

An Incorporated Use of Fuzzy Logic Toolbox and Modelica Library to Design SSSC Damping Controller

Boonrat Somritvanitcha¹, Non-member,
Issarachai Ngamroo¹, Member, and Komsan Hongesombut², Non-member

ABSTRACT

ObjectStab is a general purpose simulation tool for power system stability studies developed by Modelica which is an object-oriented modeling language. It provides enough modeling flexibility to allow addition or modification of new power system components. This paper describes an incorporated use of fuzzy logic toolbox in Matlab/Simulink and ObjectStab library to enhance the application of this library into fuzzy control design environment. The example provided here is the modeling of the static synchronous series compensator (SSSC) which is the new device developed in the ObjectStab. In addition, the interface of ObjectStab with Matlab/Simulink for an SSSC damping controller design by fuzzy logic toolbox is explained step by step. Simulation studies in a multi-machine power system confirm the effectiveness of the designed fuzzy controller.

Keywords: Modelica, ObjectStab, Matlab, Power system modeling, Fuzzy control

1. INTRODUCTION

Conventionally, the sophisticated simulation tools for power system stability studies such as PSS/E, SIMPOW, DigSilent and EuroStag etc., have gained worldwide renown. While most of these tools are very efficient in computation and reasonably user-friendly, their components models are often encapsulated and difficult or impossible to examine and modify. Additionally, most of simulators often require substantial training in order to use them productively. These tools, therefore, are not suitable for academic and educational uses. The transparent and flexible component modelings are significantly required. With recent technology, the modeling problem can be made easier by use of Modelica language [1-3]. Modelica is an object-oriented equation-based programming language. Constructing the model by Modelica yields

two solutions; (1) a large model library with numerous extended models, (2) the use of a simulation interface that allows the user to add easily the new models to the program library. These features are suitable for development of new power system models using an object-oriented way. In Modelica, the library supported for power system stability studies is ObjectStab [4-6]. The aim of the ObjectStab library is to provide the environment within which users can quickly get started and to provide enough modeling flexibility to allow addition or modification of new power system components such as generators, loads, Flexible AC Transmission Systems (FACTS) devices, and control system models. The ObjectStab environment is completely open in contrast to many modeling tools that have a fixed set of component models and proprietary methods for introducing new components. Users of ObjectStab can easily introduce components that match the user's own and unique needs. Therefore, it lets the users build their own system in a very simple way. Besides, one of the most benefits of using ObjectStab is that all models can be exported for use in Matlab/Simulink, then a full set of Matlab Tools can be applied. This powerful characteristic significantly enhances the performance of the ObjectStab for design of power system damping controllers. On the other hand, applications of fuzzy logic theory to the electric power engineering have drawn tremendous attentions from researchers [7,8]. Fuzzy logic controllers are rapidly becoming viable alternatives for classical linear controllers. A fuzzy logic controller has several advantages compared to those from many conventional controllers; for instance, it offers a convenient way of designing a controller about the process being controlled from operator's experiences and expert's knowledge. This approach can enhance performance, adaptive property, reliability and robustness of the controllers better than conventional linear controllers.

The contribution of this paper is to develop a design method for power system damping controller by incorporating ObjectStab to use with a fuzzy logic toolbox in Matlab/Simulink. The example provided here is the modeling of the static synchronous series compensator (SSSC) [9,10] which is the new device developed in the ObjectStab. In fuzzy control design,

Manuscript received on December 20, 2005; revised on March 5, 2006.

¹The authors are with the Department of Electrical Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand; E-mail: ngamroo@yahoo.co.th

²The author is with the Power System Technology Group, Tokyo Electric Power Company, Yokohama, Japan; E-mail: Hongesombut.Komsan@tepco.co.jp

the simple approach to extract membership functions and fuzzy logic control rules based on observed signals is proposed. The interface of ObjectStab with Matlab/Simulink for fuzzy control design is described step by step. To validate the effectiveness of the designed fuzzy controller, several simulation studies have been performed on a two-area four-generator power system. Simulation results show that the proposed fuzzy controller of SSSC can effectively and robustly enhance the damping of power system oscillations.

2. OVERVIEW OF OBJECTSTAB LIBRARY FOR POWER SYSTEM SIMULATION

As the network of power system grows day by day, the existing power system models have to be upgraded or changed to meet new requirements. In conventional software tools, a minor change in power system models may need to upgrade an entire program. As a result, model modifications and evolutions may require a considerable time period. Object-oriented modeling is a technique to solve complex problems of modeling and simulation. It also provides a structured, computer-supported way of doing mathematical and equation based modeling. One of the major benefits of object-oriented modeling is the ability to extend the behavior and properties of an existing class. Recently, Modelica which is an object-oriented based language, has been paid attention by many researchers [1-3]. way of describing systems and their behavior of Modelica is at higher level of abstraction than the usual object-oriented programming since some implementation detailed can be omitted. Classes can be considered as blueprints for creating objects. Variables, equations and function definitions can be inherited among classes. Specifying behavior is primarily done through equations instead of via methods.

