6. References

[1] J.S. Czuba, L.N. Hannet and J.R. Willis, "Implementation of power system stabilizer at Ludington pumped storage planr", IEEE Trans. PWRS, Vol. 1, No. 1, PP. 121-128, 1986.

[2] P. Kundor, D.C. Lee and H.M. Zein El-Din, "Power system stabilizer for thermal units: analytical techniques and on-site validation", IEEE Trans. PAS, Vol. 100, PP. 81-95, 1981.

[3] E.V. Larsen and D.A. Swann, "Applying Power System Stabilizers", IEEE Trans. PAS, Vol. 100, PP. 3017-3046, 1981.

[4] P.M. Anderson and A.A. Fouad, Power System Control and Stability, Iowa State University Press, 1977.

[5] F.P. deMello and C. Concordia, "Concepts of Synchronous Machine Stability as Affected by Excitation Control", IEEE Trans. PAS, Vol. 88, PP. 316-329, 1969.

[6] Y.Y. Hsu and C.Y. Hsu, "Design of a Proportional- Integral Power System Stabilizer", IEEE Trans. PWRS, Vol. 1, No. 2, PP. 46-53, 1986.

[7] A. Chandra, O.P. Malik and G.S. Hope, "A Self Tuning Controller for the Control of Multi Machine Power Systems", IEEE Trans. PWRS, Vol. 3, No. 3, PP. 1065-1071, 1988.

[8] A. Ghosh, G. Ledwich, O.P. Malik and G.S. Hope, " Power System Stabilizer Based on Adaptive Control Techniques", IEEE Trans. PAS, Vol. 103, PP. 1983-1986, 1984.

[9] W. Eric Norum and K.E. Bollinger, "Lab and Filed Tests of a Self-Tuning Power System Stabilizer", Paper No. 92 SM 477-0 EC, Presented at the IEEE Power Engineering Society, 1992 Summer Meeting, 1992.

[10] Y.Y. Hsu and k.L. Liou, "Design of Self-Tuning PID Power System Stabilizers for Synchronous Generators, " IEEE Trans. EC, Vol. 2, PP. 343-348, 1987.

[11] G.P. Chen, O.P. Malik, G.S. Hope, Y.H. Qin and G.Y. Xu, "An Adaptive Power System Stabilizer Based on the self-Optimizing Pole Shifting Control Strategy", IEEE Trans. EC, Vol. 8, No.4, 1993.

[12] N.C. Pahalawaththa, G.S. Hope and O.P. Malik, "Multivariable Self-Tuning Power System Stabilizer Simulation and Implementation Studies", IEEE Trans. EC, Vol. 6, No. 2, 1991.

[13] Y.Y. Hsu and C.R. Chen, "Tuning of Power System Stabilizers Using an Artificial Neural Network", IEEE Trans. EC, Vol. 6, No. 4, 1991.

[14] D.E. Rumelhurt, G.E. Hinton and R.J. Williams, "Learning Internal Representations by Error Propagation" in D.E. Rumelhurt and J.L. McClelland, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations, MIT Press, 1986.

[15] R.P. Lippmann, " An Introduction to Computing with Neural Nets", IEEE ASSP Mag., PP. 4-22, April 1987.

[16] D.R. Hose and B.G. Horne, " Progress in Supervised Neural Networks", IEEE Signal Processing Mag., PP. 8-39, Jan. 1993.

[17] B. Widrow and M.A. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline, and "Backpropagation", IEEE Proc., Vol. 78, No. 9., Sep. 1990.

[18] P.J. Werbos, "Backpropagation Through Time: what it Does and How to Do it" IEEE Proc., Vol. 78, No. 10, Oct. 1990.

[19] K.S. Narendra, and K. Parthasarathy, "Identification and Control of Dynamical Systems using Neural Networks", IEEE Trans. NN, Vol. 1., No. 1, PP.4-27, 1990.

[20] D.H. Nguyen, and B. Widrow, "Neural Networks for Self-Learning Control Systems," IEEE CS Mag., PP. 18-23. Apr. 1990.

