

01 Jan 2007

Adaptive Neural Network Based Stabilizing Controller Design for Single Machine Infinite Bus Power Systems

Wenxin Liu

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

Donald C. Wunsch
Missouri University of Science and Technology, dwunsch@mst.edu

David A. Cartes

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/engman_syseng_facwork/202

Follow this and additional works at: https://scholarsmine.mst.edu/engman_syseng_facwork



Part of the [Computer Sciences Commons](#), [Electrical and Computer Engineering Commons](#), and the [Systems Engineering Commons](#)

Recommended Citation

W. Liu et al., "Adaptive Neural Network Based Stabilizing Controller Design for Single Machine Infinite Bus Power Systems," *DCDIS A Supplement, Advances in Neural Networks*, vol. 14, no. S1, pp. 494-502, Watam Press, Jan 2007.

This Article - Journal is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Engineering Management and Systems Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

Adaptive Neural Network Based Stabilizing Controller Design for Single Machine Infinite Bus Power Systems

Wenxin Liu¹, Jagannathan Sarangapani², Ganesh K Venayagamoorthy², Donald C Wunsch II²
Mariesa L Crow², and David A. Cartes¹

Abstract—Power system stabilizers are widely used to generate supplementary control signals for the excitation system in order to damp out the low frequency oscillations. In power system control literature, the performances of the proposed controllers were mostly demonstrated using simulation results without any rigorous stability analysis. This paper proposes a stabilizing neural network (NN) controller based on a sixth order single machine infinite bus power system model. The NN is used to approximate the complex nonlinear dynamics of power system. Unlike the other indirect adaptive NN control schemes, there is no offline training process and the NN can be directly used online and learn through time. Magnitude constraint of the activators is modeled as saturation nonlinearities and is included in the Lyapunov stability analysis. The new NN controller design is compared with conventional power system stabilizers (CPSS) whose parameters are fine tuned by particle swarm optimization (PSO). Simulation results demonstrate that the proposed NN controller design can successfully damp out power system oscillations. The control algorithms of this paper can also be applied to other similar nonlinear control problems.

Key words—stabilizing control, power systems, neural networks, and particle swarm optimization.

I. INTRODUCTION

GENERATORS in power systems are equipped with voltage regulators to control the terminal voltage. It is known that the voltage regulator has a detrimental impact upon the dynamic stability of the power systems. During a change in operating condition, oscillations of small magnitude and low frequency often persist for long periods of time and in some cases even present limitations on power transfer capability. The issue of power system stabilizing control has received a great deal of attention since 1960's. Power system stabilizers (PSSs) are designed to generate supplementary control signal in the excitation system to damp out low frequency oscillations [1].

Previous works on stabilizing control are based on linearized models. For example, the widely used conventional power system stabilizer (CPSS) is designed using the theory of phase compensation and introduced as a

lead-lag compensator. To have the CPSS provide good damping over wide operating conditions, its parameters need to be fine tuned in response to all kinds of oscillations, which is a time-consuming job. To simplify this process, intelligent optimization algorithms (such as simulated annealing [2], genetic algorithm [3], and tabu search [4]) are applied to offline determining the "optimal parameters" of CPSS by optimizing an eigenvalue based cost function. In the past decade, fuzzy logic and NN were applied online to adjust the parameters of CPSS based on the knowledge gained by offline training. Since power systems are highly nonlinear systems, with configurations and parameters changing with time, the designs based on linearized model cannot guarantee their performances in practical operating environment. Thus, adaptive controller designs based on nonlinear models are required for the power system [5].

In recent years, stabilizing control schemes using NN and fuzzy logic have been proposed. Most of the papers only demonstrated the effectiveness of the controller design via simulation but the stability analyses were not carried out. The reason for the lack of stability analysis is partly due to the complexity of the power systems. Anyway, industry will still prefer a controller designs that are designed through rigorous stability analysis. To address this problem, certain controller designs have appeared based on feedback linearization or differential geometric theory [6-8]. Some of these papers are based on simplified models, which overlook the complex dynamics of practical system. Furthermore, feedback linearization requires the system model to be known exactly, imprecise model will greatly degrade the performance. While practical power system models are very difficult to be known exactly, this requirement can seldom be satisfied. Since the stabilizing and voltage controllers are all implemented in the excitation system, there is the possibility for these two kinds of controls to interact with each other. But few papers show the performance of voltage control under stabilizing controls

The paper tries to overcome the above mentioned problems by designing a stable adaptive neural network controller. The controller design is based on a sixth-order single machine infinite bus power system. Since the complex nonlinearity can be approximated using a neural network, the requirement on precise model is released. The weight updating rule of the NN is an unsupervised version of backpropagation through time and the initial weights of NN can be directly set to zero to avoid the time consuming offline training process. Since practical operating conditions

¹Wenxin Liu and David A. Cartes are with the Center for Advanced Power Systems, Florida State University, Tallahassee, 32310, USA (e-mails: wliu and dave@caps.fsu.edu)

²Jagannathan Sarangapani, Ganesh K. Venayagamoorthy, Donald C. Wunsch II, and Mariesa L. Crow are with the Department of Electrical and Computer Engineering, University of Missouri - Rolla, MO 65401 USA, (e-mails: {sarangap, ganeshv, dwunsch, and crow}@umr.edu)

require the magnitude of control signal to be within certain limit, this paper also investigate the stability of the closed loop system when the magnitude of the ideal control signal overshoot the limit.

