Proc. 2002 International Joint Conference on Neural Networks, 2002 World Congress on Computational Intelligence, Honolulu, Hawaii, pp. 2132-2137, May 12-17, 2002.

Power System Stabilization Using A Free-Model Based Inverse Dynamic Neuro Controller

Kwang Y. Lee, *Fellow IEEE*, Hee-Sang Ko

Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802

Abstract--**This paper presents an implementation of power system stabilizer using inverse dynamic neuro controller. Traditionally, mutilayer neural network is used for a universal approximator and applied to a system as a neuro-controller. In this case, at least two neural networks are required and continuous tuning of neuro-controller is required. Moreover, training of neural network is required considering all possible disturbances, which is impractical in real situation.**

In this paper, Inverse Dynamic Neuro Model (IDNM) is introduced to avoid this problem. Inverse Dynamic Neuro Controller consists of IDNM and Error Reduction neuro Model (ERNM). Once the IDNM is trained, it does not require retuning for cases with other types of disturbances. The controller is tested for one machine and infinite-bus power system for various operating conditions.

Index Terms--**Free model, Power system stabilization, Inverse dynamics, Neuro controller.**

I. INTRODUCTION

Power plants interconnected to a power system are complex, fast acting multivariable systems with high nonlinearity and a wide range of time constants in different control loops. Interconnected to a power system, power plants operate over a wide range of operating conditions, from lagging to leading phase, and are subjected to different types of disturbances, such as changes in terminal voltage and transformer tap positions, adjustment of operating conditions by changing inputs to governor and excitation systems, and short circuits on transmission lines. The characteristics of power plants vary as conditions change, but the outputs have to satisfy the requirements in power system operation. The ultimate goal of the power plant control is a stable operation of the power system in spite of a wide of range of the plant operation.

Considerable efforts have been made to synthesize Power System Stabilizers (PSS) for power systems, most of which are based on the deMello and Concordia's pioneering work [1,2]. In their work, a linearized model is used to find a proper set of parameters in a fixed structured PSS. Linear optimal control and modern control theories were also introduced to improve the dynamic performance of power systems under the uncertainty in power system models [3,4,5]. These techniques, however, depend on the model accuracy, which is less reliable as the power system becomes larger.

Adaptive techniques have been also employed in the PSS design for wide range operations [5-13]. In real applications, however, they have a shortcoming of immense calculation in every sampling period.

Recently, Artificial Neural Networks (ANN) have attracted attention of power system engineers. There have been a great deal of researches reported on ANN and their application to control systems [14,16]. Inverse Dynamic Neural Network (IDNN) was introduced in [14], which requires Error Reduction Network consisting of three IDNNs, and [15] requires the training of the neural network considering all possible conditions. Poggio and Girossi have revealed that the learning of an ANN is equivalent to synthesizing an associative memory between input space and output space [17]. It was also regarded as an equivalent problem to the estimation of an input-output transformation using a given input-output data set. Nguyen and Widrow show that the neural network can be applied to control a highly nonlinear system [18]. Iiguni and Sakai used an ANN as an auxiliary controller for conventional LQ controller to compensate for the nonlinearities existing in the control system [19]. There has been a great deal of literature in which the learning ability of ANNs was exploited and applied to the PSS problems. Hsu and Chen [20] proposed a real time self-tuning approach, where they used ANN to tune the parameters of conventional proportional-integral type PSS. However, since their approach requires a mathematical model of the controlled system for a wide range of operating conditions, it bears a shortcoming in large scale system application. Wu, Hogg and Irwin [21] presented a hierarchically structured neuro-PSS whose approach is similar to that of Nguyen and Widrow [18]. The neuro-PSS consists of two subnets; one for input-output mapping, and the other for control. Kennedy and Quintana [22] suggested an inverse controller using ANN. In their work, an inverse dynamic relationship is represented in the state space and is trained by an ANN that is used as an inverse controller.