The development of Modelica package for power system simulations in this paper is initially motivated by the ObjectStab- A Modelica library for power system stability study of M. Larsson [4,5]. The ObjectStab has successfully applied in many applications [4-6]. The ObjectStab library contains the following models:

- Generator with constant frequency and voltage as slack or PV node or using third- or sixth-order dq model with detailed generator model of exciter and governor control systems or as quasi-steady state equivalents.
- Reactive power compensation device; shunt reactors, shunt capacitances, and series capacitances.
- Transmission lines in series with impedance and pi-link representation.
- Transformer with neglecting losses by magnetization and hysteresis currents.
- Static and dynamic loads
- Buses, faulted bus and faulted line with fault

impedance.

- Measurement instruments of voltage, current, active and reactive powers.

Modelica connectors are instances of connector classes, which define the variables that are part of the communication interface. Thus, connectors specify external interface for interaction. There are basically two kinds of variables in connectors, non-flow variables and flow variables. Two different types of coupling, which are (1) Equality coupling for non-flow variables, (2) Sum-to-zero-coupling according to Kirchoff's current law for flow variables, are established by connections depending on whether the variables in the connected connectors are non-flow or flow variable declared using *flow* prefix.

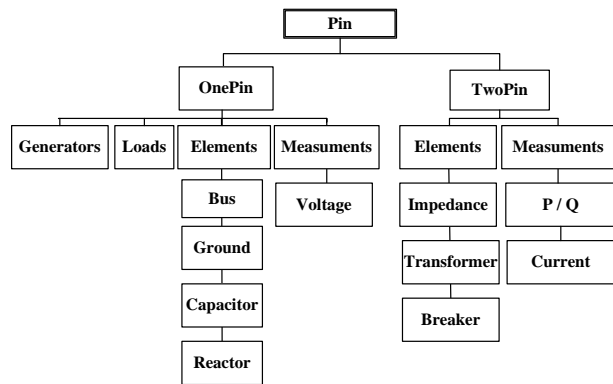


Fig.1: Classification of model components

In the Modelica package, a connector named Pin is described by phasor representation as in the ObjectStab[4]:

$$I = i_a + j i_b \quad (1)$$

$$V = (1 + v_a) + j v_b \quad (2)$$

Using the above definition for a connection point of power system components, an object named Pin can be constructed as a base class for power system connectors in such that voltages are non-flow variables and currents are flow variables. All power system models are treated as objects that are derived from its base class. The classification of the models has been done depending on the number of ports. Fig.1 shows the classification of models in the new developed Modelica package for power system simulations.

3. MODELING OF SSSC IN OBJECTSTAB

The example provided in this paper is the modeling of the static synchronous series compensator (SSSC) [9-11] which is the new device developed in the ObjectStab. SSSC is a FACTS device which is connected in series with a transmission line as depicted in Fig.2. An SSSC consists of voltage-sourced converter, capacitor and transformer. In principle, an SSSC is capable of interchange of active and reactive power

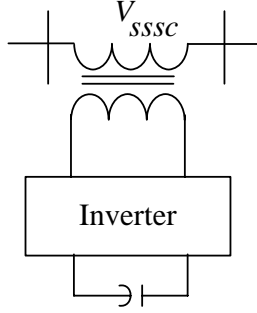


Fig.2: Configuration of SSSC

if sufficient energy is supplied. Here, only reactive power compensation is intended. Therefore, an SSSC is controlled by injecting a voltage with controllable magnitude and phase in the way that the vector of injected voltage is perpendicular to the line current. In physical modeling, an SSSC can be modeled as a series-connected voltage source in series with a reactance. Consider the free body diagram as shown in Fig.3, an SSSC model consists of an impedance object XS and a controllable voltage source object V_sssc connected in series with a line impedance XT. The SSSC model is defined using the TwoPin as a base class and thereby inherit its attributes. Equations that describe the specific physical characteristics of the relation between voltage and current for impedance model are shown below.

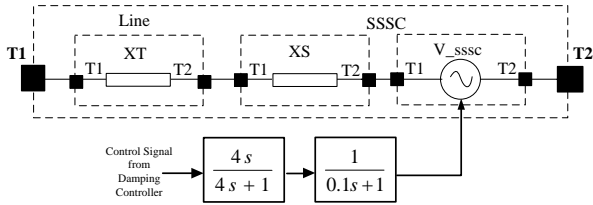


Fig.3: Free body diagram of an SSSC

```

model Impedance "Impedance model"
  extends Base.TwoPin;
  parameter Base.Resistance R=0.0 "Resistance";
  parameter Base.Reactance X=0.1 "Reactance";
equation
  [T1.va - T2.va; T1.vb - T2.vb] = [R, -X; X, R]
    * [T1.ia; T1.ib];
  [T1.ia; T1.ib] + [T2.ia; T2.ib] = [0; 0];
end Impedance

