Power System Stabilizer Using a New Recurrent Neural Network for Multi-Machine

Chun-Jung Chen, Tien-Chi Chen, Jin-Chyz Ou

***Abstract* -This paper presents a power system stabilizer (PSS) for multi-machines using a new two-layer recurrent neural network (RNN) , which is called the recurrent neural network power system stabilizer (RNNPSS) in order to damp the oscillations of the multi-machines power system. The RNNPSS consists of a recurrent neural network identifier (RNNI) and a recurrent neural network controller (RNNC). The RNNI tracks the dynamics characteristics of the plant, and the RNNC to damp the system's low frequency oscillations. The RNN consists of an input layer and an output layer. Each neuron in the input layer is a recurrent one which is connected to oneself and other neurons, and then connected to the output layer. The proposed RNNPSS were simulated for three machines generator, the results demonstrate that the effectiveness of the proposed RNNPSS and reduce its sensitivity to system disturbances. The operating range was demonstrated better than the traditional PSS does.**

Index terms – Recurrent neural network, power system stabilizer, identifier and controller, RNNPSS [1]

I. INTRODUCTION

The power systems are complex nonlinear systems and often exhibit low-frequency oscillations due to dynamic loading and/or sudden mistake events on the transmission of power systems. It is especially happened to be nonlinear and extremely unstable for multi-machine system. Since the power system is highly non-linear dynamic, the PSS provides supplementary control signal to the excitation system and governor system of the generating unit to damp these oscillations and improve generator's dynamic performance. In recent decades, the conventional power system stabilizer (CPSS) such as the linear quadratic regulator PSS, self-tuning PSS, variable structure PSS and adaptive PSS have been proposed to provide optimum damping to the system oscillations [1]. The linear quadratic regulator PSS and pole assignment with least distance summation criterion methods are proposed to solve the state feedback gain in order to obtain the optimal poles and good transient response. The optimal pole shifting method for improving the oscillations of PSS controlled by adaptive optimal pole assignment have also been used in recent years [2]. The CPSS shows the following disadvantages: 1) The CPSS is based on linear model, but the power system doesn't. 2) Time consuming tuning is longer. 3) CPSS cannot provide the desired performance in the entire operating range.

The PSS based on neural network have been developed to damp the oscillation on power system in recent years [3]-[9]. The advantages of the PSS based on neural network include that it exhibits adaptive abilities in a network learning process, high reliability, good fault tolerance, and better performance on non-linear power system. Neural networks can be applied to control and identify the nonlinear systems since they approximate any desired degree of accuracy with a wide range of nonlinear model. In addition, the neural networks controller and identifier can be implemented in parallel and, therefore, shorter computational time. It also can avoid the system complexity and effectively damp the system oscillation when power system dynamics are complex. The PSS based on neural network can be classified as feed-forward neural network PSS (FNNPSS) and recurrent neural network PSS (RNNPSS). The FNNPSS can approximate any continuous functions closely. However, a FNNPSS is a static mapping. To deal with dynamical problems, a FNNPSS requires a large number of neurons to represent dynamical responses in the time domain. Moreover, the internal information of a FNNPSS cannot be utilized for weight updates and the function approximation is sensitive to the training data. On the other hand, a RNNPSS is a dynamic mapping. A RNNPSS can capture the dynamic response of a system without delays caused by external feedback because the recurrent neuron has an internal feedback loop. Thus, the RNNPSS demonstrates good control performance in the presence of uncertainties.

In this research, a new two-layer RNNPSS with a RNNC and RNNI was designed and applied to damp the oscillations of the power system with three machine generators. The RNNI, a two-layer neural network, was used to provide real-time adaptive identification of the unknown power system dynamics. The RNNC, a two-layer neural network, was used to produce an adaptive control force so that the power system states could accurately track the reference states. A back-propagation algorithm was used as the learning algorithm, in order to automatically adjust the parameters of the RNNC and RNNI. Each RNNPSS was designed to damp one of the generators respectively. The theoretical analysis and simulation results demonstrate that the effectiveness of the proposed RNNPSS and reduce its sensitivity to power system's parameter variations and disturbances.