[21] H. Hagan and M.B. Menhaj, "Training Feed Forwark Networks with Marquardt Algorithm", IEEE Trans. NN, Vol. 5, No. 6, 1994.

[22] M.B. menhaj and J. Shahrouzi, "Short Term Load Forecasting Using Layered Neural Nets With Marquardt Algorithm", Submitted to ICEE95, Iran-Tehran, 1995.

[23] IEEE Committee Report, "Excitation System Models for Power System Stability Studies", IEEE Trans. PAS, Vol. 100, No. 2, PP. 494-509, 1981.

[24] IEEE Committee Report, "Dynamic Models for Steam and Hydro Turbines in Power System Studies", IEEE Trans. PAS, Vol. 92, No. 6, PP. 1904-1915, 1973.

In the next stage, we use a MLP to identify the TPIPSS including both the PIPSS and the ANN in Figure 5 as a whole dynamic system. The training data are produced by running many simulations on the complete plant with different input disturbances such as changing voltage and power references and various faults on the system. From the simulation results, the $\Delta\omega$, ΔV_t, and U_{PSS} signals and some of their past time instances are extracted into separate data files. The neural network is then trained off-line using these data. After many trials and errors we finally came up with a structure of NN(8-10-1) as a Neuro-PSS which shows an excellent performance characteristics. The input variables to this NPSS constitute a 8x1 vector as:

$$\underline{P} = [\Delta\omega(k), \Delta\omega(k-1), \Delta\omega(k-2), \Delta\omega(k-3), \Delta V_t(k), \Delta V_t(k-1), \Delta V_t(k-2), \Delta V_t(k-3)]^T \qquad (14)$$

and the output is U_{PSS} (K).

Figures 7 through 9 illustrate the dynamic response of the system equipped with this NPSS when subjected to different disturbances. For comparison, the responses of the system with and without the TPIPSS are also shown in those Figures.

These Figures show that our proposed Neuro-PSS is capable of stabilizing the system the same way the TPIPSS can. In some cases (see Figures 7 and 8) the NPSS shows even better performance than TPIPSS.

5. Conclusion

In this paper a Neuro-PSS using a multilayer feedforward neural network with Marquardt training algorithm has been used to control and to stabilize a synchronous generator connected to a remote bus. Simulation results show remarkable capabilities of our proposed method. In fact the Neuro-PSS, as a nonlinear controller, is capable of stabilizing the generator dynamic oscillations under different types of disturbances and different operating conditions of the generator (leading or lagging power factor) without online tuning or adaptation. The input signals of the proposed PSS are the generator voltage and speed which are easily accessible in a real generating plant. The authors believe that the performance and characteristics of the proposed Neuro-PSS could still be enhanced by development and application of efficient on-line training methods for Neural networks.

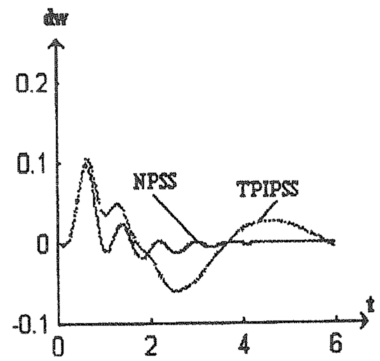

Figure 8: Performance of PSSs subjected to a 30% ref. voltage increase.

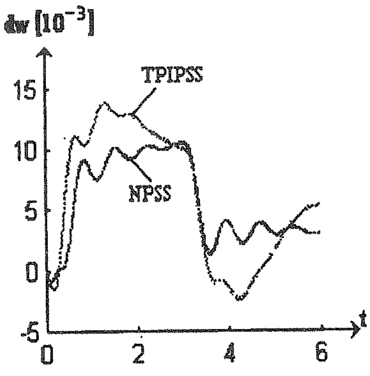

Figure 9: Performance of PSSs subjected to a 50% ref. power increase.