```

The following equations of V_sssc model define the characteristic of series injected voltage which can either lead or lag the line current by 90 degrees. The magnitude of injected voltage is controlled by an external input which can be obtained from transfer function blocks. The source code of the SSSC voltage source is shown below.

```

model V_sssc
  extends ObjectStab.Base.TwoPin;
  Real id, iq;
  Real VS=u1.signal[1];
  Real itheta, vsttheta;

```

```

  Real vsd(start=0), vsq(start=0);
  Modelica.Blocks.Interfaces.InPort u1;
  equation
  when sample(0, 0.005) then
    id = pre(T1.ia);
    iq = pre(T1.ib);
  end when;
  itheta = Modelica.Math.atan2(iq, id);
  vsd = abs(VS)*Modelica.Math.sin(itheta);
  vsq = abs(VS)*Modelica.Math.cos(itheta);
  if VS >= 0 then
    (1 + T2.va) = (1 + T1.va) - vsd;
    T2.vb = T1.vb + vsq;
  vsttheta = Modelica.Math.atan2(vsq, -vsd);
  else
    (1 + T2.va) = (1 + T1.va) + vsd;
    T2.vb = T1.vb - vsq;
    vsttheta = Modelica.Math.atan2(-vsq, vsd);
  end if;
  T1.ia = -T2.ia;
  T1.ib = -T2.ib;
  initial equation
  itheta = 0;
  end V_sssc

```

Having collected components, we can put all objects together to model an SSSC as shown below. It can be noticed that the physical connections in the real circuit correspond to the logical connections in the free body diagram. The complete Modelica code of the sssc model that is given below, has two parts. The first part “model sssc” connects voltage source V_sssc with XS. The second part “model SSSC_and_Line” connects V_sssc and XS in series with a transmission line impedance XT as illustrated in Fig 3.

```

model sssc
  extends ObjectStab.Base.TwoPin;
  parameter ObjectStab.Base.Resistance R=0.0 "Resistance";
  parameter ObjectStab.Base.Reactance X=0.19;
  Modelica.Blocks.Interfaces.InPort u;
  System4Gen.models.V_sssc V_sssc1;
equation
  connect(T1, xs.T1);
  connect (V_sssc1.T2,T2);
  connect (V_sssc1.T1,xs.T2);
  connect(u,V_sssc1.u1);
end sssc;

model SSSC_and_Line
  extends ObjectStab.Base.TwoPin;
  parameter ObjectStab.Base.Resistance R=0.0 "Resistance";
  parameter ObjectStab.Base.Reactance X=0.1 "Reactance";
  System4Gen.models.sssc sssc1;
equation
  connect(XT.T1, T1) ;
  connect(TF1.outPort, TF2.inPort);
  connect(TF2.outPort, Limiter1.inPort);
  connect(TF1.inPort, Dln);
  connect(sssc1.T2, T2);
  connect(sssc1.T1, XT.T2);
  connect(Limiter1.outPort, sssc1.u)
  initial equation
  TF1.y = 0;
  der(TF2.y) = 0;
end SSSC_and_Line;

```

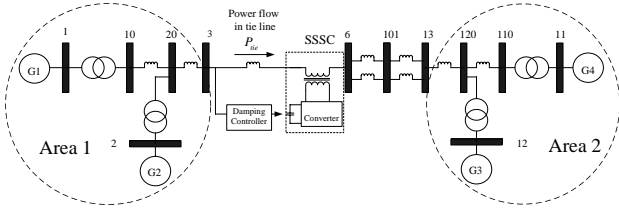


Fig.4: Power system model

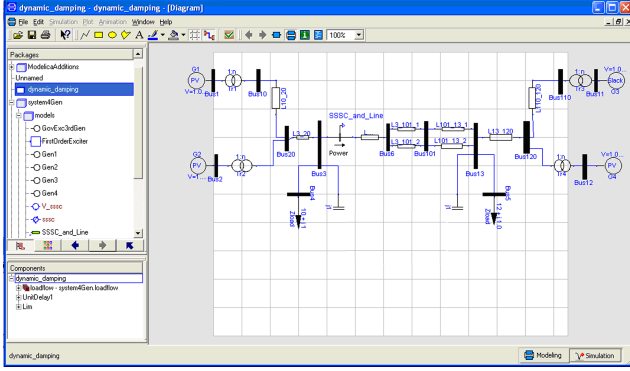


Fig.5: Power system model established in Dymola

4. INCORPORATING OBJECTSTAB AND FUZZY LOGIC TOOLBOX

In this part, the performance of the proposed SSSC model established in ObjectStab is evaluated by simulation study. The interface of ObjectStab with Matlab/Simulink is described. Subsequently, the damping controller design of SSSC using fuzzy logic toolbox [12] is carried out.

