A Generalized Neuron Based Adaptive Power System Stabilizer for Multimachine Environment

D. K. Chaturvedi, Dayalbagh Educational Institute
O. P. Malik

Available at: https://works.bepress.com/dk_chaturvedi/14/
Generalized Neuron-Based Adaptive PSS for Multimachine Environment

D. K. Chaturvedi and O. P. Malik, Fellow, IEEE

Abstract—Artificial neural networks can be used as intelligent controllers to control nonlinear, dynamic systems through learning, which can easily accommodate the nonlinearities and time dependencies. Taking advantage of the characteristics of a generalized neuron (GN), that requires much smaller training data and shorter training time, a GN-based adaptive power system stabilizer (GNAPSS) is proposed. It consists of a GN as an identifier, which predicts the plant dynamics one step ahead, and a GN as a controller to damp low frequency oscillations. Results of studies with a GN-based PSS on a five-machine power system show that it can provide good damping of both local and inter-area modes of oscillations over a wide operating range and significantly improve the dynamic performance of the system.

Index Terms—Adaptive PSS, generalized neuron controller, neural network, on-line training.

I. INTRODUCTION

A SUPPLEMENTARY control signal in the excitation system and/or the governor system of a generating unit can be used to provide extra damping for the system and thus improve the dynamic performance [1]. Power system stabilizers (PSSs) aid in maintaining power system stability and improve dynamic performance by providing a supplementary signal to the excitation system. This is an easy, economical and flexible way to improve power system stability in interconnected ac power systems. Over the past few decades, PSSs have been extensively studied and successfully used in the industry.

The commonly used PSS (CPSS) was first proposed in the 1950s based on a linear model of the power system at some operating point to damp the low-frequency oscillations in the system [2], [3].

The most commonly used PSS, referred to as the CPSS, is a fixed-structure, fixed-parameter device. It is designed using the classical linear control theory to enhance the stability by damping the unstable oscillatory modes of oscillations. These modes of oscillation are characterized by low mechanical natural frequencies in the range of 0.3–2.0 Hz. CPSS uses a lead/lag compensation network to compensate for the phase shift caused by the low frequency oscillation of the system. By appropriately tuning the parameters of a lead/lag compensation network, it is possible to make a system have desired damping characteristics.

Power systems are nonlinear systems. Their configuration and parameters change with time. The linearized system models used to design the conventional PSSs are valid only at the operating point that is used to linearize the system. As fixed parameter controller, CPSS cannot provide optimal performance under wide operating conditions.

Extensive research has been done in the past to design PSSs with improved performance [1]–[9]. One of the good solutions is an adaptive controller. This type of stabilizer can adjust its parameters on-line according to the operating conditions. Intensive studies have shown that the adaptive stabilizer can not only provide good damping over a wide operating range, but more importantly, does not raise any coordination problem among stabilizers. More recently, artificial neural networks (ANNs) and a fuzzy set theoretic approach have been proposed for power system stabilization problems [8]–[15]. Both techniques have their own advantages and disadvantages. An integration of these approaches can give improved results.

The common neuron model has been modified to obtain a generalized neuron (GN) model using fuzzy compensatory operators as aggregation operators to overcome the problems such as large number of neurons and layers required for complex function approximation, which not only affect the training time but also the fault tolerant capabilities of the ANN [16]. Application of this GN as an adaptive PSS (APSS) in the multimachine environment is described in this paper.

II. ADAPTIVE GN-BASED POWER SYSTEM STABILIZER

An APSS estimates some uncertainty within the system, then automatically designs a controller for the estimated plant uncertainty. In this way the control system uses information gathered on-line to determine exactly the current state of the plant so that good control can be achieved [5], [17]–[21]. Most APSSs use self-tuning adaptive control scheme, as it is one of the most effective adaptive control schemes. The structure of a self-tuning indirect APSS includes: an on-line parameter identifier (approximator) and a control strategy (system controller).

At each sampling period, a mathematical model is identified on-line to track the dynamic behavior of the plant. Assuming the identified model represents the system exactly, the control signal can be calculated based on on-line identified parameters. There are several control strategies that can be used in the self-tuning adaptive control, such as minimum variance, generalized minimum variance, pole assignment, pole shift [5], [17]–[21], ANN based [8], [10], [13], [14], [22], [23], and fuzzy system based [11], [12], [15].

