

Adaptive Neural Network Based Power System Stabilizer Design

Wenxin Liu, Ganesh K. Venayagamoorthy, Donald C. Wunsch II

Department of Electrical and Computer Engineering
University of Missouri – Rolla
Rolla, MO 65401
USA

Abstract – Power system stabilizers (PSS) are used to generate supplementary control signals for the excitation system in order to damp the low frequency power system oscillations. To overcome the drawbacks of conventional PSS (CPSS), numerous techniques have been proposed in the literature. Based on the analysis of existing techniques, this paper presents an indirect adaptive neural network based power system stabilizer (IDNC) design. The proposed IDNC consists of a neuro-controller, which is used to generate a supplementary control signal to the excitation system, and a neuro-identifier, which is used to model the dynamics of the power system and to adapt the neuro-controller parameters. The proposed method has the features of a simple structure, adaptivity and fast response. The proposed IDNC is evaluated on a single machine infinite bus power system under different operating conditions and disturbances to demonstrate its effectiveness and robustness.

Keywords: Neural Networks, Power System Stabilizer, On-line Training, Indirect Adaptive Control.

I. INTRODUCTION

Power system stabilizers are used to generate supplementary control signals for the excitation system in order to damp the low frequency inter-area and intra-area oscillations [1]. Conventional power system stabilizer is widely used in existing power systems and has made a contribution in enhancing power system dynamic stability. The parameters of CPSS are determined based on a linearized model of the power system around a nominal operating point where they can provide good performance. Since power systems are highly nonlinear systems, with configurations and parameters that change with time, the CPSS design based on the linearized model of the power system cannot guarantee its performance in a practical operating environment.

To improve the performance of CPSSs, numerous techniques have been proposed for their design, such as using intelligent optimization methods (simulated annealing, genetic algorithm, tabu search) [2]-[4], fuzzy [5]-[6], neural networks and many other nonlinear control techniques. The intelligent optimization algorithms are used to determine the optimal parameters for CPSS by optimizing an eigenvalue based cost

function in an offline mode. Since the method is based on a linearized model and the parameters are not updated online, therefore they lack satisfactory performance during practical operation. The rule-based fuzzy logic control methods are well known for the difficulty in obtaining and adjusting the parameters of the rules especially online. Recent research indicates that more emphasis has been placed on the combined usage of fuzzy systems and other technologies such as neural networks to add adaptability to the design [7]. Currently, most of the nonlinear control based methods use simplified models to decrease complexity of the algorithms. Considering the complexity of practical power systems, more realistic model with less computation time is required for effective robust control over a wide range of operating conditions.

Since neural networks have the advantages of high computation speed, generalization and learning ability, they have been successfully applied to the identification and control of nonlinear systems. The work on the application of neural networks to the PSS design so far includes online tuning of CPSS parameters [8]-[9], the implementation of inverse model control [10]-[11], direct control and indirect adaptive control [12]-[15]. The online tuning of CPSS parameters and the inverse model control do not update the weights of neural networks online so their performances highly depend on the quality of offline training samples which are difficult to obtain. Since the plant model is not used in the direct adaptive neural network control structure, computation time is greatly decreased. But there is no accurate way to directly evaluate the performance of the controller, especially when the system parameters are changing over time; therefore, this is not the most effective control technique.

Paper [12] has shown that an indirect adaptive PSS can be designed using two input signals, the speed deviation and the power deviation to a neural network controller. In this paper, the authors show that it is possible to design an effective online trained PSS using only the speed deviation as an input to a neuro-controller. The indirect neural network control design consists of two neural networks, namely the neuro-controller and the neuro-identifier. The neuro-controller is used to generate the stabilizing supplementary control signal to the plant and the neuro-identifier is used to provide a dynamic model of the plant to evaluate and update the weights of the neuro-controller. Since both the neuro-identifier and the neuro-controller are updated online, the controller can adapt to changes in the system configuration and condition.

The power system model is described in section II. The design of the indirect adaptive neural network controller is described in section III. The training process of the IDNC is described in section IV. Some simulation results are provided in section V.

