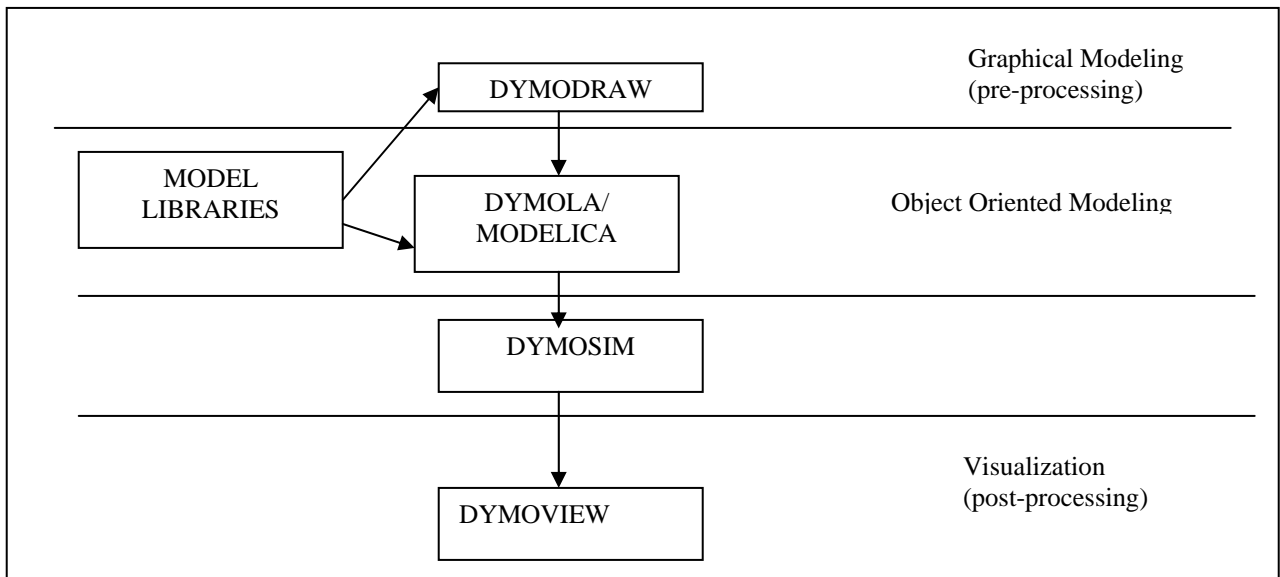


Figure 2: Architecture of DYMOLA

3. Modeling of Basic Components in DYMOLA:

Among the various components required for the modeling PE circuits/systems some will have certain common properties [6]. For e.g. Resistor, Diode, Voltage Source etc all have two connecting nodes by which they are connected to other components. Hence an object called “TwoPinComp” can be modeled and basic components like resistors etc can be modeled as the child objects of this parent object . Thus the class inheritance property of object-oriented modeling is used. Similarly other parent objects like “ThreePinComp”, “FourPinComp” etc are defined. The DYMOLA source code for these components is shown in Examples 1-3.

```
model class TwoPinComp
  cut in1(vin1/in1)
  cut in2(vin2/-in1)
end
```

Example 1:Basic Model for a Two Pin Component in DYMOLA

```
model class ThreePinComp
  cut in1(vin1/in1)
  cut in2(vin2/-in1)
  cut cont(Gat)
end
```

Example 2:Basic Model for a Three Pin Component in DYMOLA

```
model class FourPinComp
  cut act(vin1/in1)
  cut ref(vin2/in2)
  cut cont(Gat)
  cut neg(vneg/ineg)
end
```

Example 3:Basic Model for a Four Pin Component in DYMOLA

Based on these parent classes various components are modeled. For e.g. thyristor is modeled as a low/high resistor during ON/OFF state respectively. It is derived from the “ThreePinComp” class. Similar philosophy is used to develop various other models, which include switches, firing circuits for these switches, electrical machines etc. The source code for a thyristor is shown in Appendix A.

4. Measurement Templates for Power Quality Analysis:

The amount of harmonic distortion in the power systems today is constantly increasing. This is due to increase in number and power of various non-linear loads. In addition various power electronic systems like controlled rectifiers, inverters, choppers etc. inject harmonics into the system. It is hence essential to monitor various power quality related aspects like harmonic distortion, ripple etc. Hence various measurement templates have been developed to facilitate the study of these issues.