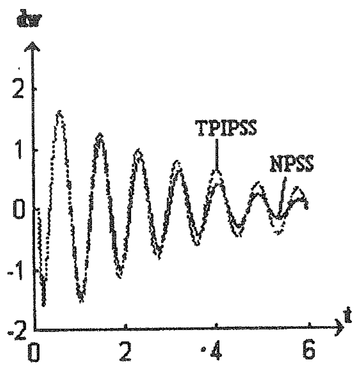

Figure 10: Performance of PSSs subjected to a fault on gen. terminal followed by a line outage in a double circuit line.

4. Application of Neural Nets To PSS

4.1. Design of the Neuro-PSS

As we mentioned before, our goal in this paper is to find a good approximation for the function G in equation (5). This is well done by means of a MLP neural network. The approach used here to design the Neuro-PSS (NPSS) is divided into two different phases. First, the method introduced in [14] is used to design a Tunable PIPSS (TPIPSS) in which a MLP neural network computes the K_I and K_P gains of the PIPSS based on the operating point of the generator. The MLP in this system is an approximation to the functions f_I and f_P in equation (1). The block diagram of this system is shown in Figure 5 below:

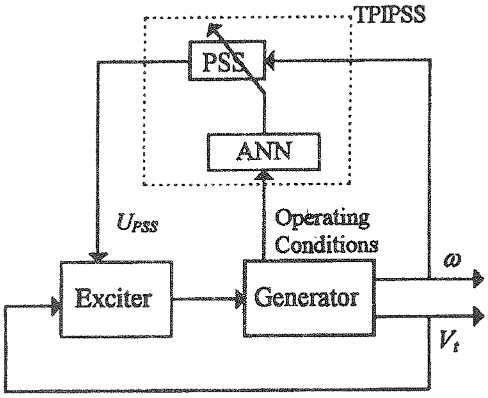

Figure 5: Block diagram of a tunable PIPSS

Before calculation of its output signal, the tunable PIPSS must know the system operating point in order to determine the K_P and K_I gains (see Figure 5).

In the second phase of our approach, we have substituted completely the TPIPSS with a MLP neural network. This neural network is an identified model of the TPIPSS and is trained on the pairs of input and output signals of the TPIPSS designed in the first phase. The MLP trained in this way represents an approximation to the function G in equation (5). Figure 6 shows the block diagram of the proposed neuro-PSS.

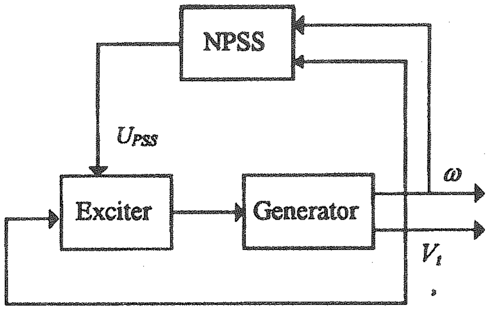

Figure 6: Block diagram of the Neuro-PSS

It should be noted that the input arguments of the function G can not be completely known in advance. By referring to [20], we know that for a system of order n, the last n time instances of outputs and inputs are sufficient as the input to the identifying neural network. In fact, the proper choice of the number ot past time instances of the variables, not exceeding n, is a matter of art; This can be found in general by a trial and error process.

4.2. Simulation Results

To illustrate the effectiveness of the proposed method, a synchronous generator connected to a remote bus shown in Figure 1 is considered. The 7th-order nonlinear model [4] is used for the generator which is equipped with a IEEE-DC1 type exciter with amplifier and excitation saturation. The turbine is a single-reheat tandom type and the governor is modeled with the general model of the steam turbine governors. All the model details and parameters are found in [4, 25, 26].

For the first stage, a NN (6-8-2) MLP neural network is trained off-line with the Marquardt algorithm to tune a PIPSS for varying operating conditions. The training data were chosen so that the complete operational range of the generator including both cases of leading and lagging power factors be covered. This neural network is then used in the block diagram of Figure 5 to tune the PIPSS adaptively.

NN(4-5-2). This convention is referred to illustrate the topology of a net.