II. POWER SYSTEM MODEL

The single machine infinite bus power system (SMIB) model used to evaluate the IDNC is shown in Fig. 1. The SMIB consists of a synchronous generator, a turbine, a governor, an excitation system and a transmission line connected to an infinite bus. The model is built in MATLAB/SIMULINK environment using the Power System Blockset [16]. In Fig. 1, P_{REF} is the mechanical power reference, P_{SV} is the feedback through the governor, T_M is the turbine output torque, V_{inf} is the infinite bus voltage, V_{TREF} is terminal voltage reference, V_T is terminal voltage, V_A is the voltage regulator output, V_F is field voltage, V_E is the excitation system stabilizing signal, $\Delta\omega$ is the speed deviation, V_{PSS} is the PSS output signal, P is the active power and Q is the reactive power at the generator terminal.

In Fig. 1, the switch S_1 is used to carry out tests on the power system with IDNC, CPSS and without PSS (with switch S_1 in position 1, 2 and 3 respectively). Switch S_2 is used to select between normal operation and training phase (position 1 and 2 respectively).

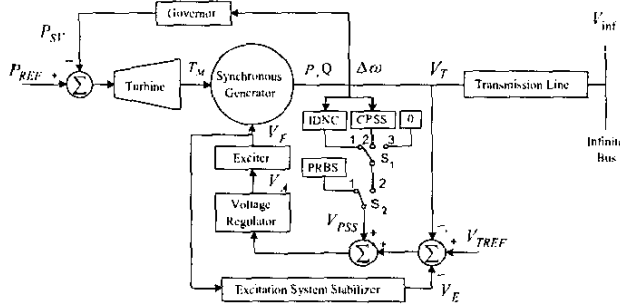


Fig. 1. System model configuration

The synchronous generator is described by a seventh order d-q axis set of equations with the machine current, speed and rotor angle as the state variables. The turbine is used to drive the generator and the governor is used to control the speed and the real power. The block diagram of a separately excited turbine and a conventional governor are shown in Fig. 2.

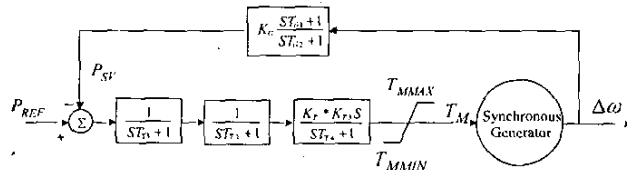


Fig. 2. Block diagram of the turbine and the governor

The excitation system for the generator is modeled according to IEEE Std. 421.5 [17]. The block diagram of the excitation system is shown in Fig. 3.

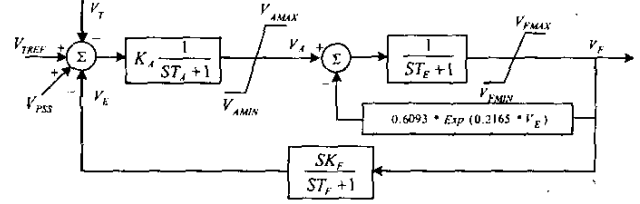


Fig. 3. Block diagram of the excitation system

The CPSS consists of two phase-lead compensation blocks, a signal washout block, and a gain block. The input signal is the rotor speed deviation $\Delta\omega$ [18]. The block diagram of the CPSS is shown in Fig. 4.

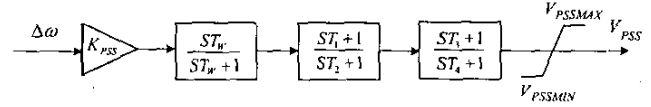


Fig. 4. Block diagram of the conventional power system stabilizer

The parameters for the generator, AVR, excitation system, turbine and governor are given in Appendix A [18]-[19].

III. DESIGN OF THE ADAPTIVE CONTROLLER

The indirect neural network controller (IDNC) design consists of two separate neural networks, namely the neuro-controller and the neuro-identifier. The structure for the training of the neuro-identifier and the neuro-controller is shown in Fig. 5. These two phases are carried out in cascade once the IDNC is connected to the plant by placing the switch S_1 at position 1 in Fig. 1. The dash lines show the backpropagation paths to update the weights of the neuro-identifier and the neuro-controller. The training processes of the neuro-identifier and neuro-controller are described in detail in section IV.