The basic concepts of average, RMS etc. of any variable (say current or voltage) are used in developing the models for the measurement templates [8]. All these models are derived from the “TwoPinComp” base class. These are used in the calculation of total harmonic distortion (THD), power factor, displacement power factor, ripple factor etc. The discrete version of the equation for average and RMS function are used. The “sample” function defined in DYMOLA is used to find the value of the

variable at a particular instant and required quantities are calculated by an iterative technique of updating the sum every at sample point. The steps involved in the building such models are listed below:

- Equations specific to the measurement function are written the equation section of the DYMODRAW window.
- Graphical icon is attached to this equation.
- The program is saved as a “class” and the base class is entered as “TwoPinComp”.

The Figure 3 shows graphical representation of the measurement template library. The source code for total harmonic distortion template is given in Appendix B.

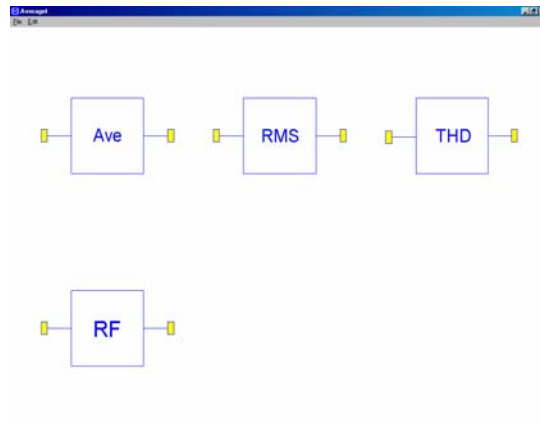


Figure 3: Measurement Library view in Dymola

5. AC/DC Power Electronic Converter

Figure 4 shows a simple three-phase full wave bridge rectifier circuit, modeled in DYMOLA using DYMODRAW as a front-end. The output voltage and current for a firing angle of 45 degrees are shown in Figure 5 (as seen in DYMOVIEV).