To test the performance of the proposed NN controller, it is compared with an “optimal” CPSS under different operating conditions. Here, “optimal” means that the parameters of the CPSS are optimized by particle swarm optimization (PSO). Simulations with different kinds of operating conditions show that the proposed NN controller can perform better than the optimal CPSS although the design parameters remain the same.

The paper is organized as follows. Section II presents a brief background on universal approximation property of neural networks and stability of nonlinear system. Section III introduces the single machine infinite bus power system model. The neural network controller design is described in section IV. The determination of the optimal CPSS parameters using PSO is explained in section V. Simulation results are provided in section VI, and finally, the conclusion in Section VII.

II. BACKGROUND

The following mathematical notions are required for system approximation using NNs and system stability in the design of an adaptive NN controller.

A. Approximation Property of NN

The commonly used property of neural networks for control is its function approximation and adaptability capacities [9]. Let $f(x)$ be a smooth function from $R^n \rightarrow R^m$, then it can be shown that, as long as x is restricted to a compact set $S \in R^n$, for some sufficiently large number of hidden-layer neurons, there exist weights and thresholds such that

$$f(x) = W^T \varphi(x) + \varepsilon(x) \quad (1)$$

where x is the input vector, $\varphi(\cdot)$ is the activation function, W is the weight matrix of the output layer and $\varepsilon(x)$ is the approximation error. Equation (1) means a neural network can approximate any continuous function in a compact set. In fact, for any choice of a positive number ε_N , one can find a neural network such that $\varepsilon(x) \leq \varepsilon_N$ for all $x \in S$. For suitable function approximation, $\varphi(x)$ must form a basis [10].

For two layer neural networks, $\varphi(x)$ is defined as $\varphi(x) = \sigma(V^T x)$, where V is the weight matrix of the first layer and $\sigma(x)$ is the sigmoid function. If V is fixed, then the only design parameter in the NN is W and this NN becomes a simple version of function link network (one layer neural network) which is easier to train. It has been shown in [11] that $\varphi(x)$ can form a basis if V is chosen randomly. The larger the number of the hidden layer neurons N_h , the smaller the approximation error $\varepsilon(x)$. Baron shows that the neural network approximation error $\varepsilon(x)$ for one-layer NN is fundamentally bounded by a term of the order $(1/n)^{2/d}$, where n is the number of fixed basis functions and d is the

dimension of the input to the NN [9]. The structure of the function link neural network is shown in Fig. 1.

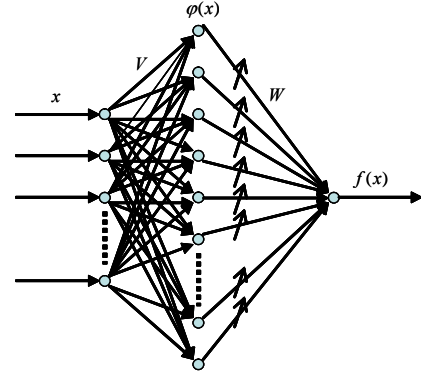


Fig. 1. Structure of function link neural network.

B. Stability of Systems

In the design of the controller, the following stability notion is needed. Consider the nonlinear system given by

$$\begin{aligned} \dot{x} &= f(x, u) \\ y &= h(x) \end{aligned} \quad (2)$$

where $x(t)$ is a state vector, $u(t)$ is the input vector and $y(t)$ is the output vector [12]. The solution to (2) is uniformly ultimately bounded (UUB) if for any U , a compact subset of R^n , and all $x(t_0) = x_0 \in U$ there exists an $\varepsilon > 0$ and a number $T(\varepsilon, x_0)$ such that $\|x(t)\| < \varepsilon$ for all $t \geq t_0 + T$.

III. MODEL OF SINGLE MACHINE POWER SYSTEM

Fig. 2 shows the configuration of the single machine infinite bus power system. The system consists of a synchronous generator, an exciter, an automatic voltage regulator (AVR) and a transmission line which connects the generator to the infinite bus.

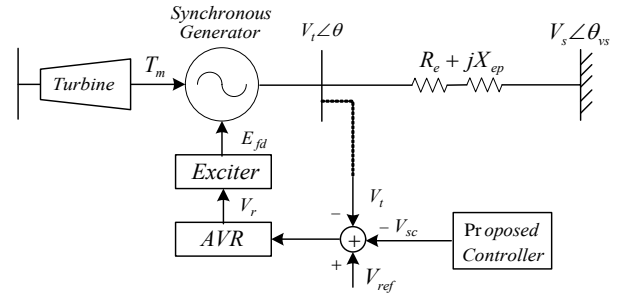


Fig. 2: Single machine infinite bus power system.