3.2. Marquardt Training Algorithm

Consider a MLP network shown in Figure 3 with the general input-output relationship described as:

$$\underline{\alpha}(\underline{P}) = f(\underline{P}, \underline{\theta}); \underline{P} \in P \tag{6}$$

where $\underline{p}$, $\underline{\alpha}$ and $\underline{\theta}$ are respectively the input, the actual output and the parameters of the network. The parameters of the network include all the network weights and biases (W_i, b_i). By training, we mean the minimization of the network error function on the set of training data. This error function could be defined as follows:

$$E(\underline{\theta}) = \sum_{q=1}^{Q} \underline{e}_q^T . \underline{e}_q = \sum_{q=1}^{Q} \sum_{j=1}^{S_M} e_q^2(j) = \sum_{i=1}^{N} e_i^2(\underline{\theta}) \tag{7}$$

where Q and s_M are respectively the size of training data set and the size the network output vector and $\underline{e}_q$ is the difference between the desired output, $\underline{t}_q$, and the actual network output, $\underline{a}(\underline{p}_q)$:

$$\begin{aligned} \underline{e}_q &= \underline{t}_q - \underline{\alpha}(\underline{P}_q) \\ &= [e_q(1) e_q(2) \cdots e_q(S_M)]^T ; q = 1, \cdots, Q \end{aligned} \tag{8}$$

Remember that the adaptive rule for adjustment of the network parameters in order to minimize the error function may be generalized as:

$$\underline{\theta}(k+1) = \underline{\theta}(k) - \alpha(k) . \nabla E(\underline{\theta}(k)) \tag{9}$$

The simplest optimization method, the steepest descent algorithm, uses the opposite direction of the gradient vector (this is the basis of the BP algorithm [15, 16]), which is a vector pointing to the descent direction of the function $E(\underline{\theta})$, and will lead to a function decrease every adaptation step. The most important deficiency of steepest descent algorithm is that it does not say anything about the size of steps to be taken, the step size $\alpha(k)$ influences the rate of parameter adjustment or learning. This makes the standard BP scheme or any other gradient search method to be very sluggish.

The Newton method, as a basic method using second order information, tries to find the best $\alpha(k)$ in each step. In this method, the parameter adjustment is governed by:

$$\underline{\theta}(k+1) = \underline{\theta}(k) - \left[\nabla^2 E(\underline{\theta}(k))\right]^{-1} . \nabla E(\underline{\theta}(k)) \tag{10}$$

One problem with Newton method is that it requires the calculation of many 2nd-derivatives. Another problem occurs when the Hessian matrix, $\nabla^2 E$, is not positive definite and the inverse does not exist. As a remedy for the first problem, we may use the Gauss-Newton modification which uses the following approximation:

$$\nabla^2 E(\underline{\theta}) \approx J^T(\underline{\theta}) . J(\underline{\theta}) \tag{11}$$

where $J(\underline{\theta})$ is the Jaccobian matrix of the error function and is defined as:

$$J(\underline{\theta}) = [J_{ij}] = \left[\frac{\partial e_i(\underline{\theta})}{\partial \theta_j}\right] \tag{12}$$

To solve the second problem, Marquardt-Levenberg modification is used, which along with the Gauss-Newton modification makes the following approximation to the Newton's adjustment rule [23, 24]:

$$\underline{\theta}(k+1) = \underline{\theta}(k) - \left[J^T(\underline{\theta}) . J(\underline{\theta}) + \mu_k . \mathbf{I}\right]^{-1} . \nabla E(\underline{\theta}) \tag{13}$$

Notic that when μ_k is large, this approaches the steepest descent algorithm with small step size. The algorithm starts with a small μ_k and increases it by a factor β if the step does not yield a smaller value for $E(\underline{\theta})$, or decreases it if we are in the right descent direction.

in the discrete-time domain.