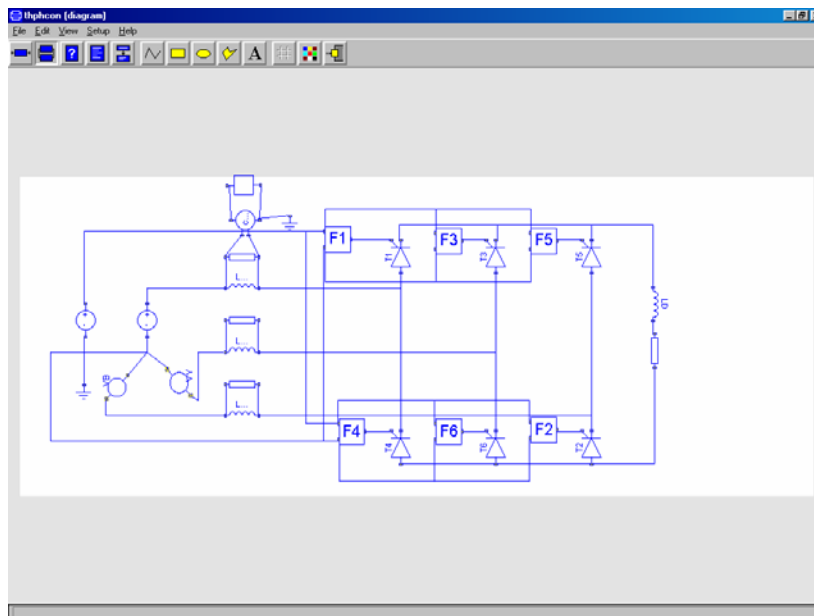


Figure 4: Three-phase bridge rectifier

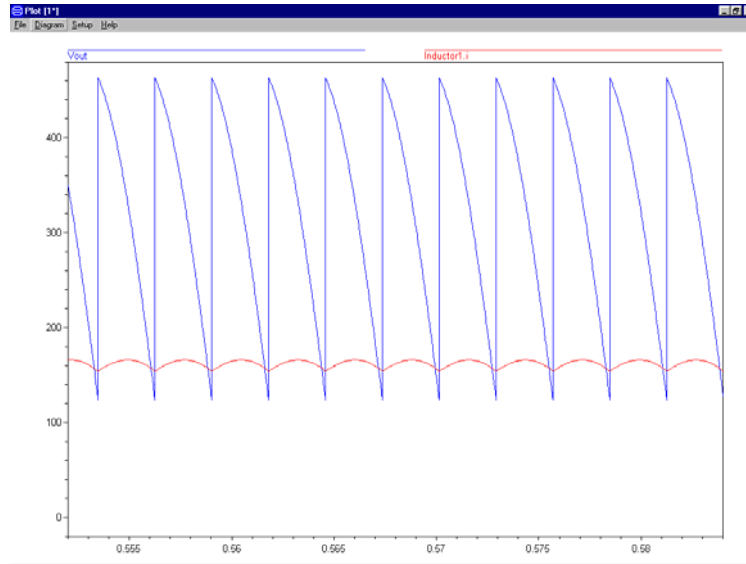


Figure 5: Output voltage and current for firing angle = 45 degrees

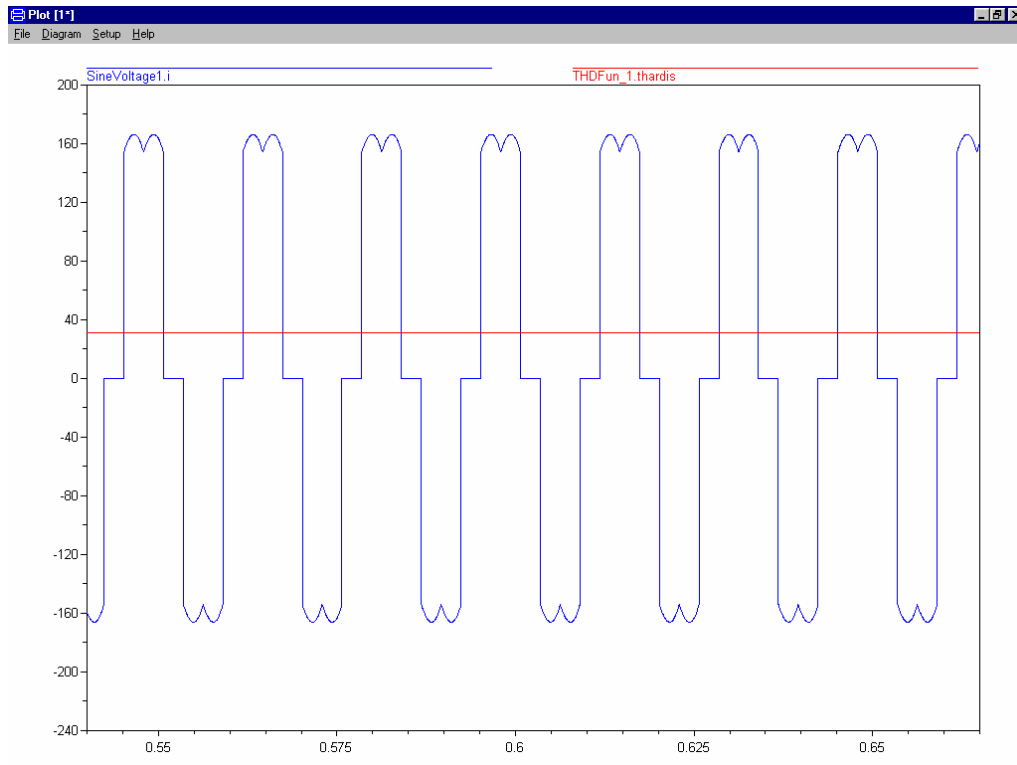


Figure 6 : Source current and THD for firing angle =45 degrees.

The output voltage and current waveforms are as expected. The source current waveform shows large amount of harmonic distortion, pre-dominantly fifth and seventh harmonics. The total harmonic distortion and source current waveform are also shown in Figure 6. The total harmonic distortion is 35%.