In above figure, T_m is the mechanical torque, E_{fd} is the field voltage, V_r is the output of the automatic voltage regulator (AVR), $V_t \angle \theta$ is the terminal voltage at the generator bus, V_{ref} is the reference signal applied to the AVR, R_e and X_{ep} form the impedance of the transmission line between the generator and infinite bus, $V_s \angle \theta_{vs}$ is the infinite bus voltage, and V_{pss} is the stabilizing control signal.

The dynamics of the single machine power system are expressed using a two axis model [13] as in (3). The first four

equations represent the dynamics of the synchronous generator, the fifth and sixth equations represent the dynamics of the exciter and AVR respectively.

$$\left\{ \begin{array}{l} \frac{d\delta}{dt} = \omega - \omega_s \\ \frac{2H}{\omega_s} \frac{d\omega}{dt} = T_m - E_d' I_d - E_q' I_q - (X_q' - X_d') I_d I_q \\ T_{q0}' \frac{dE_d'}{dt} = -E_d' + (X_q - X_q') I_q \\ T_{d0}' \frac{dE_q'}{dt} = -E_q' - (X_d - X_d') I_d + E_{fd} \\ T_e \frac{dE_{fd}}{dt} = -E_{fd} + V_r \\ T_a \frac{dV_r}{dt} = -V_r + K_a (V_{ref} - V_t + V_{pss}) \end{array} \right. \quad (3)$$

where, I_d , I_q and V_t are subjected to the constraints of (4) and (5) respectively:

$$\left\{ \begin{array}{l} 0 = R_e I_d - (X_q' + X_{ep}) I_q - E_d' + V_s \sin(\delta - \theta_{vs}) \\ 0 = R_e I_q + (X_d' + X_{ep}) I_d - E_q' + V_s \cos(\delta - \theta_{vs}) \end{array} \right. \quad (4)$$

$$V_t = \sqrt{V_d^2 + V_q^2} \quad (5)$$

with

$$\left\{ \begin{array}{l} V_d = R_e I_d - X_{ep} I_q + V_s \sin(\delta - \theta_{vs}) \\ V_q = R_e I_q + X_{ep} I_d + V_s \cos(\delta - \theta_{vs}) \end{array} \right. \quad (6)$$

In above equations, besides the variables appeared in Fig. 2, δ is the rotor angle in radian, ω is the speed in radian per second, E_d' and E_q' are the internal transient voltage in per units. Table I shows the value of the parameters in above model.

TABLE I
SYSTEM PARAMETERS

$H=3.01$	$X_d=1.3125$	$X_q=1.2578$	$X_d'=0.1813$
$X_q'=0.25$	$T_{d0}=5.89$	$T_{q0}=0.6$	$T_e=0.314$
$K_a=20$	$T_a=0.2$	$R_e=0.025$	$X_{ep}=0.085$

The control objective is to stabilize the speed ω to $\omega_s=2\pi f$ for different operating conditions and disturbances. The system can be linearized via input-output feedback [14]. Since the control objective is speed ω , this is a single input single output control problem. Define the speed deviation as $e = \Delta\omega = \omega - \omega_s$, then the control objective is to regulate e to zero. In order to get the expression of the speed deviation with respect to the control signal, we need to differentiate e several times until the control signal appears. The process is shown as below:

$$\begin{aligned} \dot{e} = \dot{\omega} &= k_6 E_d'^2 + k_7 E_q'^2 + k_8 E_d' E_q' + k_9 E_d' \sin \delta + k_{10} E_d' \cos \delta + k_{16} \\ &+ k_{11} E_q' \sin \delta + k_{12} E_q' \cos \delta + k_{13} \sin^2 \delta + k_{14} \cos^2 \delta + k_{15} \sin \delta \cos \delta \end{aligned} \quad (7)$$

$$\begin{aligned} \ddot{e} = \ddot{\omega} &= 2k_6 E_d' \dot{E}_d' + 2k_7 E_q' \dot{E}_q' + k_8 (\dot{E}_d' E_q' + E_d' \dot{E}_q') \\ &+ k_9 (\dot{E}_d' \sin \delta + E_d' \cos \delta \dot{\delta}) + k_{10} (\dot{E}_d' \cos \delta - E_d' \sin \delta \dot{\delta}) \\ &+ k_{11} (\dot{E}_q' \sin \delta + E_q' \cos \delta \dot{\delta}) + k_{12} (\dot{E}_q' \cos \delta - E_q' \sin \delta \dot{\delta}) \\ &+ (k_{13} - k_{14}) \dot{\delta} \sin 2\delta + k_{15} \dot{\delta} \cos 2\delta \end{aligned} \quad (8)$$