II. THE PROPOSED RNN PSS

Since the RNNPSS controls each generator respectively. The application of RNNPSS for one of the three machines is shown in Fig. 1. The architecture of RNNPSS consists of a RNNI and RNNC. The RNNI takes the responsible to track the characteristics of the power system, and the RNNC is to provide an adaptive supplementary signal to excitation system or governor system of the generating units to damp the oscillations of power system.

Chun-Jung Chen is currently with the Faculty of Electrical Engineering, Kun Shan University, Tainan, Taiwan. He can be contacted at: ivanchen@mail.ksu.edu.tw.
Tien-Chi. Chen is currently with the Faculty of Engineering Science, National Cheng Kung University, Tainan, Taiwan. He can be contacted at: Tchichen@mail.ncku.edu.tw.
Jin-Chyz Ou is currently with the Faculty of Electronic Engineering, Kun Shan University, Tainan, Taiwan. He can be contacted at: ou@mail.ksu.edu.tw.

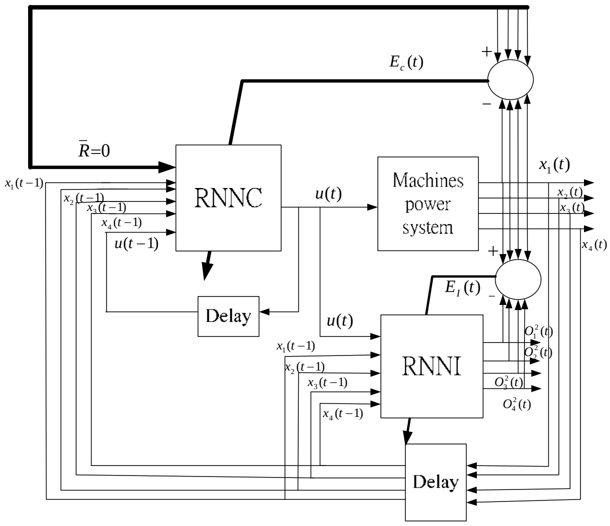

Fig. 1 The configuration of RNNPSS and power system

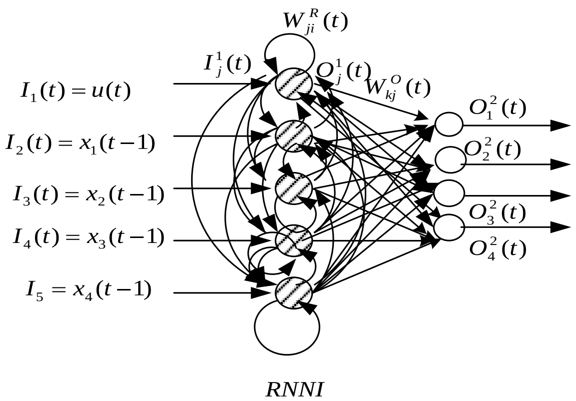

Fig.2 The network structure of the RNNI

1. The RNNI

The RNNI which tracks the dynamic characteristic trajectory of the plant, the network structure of the RNNI is shown in Fig. 2. The RNNI consists of a two-layered network structure, input layer and output layer. The input layer and output layer have m_I neurons and n_I neurons, respectively, which are setted as $m_I = 5$ and $n_I = 4$. The input and output of the *j*th neuron of the input layer are denoted by $I_j^1(t)$ and $O_j^1(t)$, $j=1,...,m_I$. The input and output of the *k*th neuron of the output layer are denoted as $I_k^2(t)$ and $O_k^2(t)$, $k=1,...,n_I$. The superscripts 1 and 2 represent the input layer (layer 1) and output layer (layer 2), respectively. The subscripts j and k represent the *j*th neuron and the *k*th neuron, respectively. The input signals of the RNNI are $I_1(t),...,I_{m_I}(t)$, which consists of $u(t)$, $x_1(t-1)$, …, $x_4(t-4)$. Each neurons of the input layer are connected with each other using recurrent weights $W_{ji}^R(t)$, $i=1,...,m_I$ and $j=1,...,m_I$. The neurons of the input layer and output layer are connected with weights $W_{kj}^O(t)$, $j=1,...,m_I$ and $k=1,...,n_I$.