6. Modeling of Basic Components in MODELICA:

Now, MODELICA is a more effective object-oriented modeling language for physical system modeling, designed to allow convenient, component-oriented modeling of complex physical systems [4], [5]. It supports acausal modeling i.e. ports are not committed to 'input' and 'output' early in the modeling process. This leads to simpler models and efficient simulation. It has built-in libraries for various types of systems like electrical circuits, power systems, multi-body systems, drive trains, hydraulics, thermodynamic systems. Hence the power electronic components, measurement templates and electric machines libraries developed in DYMOLA are now converted to MODELICA. These in conjunction with the power system library of MODELICA viz. ObjectStab can be employed in analysis of various power system related issues like power quality, harmonic distortion, VAR compensation.

```
partial model TwoPin "Component with two pins"  
  SIunits.Voltage v "Voltage drop between the two pins (= p.v - n.v);"  
  SIunits.Current i "Current flowing from pin p to pin n";  
  Modelica.Electrical.Analog.Interfaces.PositivePin p "Positive pin"  
  Modelica.Electrical.Analog.Interfaces.NegativePin n "Negative pin"  
  equation  
    v = p.v - n.v;  
  end TwoPin;
```

Example 4: Basic model for Two Pin Component in MODELICA

```
partial model ThreePin "Component with three pins"  
  SIunits.Voltage v "Voltage drop between the two pins (= p.v - n.v);"  
  SIunits.Current i "Current flowing from pin p to pin n";  
  Modelica.Electrical.Analog.Interfaces.PositivePin p "Positive pin"  
  Modelica.Electrical.Analog.Interfaces.NegativePin n "Negative pin"  
  Modelica.Electrical.Analog.Interfaces.Pin g "Control Pin"  
  equation  
    v = p.v - n.v;  
    0 = p.i + n.i;  
    i = p.i;  
  end ThreePin;
```

Example 5: Basic model for Three Pin Component in MODELICA

```
partial model FourPin "Component with four pins"  
  SIunits.Voltage v1 "Voltage drop between the two pins (= p1.v - n1.v);"  
  SIunits.Current i1 "Current flowing from pin p1 to pin n1";  
  SIunits.Voltage v2 "Voltage drop between the two pins (= p2.v - n2.v);"  
  SIunits.Current i2 "Current flowing from pin p2 to pin n2";  
  Modelica.Electrical.Analog.Interfaces.PositivePin p1 "Positive pin"  
  Modelica.Electrical.Analog.Interfaces.NegativePin n1 "Negative pin"  
  Modelica.Electrical.Analog.Interfaces.NegativePin n1 "Control pin"  
  Modelica.Electrical.Analog.Interfaces.NegativePin n1 "Negative pin"  
  end FourPin;
```

Example 6: Basic model for Four Pin Component in MODELICA

As in DYMOLA, in MODELICA also one defines some basic classes which are equivalent to “TwoPinComp” etc. These are called “partial models” in MODELICA. The “extends” statement is used to derive other models from these. This is done by adding the equations specific to the model in the equation layer. The MODELICA source code for basic “TwoPin”, “ThreePin”, “FourPin” partial models are shown in Examples 4-6.

One important thing that can be noted, by comparing examples 1,2 and 3 with examples 4, 5 and 6 respectively, is the way in which DYMOLA and MODELICA define the interconnections between the components. DYMOLA uses the concept of the “Cut” while MODELICA uses the concept of “Pin” for the same. Hence the “across” and “through” variables of the “Cut” are replaced with the “current” and “voltage” of the “Pin” respectively. When the DYMOLA libraries were converted to MODELICA, the models would work only with other such converted components. It was not possible to interface these with the standard library components. This was due the differences in the interconnections.

There were other differences between MODELICA and DYMOLA that were observed during the process of conversion of models. To make the electrical machines models compatible with the mechanical interfaces library, additional equations were added to the machine models. MODELICA has a standard mechanical connector – flange that has torque and the angular position as the parameters. The DYMOLA model gave the output at the mechanical pin as torque and speed. Hence the equation for angular position as the integral of the speed was introduced in all machine models.

There are few syntactical differences in two modeling languages. MODELICA does not allow comparison of “Real” and “Boolean” variables in a logical expression. Wherever such situation arose, dummy Boolean variable was introduced. Also the manner in which MODELICA handles various looping structures is different from DYMOLA. Hence certain changes had to be made in the code.