$$\begin{aligned} \ddot{e} = \ddot{\omega} &= 2k_6 \dot{E}_d'^2 + 2k_7 \dot{E}_q'^2 + 2k_8 \dot{E}_d' \dot{E}_q' \\ &+ k_9 (\dot{E}_d' \cos \delta \dot{\delta} + \dot{E}_d' \cos \delta \dot{\delta} - E_d' \sin \delta \dot{\delta}^2 + E_d' \cos \delta \dot{\omega}) \\ &- k_{10} (\dot{E}_d' \sin \delta \dot{\delta} + \dot{E}_d' \sin \delta \dot{\delta} + E_d' \cos \delta \dot{\delta}^2 + E_d' \sin \delta \dot{\omega}) \\ &+ k_{11} (\dot{E}_q' \cos \delta \dot{\delta} + \dot{E}_q' \cos \delta \dot{\delta} - E_q' \sin \delta \dot{\delta}^2 + E_q' \cos \delta \dot{\omega}) \\ &- k_{12} (\dot{E}_q' \sin \delta \dot{\delta} + \dot{E}_q' \sin \delta \dot{\delta} + E_q' \cos \delta \dot{\delta}^2 + E_q' \sin \delta \dot{\omega}) \\ &+ (k_{13} - k_{14}) \dot{\omega} \sin(2\delta) + 2(k_{13} - k_{14}) \cos(2\delta) \dot{\delta}^2 \\ &+ k_{15} \dot{\omega} \cos(2\delta) - 2k_{15} \sin(2\delta) \dot{\delta}^2 + 2k_6 E_d' \ddot{E}_d' + k_8 \ddot{E}_d' E_q' + k_9 \ddot{E}_d' \sin \delta \\ &+ k_{10} \ddot{E}_d' \cos \delta + 2k_7 E_q' \ddot{E}_q' + k_8 E_d' \ddot{E}_q' + k_{11} \ddot{E}_q' \sin \delta + k_{12} \ddot{E}_q' \cos \delta \end{aligned} \quad (9)$$

with,

$$\begin{aligned} \ddot{E}_d' &= k_{17} \dot{E}_d' + k_{18} \dot{E}_q' + k_{19} \cos \delta \dot{\delta} - k_{20} \sin \delta \dot{\delta} \\ \ddot{E}_q' &= k_{21} \dot{E}_d' + k_{22} \dot{E}_q' + k_{23} \cos \delta \dot{\delta} - k_{24} \sin \delta \dot{\delta} + k_{25} \dot{E}_{fd} \end{aligned} \quad (10)$$

$$e^{(5)} = \omega^{(4)} = f(\bar{x}) + g(\bar{x}) V_{pss} \quad (11)$$

where the definition of $k_1 \sim k_{25}$ can be found in the Appendix, $f(\bar{x})$ and $g(\bar{x})$ are both functions of system states and their different orders of derivatives, and $g(x)$ is defined in (12).

$$g(\bar{x}) = k_{25} k_{26} (2k_7 E_q' + k_8 E_d' + k_{11} \sin \delta + k_{12} \cos \delta) \quad (12)$$

where \bar{x} stands for the original state variables δ , ω , E_d' , E_q' , E_{fd} , and V_r .

The expression of $f(\bar{x})$ is not given here because both $f(\bar{x})$ and $g(\bar{x})$ will be approximated using one NN in the following controller design, but the expression of $g(\bar{x})$ is necessary because it has to satisfy the two assumptions in Section IV.

Since ω is chosen as the control objective and the control signal V_{pss} appears at the fourth order of ω . The transformed control system model is of fourth-order rather than the original sixth-order. Since the uncontrolled state δ satisfies $\dot{\delta} = \omega - \omega_s$, δ is bounded when ω is stabilized to ω_s . So even if δ is not considered in the transformed control system model, the system is still stable.

If the new state variables are defined as

$$\bar{e} = [e_1, e_2, e_3, e_4]^T = [\Delta\omega, \dot{\omega}, \ddot{\omega}, \ddot{\omega}]^T \quad (13)$$

Then the error dynamics of the system can be expressed into the *Brunovsky Canonical Form* as

$$\begin{aligned}
\dot{e}_1 &= e_2 \\
\dot{e}_2 &= e_3 \\
\dot{e}_3 &= e_4 \\
\dot{e}_4 &= f(\bar{x}) + g(\bar{x})u + d
\end{aligned} \tag{14}$$

where u stands for the control signal and d stands for a bounded disturbance with $|d| \leq d_M$, \bar{x} stands for $\delta, \omega, E_q', E_{fd}, V_r$.

IV. STABILIZING CONTROLLER DESIGN

A. Assumptions

Assumption 1: $g(\bar{x})$ is bounded and the sign is known to be either positive or negative. Without losing generality, $g(\bar{x}) > 0$ is assumed. Furthermore, there exists two positive constants g_m and g_M , such that $g_M > g(\bar{x}) > g_m > 0$.