The input to neuro controller is a full state vector of the controlled plant, which is somewhat impractical in real world. Zhang and coworkers suggested a PSS using an inverse input-output mapped ANN [15]. In their paper, inverse control is well explained, but it requires a protection scheme for large control input when the plant is of non-minimum phase.

0-7803-7278-6/02/$10.00 ©2002 IEEE

Most of the neuro controllers, with few exceptions, have a common feature, that is, using two ANNs; one for system identification and the other for controller. The ANN identifier is trained first with input-output data to emulate the controlled system. After training of the identifier, it is used as an error propagation tunnel in the ANN controller training. The training of the controller is performed by minimizing a cost function. This burden of training two neural networks is a drawback of conventional neuro-control approaches. After the completion of neuro-controller training, it is difficult to change the control performance because it requires additional training of the neuro-controller. In general, the training of a controller is more difficult then that of an identifier. This is another drawback.

In this paper, an inverse input-output relationship of the controlled system is identified by an Inverse Dynamic Neuro Model (IDNM) based on the free model concept [23,24]. The training procedure of the IDNM uses the Levenberg-Marquardt method [25], where the desired output of IDNM is not the system output but the control input. After training, the IDNM is used as an inverse controller, and additional training of the controller is not needed. Instead, the Error Reduction Neuro Model (ERNM) is introduced to minimize the modeling error of the IDNM.

II. INVERSE DYNAMIC NEURO CONTROLLER

A. Inverse Dynamic Neuro Model

In this work, the free-model concept [23,24] is applied to find an Inverse Dynamic Neuro Model (IDNM) using input-output data only. After the IDNM is obtained, it replaces an existing controller. In the free model, the data used has incremental forms using backward difference operators. Such data forms are from the concept of the free model, in that an unknown system can be identified if the differences, such as position, velocity and acceleration, are known.

The inverse dynamic neuro controller is developed as an alternative controller for the existing controller, which is the Power System Stabilizer. The controller employs IDNM and Error Reduction Neuro Model (ERNM); IDNM as a feedforward controller which increases the performance of the system and improves the tracking error, and ERNM as a feedback controller which takes care of the stability.

In general, the output of a system can be described with a function or a mapping of the plant input-output history. For a single-input single-output (SISO) discrete-time system, the mapping can be written in the form of a nonlinear function as follows:

$$\begin{aligned} y(k+1) = f(&y(k), y(k-1), \dots, y(k-n), \\ &u(k), u(k-1), \dots, u(k-m)) \end{aligned} \quad (1)$$

Using the free-model concept, this can be equivalently represented as following:

$$\begin{aligned} y(k+1) = f(&y(k), \Delta y(k), \dots, \Delta^n y(k), u(k), \\ &u(k-1), \Delta u(k-1), \dots, \Delta^m u(k-1)), \end{aligned} \quad (2)$$

where Δ^i is the backward difference operator defined as

$$\Delta^i f(k) = \Delta^{i-1} f(k) - \Delta^{i-1} f(k-1), \quad \Delta^0 f(k) = f(k). \quad (3)$$

Solving for the control, (2) can be represented as following:

$$\begin{aligned} u(k) = g(&y(k+1), y(k), \Delta y(k), \Delta^2 y(k), \dots, \Delta^n y(k), \\ &u(k-1), \Delta u(k-1), \Delta^2 u(k-1), \dots, \Delta^m u(k-1)), \end{aligned} \quad (4)$$

which is a nonlinear inverse mapping of (2).

The objective of the control problem is to find a control sequence which will drive the system to an arbitrary reference trajectory. This can be achieved by replacing $y(k+1)$ in (4) with $y_r(k+1)$, the reference trajectory.

In general, the IDNM can be represented by a neural network. In Fig. 1, the training mode is introduced, where Δ denotes the vector of difference operators defined in (3). In this process, the closed-loop identification is required since the plant may not be stable initially. Pseudo-Random Binary Signal (PRBS) is applied for the closed-loop system identification [26] to collect input-output data used in training mode. In Fig. 2, it is shown how the IDNM is applied as a controller in the system, where a supplementary signal $\hat{E}(k)$ is applied in order to compensate for modeling error.