D. K. Chaturvedi was with the University of Calgary, Calgary, AB T2N 1N4, Canada. He is now with Faculty of Engineering, Dayalbagh Educational Institute, Dayalbagh, Agra 282005, India (e-mail: dke_foe@rediffmail.com).
O. P. Malik is with Department of Electrical and Computer Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: malik@enelec.ucalgary.ca).
Digital Object Identifier 10.1109/TPWRS.2004.840410
The ANN-based APSS gives quicker control action due to parallel processing once it is properly trained. The main problems with ANN-based PSS are the selection of ANN structure and neuron type, large training data and large training time required. To overcome this problem, a GN is developed as given in the Appendix. Taking the benefits of GN, an APSS is developed consisting of GN identifier and GN controller.

The GN model is much less complex compared to a three-layered ANN proposed earlier for PSS [8], [22], [23]. These ANNs were 20-20-1, 35-1, and 30-10-1. Taking, for illustration purposes, an ANN with one hidden layer and much smaller number of neurons, a comparison of the structural complexity associated with ANN and GN model is given in Table I [24].

It is seen from Table I that the number of interconnections for a GN model is much smaller in comparison to an ANN. Hence, the number of unknown weights is reduced drastically, which ultimately reduces the training time and training data required.

**A. GN Identifier**

The problem of identification includes setting up a suitably parameterized identification model and adjusting the parameters of the model to optimize a performance function based on the error between the plant and the identified model outputs.

A schematic diagram of the GN-based plant identifier using forward modeling is shown in Fig. 1. A GN identifier is placed in parallel with the system and has the following inputs:

\[ X_i(t) = [\omega \text{-vector}, u \text{-vector}] \]  

where

\[ \omega \text{-vector} = [\omega(t), \omega(t-T), \omega(t-2T), \omega(t-3T)] \]
\[ u \text{-vector} = [u(t-T), u(t-2T), u(t-3T)] \]

where \( T \) is the sampling period, \( \omega \) is angular speed deviation from synchronous speed in rad/s, and \( u \) is the controller output.

The dynamics of the change in angular speed of the synchronous generator can be viewed as a nonlinear mapping with the inputs mentioned in (1) and could be mathematically written as

\[ \omega(t + T) = f_i(X_i(t)) \]  

where \( f_i \) is a nonlinear function.

Therefore, the GN-identifier for the plant can be represented by a nonlinear function \( F_i \)

\[ \omega_i(t + T) = F_i(X_i(t), W_i(t)) \]

where \( W_i(t) \) is the matrix of GN identifier weights at time instant \( t \).

**B. Training of GN-Identifier**

Training of an ANN is a major exercise, because it depends on various factors such as the availability of sufficient and accurate training data, suitable training algorithm, number of neurons in the ANN, number of ANN layers and so on. The GN identifier with only one neuron is able to cope with the problem complexity, as the selection of the number of neurons and layers is not required. Training of the proposed GN identifier has two steps: off-line training and on-line update using error back propagation.

In off-line training, the GN identifier is trained for a wide range of operating conditions i.e., output ranging from 0.1 to 1.0 p.u. and the power factor ranging from 0.7 lag to 0.8 lead. Similarly, a variety of disturbances is also included in the training, like change in reference voltage, input torque variation, transmission line outage, and three-phase fault on one circuit of the double circuit transmission line. Off-line training data for the GN identifier has been acquired from the system controlled by the CPSS although any suitable controller can be used. The training data is normalized in the range 0.1 to 0.9, so that if the input variables go beyond this range, the controller can still handle it.

The error between the system and the GN identifier output at a unit delay is used as the GN identifier training signal. The error square is used as the performance index

\[ J_i(t) = \frac{1}{2} (\omega_i(t) - \omega(t))^2. \]  

The weights of the GN identifier are updated as

\[ W_i(t) = W_i(t - T) + \Delta W_i(t) \]

where \( \Delta W_i(t) \), the change in weight depending on the instantaneous gradient, is calculated by

\[ \Delta W_i(t) = -\eta_i J_i(t) \frac{\partial J_i(t)}{\partial W_i(t)} + \alpha \Delta W_i(t - T) \]

where

\( \eta_i \) learning rate for GN identifier;
\( \alpha_i \) momentum factor for the GN identifier.

The off-line training is performed with 0.1 learning rate and 0.4 momentum factor. If large learning rate is chosen, then due
is the matrix of neural controller weights at time instant \( t \). The matrix is de-normalized to get the actual control action and then sent to the plant and the GN-identifier simultaneously.

\[
\omega(t + T) = f_i(X_i(t)) \approx \omega_i(t + T) = F_i(X_i(t), W_i(t))
\]

the proposed GN-identifier will be connected to the power system for on-line update of weights. Two illustrative results for the on-line performance of the GN identifier are shown in Figs. 2 and 3, with a sampling interval of 30 ms.