Assumption 2: The derivative of $g(\bar{x})$ is bounded, which means there exist a positive constant g_{dM} , such that $|\dot{g}(\bar{x})| \leq g_{dM}$.

Remark: Based on the definitions of (12) and the constants in the Appendix, the above two assumptions hold for this single machine power system. These assumptions are future investigated by simulate the system under different kinds of operating conditions.

B. Neural Network Controller Design

Defining the filtered error r as

$$r = [\Lambda^T \quad 1]\bar{e} \tag{15}$$

where $\Lambda = [\lambda_1 \quad \lambda_2 \quad \lambda_3]^T$ is an appropriately chosen coefficient vector such that $e \rightarrow 0$ as $r \rightarrow 0$, (i.e. $s^3 + \lambda_3 s^2 + \lambda_2 s + \lambda_1$ is Hurwitz).

Differentiating (15) and substituting (14) to get

$$\dot{r} = [0 \quad \Lambda^T]\bar{e} + f(\bar{x}) + g(\bar{x})u + d \tag{16}$$

According to the theory of feedback linearization [15], the desired control signal can be chosen as

$$u^* = -K_v r - \frac{1}{g(\bar{x})}(f(\bar{x}) + [0 \quad \Lambda^T]\bar{e}) \tag{17}$$

where K_v is a selected positive constant.

According to the NN approximation theory, there exists a NN that can approximate the second term of (17) within designated precision, such that

$$W^T \Phi(\bar{x}, \bar{e}) = -\frac{1}{g(\bar{x})}[f(\bar{x}) + [0 \quad \Lambda^T]\bar{e}] + \varepsilon \tag{18}$$

where W is a constant weight, ε is the approximation error that is bounded by $|\varepsilon| \leq \varepsilon_N$.

Defining the ideal control signal u (without magnitude constraint) as

$$u = -K_v r + \hat{W}^T \Phi(\bar{x}, \bar{e}) \tag{19}$$

where the design parameter K_v is a positive constant.

The actual stabilizing control signal V_{pss} applied to the power system is given by:

$$V_{pss} = \begin{cases} u & \text{while } |u| \leq u_{\max} \\ u_{\max} \text{sign}(u) & \text{while } |u| > u_{\max} \end{cases} \tag{20}$$

where u_{\max} is the maximum allowed control signal magnitude. The structure of the controller is shown in Fig. 3. It has multi-loop structure with an inner nonlinear adaptive NN loop used to estimate the nonlinear dynamics of the single machine power system and an outer PD tracking loop. The next step is to determine an appropriate weight updating rule so that the closed-loop stability of the control system can be guaranteed. The performance of the proposed adaptive neural network controller is described by theorem I.

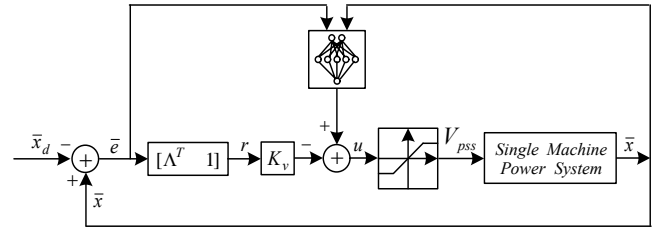


Fig. 3. Neural network feedback linearizing controller

Assume W is bounded by W_{\max} , that is, $\|W\| \leq W_{\max}$.

Rearranging (18) as an expression of $f(\bar{x})$ and substitute which into (16) to get

$$\dot{r} = -K_v g(\bar{x})r - g(\bar{x})\tilde{W}^T \Phi(\bar{x}, \bar{e}) + g(\bar{x})\varepsilon + d \tag{21}$$

where \tilde{W} is the error in weight approximation defined as:

$$\tilde{W} = W - \hat{W} \tag{22}$$

Theorem 1: Assume the unknown disturbance d and the weight approximation error ε are bounded by some known constants such that $|d| \leq d_N$ and $|\varepsilon| \leq \varepsilon_N$ respectively. Selecting the weight updating rule as

$$\dot{\hat{W}} = -\Gamma r \Phi(\bar{x}, \bar{e}) - \alpha \Gamma \|r\| \hat{W} \tag{23}$$

where $\alpha, \Gamma > 0$ are the adaptation gains and the K_v is given as follows

$$K_v > \frac{g_{dM}}{2g_m^2} \tag{24}$$

Then the filtered error r and the weight estimation error \tilde{W} are uniformly ultimately bounded.

Proof: The proof is given for two cases.