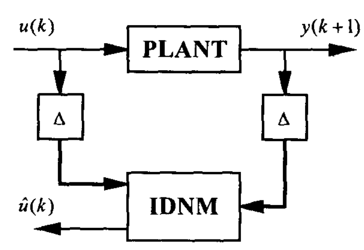

Fig. 1. Training mode of IDNM.

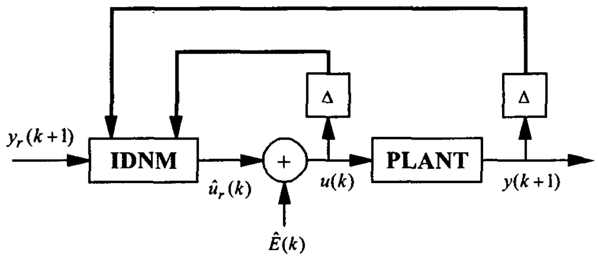

Fig. 2. Control mode of IDNM.

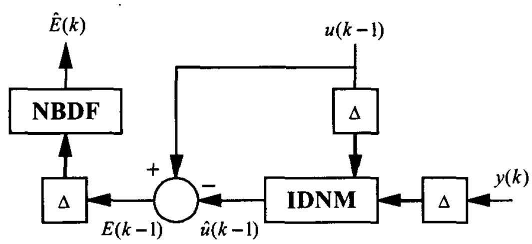

Fig. 3. The Error Reduction Neuro Model.

B. Error Reduction Neuro Model

For better accuracy, it is required to consider a wide range of operating conditions and disturbances in designing the controller. In real application, however, it is impractical to consider all operating conditions and disturbances. Therefore, when the system is operating under conditions and disturbances that the IDNM has never learned, error between the IDNM and real inverse dynamics inevitably exists even though the IDNM learning may have been completed for previously given data set.

Suppose the IDNM is trained for one operating condition and then it is to be used for some other operating condition without retuning its parameters. In Fig. 2, the output of IDNM $\hat{u}_r(k)$ is no longer the same as $\hat{u}(k)$ in Fig. 1. In other words, the output of IDNM, $\hat{u}_r(k)$, in Fig. 2 will not be the same as the desired $u(k)$ for a different operating condition since $y_r(k+1)$ is not the same as $y(k+1)$ used in the training mode of IDNM. Therefore, there exists a modeling error $E(k)$ at time k. The system input considering the modeling error becomes as following:

$$u(k) = \hat{u}_r(k) + \hat{E}(k), \tag{5}$$

where $\hat{E}(k)$ is the estimate of the modeling error $E(k)$.

The error can be estimated by extrapolating the previous error using the Newton-backward-difference formula (NBDF) [27] as follows:

$$\hat{E}(k) = \sum_{r=1}^{l} (-1)^r \binom{r}{l+1} \Delta^r E(k-1), \tag{6}$$

where

$$E(k-1) = u(k-1) - \hat{u}(k-1),$$
$$\Delta E(k-1) = E(k-1) - E(k-2).$$

Here, $E(k-1)$ is the modeling error at time $k-1$, Δ^r is the backward difference operator defined in (3), l is the extrapolation order, and the binomial-coefficient notation is defined as

$$\binom{s}{k} = \frac{s(s-1)\cdots(s-k+1)}{k!}.$$

Fig. 3 shows the design of the Error Reduction Neuro Model (ERNM) using the IDNM, and NBDF.

In Fig. 2, IDNM is fixed after training and it acts like a feedforward controller. The compensation of the modeling error and disturbances are taken care of by the ERNM, which acts like a feedback controller.

III. SIMULATION TEST

The proposed Inverse Dynamic Neuro Controller is tested in a one-machine and infinite-bus power system (Appendix).