To verify the SSSC model, a two-area four generators power system [13] in Fig. 4 is employed. The synchronous generator is represented by the 3rd-order model. Every generator is equipped with an automatic voltage regulator, a power system stabilizer, and a governor. An SSSC is connected in series with a transmission line between buses 3 and 6. The construction of power system modeling and the design procedure of fuzzy logic based-damping controller of SSSC are explained step by step as follows.

Step 1. Construct the single-line diagram of the study system by Dymola program [14] as delineated in Fig. 5. In the subwindow "Packages", create dynamic models of SSSC as explained in Part 3. Perform load flow to obtain the initial values of system variables.

Step 2. Establish the Modelica code as shown in Fig. 6, to define user's created models, i.e. SSSC. Then, declare the location of SSSC, the faulted bus, the fault impedance in the system of Fig. 5. Determine the input and output variables for interfacing Dymola with Simulink. These variables will appear at the input and output terminals of DymolaBlock in the step 3. In addition, the user can define new equations to calculate the interested variables such as the speed difference between generators G1 and G3 (w_{13}

$= G1.w - G3.w$) etc.

Step 3. Connect the power system model installed with SSSC embedded in the DymolaBlock with the damping controller in Simulink as demonstrated in Fig. 7. The DymolaBlock can be found in Simulink's library. The damping controller is modeled by the lead/lag controller with two compensation blocks and a controller gain. In addition, a limiter $-0.2 \leq \Delta U_{SSSC} \leq 0.2$ is equipped at the output terminal of the lead/lag controller. Consequently, the power system with a lead/lag controller of SSSC can be obtained. The control parameters of lead/lag controller are appropriately tuned so that the lead/lag controller provides satisfactory damping effect.

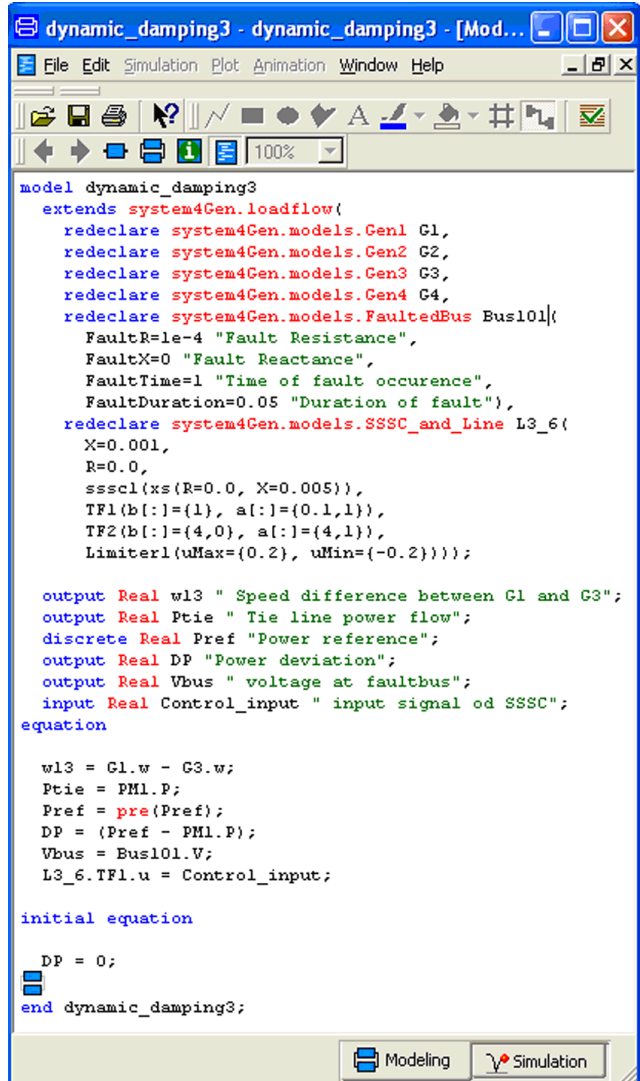


Fig.6: Modelica code for power system model

Step 4. In this step, the fuzzy logic-based PI controller [7,8] of SSSC is designed by the input and output signals observed from the lead/lag controller of SSSC in Fig 7. The structure of the fuzzy logic-based PI controller is depicted in Fig. 8. It consists of a fuzzy controller connected in series with a PI controller with two input signals (ΔP_{tie} and $\int \Delta P_{tie} dt$)