Case 1: $|u| \leq u_{\max}, V_{pss} = u$

▪ *Filtered Error Bound*

Selecting the Lyapunov function candidate $V \in R$ as given in [16]

$$V = \frac{r^2}{2g(\bar{x})} + \frac{1}{2} \tilde{W}^T \Gamma^{-1} \tilde{W} \quad (25)$$

Differentiating V gives

$$\dot{V} = \frac{r\dot{r}}{g(\bar{x})} - \frac{\dot{g}(\bar{x})r^2}{2g(\bar{x})^2} + \tilde{W}^T \Gamma^{-1} \dot{\tilde{W}} \quad (26)$$

Substituting the error dynamics (21) into (26) gives

$$\dot{V} = -(K_v + \frac{\dot{g}(\bar{x})}{2g(\bar{x})^2})r^2 + \tilde{W}^T [\Gamma^{-1} \dot{\tilde{W}} - r\Phi(\bar{x}, \bar{e})] + r\varepsilon + \frac{rd}{g(\bar{x})} \quad (27)$$

Substituting (23) into (27) gives

$$\dot{V} = -(K_v - \frac{g_{dM}}{2g_m^2})r^2 + \alpha|r|\tilde{W}^T(W - \tilde{W}) + r\varepsilon + \frac{rd}{g(\bar{x})} \quad (28)$$

Rewriting (28) as follows

$$\begin{aligned} \dot{V} &\leq -(K_v - \frac{g_{dM}}{2g_m^2})r^2 - \alpha|r|[\|\tilde{W}\| - \frac{W_{\max}}{2}]^2 - \frac{W_{\max}^2}{4} + |r|(\varepsilon_N + \frac{d_M}{g_m}) \\ &\leq -|r|[(K_v - \frac{g_{dM}}{2g_m^2})r] - [\frac{W_{\max}^2}{4}\alpha + \varepsilon_N + \frac{d_M}{g_m}] - \alpha|r|[\|\tilde{W}\| - \frac{W_{\max}}{2}]^2 \\ &\leq -|r|[(K_v - \frac{g_{dM}}{2g_m^2})r] - [\frac{W_{\max}^2}{4}\alpha + \varepsilon_N + \frac{d_M}{g_m}] \end{aligned} \quad (29)$$

According to (24), the bound of the filtered error can be expressed as (30). Because if (30) is not satisfied, then \dot{V} will be negative, thus, the filter will remain in the bound given in (30).

$$\|r\| \leq \frac{\frac{W_{\max}^2}{4}\alpha + \varepsilon_N + \frac{d_M}{g_m}}{K_v - \frac{g_{dM}}{2g_m^2}} \quad (30)$$

▪ *Weight Estimation Error Bound*

Choosing the Lyapunov function the same as (25), now reevaluate \dot{V}

$$\begin{aligned} \dot{V} &= -Kr^2 + \alpha|r|[-\tilde{W}^T \tilde{W} + \tilde{W}^T W] - r\varepsilon + \frac{rd}{g(x)} \\ &\leq -Kr^2 - \alpha|r|[\|\tilde{W}\|^2 - W_{\max}\|\tilde{W}\| - \frac{\varepsilon_N + \frac{d_M}{g_m}}{\alpha}] \end{aligned} \quad (31)$$

It can be seen that $\dot{V} < 0$ as long as

$\|\tilde{W}\|^2 - W_{\max}\|\tilde{W}\| - \frac{\varepsilon_N + \frac{d_M}{g_m}}{\alpha} > 0$. Similar to (30), bound of the weights estimation error is given by

$$\|\tilde{W}\| \leq \frac{W_{\max} + \sqrt{W_{\max}^2 + \frac{4(\varepsilon_N + \frac{d_M}{g_m})}{\alpha}}}{2} \quad (32)$$

Case 2: $|u| > u_{\max}, V_{pss} = u_{\max} \text{sign}(u)$

Defining $\Delta u = V_{pss} - u$, with Δu satisfying

$|\Delta u| \leq \Delta u_{\max}$. Substituting $V_{pss} = u + \Delta u$ into (16), similarly, gives

$$\dot{r} = -K_v g(\bar{x})r - g(\bar{x})\tilde{W}^T \Phi(\bar{x}, \bar{e}) + g(\bar{x})\varepsilon + d + g(\bar{x})\Delta u \quad (33)$$

▪ *Filtered Error Bound*

Choosing the Lyapunov function candidate the same as (25) and substituting (33) into (27) results in

$$\begin{aligned} \dot{V} &= -(K_v + \frac{\dot{g}(\bar{x})}{2g(\bar{x})^2})r^2 + \tilde{W}^T [\Gamma^{-1} \dot{\tilde{W}} - r\Phi(\bar{x}, \bar{e})] \\ &\quad + r\varepsilon + \frac{rd}{g(\bar{x})} + r\Delta u \\ &\leq -|r|[(K_v - \frac{g_{dM}}{2g_m^2})r] - \frac{W_{\max}^2}{4}\alpha - \varepsilon_N - \frac{d_M}{g_m} - \Delta u_{\max} \end{aligned} \quad (34)$$

According to (24) and (34), the bound of the filtered error can be expressed as (35).