In the training of IDNM, Fig. 4 shows the architecture of IDNM, where activation function f_j is the *tanh* and F_0 is 1. Since the main purpose in the designing of controller is to control a system in simple way, the five inputs are used in the training of IDNM such as $\{y_r(k+1), y(k), \Delta y(k), u(k-1), \Delta u(k-1)\}$.

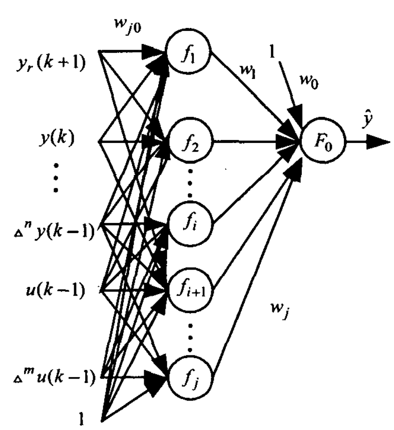

Fig. 4. The architecture of IDNM.

Fig. 5 shows the collected input-output data which are used to train an IDNM in training mode. Fig. 6 shows the validation of Inverse Dynamic Neuro Model after it is trained by Levenberg-Marquardt method. The system input or controller output is validated in Fig. 6. The sampling time of 0.01 sec. is used to train an IDNM.

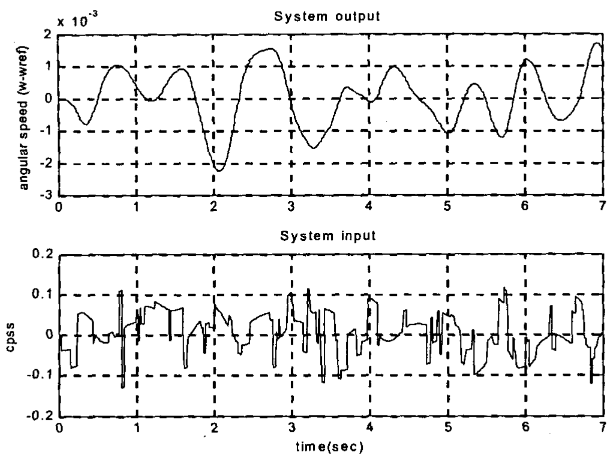

Fig. 5. The system input and output data in closed-loop.

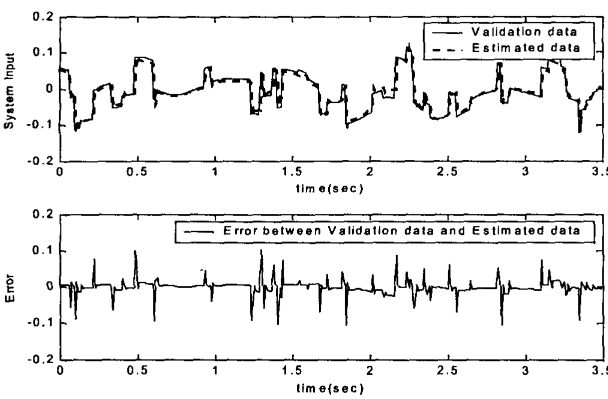

Fig. 6. The validation between estimation data and validation data.

The input-output data set used to train IDNM is $\{y_r(k+1), y(k), \Delta y(k), u(k-1), \Delta u(k-1)\}$ and the fitness can be calculated as following:

$$fit = 100 \times \frac{1 - norm(y - \hat{y})}{norm(y - mean(y))}, \qquad (7)$$

where y is the real data or validation data and $\hat{y}$ is the estimated data in validation mode. The *fitness* is 59.6532%. The trained IDNM is not trained again for other disturbances.

Case 1: A Single Disturbance

Generator model and operating points are given in Appendix. The torque angle deviation δ is applied as a disturbance. The initial torque angle δ is increased by 0.7 [p.u.]. In Fig. 7, the IDNC is compared with a conventional PSS (CPSS) and IDNM in control mode without ERNM. Moreover, In Fig. 8, the control inputs $\hat{u}_r(k), \hat{E}(k)$, and $u(k)$ in (5) are shown.