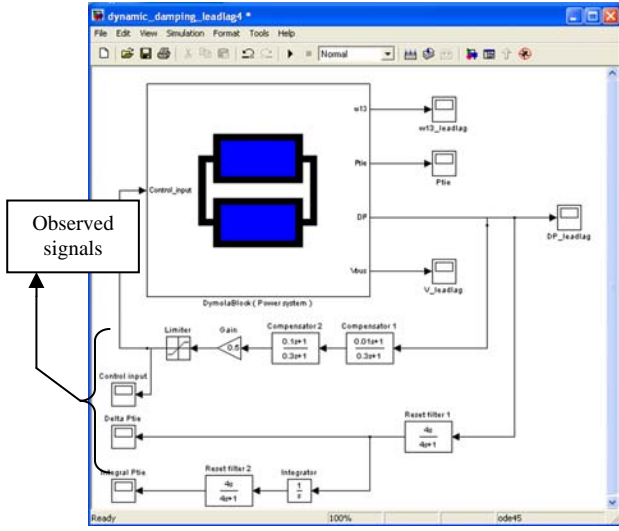


Fig.7: Simulink model of power system with lead/lag

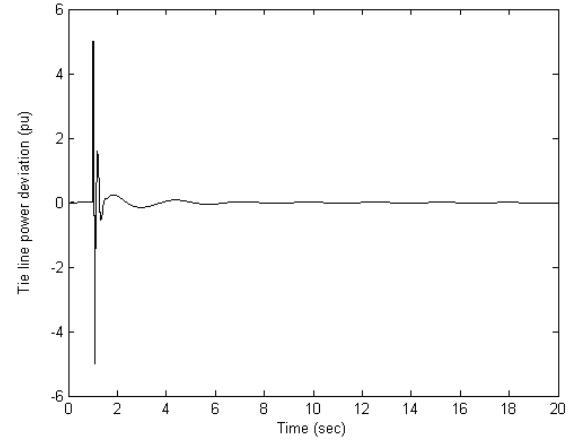


Fig.9: Input signal 1: a tie line power deviation signal

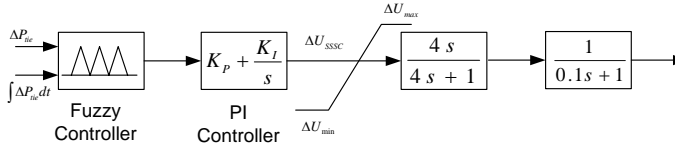


Fig.8: SSSC with a fuzzy logic-based PI controller

and one output signal (ΔU_{SSSC}), where ΔP_{tie} is a tie line power deviation signal, $\int \Delta P_{tie} dt$ is an integration of tie line power deviation signal and ΔU_{SSSC} is an output signal of damping controller. The PI gains are appropriately set at $K_p = 2$ and $K_i = 0.5$. The same limiter $-0.2 \leq \Delta U_{SSSC} \leq 0.2$ is also installed at the output terminal of the fuzzy controller.

In fuzzy control design, important steps are how to obtain the membership functions and control rules.

Typically, the membership functions and the control rules are obtained by expert's experiences where they are synthesized by the choices of the input-output membership functions and the rule bases. This study presents a simple, general and practical approach to extract membership functions and fuzzy logic control rules based on observed signals. Based on simulation results at base case condition, two input signals and one output signal detected from the SSSC with lead/lag controller are shown in Figs.9, 10 and 11, respectively. After obtaining input-output responses from a lead/lag controller, we can determine the universe of discourse for membership functions of each input and output signal roughly. For simplicity, triangle and trapezoidal shaped membership functions have been selected to describe all linguistic variables. Figs. 12 (a)–(c) illustrate the membership functions of two input variables and one output variable. Each membership function consists of two trapezoidal and five triangular memberships. The input and output signals are divided into seven linguistic

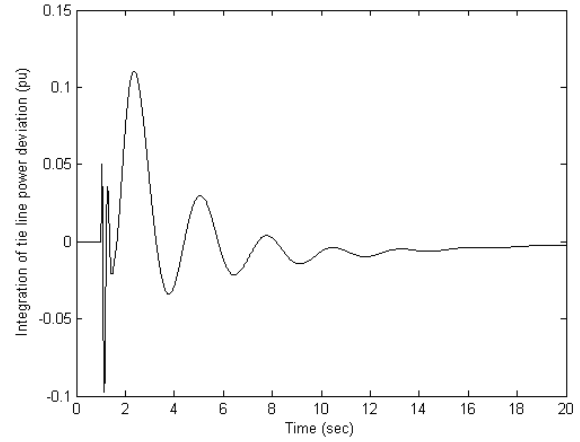


Fig.10: Input signal 2: an integration of tie line power

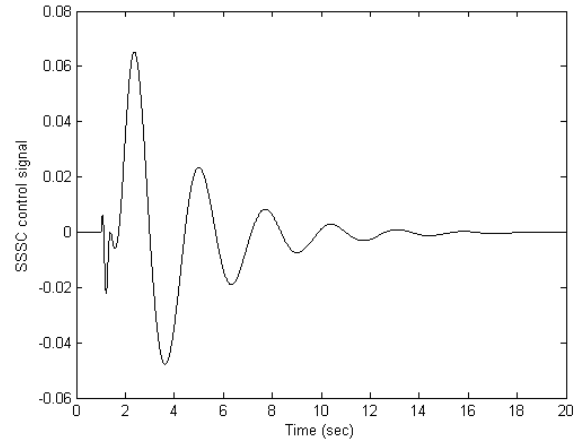


Fig.11: Output signal: an output signal of damping

tic variables using the fuzzy set notations as defined in Fig. 12.