After taking into account all these factors, the models have been used with the standard library components. This allowed the use of the available power system library viz. ObjectStab [10] and study the various power quality issues. Figure 7 shows typical power system components available in MODELICA.

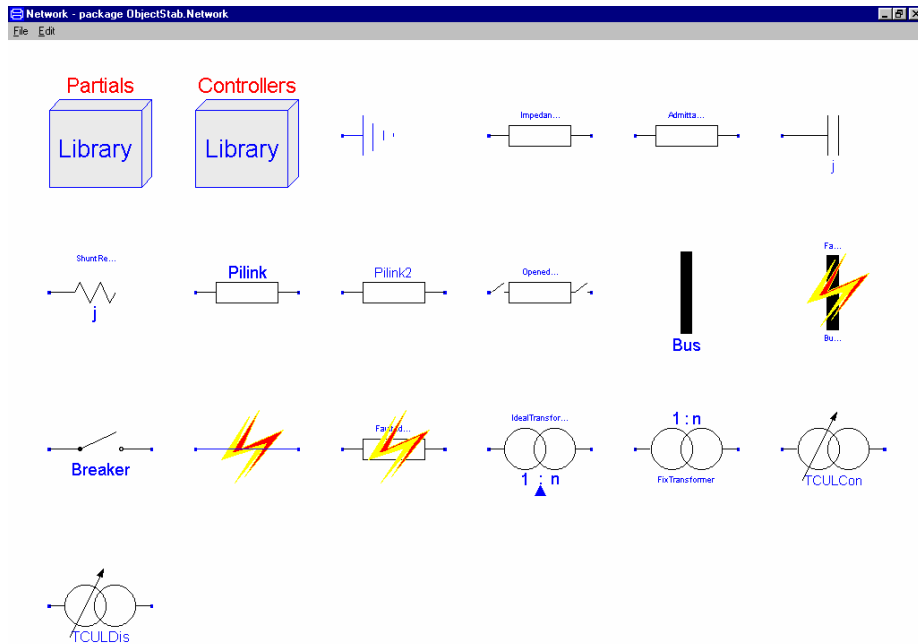


Figure 7: Typical power system components available in ObjectStab

The Table 1 shows the various models developed.

TwoPinComp	ThreePinComp	Others
Diode	Thyristor	DC Machine
Two Pin Switch	Thyristor Firing Circuit	Induction Machine
Measurement Templates	Three Pin Switch	
	PWM Firing Circuit	

Table 1: Various Models Developed

7. AC/DC Power Electronic Converter with Harmonic Filters:

Figure 8 shows a simple three-phase full wave bridge rectifier circuit, modeled in MODELICA. A fifth and seventh harmonic filter is introduced in all the three phases. The output voltage and current waveforms are similar as in the previous case (Figure 5). The total harmonic distortion shows a marked reduction as expected. The source current waveform also shows marked improvement (Figures 9 & 10). The reduction in THD is less as compared to the previous case.

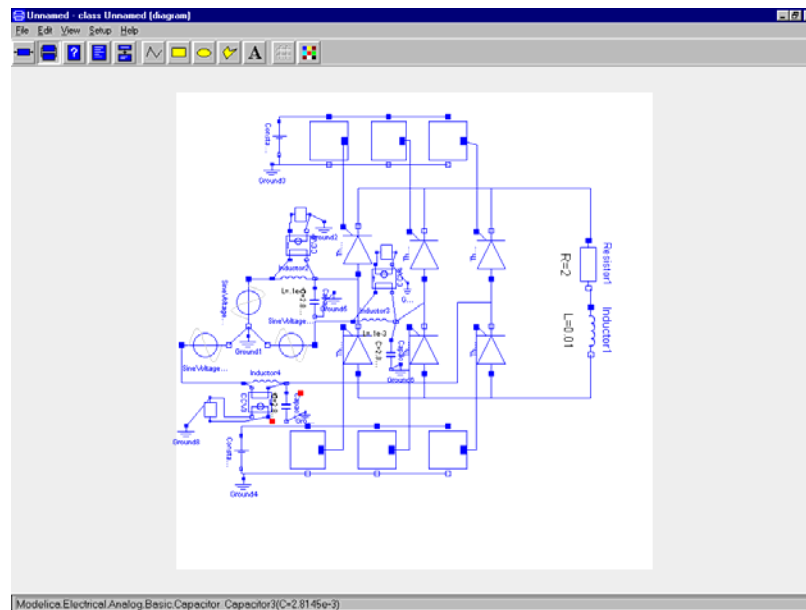


Figure 8: Three-phase rectifier with harmonic filter

The Figure 11 shows the effect of introduction of harmonic filters in the phases. The value of harmonic distortion reduces from 35% (without filter) to 31% (with 7th harmonic filter) to 25% (with 5th harmonic filter). The improvement in the source current waveform is also seen. These results obtained using MODELICA are validated from the previous results derived from DYMOLA based models.