The mathematical operation of the RNNI can be expressed as follows.

(a) Input layer:

The input and output of the *j*th neuron of the input layer can be expressed as:

$$I_j^1(t) = I_j(t) + \sum_{i=1}^{m_I} W_{ji}^R(t) O_i^1(t-1), \quad j = 1,.....,m_I \tag{1}$$

and

$$O_j^1(t) = f(I_j^1(t)) = \frac{e^{I_j^1(t)} - e^{-I_j^1(t)}}{e^{I_j^1(t)} + e^{-I_j^1(t)}}, \quad j = 1,.....,m_I \tag{2}$$

(b) Output layer:

The input and output of the *k*th neuron of the output layer are equal and can be expressed as:

$$O_k^2(t) = I_k^2(t) = \sum_{j=1}^{m_I} W_{kj}^O(t) O_j^1(t), \quad k = 1,.....,n_I \tag{3}$$

Due to the properties of guaranteed convergence, and minimizing the performance function, an error function for the RNNI is defined as

$$E_I(t) = \frac{1}{2}\sum_{k=1}^{n_I}\left(x_k(t) - O_k^2(t)\right)^2 \tag{4}$$

where $x_k(t), k = 1,...,n_I$ are the system outputs.

The weights $W_{kj}^O(t)$ and $W_{ji}^R(t)$ can be adjusted using the steepest descent algorithm, which is expressed as

$$W_{kj}^O(t+1) = W_{kj}^O(t) + \Delta W_{kj}^O(t) = W_{kj}^O(t) - \eta_I^O \frac{\partial E_I(t)}{\partial W_{kj}^O(t)} \tag{5}$$

$$W_{ji}^R(t+1) = W_{ji}^R(t) + \Delta W_{ji}^R(t) = W_{ji}^R(t) - \eta_I^R \frac{\partial E_I(t)}{\partial W_{ji}^R(t)} \tag{6}$$

where η_I^O and η_I^R and are the learning rates of the RNNI.

The gradient of error $E_I(t)$ with respect to the weights $W_{kj}^O(t)$ and $W_{ji}^R(t)$ are represented as

$$\frac{\partial E_I(t)}{\partial W_{kj}^O(t)} = -\left(x_k(t) - O_k^2(t)\right)O_j^1(t) \tag{7}$$

$$\frac{\partial E_I(t)}{\partial W_{ji}^R(t)} = -\sum_{k=1}^{n_I}\left[x_k\left((t) - O_k^2(t)\right)W_{kj}^O(t)\left(1 - \left(O_j^1(t)\right)^2\right)O_j^1(t-1)\right] \tag{8}$$