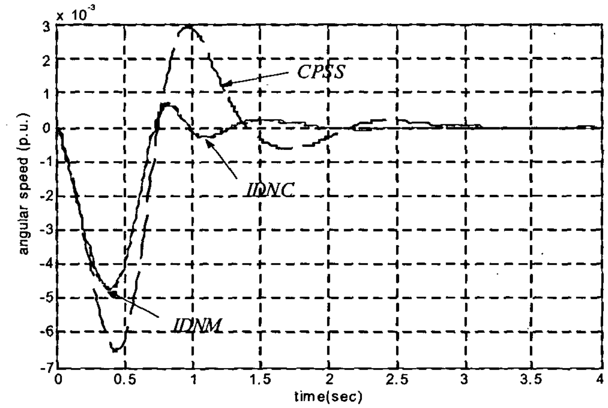

Fig. 7. The comparison among the CPSS, IDNM, and IDNC.

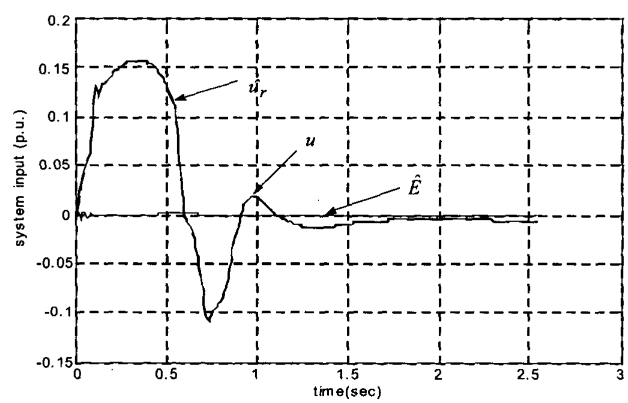

Fig. 8. The control inputs of $\hat{u}_r(k), \hat{E}(k)$, and $u(k)$.

It is shown that the IDNC performs better than the CPSS, although it was not trained with the new disturbance.

Case 2: Multiple Disturbance

In this case, there are two disturbances: one is the torque angle deviation δ which is increased by 0.5 [p.u.] from initial operating point, 1.2876, from 0 sec. to 7 sec. After that,

0-7803-7278-6/02/$10.00 ©2002 IEEE

the initial torque T_m, 1.1, is decreased by 20%. Fig. 9 shows the comparison of the output among IDNC, IDNM without ERNM, and CPSS. The IDNM is the same in the used in the previous case1, and the sampling time is 0.01 sec. It is also shown that the IDNC performance is much better than the CPSS even if the operating conditions of δ and T_m are changed.

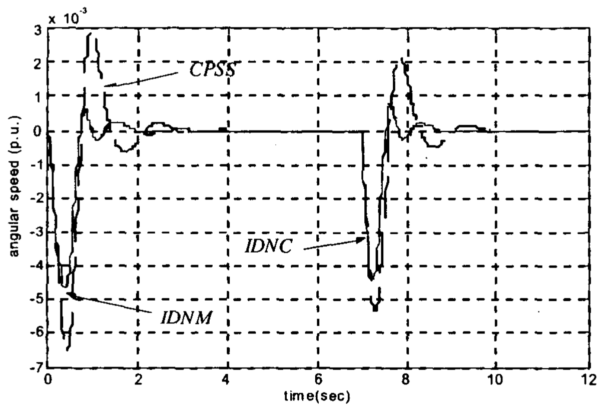

Fig. 9. The comparison among CPSS, IDNM, and IDNC.

Case 3: Network Disturbance

In this case, a three-phase fault is created in near infinite bus. The simulation is as following:

Fault occurred at 0 sec., lasted for 0.05 sec., faulted lines are removed until 0.25 sec., and then reclosed.