$$|r| \leq \frac{\frac{W_{\max}^2}{4}\alpha + \varepsilon_N + \frac{d_M}{g_m} + \Delta u_{\max}}{K_v - \frac{g_{dM}}{2g_m^2}} \quad (35)$$

▪ *Weight Estimation Error Bound*

Similar to case 1, the weight estimation bound is given by

$$\|\tilde{W}\| \leq \frac{W_{\max} + \sqrt{W_{\max}^2 + \frac{4(\varepsilon_N + \frac{d_M}{g_m} + \Delta u_{\max})}{\alpha}}}{2} \quad (36)$$

Remark 1: In the adaptive control literature, the unboundedness of parameter estimates when persistence of excitation (PE) fails to hold is known as "parameter drift". This phenomenon has been referred to as "weight overtraining" in the NN literature. The PE condition ensures that parameter drift does not occur. However, it is difficult to

verify or guarantee the PE condition. Hence this theorem relaxes the PE condition [12].

Remark 2: The weights of the hidden layer are randomly chosen initially between 0 and 1 and fixed, therefore, not adapted. The initial weights of the output layer are just set to zero and then adapted online according to (23). There is no preliminary off-line learning phase, and stability will be provided by the outer tracking loop until the NN learns. This is a significant improvement over other NN control techniques where one must find some initial stabilizing weights, generally a difficult task for complex nonlinear systems over a wide range of operating conditions.

Remark 4: A single NN is used to approximate both nonlinearities of $f(\bar{x})$ and $g(\bar{x})$ with an expression shown in (18), avoiding the use of two neural networks to approximate $f(\bar{x})$ and $g(\bar{x})$ separately. This results in a well defined compact controller structure.

Remark 5: According to (35) and (36), it can be seen that this is a local stability result because of the bounded control input. It can also be seen from these equations that the error bounds are proportional to Δu_{max} . Larger u_{max} will result in a larger error bound. However, the tracking error bound can be made arbitrarily small by increasing K_v [17].

V. OPTIMAL CPSS PARAMETERS WITH PARTICLE SWARM OPTIMIZATION

Fig. 4 shows the typical block diagram of a CPSS recommended by IEEE [18]. Usually the parameters of the two lead lag compensator blocks are the same ($T_1 = T_3$, $T_2 = T_4$), thus the tunable parameters of CPSS are T_1 , T_2 , T_5 , T_6 , and K_{pss} . To have the CPSS provide good damping over wide operating conditions, these parameters need to be fine tuned in response to different kinds of disturbances, which is a time-consuming job. To compare the proposed NN controller design with the best possible performance of CPSS, particle swarm optimization is used in this paper to find the best parameters for CPSS.

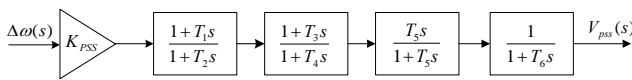


Fig. 4. Configuration of CPSS suggested by IEEE Std 421.5

PSO is one of the recent evolutionary computation techniques. Like the other evolutionary computation techniques, PSO is a population-based search algorithm and is initialized with a population of random solutions, called particles. Unlike in the other evolutionary computation techniques, each particle in the PSO is also associated with a velocity. Particles fly through the search space with velocities which are dynamically adjusted according to their historical behaviors. Therefore, the particles have a tendency to fly towards better and better solutions over the course of search process [19].

The PSO algorithm is simple in concept, easy to implement

and computational efficient. The PSO algorithm used in this paper is described as below:

The velocity of a particle in the dimension d is updated as follows:

$$v_{id} = wv_{id} + c_1 rand_1(.) (p_{id} - x_{id}) + c_2 rand_2(.) (p_{gd} - x_{id}) \quad (37)$$

The position of a particle in the dimension d is updated as follows:

$$x_{id} = x_{id} + v_{id} \quad (38)$$

where w , c_1 , and c_2 are the inertia weight, cognitive acceleration and social acceleration constants respectively, and $rand_1(.)$ and $rand_2(.)$ are two random functions in the range of $[0, 1]$; $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ represents the i th particle; $p_i = (p_{i1}, p_{i2}, \dots, p_{id})$ represents the best previous position (the position giving the best fitness value - p_{best}) of the i th particle; p_g represents the best particle among all the particles in the population (g_{best}); $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$ represents the velocity of the particle i [19].

Equation (37) consists of three parts. The first part is the momentum part. The velocity can't be changed abruptly. It is changed from the current velocity. The second part is the "cognitive" part which is learned from its own experience. The third part is the "social" part which is learned from group experience [19].

For the optimization of CPSS, there are five parameters, thus the dimension of x_i is 5. The range of the five parameters are as follows, $T_1 \in [0.1 \ 1]$, $T_2 \in [0.01 \ 0.1]$, $T_5 \in [1 \ 10]$, $T_6 \in [0.001 \ 0.01]$, and $K_{pss} \in [0.01 \ 0.1]$. the population size is chosen to be 10. The values for the positive constant w , c_1 , and c_2 are 0.8, 2, and 2 respectively.