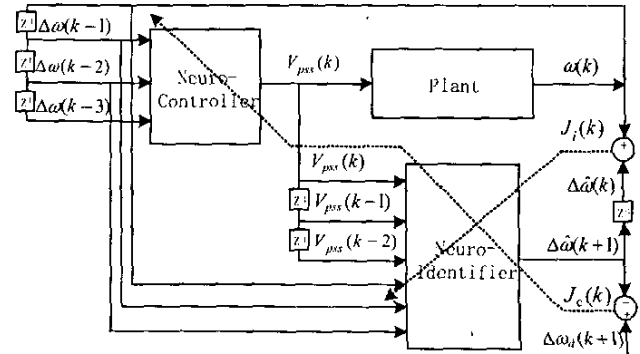


Fig. 5. Structure of the indirect neural network controller

Since the two training processes are shown in one figure, it is necessary to clarify the time step definition. Both $V_{pss}(k)$ and $\Delta\omega(k)$ signals are sampled at time step k , but $\Delta\omega(k)$ is not the response for the control signal $V_{pss}(k)$. Due to the time lag property of the plant, the impact of the control signal $V_{pss}(k)$ is reflected in the next time sample of the output signal $\Delta\omega(k+1)$. The following section describes the neuro-identifier and neuro-controller inputs and outputs for the online operation.

A. Neuro-Identifier

The neuro-identifier is developed using the series-parallel Nonlinear Auto Regressive Moving Average (NARMA) model [20]. The model output \hat{y} at time $k+1$ depends on both past n values of output and m past values of input. The neuro-identifier output equation takes the form given by (1).

$$\hat{y}(k+1) = f \left[\begin{array}{l} y(k), y(k-1), \dots, y(k-n+1) \\ u(k), u(k-1), \dots, u(k-m+1) \end{array} \right] \quad (1)$$

Where $y(k)$ and $u(k)$ represent the output and input of the plant to be controlled at time k . For this particular system, y , u and \hat{y} are the speed deviation $\Delta\omega$ of the plant, the output of the neuro-controller V_{pss} and the estimated plant output $\Delta\hat{\omega}(k)$ by the neuro-identifier respectively. Here both m and n are chosen to be 2. One reason for choosing three time step values is because a third order model of the system is sufficient for the study of transient stability. The other reason is that more time delays means more computation and one author's previous work verified that three time delays is enough for this kind of problem [Venayagamoorthy, 2001b]. The neuro-identifier is a multi-layer feedforward network trained with backpropagation (BP) algorithm. The numbers of neurons in the input, hidden and output layers are six, ten and one respectively. All the inputs and outputs signals of the neuro-identifier are normalized to the range of $[-1, 1]$.

B. Neuro-Controller

The neuro-controller is also a multi-layer feedforward network trained with BP algorithm. The number of neurons in input, hidden and output layers is three, six and one respectively. The inputs to the neuro-controller are the speed deviation $\Delta\omega$ and its two previous values and the output of the neuro-controller is the control signal V_{pss} . The number of neurons in the hidden layers of both the neuro-identifier and the neuro-controller are arrived at heuristically.

IV. TRAINING PROCESS

A. Pre-control Phase

During this phase, the switch S_2 is at position 2 in Fig. 1. The PRBS signal is added to the voltage regulator when the plant is operating at steady state. The addition of PRBS signals excites all possible dynamics of system. The values of

the random signals are distributed between -0.1 and 0.1 which is the typical output range limit of a CPSS.

1) Training of the neuro-identifier

The training process of the neuro-identifier during pre-control is shown in Fig. 6. The inputs to the neuro-identifier during this phase are $[\Delta\omega(k-1), \Delta\omega(k-2), \Delta\omega(k-3), V_{pss}(k-1), V_{pss}(k-2), V_{pss}(k-3)]$ and its output is $\Delta\hat{\omega}(k)$. The desired output is the output of the plant $\Delta\omega(k)$. The cost function for training the neuro-identifier is given by:

$$J_i(k) = \frac{1}{2} e_i(k)^2 = \frac{1}{2} [\Delta\omega(k) - \Delta\hat{\omega}(k)]^2 \quad (2)$$

Fig. 7 is used to show training process for the neuro-identifier. It can be seen that the neuro-identifier gives good estimation of the output of the plant after training for a period of 4 seconds.