The effect of a line removal is simulated by changing the line impedance Z to $Z \times 2$. Figs. 8 shows the comparison of the system output among IDNC, IDNM, and CPSS. The same IDNM found in Case 1 is used and the sampling time is 0.01 sec. In the network disturbance, the IDNC also performs much better than the CPSS even if it was not trained with this type of disturbance.

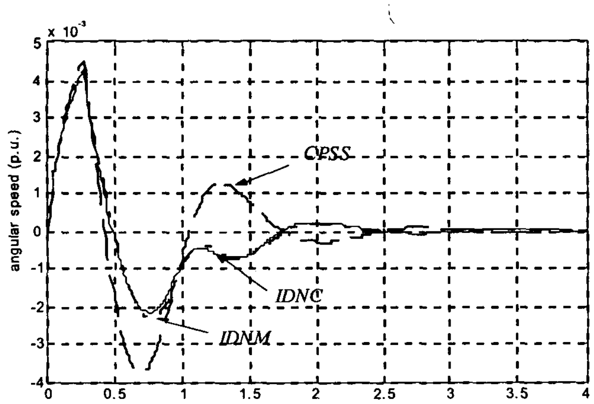

Fig. 10. The comparison among CPSS, IDNM, and IDNC.

IV. CONCLUSION

In this paper, the inverse dynamic neuro controller (IDNC) is presented. The inverse dynamic neuro model (IDNM) is derived using the free-model concept. The input-output data set are collected in closed-loop since the system is unstable initially. Pseudo-Random Binary Signal is applied in the closed-loop system.

There are some advantages of using the proposed IDNM. One is that a trained artificial neural network is used for IDNM, and it is not required to retrain it for other disturbances. The other is that it has Error Reduction Neuro Model (ERNM) to reduce the modeling error for various operating conditions. Once the IDNM is trained in a training mode, it is applied in control mode without requiring retraining, while, in other inverse dynamic neural network (IDNN) approaches, well trained IDNNs are required. In other words, it is required to retrain neural networks when other disturbances are introduced in the system.

The IDNC was implemented in a one-machine infinite-bus power system. It was tested in various operating conditions and compared with the conventional PSS. In all cases, the IDNC out-performed the conventional PSS and thus demonstrated the usefulness of IDNM based controller design.

V. APPENDIX

One-Machine Infinite-Bus (OMIB) Power System [28]

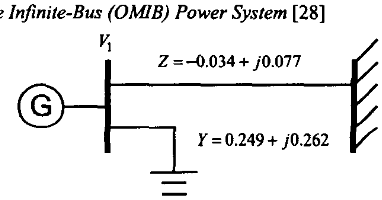

Fig. A.1. One-machine infinite-bus model.

Machine Models (generator, turbine, governor and exciter):

$$\frac{d\delta_i}{dt} = \omega_b(\omega_i - \omega_0)$$

$$M_i \frac{d\omega_i}{dt} = (T_{Mi} - P_{ei} - D_i(\omega_i - \omega_0))$$

$$T'_{doi} \frac{dE'_{qi}}{dt} = (E_{fdi} - E'_{qi} - (X_{di} - X'_{di}) I_{d_i})$$

$$T_{Ai} \frac{dE_{fdi}}{dt} = (K_{Ai}(V_{refi} - V_i + C_{pssi}) - E_{fdi})$$

$$T_{Ci} \frac{dT_{Mi}}{dt} = (F_{hpi} U_{gi} - T_{Mi} + T_{Mri})$$

$$T_{gi} \frac{dU_{gi}}{dt} = (K_{gi}(\omega_{refi} - \omega_i) - U_{gi})$$

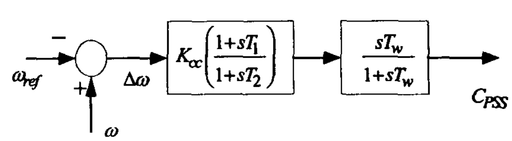

Fig. A.2. Conventional power system stabilizer model.