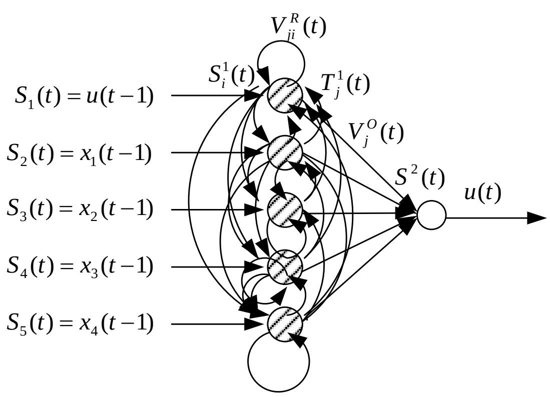

Fig. 3 The network structure of the RNNC

2. The RNNC

The network structure of the RNNC is shown in Fig. 3. The RNNC consists of a two-layered network structure, input layer and output layer. The input layer and output layer have m_C neurons and one neuron, respectively, which are setted as $m_C = 5$. The input and output of the *j*th neuron of the input layer are denoted by $S_j^1(t)$ and $T_j^1(t)$, $j=1,\ldots,m_C$. The output layer has one neuron, which input and output are denoted as $S^2(t)$ and $u(t)$. The superscripts and subscripts are the same as those of the RNNI. The input signals of the RNNC are $S_1(t),\ldots,S_{m_C}(t)$, which consists of $u(t-1), x_1(t-1), \ldots, x_4(t-1)$. Each neurons of the input layer are connected with each other using recurrent weights $V_{ji}^R(t)$, $i=1,\ldots,m_C$ and $j=1,\ldots,m_C$. The neurons of the input layer and output layer are connected with weights $V_j^O(t)$, $j=1,\ldots,m_C$.

The mathematical operation of the RNNC can be expressed as follows.

(a) Input layer:

The input and output of the *j*th neuron of the input layer can be expressed as:

$$S_j^1(t) = S_j(t) + \sum_{i=1}^{m_C} V_{ji}^R(t) T_i^1(t-1), \quad j = 1,\ldots\ldots,m_C \tag{9}$$

and

$$T_j^1(t) = f(S_j^1(t)) = \frac{e^{S_j^1(t)} - e^{-S_j^1(t)}}{e^{S_j^1(t)} + e^{-S_j^1(t)}}, \quad j = 1,\ldots\ldots,m_C \tag{10}$$

(b) Output layer:

The output layer has one neuron, which input and output are equal and can be expressed as:

$$u(t) = S^2(t) = \sum_{j=1}^{m_C} V_j^O(t) T_j^1(t), \tag{11}$$

Due to the properties of guaranteed convergence, and minimizing the performance function, an error function for the RNNC is defined as

$$E_C(t) = \frac{1}{2}\sum_{k=1}^{m_C}\left(r_k(t) - x_k(t)\right)^2 \tag{12}$$

where $x_k(t)$ and $r_k(t)$ are the system outputs and reference commands.

The weights $V_{kj}^O(t)$ and $V_{ji}^R(t)$ can be adjusted using the steepest descent algorithm, which is expressed as

$$V_j^O(t+1) = V_j^O(t) + \Delta V_j^O(t) = V_j^O(t) - \eta_C^O \frac{\partial E_C(t)}{\partial V_j^O(t)} \tag{13}$$

$$V_{ji}^R(t+1) = V_{ji}^R(t) + \Delta V_{ji}^R(t) = V_{ji}^R(t) - \eta_C^R \frac{\partial E_C(t)}{\partial V_{ji}^R(t)} \tag{14}$$

where η_C^O and η_C^R and are the learning rates of the RNNC.

The gradient of error $E_C(t)$ with respect to the weights $V_{kj}^O(t)$ and $V_{ji}^R(t)$ are represented as

$$\frac{\partial E_C(t)}{\partial V_j^O(t)} = -\sum_{k=1}^{n_C}\left(\left(r_k(t) - x_k(t)\right) W_{k1}^O(t)\left(1 - \left(O_1^1(t)\right)^2\right)\right) \cdot T_j^1(t) \tag{15}$$

$$\frac{\partial E_C(t)}{\partial V_{ji}^R(t)} = -\sum_{k=1}^{n_c}\left(\left(r_k(t) - x_k(t)\right) W_{k1}^O(t)\left(1 - \left(O_1^1(t)\right)^2\right)\right) \cdot V_j^O(t)\left(1 - \left(T_j^1(t)\right)^2\right) T_i^1(t-1) \tag{16}$$