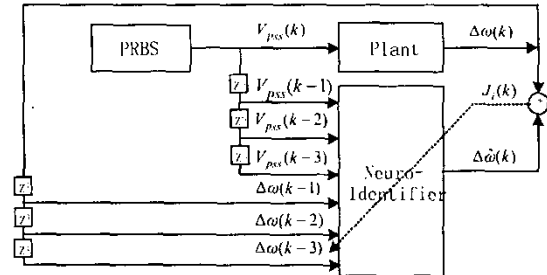


Fig. 6. Training of the neuro-identifier during pre-control

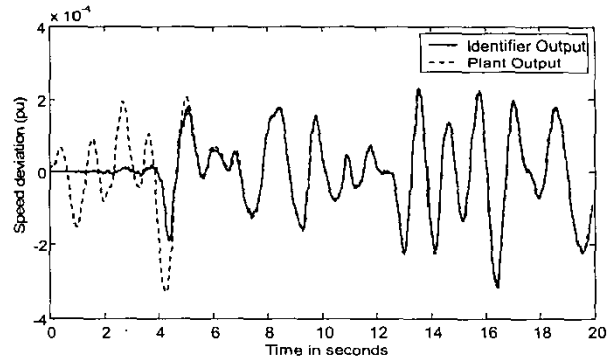


Fig. 7. Estimated and actual speed deviation of the plant

2) Training of the neuro-controller

The training process of the neuro-controller is shown in Fig. 8. The training of the neuro-controller takes place with the training of the neuro-identifier in cascade. During this stage, the input to the neuro-controller is $[\Delta\omega(k-1), \Delta\omega(k-2), \Delta\omega(k-3)]$ and the output is \hat{V}_{pss} , which is then fed to the neuro-identifier and evaluated against the desired output. The desired control signal is calculated through the neuro-identifier by comparing the output of the neuro-identifier with the desired system response (since the speed is expected to be

constant all the time, the desired speed deviation is set to zero all the time) and back propagating the error through the neuro-identifier. The desired control signal is calculated according to:

$$V_{pss}^d(k) = V_{pss}(k) + \frac{\partial J_c(k+1)}{\partial V_{pss}(k)} \quad (3)$$

During this phase the weights of the neuro-identifier are not changed. The cost function used to train the neuro-controller is given by:

$$J_c(k+1) = \frac{1}{2} e_c(k+1)^2 = \frac{1}{2} [\Delta\omega_d(k+1) - \Delta\hat{\omega}(k+1)]^2 \quad (4)$$

The cost functions (2) and (4) are also used for the training of the neuro-identifier and the neuro-controller during post-control phase. The trainings of the neuro-controller and the neuro-identifier continue for different operating points and system configurations until the weights of the neuro-controller converge.

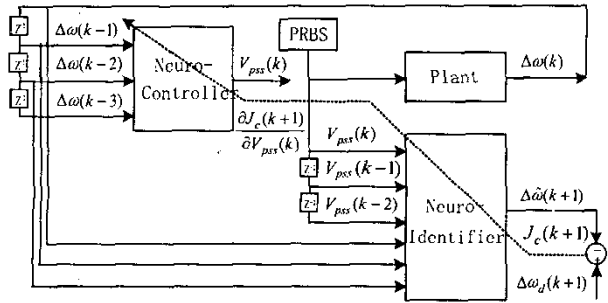


Fig. 8. Training of the neuro-controller during pre-control

B. Post-control phase

During this process, the switch S_2 in Fig. 1 is at position 1. The on-line training process includes three steps, which are the training of the neuro-identifier, training of the controller and, calculating and applying the control. The three stages are described below.

Before the training process of post-control, the system are simulated for four time steps, so as to get the time delayed value of the control signals - $V_{pss}(k-1)$, $V_{pss}(k-2)$, $V_{pss}(k-3)$ and system-states $\Delta\omega(k)$, $\Delta\omega(k-1)$, $\Delta\omega(k-2)$, $\Delta\omega(k-3)$.