C. GN Controller

A schematic diagram of the GN controller is shown in Fig. 4. The last four values of the angular speed of the synchronous machine, sensed at fixed time intervals (30 ms), are used as input to the GN controller. Besides the angular speed, past three control actions are also given to the GN controller as inputs. These inputs are normalized in the range 0.1–0.9. The output of the GN-controller is the control signal \( u(t) \)

\[
u(t) = F_c(X_i(t), W_c(t)) \quad (6)
\]

where \( W_c(t) \) is the matrix of neural controller weights at time instant \( t \). The \( u(t) \) is de-normalized to get the actual control action and then sent to the plant and the GN-identifier simultaneously.

D. Training of GN Controller

Training of the proposed GN controller also has two steps—off-line training and on-line update. In off-line training, the GN controller is trained for a wide range of operating conditions and a variety of disturbances. Data for off-line training of the GN controller has been acquired from the system controlled by the CPSS, which is tuned for each operating condition.

The performance index of the neurocontroller is

\[
J_c(t) = \frac{1}{2} [\omega_i(t + T) - \omega_d(t + T)]^2 \quad (7)
\]

where \( \omega_d(t + T) \) is the desired plant output at time instant \( t + T \); in this study, it is set to be zero.

The weights of the GN controller are updated as

\[
W_c(t) = W_c(t - T) + \Delta W_c(t) \quad (8)
\]
where $\Delta W_c(t)$, the change in weight depending on the instantaneous gradient, is calculated by

$$\Delta W_c(t) = -\eta_c \omega_i(t+T) \frac{\partial L_c(t)}{\partial \omega_i(t)} \frac{\partial \omega_i(t)}{\partial W_c(t)} + \alpha_c \Delta W_c(t-T) \quad (9)$$

where

- $\eta_c$ learning rate for the GN controller;
- $\alpha_c$ momentum factor for the GN controller.

Initially, the proposed controller is trained off-line. Off-line training is started with small random weights ($\pm 0.01$) and then updated with learning rate ($\eta_c = 0.1$) and momentum factor ($\alpha_c = 0.4$). Once the off-line training is over, it is connected to the power system. To further improve the controller performance and handle the unforeseen situations, the proposed GN-controller is trained on-line. In on-line training of the GN-controller, the expected error is calculated from the one-step ahead predicted output of the GN-identifier. The expected error is then used to update the weights on-line. Parameters of the GN identifier and controller are adjusted every sampling period. The value of the learning rate and momentum factor is taken as smaller, i.e., 0.01 and 0.05. This allows the controller to track the dynamic variations of the power system and provide the best control action.

### III. Simulation Results

#### A. Single-Machine Infinite Bus System

Performance of the proposed GN-based APSS was first investigated on a synchronous generator connected to a constant voltage bus through two parallel transmission lines. Parameters of the machine, exciter, AVR, and the CPSS are given in the Appendix. The generator was simulated by a seventh-order nonlinear model.

1) **CPSS Parameter Tuning**: With the generator operating at $P = 0.9$ p.u. and $Q = 0.4$ p.u. lag, a 100 ms three-phase to ground fault was applied at $0.5$ s at the generator bus. The CPSS was carefully tuned to yield the best performance under these conditions and its parameters were kept fixed for the comparative studies.

2) **Performance of GN APSS (GNAPSS)**: Illustrative results for the GNAPSS and the CPSS for a 100 ms three-phase to ground fault at generator bus, line removal, and step change in torque disturbance under different operating conditions are shown in Fig. 5.
B. Multimachine System

A five-machine power system without infinite bus, Fig. 6, that exhibits multimode oscillations, is used to study the performance of the previously trained GNAPSS. In this system, generators #1, #2, and #4 are much larger than generators #3 and #5. All five generators are equipped with governors, AVRs and exciters. This system can be viewed as a two area system connected through a tie line between buses #6 and #7. Generators #1 and #4 form one area and generators #2, #3, and #5 form another area. Parameters of all generators, transmission lines, loads and operating conditions are given in the Appendix. Under normal operating conditions, each area serves its local load and is almost fully loaded with a small load flow over the tie line.