III. COMPUTER SIMULATIONS

The state equation of a three-machine power system was given as [2]

$$\dot{x}(t) = Ax(t) + Bu(t) + Tw(t).$$

where x the state vector, u is the input vector, and w is the disturbance vector, are expressed as follows:

$$x(t) = [\Delta\delta_1(t), \Delta\omega_1(t), \Delta e_{q1}(t), \Delta e_{FD1}(t), \Delta\delta_{12}(t), \Delta\omega_2(t), \Delta e_{q2}(t), \Delta e_{FD2}(t), \Delta\delta_3(t), \Delta\omega_3(t), \Delta e_{q3}(t), \Delta e_{FD3}(t)]^T$$

$$u = [u_1, u_2, u_3]^T$$

$$w = [\Delta V_{ref1}, \Delta T_{m1}, \Delta V_{ref2}, \Delta T_{m2}, \Delta V_{ref3}, \Delta T_{m3}]^T$$

The system matrix A, B and T are given as follows [2, 10]：

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

$$A_{11} = \begin{bmatrix} 0.0 & 377.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -0.147 & 0.039 & -0.013 & 0.0 & 0.022 & 0.004 \\ -0.266 & -3.393 & -0.922 & 1.0 & -0.087 & 0.754 \\ -30.1 & -309.14 & -60.943 & -20.0 & 24.599 & -91.99 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 377.0 \\ 0.004 & -0.034 & -0.087 & 0.0 & -0.149 & 0.032 \end{bmatrix}$$

$$A_{12} = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.024 & 0.0 & -0.3 & 1.1 & 0.1 & 0.0 \\ -3.501 & 0.0 & 62.1 & -1675.0 & -10.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -0.008 & 0.0 & 0.079 & -0.0028 & 0.0 & 0.0 \end{bmatrix}$$

$$A_{21} = \begin{bmatrix} 0.121 & 1.313 & 0.021 & 0.0 & -1.6 & -1.885 \\ -18.48 & -64.47 & -12.55 & 0.0 & 106.09 & -516.11 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.001 & -0.017 & -0.003 & 0.0 & 0.017 & -0.01 \\ 0.083 & 0.0 & -0.002 & 0.0 & 0.22 & 0.0 \\ -10.1 & -33.934 & -6.78 & 0.0 & 1.7 & -46.37 \end{bmatrix}$$

$$A_{22} = \begin{bmatrix} -0.21 & 1.0 & 0.46 & 0.754 & 0.06 & 0.0 \\ -21.67 & -20.0 & 16.99 & -171.91 & -11.41 & 0.0 \\ 0.0 & 0.0 & 0.0 & 377.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.056 & -0.017 & -0.009 & 0.0 \\ 0.011 & 0.0 & -1.2 & -1.131 & -0.197 & 1.0 \\ -2.1 & 0.0 & 7.0.1 & -893.49 & -54.4 & -20.0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 800 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 900 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1000 \end{bmatrix}^T$$

$$T = \begin{bmatrix} 0 & 0 & 0 & 800 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.217 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 900 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0147 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1000 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.178 & 0 & 0 \end{bmatrix}^T$$

The comparison of the PPS was simulated for three types of stabilizer-LQR stabilizer [2], FNNPSS and the proposed RNNPSS.

(1) The RNNPSS

The learning rates were chosen as $\eta_I^R = \eta_I^O = \eta_c^R = \eta_c^O = 0.1$, Initial weight values, $W_{ji}^R(t)$, $W_{kj}^O(t)$ and $V_{cji}^R(t)$, $V_{cj}^O(t)$, are all chosen arbitrarily between 0.1~0.5.

(2) The FNNPSS

The learning rates and initial weight values are the same as those of the RNNPSS.