1) Training of the neuro-identifier

Figure 9 shows the training process of the neuro-identifier during post control. So far the neuro-identifier is only trained around some stable operating point, it is necessary for it to be trained online to adapt to the cases when system states change. During this phase, the input and desired output of the neuro-identifier are $[\Delta\omega(k-1), \Delta\omega(k-2), \Delta\omega(k-3), V_{pss}(k-1), V_{pss}(k-2), V_{pss}(k-3)]$ and $\Delta\omega(k)$ respectively, where $\Delta\hat{\omega}(k)$ is the most current system state.

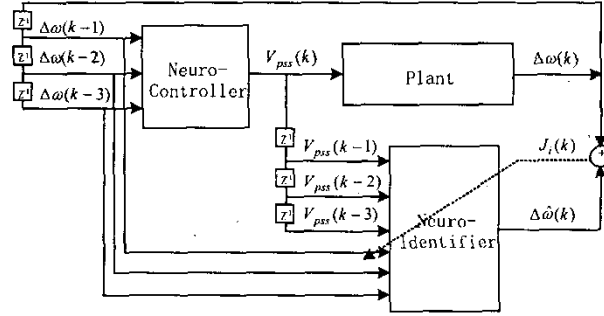


Fig. 9. Training process of the neuro-identifier

2) Training of the neuro-controller

Figure 10 shows the training process of the neuro-controller. During this phase, the neuro-controller inputs are $[\Delta\omega(k-1), \Delta\omega(k-2), \Delta\omega(k-3)]$ and the control signal is $V_{pss}(k)$. Inputs to the neuro-identifier are $[V_{pss}(k), V_{pss}(k-1), V_{pss}(k-2), \Delta\omega(k), \Delta\omega(k-1), \Delta\omega(k-2)]$, and its output is $\Delta\hat{\omega}(k+1)$, which is compared with the desired plant output $\Delta\omega_d(k+1)$. The cost function in (4) is used to update the weights of the neuro-controller.

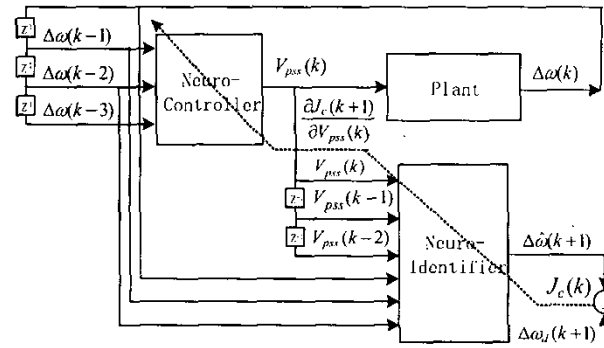


Fig. 10. Training process of the neuro-controller

3) Applying the control signal

At the next time step, a new control signal $V_{pss}(k+1)$ is calculated by simulating the neuro-controller with $[\Delta\omega(k), \Delta\omega(k-1), \Delta\omega(k-2)]$ and applied to the plant. The IDNC training process is repeated according to 1) to 3) for the entire operation of the plant.

V. SIMULATION RESULTS

To evaluate the performance of the neuro-controller, the system response of the IDNC is compared with the cases where there is no PSS and with a CPSS in the system. The comparison is carried out under different kinds of operating points ($P=0.401pu$, $Q=0.017pu$ and $P=0.602pu$, $Q=0.088pu$) and disturbances. These disturbances are the three phase short circuit at the infinite bus, step changes in the terminal voltage reference and a simulated transmission line fault.

A. Simulation Results ($P=0.401pu$, $Q=0.017pu$)

Figure 11 is the comparison of the system responses under a 50ms three phase short circuit fault occurring at 0.2 second. It can be seen that CPSS has better damping of the speed deviation than when there is no CPSS in the system but IDNC has the best damping. Figure 12 is the comparison of the system response to a 5% step change in V_{ref} ($1.05pu$ to $1.1025pu$) at 0.2 second and 5% decrease ($1.1025pu$ to $1.05pu$) at 5.2 second. Again, the IDNC provides the best damping. Figure 13 is the comparison of the system responses to a simulated transmission line fault. During this case, the impedance of the transmission line changes from $Z_1=0.05 + j1.5pu$ to $Z_2=0.025 + j0.7559pu$ at 0.2 second. Again, the IDNC provides the best damping of the three cases.