Table A.1. The Parameters of PSS

T_1	T_2	T_w	K_{CC}
0.685	0.1	3	7.091

Table A.2. Exciter Data

T_A (sec)	K_A (p.u)
0.05	25

Table A.3. Generator, Turbine, and Governor data (e'_q Model)

M	T'_{do}	x_d	x_q	x'_d	T_c	F_{hp}	K_g	T_g
9.26	7.76	0.973	0.55	0.19	0.1	1	10	0.1

Table A.4. Operating points

V_{int}	δ	E_{fd}	T_m	E'_q	v_d	v_q
1.05	1.2876	1.4219	1.1	1.0197	0.5022	0.9221

VI. ACKNOWLEDEMENT

This work is supported in part by the National Science Foundation under Grants, Free-Model Based Intelligent Control of Power Plants and Power Systems (ECS-9705105) and Development of Power System Intelligent Coordinated Control (INT-9605028)

VII. REFERENCES

[1] F. P. deMello and C.A.Concordia, "Concept of Synchronous Machine Stability as Affected by Excitation Control," *IEEE Trans. on PAS*, Vol. PAS-103, pp. 316-319, 1969.

[2] F. P. deMello, P. J. Nolan, T. F. Laskowoski, and J. M. Undrill, "Coordinate application of stabilizers in multimachine power system," *IEEE Trans. on PAS*, pp. 892-901, May/June 1980.

[3] Doi, S. Abe, "Coordinated Synthesis of Power System Stabilizers in Multimachine Power Systems," *IEEE Trans. on PAS*, Vol 103, pp. 1473-1479, June 1984.

[4] T. L. Hwang, T. Y. Hwang, and W. T. Yang, "Two-level Optimal Output Feedback Stabilizer Design," *IEEE Trans. on PWRS*, Vol. 6, No.3, pp. 1042-1047, August 1991.

[5] M. R. Khaldi, A. K. Sarkar, K. Y. Lee, and Y. M. Park, "The Model Performance Measure for Parameter Optimization of Power System Stabilizers," *IEEE Trans. on Energy Conversion*, Vol. 8, No.4, pp. 660-666, December 1993.

[6] K. T. Law, D. J. Hill, and N. R. Godfrey, "Robust Controller Structure for Coordinate Power System voltage Regulator and Stabilizer Design," *IEEE Trans. on Control System Technology*, Vol. 2, No. 3, pp. 220-232, September 1994.

[7] A. Ghosh, G. Ledwich, O. P. Malik, and G. S. Hope, "Power System Stabilizer Based on Adaptive Control Techniques," *IEEE Trans. on PAS*, Vol. 103, pp. 1983-1989, August 1984.

[8] S. J. Chang, Y. S. Chow, O. P. Malik, and G. S. Hope, "An Adaptive Synchronous Machine Stabilizer," *IEEE Trans. on PWRS*, Vol. 1, pp. 101-109, August 1986.

[9] D. A. Pierre, "A Perspective on Adaptive Control of Power Systems," *IEEE Trans. on PWRS*, Vol. 2, pp. 387-396, May 1987.

[10] W. Gu and K. E. Bollinger, "A Self-tuning Power System Stabilizer for Wide-range Synchronous Generator Operation," *IEEE Trans. on PWRS*, Vol. 4, No. 2, pp. 1191-1199, August 1989.

[11] J. Y. Fan, T. H. Ortmeyer, and R. Mukundan, "Power System Stability Improvement with Multivariable Self-tuning Control," *IEEE Trans. on PWRS*, Vol. 5, No. 5, pp. 227-234, February 1990.

[12] A. A. Ghandakly and A. M. Farhoud, "A Parametrically Optimized Self-Tuning Regulator for Power System Stabilizer," *IEEE Trans. on PWRS*, Vol. 7, pp. 1245-1250, August 1992.

[13] O. P. Malik and C. Mao, "An Adaptive Optimal Controller and Its Application to an Electric Generating Unit," *International Journal of Electric Power & Energy Generating Unit*," Vol. 15, pp. 169-178, 1993.