(3) The LQR stabilizer [2,10]

The control input was $u = -Kx$.

$$K = [K_{11} \quad K_{12}].$$

where

$$K_{11} = \begin{bmatrix} 2.4104 & -76.5612 & 3.5553 & 0.0725 & -1.2351 & -211.8305 \\ 0.1960 & 3.3675 & 0.0364 & 0.0005 & -0.0726 & -20.1852 \\ -0.4233 & -12.7616 & 0.0344 & 0.0004 & -0.2825 & -9.4848 \end{bmatrix}$$

$$K_{12} = \begin{bmatrix} 0.0411 & 0.0005 & 0.3662 & -136.2018 & 0.0268 & 0.0003 \\ 3.1244 & 0.0640 & 0.0414 & -5.8995 & -0.0057 & -0.0001 \\ 0.0002 & -0.0001 & 0.0129 & -193.8797 & 3.1371 & 0.0617 \end{bmatrix}$$

The poles of the closed-loop system $A - BK$ are

$-40.6712 \pm j38.8362$, $-38.8845 \pm j36.5313$,
$-36.5320 \pm j34.5999$, $-1.7979 \pm j7.8165$,
$-0.8126 \pm j4.3389$, $-0.6283 \pm j7.5563$.

Figures 4, 5 and 6 show the responses of state $\Delta\omega_1$, $\Delta\omega_2$ and $\Delta\omega_3$ respectively using the LQR stabilizer, FNNPSS and RNNPSS for the disturbance $w = [0 \quad 0.01]$ pu. These figures indicate that the proposed RNNPSS has better performance than the LQR stabilizer and FNNPSS.

IV. CONCLUSIONS

The RNNPSS proposed in this paper shows the faster convergence than the FNNPSS and LQR stabilizer in a three-machine power system. Because the proposed neural network consists of only two layers, the time of updating the weights is faster than the conventional three layer neural networks. The RNNPSS can avoid the system complexity and effectively damp the system oscillation when plant dynamics are complex and nonlinear.

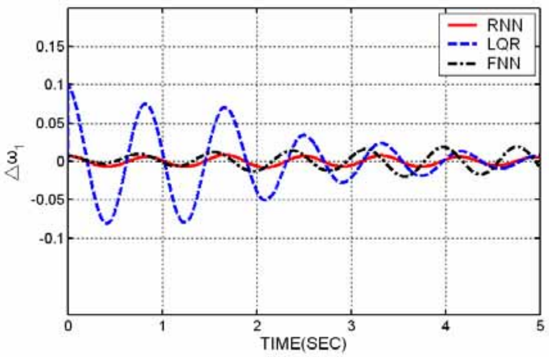

Fig..4 The responses of state $\Delta\omega_1$ using the LQR stabilizer, FNNPSS and RNNPSS for the disturbance $w_1 = [0 \quad 0.01]$ pu.

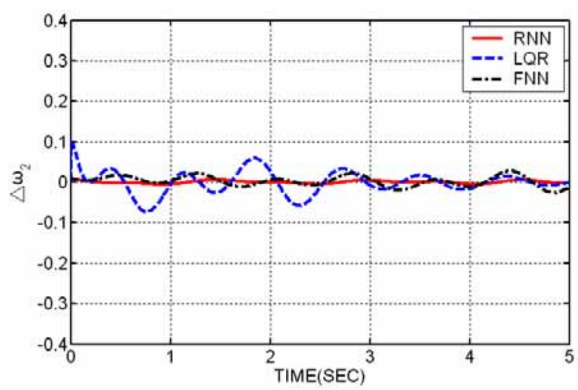

Fig. 5 The responses of state $\Delta\omega_2$ using the LQR stabilizer, FNNPSS and RNNPSS for the disturbance $w_2 = [0 \quad 0.01]$ pu.

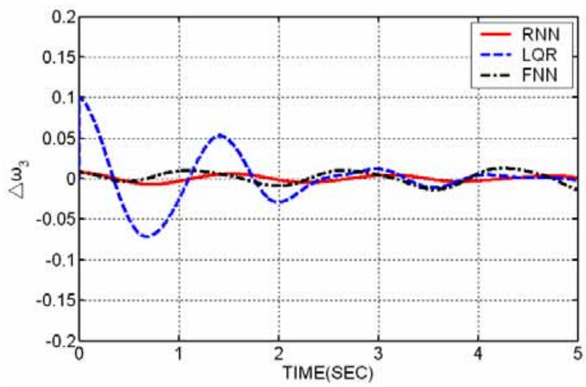

Fig. 6 The responses of state $\Delta\omega_3$ using the LQR stabilizer, FNNPSS and RNNPSS for the disturbance $w_3 = [0 \quad 0.01]$ pu.