B. Simulation Results ($P=0.602pu$, $Q=0.088pu$)

Figure 14 is a comparison of the system responses under a 100ms three phase short circuit fault occurring at the infinite bus. Figure 15 is the comparison of the system response to 10% step change in V_{ref} ($1.05pu$ to $1.155pu$) at 0.2 second and 10% decrease ($1.155pu$ to $1.05pu$) at 5.2 second. And Fig. 16 is the comparison of the system responses to a simulated transmission line fault. The impedance of the transmission line changes from $Z_1=0.05 + j1.5pu$ to $Z_2=0.025 + j0.7559pu$ at 0.2 second. For these tests, the IDNC has the best performance even though the operating point is different.

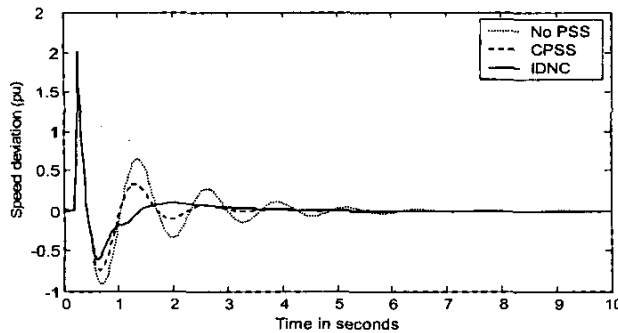


Fig. 11. System response to a three phase short circuit fault ($P=0.401pu$, $Q=0.017pu$).

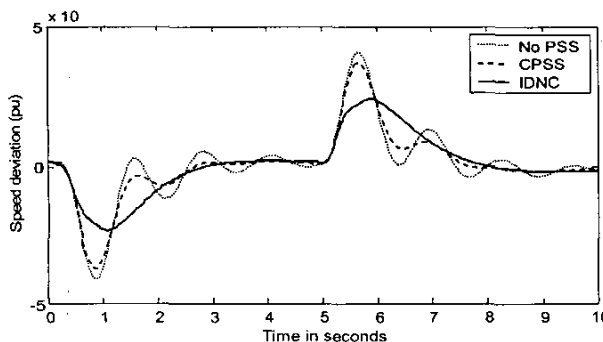


Fig. 12. System response to 5% step changes in the reference of terminal voltage ($P=0.401pu$, $Q=0.017pu$).

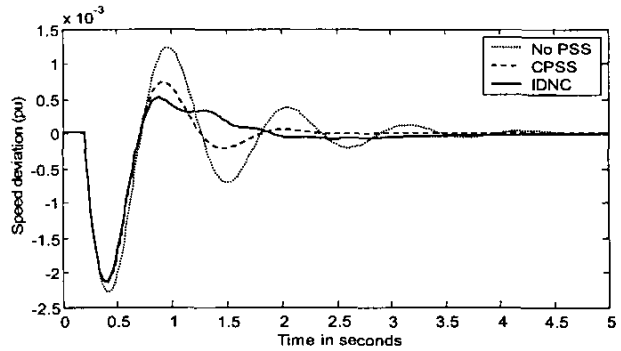


Fig. 13. System response to a simulated transmission line fault ($P=0.401pu$, $Q=0.017pu$).

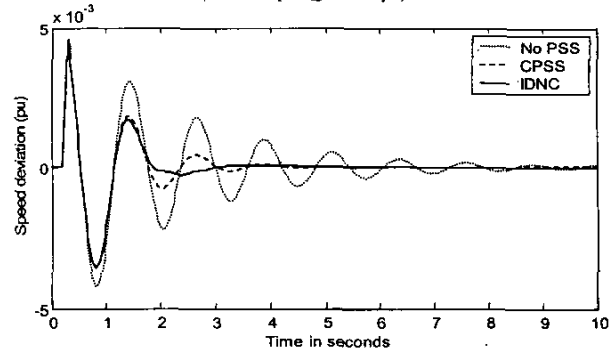


Fig. 14. System response to a three phase short circuit fault ($P=0.602pu$, $Q=0.088pu$).