1) Performance of GNAPSS With Torque Disturbance:
   a) Simulation Studies With GNAPSS Installed on One Generator: The GNAPSS is trained for a single machine infinite bus system and the same parameters (weights) are used for GNAPSS with multimachine system. The proposed GNAPSS is installed only on generator #3 and CPSSs with the following transfer function are installed on generators #1 and #2:

\[
   u_{\text{FSS}} = K_s \frac{sT_5}{1 - sT_5} \left(1 + sT_1\right) \left(1 + sT_3\right) \Delta P_e(s). \tag{10}
\]

Parameters of the CPSS are tuned carefully so that the CPSS has almost the same performance as the GNAPSS. The following parameters are set for the CPSS for all studies in the multimachine environment:

\[
   K_s = 0.2, T_1 = T_3 = 0.07, T_2 = T_4 = 0.03, T_5 = 2.5.
\]

Speed deviation of generator #3 is sampled at a fixed time interval of 30 ms. The system response is shown in Fig. 7 for the operating conditions given in the Appendix. Each part of the figure shows the difference in speed between two generators.

   b) GNAPSS Installed on Three Generators: In this test, GNAPSSs are installed on generators #1, #2, and #3. A 30\% step decrease in mechanical input torque reference of generator #3 is applied at 1 s and it returns to its original level at 10 s. The simulation results of only GNAPSSs and only CPSSs applied at generators #1, #2, and #3 are shown in Fig. 8. It is seen from the results that both modes of oscillations are damped out very effectively.

2) Three-Phase to Ground Fault: In this test, a three-phase to ground fault is applied at the middle of one transmission line between buses #3 and #6 at 1 s and the faulty line is removed 100 ms later. At 10 s, the faulty line is restored successfully. The GNAPSSs are installed on all five generators. The system responses are shown in Fig. 9. The results with CPSSs installed on the same generators are also shown in the same figures. From the system responses, it can be concluded that although the CPSS...
can damp the oscillations caused by such a large disturbance; the proposed GNAPSS has much better performance.

3) Coordination Between GNAPSS and CPSS: The advanced PSSs would not replace all CPSSs being operated in the system at the same time. Therefore, the effect of the GNAPSS and CPSSs working together needs to be investigated. In this test, the proposed GNAPSS is installed on generators #1 and #3 and CPSSs on generators #2, #4, and #5. The operating conditions are the same as given in the Appendix. A 0.3-p.u. step decrease in the mechanical input torque reference of generator #3 is applied at 1 s and returns to its original level at 10 s. The system responses are shown in Fig. 10. The results demonstrate that the two types of PSSs can work cooperatively to damp out the oscillations in the system. The proposed GNAPSS input

Fig. 8. System response for change in $T_{ref}$ with only GNAPSS and only CPSS installed on G1, G2, and G3.

Fig. 9. System response with only GNAPSS and only CPSS installed on all five machines for three-phase to ground fault.
signals are local signals. The GNAPSS coordinates itself with the other PSSs based on the system behavior at the generator terminals.

IV. CONCLUSION

A GN can incorporate the nonlinearities involved in the system. It uses only one neuron and is trained using back-propagation learning algorithm. Because it has a much smaller number of weights than the common multilayer feedforward ANN, the training data required is drastically reduced. Training time is also significantly reduced because the number of weights to be determined is much smaller than for an ANN.

A GN has been employed to perform the function of a PSS to improve the stability and dynamic performance of the single machine infinite bus as well as a multimachine power system. Simulation studies described in the paper show that the performance of the GNAPSS provides very good performance. The effectiveness of the GNAPSS to damp multimode oscillations in a five-machine power system provides satisfactory results and it can cooperate with other GNAPSSs or CPSSs.

APPENDIX

GENERALIZED NEURON MODEL

The sigmoidal thresholding function and an ordinary summation or product as aggregation functions in the existing neuron models [25], [26] fail to cope with the nonlinearities involved in real life problems. To deal with these, the proposed model has both sigmoidal and Gaussian functions with weight sharing. The GN model has flexibility at both the aggregation and threshold function level to cope with the nonlinearity involved in the type of applications dealt with, as shown in Fig. 11. The neuron has both $\Sigma$ and $\pi$ aggregation functions. The $\Sigma_1$ aggregation function has been used with the sigmoidal characteristic function ($f_1$) while the $\pi$ aggregation function has been used with the Gaussian function ($f_2$) as a characteristic function.

The output of the $\Sigma_1$ part with sigmoidal characteristic function of the GN is

$$O_{\Sigma} = f_1(s_{\text{net}}) = \frac{1}{1 + e^{-\lambda_{s} s_{\text{net}}}}$$

where

$$s_{\text{net}} = \sum W_{\Sigma} X_i + X_{o\Sigma}$$

and $\lambda_s$ is the gain scale factor for the $\Sigma$ part.