Next, the determination of control rules is appropriately adjusted so that the designed fuzzy controller provides satisfactory damping effects. For the case of two-input, one-output system, the fuzzy rule set can be shown graphically in Table 1 where every cell shows the output membership function of a fuzzy rule with the relationship between inputs 1 and 2. In the fuzzy logic toolbox of Matlab, the rules are built from the *if-then* statement, that is *if input 1 and input 2 then output 1*. For example, consider the third row and fourth column in Table 1, that means, if input 1 is SN and input 2 is Z, then output is SN.

Table 1: Rule Base

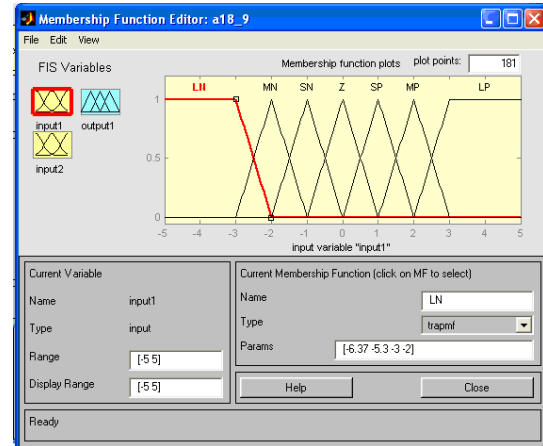
IF2 \ IF1	LN	MN	SN	Z	SP	MP	LP
LN	LN	LN	MN	SN	MN	MN	LN
MN	LN	MN	MN	MN	SN	Z	MN
SN	LN	MN	SN	SN	Z	SP	MP
Z	LN	SN	Z	Z	Z	MP	MP
SP	MN	SN	SN	SP	SP	MP	LP
MP	LN	Z	SP	MP	MP	SP	MP
LP	Z	MP	MP	SP	MP	MP	LP

Step 5. Take the fuzzy controller and the PI controller blocks from Simulink library. Connect them in series with the DymolaBlock as shown in Fig. 13. Consequently, a simulink model of power system with fuzzy controller of SSSC is achieved. Dynamic simulation studies based on this model can be performed.

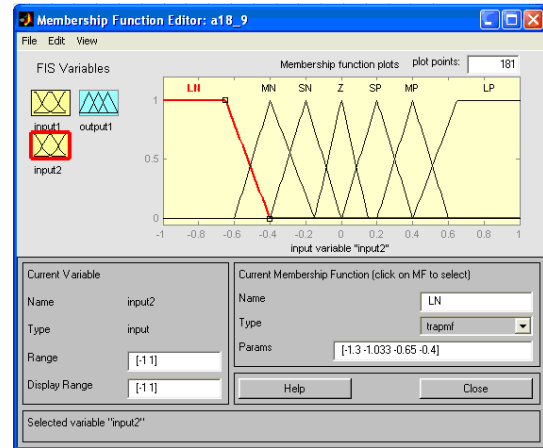
5. SIMULATION RESULTS

The effects of the designed fuzzy logic controller of SSSC are evaluated based on three case studies in Table 2. Note that the fuzzy controller is designed at the based loading condition (case 1). The effectiveness of the SSSC with fuzzy controller is compared to that of the SSSC with lead/lag compensator. Figs. 14-16 show differences of generator speeds between G1 and G3 of case studies 1-3, respectively.

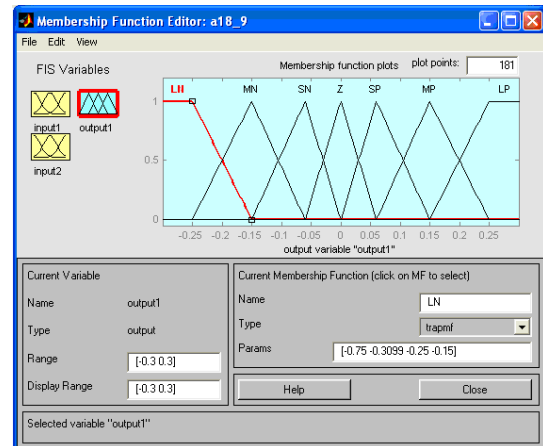
First, at the normal tie-line power flow in case 1, the simulation results of the difference speed between generators 1 and 3 which implies the inter-area oscillation between both areas are shown in Fig. 14. After the three phase fault occurred at bus 101, the generator oscillations are stabilized effectively by SSSC with either lead/lag controller or fuzzy controller. Because



(a) Membership function for input1



(b) Membership function for input2



(c) Membership function for output

LN=Large Negative, MN=Medium Negative, SN=Small Negative, Z=Zero, SP=Small Positive, MP=Medium Positive, LP=Large Positive.