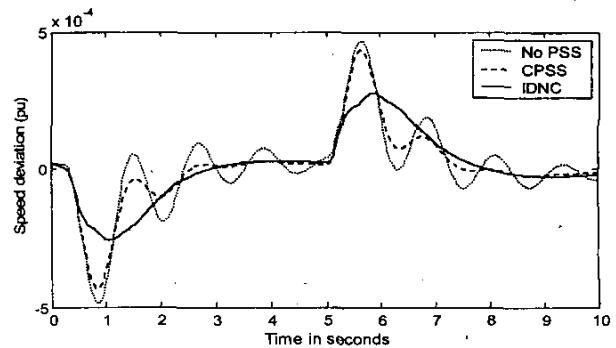


Fig. 15. System response to 5% step changes in the reference of terminal voltage ($P=0.602pu$, $Q=0.088pu$).

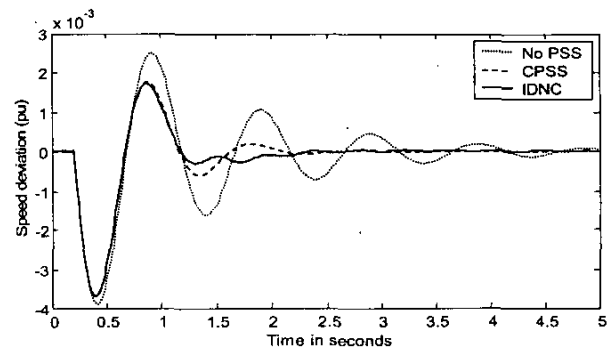


Fig. 16. System response to a simulated transmission line fault ($P=0.602pu$, $Q=0.088pu$).

From these figures we can see that the IDNC can provide better results compared with that of CPSS under all of the three cases.

VI. CONCLUSION

To overcome the drawbacks of conventional power system stabilizers, an adaptive indirect neural network control based power system stabilizer design is presented in this paper. The proposed method is evaluated on a single machine infinite bus power system. The design of the IDNC is based on only the speed deviation of the generator. Therefore, the computations involved in the neural network design are minimal. This is desirable for practical hardware implementation on the power station platforms. Simulation results for different kinds of disturbances and operating conditions demonstrate the effectiveness and robustness of the IDNC. Such a nonlinear adaptive PSS will yield better and fast damping under small and large disturbances even with changes in system operating conditions. Better and fast damping means that generators can operate more close to their maximum generation capacity. Thus, ensuring that generators remain stable under sever faults such as three phase short circuits. This means that more power generated per invested dollar.

REFERENCE

[1] Larsen, E.V. and Swann, D.A., "Applying power system stabilizers, Part I, II, III," IEEE Transaction on Power Apparatus and Systems, PAS-100 No. 6, pp. 3017-3041, 1981.

[2] Abido, M.A., "Robust design of multi-machine power system stabilizers using simulated annealing," IEEE Transaction on Energy Conversion, Vol. 15, No. 3, pp. 297-304, 2000.

[3] Do Bomfim, A.L.B., Taranto, G.N. and Falcao, D.M., "Simultaneous tuning of power system damping controllers using genetic algorithms," IEEE Transaction on Power Systems, Vol. 15, No. 1, pp. 163-169, 2000.

[4] Abdel-Magid, Y.L., Abido, M.A. and Mantaway, A.H., "Robust tuning of power system stabilizers in multi-machine power systems," IEEE Transactions on Power Systems, Vol. 15, No. 2, pp. 735 -740, 2000.

[5] El-Metwally, K.A., Hancock, G.C. and Malik, O.P., "Implementation of a fuzzy logic PSS using a micro-controller and experimental test results," IEEE Transaction on Energy Conversion, Vol. 11, No. 1, pp. 91 -96, 1996.

[6] Hariri, A. and Malik, O.P., "A fuzzy logic based power system stabilizer with learning ability," IEEE Transaction on Energy Conversion, Vol. 11, No. 4, pp. 721 -727, 1996.

[7] Hiyama, T. and Tomsovic, K., "Current status of fuzzy system applications in power systems," Proceedings of the IEEE SMC99, Tokyo, Japan, pp. 527-532, 1999.