V. REFERENCES

[1] W.C. Chan, and Y.Y. Hsu, "An Optimal Variable Structure Stabilizer for Power System Stabilization," *IEEE Trans.*, Vol. PAS-102, pp. 1738-1746, 1983.

[2] Tien-Chi Chen. and L. R. Chang-Chien., "Optimal Pole assignment to Stabilize a Power System" *Journal of Control Systems and Technology*, Vol. 4, No. 2, pp. 119-126 (1996).

[3] Ruhua You, Hassan J. Eghbali, and M.Hashem Nehrir. "An Online Adaptive Neuro-Fuzzy Power System Stabilizer for Multi-machine Systems" *IEEE Transaction on Power Systems*, Vol. 18, NO.1,Feb, 2003, pp.128-135.

[4] F. Shamsollahi and O. P. Malik, " An adaptive Power Stabilizer Using On-Line Trained Neural Networks," *IEEE Trans. Energy Conversion*, Vol. 12, No. 4, Dec. 1997, pp. 382-387.

[5] Dong. Xu and Ren-mu. He. "ANN Based Multiple Power System Stabilizers Adaptive and Coordinates Control". *PowerCon 2002. International Conference Proceeding*, Vol 1, 13-17. October 2002, pp. 361-364.

[6] D. K. Chatuevedi and O. P. Malik, "Experimental Studies With a Generalized Neuron-Based Power System Stabilizer". *IEEE Trans. Power System*, Vol. 19, NO. 3, August 2004.

[7] M. M. Salem "Studies on a Multi-Machine Power System With a Neural Network Based Excitation Controller", *Power Engineering Society meeting*, ***2000 IEEE***, Vol 1, July 2000, pp. 105-110

[8] J. He and O. P. Malik, "An Adaptive Power System Stabilizer Based on Recurrent Neural Networks" *IEEE Trans. Energy Conversion*, Vol. 12, No. 4, Dec. 1997, pp. 413-419.

[9] Payman Shamsollahi and Om P. Malik, "Application of Neural Adaptive Power System Stabilizer in a Multi-Machine Power System," *IEEE Trans. Energy Conversion*, Vol. 14, No. 3, Sept. 1999, pp. 731-736

[10] Fleming, R.J., Mohan M.A. and K. Parvatisam, "Selection of parameters of Stabilizers in Multimaching Power System," *IEEE Trans. Power Syst.*, Vol. PAS-100, pp. 2329-2333, 1981

VI. BIOGRAPHIES

Chun-Jung Chen was born in, Taiwan, R.O. C., in 1956. He received his B. S. and M. S. from National Cheng Kung University and E-Shou University in 1984 and 1992 respectively. Presently, he is working forwards his Ph. D. degree at the Cheng Kung University, Taiwan. He is at present a Lecturer at the Kun Shan University. His research interests include power system stabilizer, artificial neural network, adaptive Control and fuzzy control.

Tien-Chi Chen was born in Tainan, Taiwan, R. O. C., in 1956. He received his B. S. and M. S. from National Taiwan University, and Ph. D degree in 1989, from National Cheng Kung University. He is at present a Professor in the Dept. of Engineering Science, Cheng Kung University, Taiwan. His research interests include servo system control, power system stabilizer, artificial neural network, adaptive system control and fuzzy control.

Jin-Chyz Ou was born in, Taiwan, R.O. C., in 1955. He received his B. S. and M. S. from Chinese Culture University and E-Shou University in 1986 and 1992 respectively. Presently, he is working forwards his Ph. D. degree at the Cheng Kung University, Taiwan. He is at present a Lecturer at the Kun Shan University. His research interests include power system stabilizer, artificial neural network, adaptive Control and fuzzy control.