From the physical behavior of the power system it is well known that there exists is a strong coupling between generator output power (P_G) and system frequency (ω) as well as between generator reactive power (Q_G) and generator terminal voltage (V_t). Therefore the relations given in (1) can be simplified to:

$$\begin{cases} K_I = g_I(\omega, V_t) \\ K_p = g_P(\omega, V_t) \end{cases} \quad (4)$$

Combining the relations in (4) with the dynamic equation of PIPSS in (3), we may conclude the following simplified yet general equation for the PIPSS dynamics:

$$\begin{aligned} U_{PSS}(K) = G\,(&\Delta\omega(k), \Delta\omega(k-1), \cdots, \\ &\Delta V_t(K), \Delta V_t(k-1), \cdots, \\ &U_{PSS}(k-1), U_{PSS}(k-2), \cdots) \end{aligned} \quad (5)$$

Kept in a general form, the equation (5) states that if we can approximate the function G, then the output signal of the PIPSS could be computed directly from the generator speed and terminal voltage. This introduces the idea that a neural network can be used instead of a classical PIPSS to stabilize the system.

3. Neural Networks

Based on an analogy with the structure of real neaurons, artificial neural networks as non-linear model-free adaptive dynamic systems offer a practical approach for real problems such as adaptive identification and control of nonlinear systems [15-21]. This remarkable ability of neural networks comes mainly from the fact that they are capable of approximating (modeling) any square integrable arbitrary nonlinear function.

3.1 Architecture

A typical neural network, namely Multilayer Feedforward or Multilayer Perceptron (MLP), is shown in Figure 3. It comprises layers of interconnected processing elements (neurons) with the output of each layer being fed only to the input of the succeeding layer, thus named multilayer feedforward.

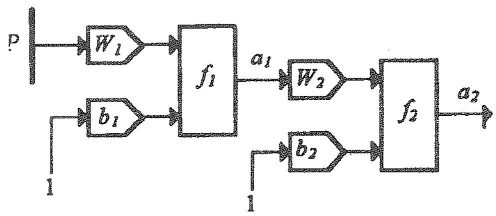

Figure 3: Block diagram of a 2-layer MLP

The basic building block of Figure 3 is shown in Figure 4, illustrating that each neuron sums all its inputs and then performs a nonlinear transfer function known as the activation function. This function can take a number of different forms. Although this is typically a sigmoidal type function, in principle any function with a bounded derivative could be used. In this paper we use the hyperbolic tangent type nonlinearity for the hidden layer and linear function for the output layer.

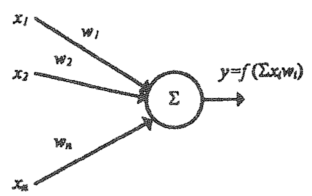

Figure 4: Simplified model of a neuron

The number of neurons in input/output layers are dependent on the respective number of inputs and outputs for the problem being considered, but the number of neurons in the hidden layers is a designer choice. A neural network with four inputs, one hidden layer with five neurons and two neurons in the output layer is denoted by the Marquardt algorithm. In section 4 a neuro-PSS is developed and its application to a synchronous generator connected to a remote bus as a sample power system is also presented. Finally section 5 concludes the paper.

2. Problem Statement

The system that we have considered in this paper is a synchronous generator connected to a remote bus through a transmission line as shown in Figure 1.

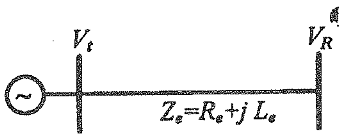

Figure 1: Single machine connected to a remote bus

Note that the remote bus emphasized in this paper is not an infinite bus; it has a varying voltage, V_R, where the subscript "R" stands for "remote". In fact the impedance Z_e and the voltage source V_R represent the Thevenin equivalent of the external network which is connected to the generator terminal. In this way, any operating condition of the generator such as faults on generator terminal, faults on remote bus or outage of a transmission line in a double circuit line could be simulated accordingly by some proper choices of Z_e and V_R.

Regarding the dynamics, the system is controlled through two different control loops. The first loop consists of exciter and AVR systems in order to control the terminal voltage, and the second loop consisting of turbine and governor, controls the system load and frequency.

To enhance the stability and to increase the system damping, the generator is also equipped with a Proportional-Integral Power System Stabilizer (PIPSS) acting as a supplementary excitation controller whose input is the speed deviation of the generator ($\Delta\omega$). This PIPSS consists of a wash-out filter and a limiter in its output. Figure 2 shows the complete diagram of the plant under study.