Step-2 The output of the $\pi$ part with Gaussian characteristic function of the GN is

$$O_{\Pi} = f_2(p_{\text{net}}) = e^{-\lambda_p p_{\text{net}}^2}$$

where

$$p_{\text{net}} = \prod W_{\Pi} X_i * X_{o\Pi}$$

and $\lambda_p$ is the gain scale factor for $\Pi$ part.
The final output of the neuron is a function of the two outputs $O_{\Sigma}$ and $O_{\Pi}$, with the weights $W$ and $(1 - W)$, respectively

$$O_{pk} = O_{\Pi} \cdot (1 - W) + O_{\Sigma} \cdot W.$$  

(13)

The neuron model described above is known as the summation type compensatory neuron model, since the outputs of the sigmoidal and Gaussian functions are summed up. Similarly, the product type compensatory neuron models may also be developed. It is found that in most of the applications, the summation-type compensatory neuron models work well [27] and is the one used for the development of the adaptive GN-based PSS.

In this paper, summation and product are used at the aggregation level for simplification, but one can take other fuzzy aggregation operators such as max, min, or compensatory operators, too. Similarly, the thresholding functions are only sigmoidal and Gaussian function for the proposed GN, but other functions like straight line, sine, cosine, etc. can also be used. The weighting factor may be associated with each aggregation function and thresholding function. During training, these weights change and decide the best functions for the GN. The learning algorithm of GN model is given in [24].

V. SYSTEM MODEL AND ITS PARAMETERS

A. Single-Machine Infinite Bus System

The generating unit is modeled by seven first-order nonlinear differential equations. Parameters for generator, Exciter, AVR, governor, CPSS, and transmission line parameters used in the simulation studies of the single machine infinite bus system are given in [8].

B. Multimachine Power System

1) Generator Parameters in p.u. on a 100-MVA Base.

For small generators #3 and #5:

$$x_{d} = 1.026, x_{q} = 0.6580, x_{d}' = 0.3390$$
$$x_{d}'' = 0.2690, x_{q}'' = 0.3350, H = 2.$$

For big generators #1, #2, and #4:

$$x_{d} = 0.1026, x_{q} = 0.0658, x_{d}' = 0.0339$$
$$x_{d}'' = 0.0269, x_{q}'' = 0.0335, H = 10.$$

Time constant for all generators

$$T_{do} = 5.67, T_{do}'' = 0.614, T_{qo}'' = 0.723.$$

2) A simplified IEEE standard type ST1A AVR and exciter model is shown in Fig. 12 and its parameters are as follows:

$$R_{C} = 0.0, \quad X_{C} = 0.0, \quad T_{f} = 0.04, \quad K_{d} = 190.0$$
$$T_{f} = 1.0, \quad K_{f} = 0.05, \quad K_{e} = 0.08, \quad T_{a} = 0.0$$
$$T_{c} = 1.0, \quad T_{c1} = 0.0, \quad T_{b} = 10.0, \quad T_{c3} = 0.0, \quad V_{oel} = 999.0.$$  

All time constants are in seconds.

Fig. 12. Schematic diagram of AVR and exciter model.

3) Governor Model:

$$g = \left[ a + \frac{b}{1 + sT_{g}} \right]$$

where $T_{g} = 0.25, a = -0.00133, b = -0.170$ for generator #1, #2 and #4.

$T_{g} = 0.25, a = -0.00015, b = -0.0150$ for generator #3 and #5.

4) Operating conditions:

<table>
<thead>
<tr>
<th>$G$</th>
<th>$G_{1}$</th>
<th>$G_{2}$</th>
<th>$G_{3}$</th>
<th>$G_{4}$</th>
<th>$G_{5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{pu}$</td>
<td>5.1076</td>
<td>8.5835</td>
<td>0.8055</td>
<td>8.5670</td>
<td>0.8501</td>
</tr>
<tr>
<td>$Q_{pu}$</td>
<td>6.8019</td>
<td>4.3836</td>
<td>0.4353</td>
<td>4.6686</td>
<td>0.2264</td>
</tr>
<tr>
<td>$V_{pu}$</td>
<td>1.0750</td>
<td>1.0500</td>
<td>1.0250</td>
<td>1.0750</td>
<td>1.0250</td>
</tr>
<tr>
<td>$\delta_{m}$</td>
<td>0.0</td>
<td>0.3167</td>
<td>0.2975</td>
<td>0.1174</td>
<td>0.3015</td>
</tr>
</tbody>
</table>

5) Load admittance (p.u.):

$$L_{1} = 7.5 - j5.0, L_{2} = 8.5 - j5.0, L_{3} = 7.0 - j4.5.$$

REFERENCES