8. Conclusions:

The use of MODELICA in modeling various power electronics and drive system components is illustrated. The application of Measurement Templates to illustrate Total Harmonic Distortion (THD) has been discussed in detail. The effectiveness of these templates in studying the various power quality issues like harmonic analysis is obvious. There is a possibility of use of these templates in online analysis and mitigation of power quality problems. These tools have been used for education purposes at Iowa State University.

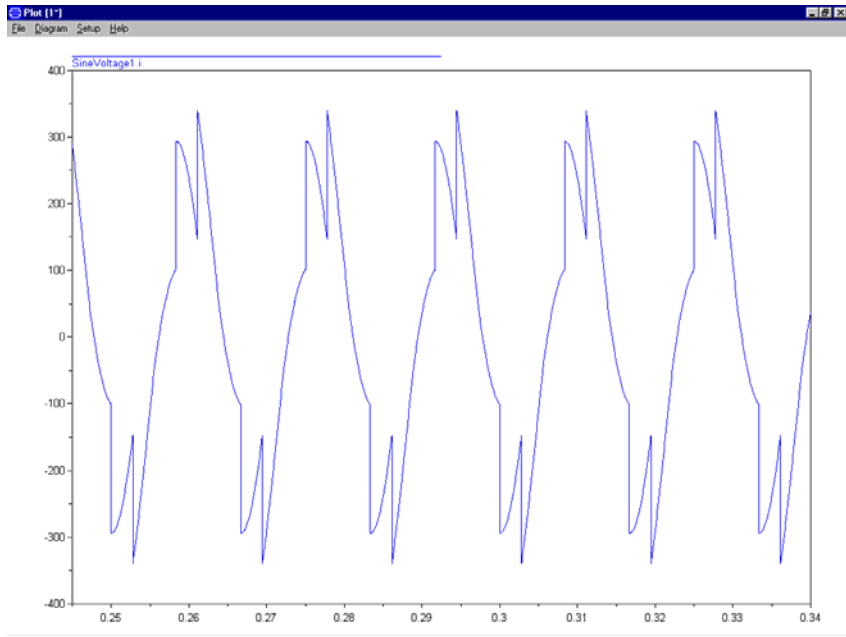


Figure 9: Source Current with 5th harmonic eliminated

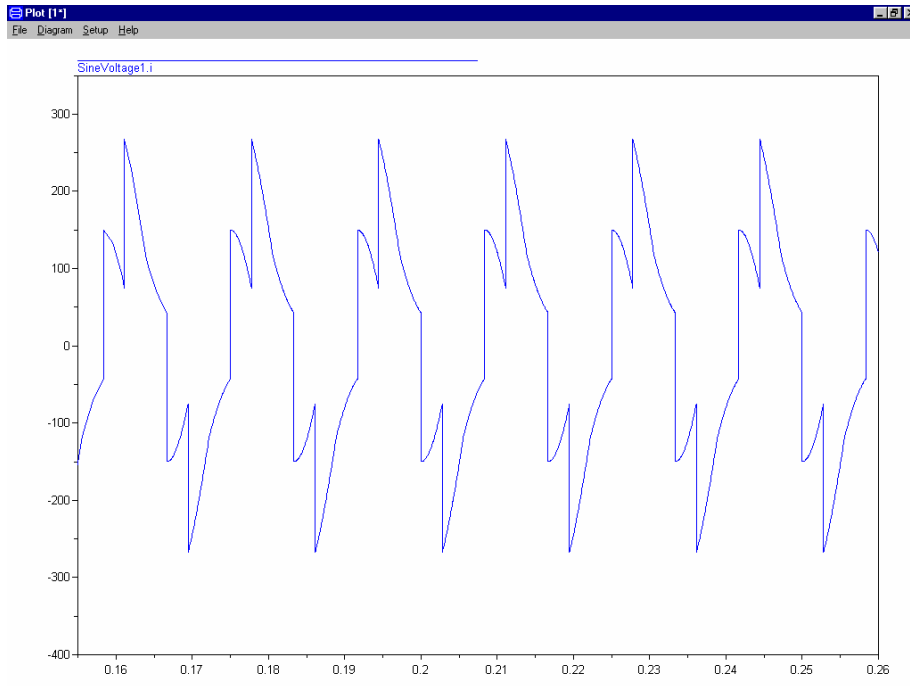


Figure 10: Source current with 7th harmonic eliminated

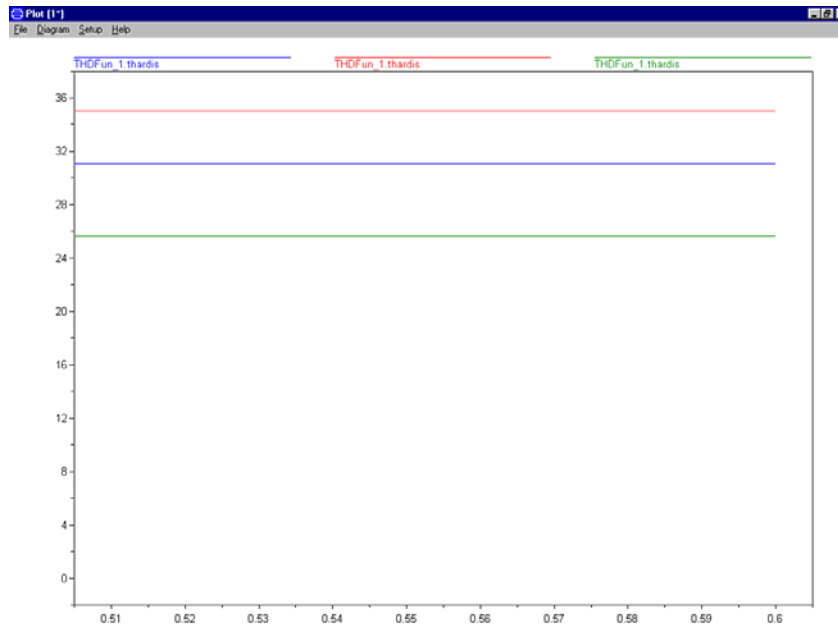


Figure 11: Total Harmonic Distortion for different cases
No Filter, 7th harmonic eliminated and 5th harmonic eliminated

REFERENCES:

1. <http://www.analogy.com/>
2. <http://www.dynasim.se>
3. Clinard, K.,-“ GOMSFE (Generic Object Models for Substation and Feeder Equipment) models of multifunctional microprocessor relays”
Power Engineering Society Summer Meeting, 1999. IEEE , Volume: 1 , 1999
Page(s): 36 -38 vol.1
4. <http://www.modelica.org/>
5. Michael Tiller, Introduction to Physical Modeling with Modelica , Kluwer Academic Publishers, Boston, USA, 2001
6. T.V.Sivakumar –“Object Oriented Modeling of Power Electronics Systems with an Emphasis on Education and Design”, PhD Thesis, IIT Madras, May 1999
7. V.V.Sastry, T.V.Sivakumar “Computer Aided Education of Power Electronics Using Object Oriented Modeling Language” EPE, Lausanne,1999
8. Rajagopalan, S.; Sastry, V.V.; Ajjarapu, V.; Venkata, S.S. -“ Object-oriented measurement templates for power electronics education and research using DYMOLA (Dynamic Modeling Laboratory)”
Computers in Power Electronics, 2000. COMPEL 2000. The 7th Workshop on , 2000
Page(s): 166 -171
9. <http://www.ee.iastate.edu/~sastry/ee452eh.htm>
10. Larsson, M. (2000). ObjectStab - A Modelica library for power system stability studies. Modelica Workshop 2000, Lund University, Sweden.