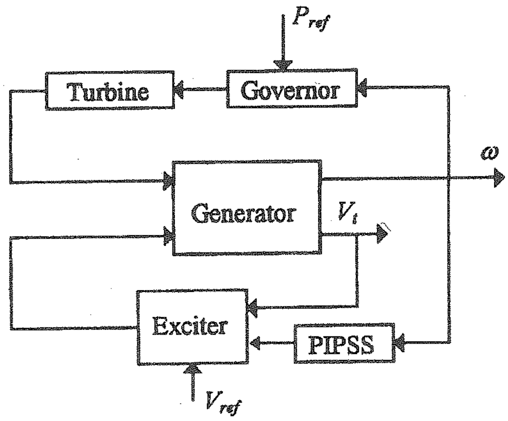

Figure 2: Block diagram of the generator control system

Based on the mathematical manipulations derived in reference [14], it can be said that there exists a nonlinear relation between the operating condition of the generator and the integral and proportional gains of the PIPSS (K_I, K_p). If we identify the operating point of the generator with the three parameters of P_G, Q_G and V_t, then the nonlinear relations could be written as follows:

$$\begin{cases} K_I = f_I(P_G, Q_G, V_t) \\ K_P = f_P(P_G, Q_G, V_t) \end{cases} \quad (1)$$

Now, if we consider the PIPSS as a dynamic system with $\Delta\omega(t)$, $U_{PSS}(t)$ as its input and output signals, respectively, and K_I and K_P parameters which vary with the operating point of the system, then the dynamic equation of the PIPSS could be shown to have the following general form,

$$U_{PSS}(t) = F(\Delta\omega, \Delta\dot{\omega}, \cdots, \dot{U}_{PSS}, \ddot{U}_{PSS}, \cdots, K_P(t), K_I(t)) \quad (2)$$

in continuous-time or equivalently

$$U_{PSS}(K) = \hat{F}(\Delta\omega(k), \Delta\omega(k-1), \ldots, U_{PSS}(k-1), U_{PSS}(k-2), \ldots, K_P(k), K_I(k)) \quad (3)$$

Design of A Neuro-PSS for Synchronous Generator

M. B. Menhaj - M. Abedi

Elect. Eng. Dept. Amirkabir Univ.

J. Shahroozi

Elect. Eng. Dept. Air Force Univ.

ABSTRACT:

This paper presents a novel approach for development of an adaptive Power System Stabilizer (PSS) using Artificial Neural Networks (ANN). The system which has been considered here is a synchronous generator connected to an external network. The generator is equipped with AVR and turbine governor compatible with IEEE standards. The neural network which has been employed here is a multilayer perceptron with Marquardt training algorithm. This algorithm is known to be the fastest training algorithm for feedforward networks [21]. The proposed neuro-PSS as a nonlinear controller, is capable of stabilizing the generator dynamic and transient oscillations under different types of disturbances and under different operating conditions as well.

1. Introduction

Dynamic stability and damping characteristics of synchronous generators are the two most important criteria that are highly concerned in power system design and operation. Power system stabilizers (PSS) have been widely employed in order to increase the stability [1-6]. Different types of PSSs are currently in use in power systems; the Lead-Lag type and the Proportional-Integral (PI) type are the most common ones. The parameters of PSSs are normally fixed at some pre-specified values which are determined under a particular operating condition.

Since the operating point of the system varies continuously due to load changes or system disturbances, the stabilizers designed for a specific operating condition will not yield satisfactory results in daily operation of the power system. This has led the researchers to the field of adaptive control in order to adjust the PSS parameters in real-time processing based upon on-line measurements [7-13].

All the different methods proposed for adaptive stabilization of synchronous generators suffer from computational deficiency; they all require plant identification in real-time before they can compute the controller parameters or the controller output. This task will be very time consuming especially for a computer with limited computational capabilities. This, in turn, has initated the use of artificial neural networks (ANN) to control and to stabilize power systems which will be discussed in sections 3 and 4.

This paper is organized as follows. Section 2 introduces the statement of the problem. Section 3 briefly describes the multilayer feedforward networks and formulates