Fig.12: Membership functions

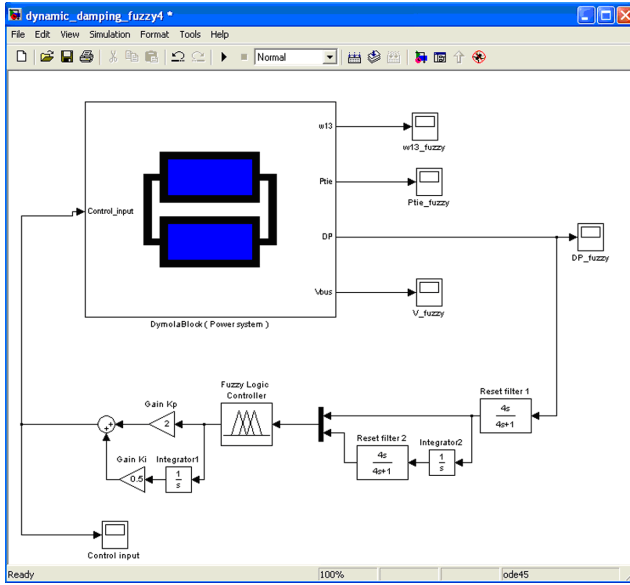


Fig.13: Simulink model

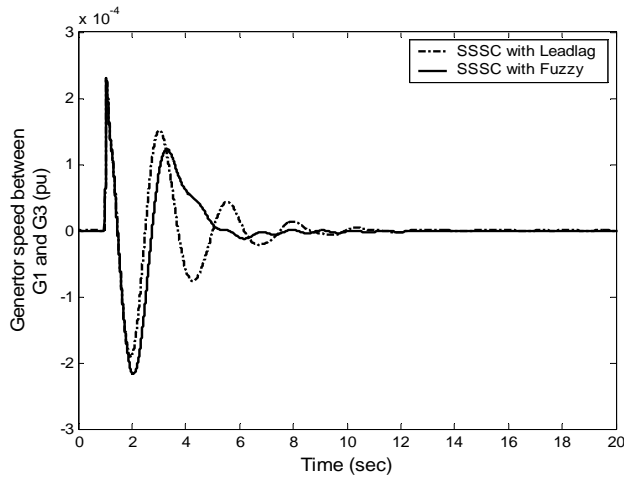


Fig.14: Simulation results of case study 1

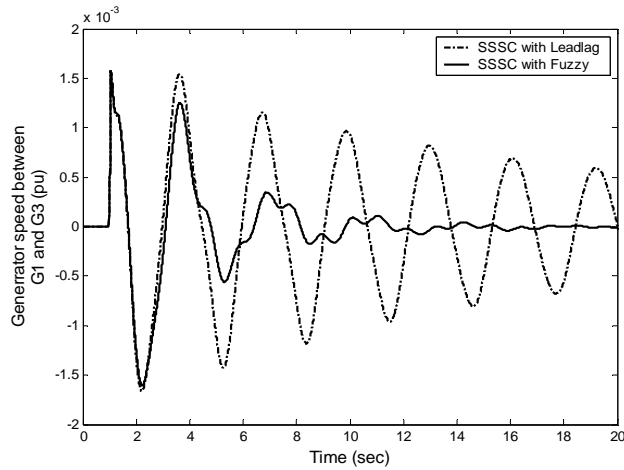


Fig.15: Simulation results of case study 2

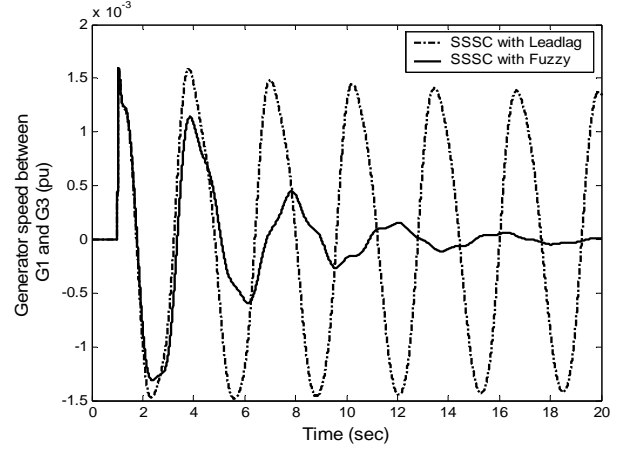


Fig.16: Simulation results of case study 3

Table 2: Case studies

Case study	pu tie line power flow (System base = 100 MW)	Applied disturbances
1	1.28	3 phase fault at Bus101 at $t=1$ sec for 50 ms
2	5.36	3 phase fault at Bus101 at $t=1$ sec for 50 ms
3	3.75	One transmission line between buses 101 and 13 is out of service and 3 phase fault at Bus101 at $t=1$ sec for 50 ms

the SSSC with fuzzy controller is designed based on the observed signals from the lead/lag controller, the performances of both controllers are almost identical.