[8] Hsu, Y.Y. and Chen, C.L., "Tuning of power system stabilizers using an artificial neural network," IEEE Transaction on Energy Conversion, Vol. 6, No. 4, pp. 612 -619, 1991.

[9] Segal, R., Kothari, M. L. and Madnani, S., "Radial basis function (RBF) network adaptive power system stabilizer," IEEE Transactions on Power Systems, Vol. 15, No. 2, pp. 722-727, 2000.

[10] Park, Y. M., Hyun, S. H. and Lee, J. H., "A synchronous generator stabilizer design using neuro inverse controller and error reduction network," IEEE Transactions on Power Systems, Vol. 11, No. 4, pp. 1969-1975, 1996.

[11] Zhang, Y., Malik, O.P., Hope, G.S. and Chen, G.P., "Application of an inverse input/output mapped ANN as a power system stabilizer," IEEE Transaction on Energy Conversion," Vol. 9, No.3, pp. 433-441, 1994.

[12] Shamsollahi, P. and Malik, O.P., "Direct neural adaptive control applied to synchronous generator," IEEE Transaction on Energy Conversion, Vol. 14, No. 4, pp. 1341-1346, 1999.

[13] Shamsollahi, P. and Malik, O.P., "Application of neural adaptive power system stabilizer in a multi-machine power system," IEEE Transaction on Energy Conversion, Vol. 14, No. 3, pp. 731-736, 1999.

[14] He, J. and Malik, O.P., "An adaptive power system stabilizer based on recurrent neural networks," IEEE Transaction on Energy Conversion, Vol. 12, No. 4, pp. 413 - 418, 1997.

[15] Young-Moon Park, Myeon-Song Choi and Lee, K.Y., "A neural network-based power system stabilizer using power flow characteristics," IEEE Transaction on Energy Conversion, Vol. 11, No. 2 , pp. 435 -441, 1996.

[16] Sybille, G., Brunelle, P., Champagne, R., Dessaint, L. and Lehuy, Hoang, "Power system blockset," Version 2.0, Mathworks Inc.

[17] IEEE, "Recommended practice for excitation system models for power system stability studies", IEEE Std. 421.5-1992, 1992.

[18] Kundur. P., Klein, M., Rogers, G.J. and Zywno, M.S., "Application of power system stabilizers for enhancement of overall system stability," IEEE Transactions on Power Systems, Vol. 4, No. 2, pp. 614 - 626, 1989.

[19] Venayagamoorthy, G.K., "Adaptive critic based neuro-controller for turbogenerators in a multi-machine power system," Ph.D. thesis. University of Natal, Durban, South Africa, 2001.

[20] Narendra, K. S. and Parthasarathy, "Identification and control of dynamical systems using neural networks", IEEE Transaction on Neural Networks, Vol. 1, No. 1, pp. 4 - 27, 1990.

APPENDIX A

Table 1: Parameters of the Single Machine Infinite Bus Power System in Fig. 1.

$T_{d0} = 6.69s$	$T_{d0}'' = 0.25s$	$X_{d0}' = 0.205pu$	$T_d' = 0.66s$
$T_q = 27ms$	$X_{d0}'' = 0.164pu$	$T_{d0}''' = 33ms$	$T_{kd} = 38ms$
$X_q = 1.98pu$	$T_d'' = 26.4ms$	$X_{d0} = 0.209pu$	$X_q'' = 0.213pu$
$T_{T1} = 0.15$	$T_{T2} = 0.594$	$T_{T3} = 0.884$	$T_{T4} = 2.662$
$T_{MIN} = 0$	$T_{MAX} = 1.2$	$K_C = 20$	$T_{G1} = 0.264$
$T_{G2} = 0.0264$	$K_A = 50$	$T_A = 0.01$	$V_{MIN} = -10$
$V_{MAX} = 10$	$T_E = 0.46$	$V_{FMIN} = 0.5$	$V_{FMAX} = 3$
$K_F = 0.1$	$T_F = 1$	$K_{STAB} = 25$	$T_W = 10$
$T_1 = 0.76$	$T_2 = 0.1$	$T_3 = 0.76$	$T_4 = 0.1$
$V_{PSS_MIN} = -0.1$	$V_{PSS_MAX} = 0.1$		