[14] Y. M. Park, S. H. Hyun, and J. H. Lee, "A synchronous Generator Stabilizer Design Using Neuro Inverse Controller and Error Reduction Network," IEEE Trans. on Power System, Vol. 11, No. 4, pp. 1969 –1975, November 1996.

[15] Y. Zhang, O. P. Malik, G. S. Hope, and G. P. Chen, "Application of an Inverse Input/Output Mapped ANN as A Power System Stabilizer," IEEE Trans. on Energy Conversion, Vol. 9, No. 3, pp. 433 –441, September 1994.

[16] Y.-M. Park, M.-S. Choi, and K. Y. Lee, "A Neural Network-Based Power System Stabilizer Using Power Flow Characteristics," IEEE Trans. on Energy Conversion, Vol. 11, No.2, pp. 435-441, June 1996.

[17] T. Poggio and F. Girosi, "Networks for Approximation and Learning, *" Proceeding of IEEE*, pp. 1481-1497, September, 1990.

[18] D. Nguyen and B. Widrow, "The Truck Backer-Upper An Example of Self-Learning in Neural Networks," *IEEE Control System Magazine*, pp. 18-23, 1990.

[19] Y. liguni and H. Sakai, "A Nonlinear Regulator Design in the Presence of System Uncertainties Using Multilayered Neural Networks," *IEEE Trans. on Neural Networks*, Vol. 3, No.4, pp. 410-417, July 1991.

[20] Y.Y. Hsu and C. R. Chen, "Tuning of Power System Stabilizers Using an ANN," IEEE Trans. on Energy Conversion, Vol. 6, pp. 612-619, 1991.

[21] Q. H. Wu, B. W. Hogg, and G. W. Irwin, "A Neural Network Regulator for Turbogenerators," IEEE Trans. on Neural Networks, Vol. 3, No. 1, January 1992.

[22] D. C. Kennedy and V. H. Quintana, "Neural Network Regulators for Sysnchronous Machines," ACC, Vol. 5, pp. 131-136, 1993.

[23] K. Y. Lee, H. -S. Ko, H. -C. Kim, J. -H. Lee, and Y. -M. Park, "A free model based intelligent controller design and its applications to power system stabilization," *IEEE Power Engineering Society Summer Meeting*, Vol. 4, pp. 1985-1989, 2000.

[24] K. Y. Lee, H. -S. Ko, H. -C. Kim, J. -H. Lee, and Y. -M. Park, "A free model based power system stabilization," *IEEE Power Engineering Society Winter Meeting*, in proceeding, pp. 1191 –1196, 2001.

[25] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, New Jersey, 1999.

[26] L. Ljung, *System Identification: Theory for the user*, Prentice-Hall, New Jersey, 1999.

[27] R. L. Burden and J. D. Faires, *Numerical Analysis*, PWS-KENT, 1989.

[28] P. Kunder, *Power System Stability and Control*, McGraw-Hill, New York, 1994.

VIII. BIOGRAPHIES

Kwang Y. Lee received B.S. degree in Electrical Engineering from Seoul National University, Korea, in 1964, M.S. degree in Electrical Engineering from North Dakota State, Fargo in 1968, and Ph.D. degree in systems science from Michigan State, East Lansing in 1971. He has been with Michigan State, Oregon State, Univ. of Houston, and the Pennsylvania State University, where he is a Professor of Electrical Engineering. His interests are power systems operation and planning, and intelligent control of power plants and power systems. He is a Fellow of IEEE and Associate Editor of IEEE Transaction on Neural Networks.

Hee-Sang Ko received B.S. degree in Electrical Engineering from Cheju National University, Korea, in 1995 and M.S. degree in Electrical Engineering from the Pennsylvania State University in 2000. He currently joined to Ph.D. program in Electrical and Computer Engineering at the University of British Columbia in 2001. His research interest is system identification, optimal and intelligent control for control and power systems.