APPENDIX A: Source Code for a Thyristor in DYMOLA

```

model class (TwoPinComp) SW1 {* (-100, -100) (100, 100)}
  {* window 0.37 0.05 0.6 0.87 }
  {* info This is a model for Switch. This is derived from
class TwoPinComp
}
  local Op = false
  parameter GOP = 1.0E-5, RCl = 1.0E-5
  {* layer icon}
  cut cont {* InputCut at (-6, 40) (14, 60)} (Gat)
  {* layer diagram}
  0=if Op then (Va-Vb)-RCl*i else i-GOp*(Va-Vb)
  new(Op) = ( (Op) and (i>0) and (Gat) ) or ( not(Op) and (u>0) and (Gat) )
  {* layer icon}
  {* line (-100, 0) (-30, 0)}
  {* line (100, 0) (32, 0)}
  {* line (-30, 0) (30, 40)}
  {* line (0, 40) (0, 20)}
  {* layer diagram}
  {* using
using c:/dymola/library/dymlib/basic.dym
using c:/dymola/library/dymlib/cut.dym
}
  {* layer icon}
  {* text "%threepinswitch" (-52, -60) (48, -80)}
end

model class (TwoPinComp) Switch1 {* (-100, -100) (100, 100)}
  {* info This is to be used as base class for Thyristor
}
  terminal Op=true
  parameter GOP = 1.E-5, RCl = 1.E-5
  0 = if Op then i-GOp*(Va-Vb) else (Va-Vb)-RCl*i
end

model class (Switch1) Thyristor {* (-100, -100) (100, 100)}
  {* info this class defines the model for Thyristor. This is
a class derived from Switch1
}
  {* layer icon}
  cut cont {* InputCut at (100, 50) (120, 70)} (Gat)
  {* layer diagram}
  when (not(Op) and i<0) or (Op and u>0 and Gat) then
    new(Op) = (not(Op) and i<0) or (Op and not((u>0 and Gat)))
  endwhen
  {* layer icon}
  {* polygon (-60, 60) (-60, -60) (60, 0) (-60, 60)}
  {* line (-100, 0) (-60, 0)}
  {* line (60, 0) (100, 0)}
  {* line (60, 60) (60, -60)}
  {* line (60, 20) (100, 60)}
  {* layer diagram}
  {* using
using c:/dymola/library/dymlib/cut.dym
}
  {* layer icon}
  {* text "%name" (-40, 80) (40, 60)}
end

```

APPENDIX B: Source Code for in Total Harmonic Distortion Function DYMOLA

```

model class THDFun {* (-100, -100) (100, 100)}
  {* window 0.41 0.12 0.39 0.58 }
  parameter M,N
  local Io,p
  local u1
  output Irms,Is1,thardis,phi,Phif,PF,DPF
  parameter k=1,Is2=0.1,MF=0.77,f=60,Isa=0.1,Isb=0.1,shift=0,shift1=0,phaseang=0
  constant pi=3.142
  {* layer icon}
  cut Is1 {* InputCut at (-110, -12) (-90, 8)} (Va/i)
  cut Is2 {* InputCut at (98, -10) (118, 10)} (Vb/-i)
  {* rectangle (-60, 60) (60, -60)}
  {* line (-100, -2) (-60, -2)}
  {* line (60, 0) (100, 0)}
  {* text "THD" (-32, 34) (36, -34)}
  {* layer diagram}
  u1=Va-Vb

  M*der(i)=u1
  Io=u1

  when Sample(10/f,0.00005) and Time<30/f then
    new(k)=k+1
    new(Is2)=Io**2+Is2
    new(Isa)=Io*cos(2*pi*f*Time+phaseang)+Isa
    new(Isb)=-Io*sin(2*pi*f*Time+phaseang)+Isb
  endwhen
  new(shift)=if Isa<0 then 90 else shift
  new(shift1)=if Isb>0 then 270 else shift
  phi=atan(Isb/Isa)*(180/3.14159) - shift1
  new(Phif)=if abs(phi)>270 then phi+360 else phi
  DPF=cos(Phif*3.142/180)
  p=k
  Irms=sqrt(Is2/k)
  Is1=1.414*sqrt(Isa**2+Isb**2)/k
  thardis=100*sqrt(Irms**2-Is1**2)/Is1
  PF=(Is1/Irms)*DPF
  {* using
  using library/dymlib/cut.dym
  }
end

```