For the heavy tie line power flow in case 2 and under the same three phase fault as in case 1, the SSSC with lead/lag controller can not provide satisfactory damping effects as depicted in Fig. 15. Inter-area oscillations slowly reach steady-state values. On the other hand, the SSSC with fuzzy controller can effectively tolerate the heavy tie-line power condition and a three-phase fault. The fuzzy controller is able to damp system oscillations effectively. System oscillations are well stabilized and gradually subside.

Finally in case 3, one transmission line is out of

service so the line impedance becomes larger while the tie line power flow is at medium level. After the 3 phase fault occurred, the SSSC with lead/lag controller can not stabilize system oscillations as demonstrated in Fig. 16. It loses the damping effects. The power system is oscillatory instability due to insufficient damping torque. In contrast, the SSSC with fuzzy controller is capable of stabilizing the power system. System responses are gradually damped out. System stability is absolutely maintained. Based on simulation results, the robustness of the SSSC with fuzzy controller against system uncertainties is significantly superior to that of the SSSC with lead/lag controller.

6. CONCLUSION

In this paper, the incorporated use of a fuzzy logic toolbox in Matlab/Simulink and an ObjectStab library has been applied to design of a fuzzy logic-based PI damping controller of SSSC. The physical model of SSSC is proposed and created by ObjectStab. The fuzzy logic control design of SSSC for power system stabilization is demonstrated. The simple method to extract membership functions and fuzzy logic control rules based on observed signals is presented. A number of simulation studies have been performed with the SSSC with fuzzy controller to test the effectiveness and robustness. Simulation results show that the SSSC with fuzzy controller is effective and robust over a wide range of operating conditions. The example given in this paper opens up a new idea of the use of a combined ObjectStab library and fuzzy control toolbox, which allows the users to design more sophisticated damping controllers.

References

- [1] Modelica and the Modelica Association Web site: <http://www.modelica.org/>
- [2] M. M. Tiller, Introduction to Physical Modeling with Modelica, Kluwer Academic Publisher, 2001.
- [3] H. Elmqvist, S. E. Mattsson and M. Otter, "Modelica-a language for physical system modeling, visualization and interaction," Proc. of the 1999 IEEE International Symposium on Computer Aided Control System Design, 1999, pp. 630-639.
- [4] M. Larsson, "ObjectStab-an educational tool for power system stability studies," IEEE Trans. on Power Systems, Vol. 19, Issue 1, 2004, pp. 56-63.
- [5] M. Larsson, "ObjectStab-a modelica library for power system stability studies," Proc. of the 2000 Modelica Workshop, 2000, pp. 13-22.
- [6] I. R. Navarro, M. Larsson and G. Olsson, "Object-oriented modeling and simulation of power systems using Modelica," Proc. of the 2000 Power Engineering Society Winter Meeting, Vol. 1, 2000, pp. 790 – 795.
- [7] W. Mielczarski, Fuzzy logic techniques in power systems, Studies in fuzziness and soft computing, Vol. 10, Physica-Verlag, 1998.
- [8] M. E. El-Hawary, Electric power applications of fuzzy systems, IEEE press series on power engineering, Wiley-IEEE Press, 1998.
- [9] L. Gyugyi, C.D. Schauder and K.K. Sen, "Static synchronous series compensator: a solid-State approach to the series compensation of transmission line," IEEE Trans. Power Delivery, Vol. 12, No. 1, 1997, 406-417.
- [10] N. G. Hingorani and L. Gyugyi, Understanding FACTS, Concepts and Technology of Flexible AC Transmission Systems, IEEE Press, 2000.
- [11] K. Hongesombut, Y. Mitani, Y. Tada, T. Takazawa and T. Shishido, "Object-Oriented Modeling for Advanced Power System Simulations," Proc. of IEEE St. Petersburg Power Tech Conference 2005.
- [12] The MathWorks, MATLAB Fuzzy Logic Toolbox, User's Guide.
- [13] P. Kundur, Power System Stability and Control, McGrawHill, 1994.
- [14] H. Elmqvist, D. Bruck, and M. Otter, Dymola, User Manual, Dynasim 2001.



Boonrat Somritvanitcha received the B.Eng degree in Electrical Engineering from Sirindhorn International Institute of Technology, Thammasat University, Bangkok, Thailand in 2006. Her research interests are in the areas of power system stability, dynamics & control.



Issarachai Ngamroo received the B.Eng degree in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand in 1992. He received M.Eng and Ph.D. degrees in Electrical Engineering from Osaka University, Osaka, Japan, in 1997 and 2000, respectively. His research interests are in the areas of power system stability, dynamics & control.



Komsan Hongesombut received the B.Eng (First Class Honors) and M.Eng degrees in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand in 1997 and 1999, respectively and Ph.D. in Electrical Engineering from Osaka University, Osaka, Japan in 2003. Currently, he is a researcher at Tokyo Electric Power Company, Tokyo, Japan. His research interests are in the areas of power system stability, dynamics & control.