To evaluate a particle (a vector of CPSS parameters), the system is simulated using the set of parameters for some kind of fault, and then the cost is calculated based on the sampled speed deviation of the generator. The cost function (fitness function) is defined as (39).

$$cost = \sum_{i=1}^n t(i) \cdot |\Delta\omega(i)| \quad (39)$$

where n is the number of samples, $t(i)$ is the time of the i th sample data, and $\Delta\omega(i) = \omega(i) - \omega_s$ is the speed deviation at time $t(i)$. The multiplication of $t(i)$ and $|\Delta\omega(i)|$ will give faster damping a lower cost.

The procedure for implementing the above PSO is as follows:

1. Initialize a population of particles with random positions and velocities in d dimensions of the problem space.
2. For each particle, evaluate the fitness/cost function.
3. Compare particle's fitness evaluation with its p_{best} . If current value is better than p_{best} , then set p_{best} equal to the current value, and p_i equals to the current location x_i in d -dimensional space.
4. Identify the particle in the neighborhood with the best success so far, and assign its index to the variable p_g .
5. Update the velocity and position of the particle according to (37) and (38).
6. Repeat step 2 until a maximum number of iterations, 100 here.

In the following section, the PSO-optimized CPSS is compared with the new NN controller design under three kinds of operating conditions, which are

- Case 1, a 200ms 3-phase short circuit fault at the infinite bus happens at 0.5s and cleared at 0.7s.
- Case 2, the operating point changes from $P_g=0.5p.u.$ and $Q_g=0.1p.u.$ to $P_g=0.7p.u.$ and $Q_g=0.2p.u.$ at 0.5s.
- Case 3, the impedance of the transmission line between the generator and infinite buses changes from $R_e=0.025$ and $X_{ep}=0.085$ to $R_e=0.05$ and $X_{ep}=0.17$ at 0.5s.

Fig. 5 shows an example of the PSO optimization process for Case 1. Table II shows the costs before and after optimization, and Table III shows the obtained optimal sets of CPSS parameters. In TABLE II and III, “Cases 1~3” stands for the optimizations corresponding to the above three Cases, and “Case All” means the optimization considers all of the three cases and the cost function is defined as (40).

$$\text{cost} = \sum_{i=1}^3 k_i \text{cost}_i \quad (40)$$

where k_i is a positive constant, in this paper, $k_1=k_2=k_3=1$.

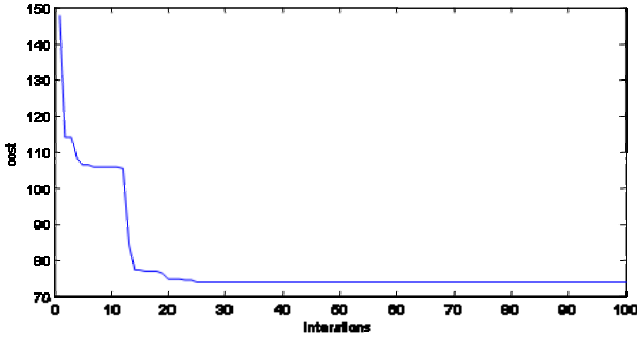


Fig. 5. Optimization process of the CPSS parameters

TABLE II
COMPARISON OF INITIAL COST AND OPTIMIZED COST

	Case 1	Case 2	Case 3	Case All
Initial cost	160.0311	20.7818	28.6531	209.4660
Final cost	73.8930	4.4444	5.8595	95.8161

TABLE III
OPTIMAL CPSS PARAMETERS TUNED BY PSO

	K_{PSS}	T_1	T_2	T_3	T_4	T_5	T_6
Case 1	0.0403	0.7827	0.0651	0.7827	0.0651	5.7049	0.0069
Case 2	0.0396	0.5226	0.0590	0.5226	0.0590	2.8453	0.0048
Case 3	0.0418	0.5047	0.0592	0.5047	0.0592	6.9329	0.0024
Case All	0.0541	0.5005	0.0521	0.5005	0.0521	5.5811	0.0080

From Table III, it can be seen that the three set of CPSS parameters optimized for three different operating conditions are different. Simulation studies also show that a CPSS optimized for some operating conditions may not work very well for other operating conditions. This is the reason why the tuning of CPSS parameters is so difficult. Because the

definition of its cost function, “Case All” can provide good performance for all of the three operating conditions. But the performance for some specific operating condition will not be the best. This fact can be observed in Section VI.

VI. SIMULATION RESULTS

The neural network has 10 inputs corresponding to the systems states, error dynamics and bias respectively.

$$[\delta, \omega, E'_d, E'_q, E_{fd}, V_r, \Delta\dot{\omega}, \Delta\ddot{\omega}, \Delta\ddot{\omega}, 1]^T \quad (41)$$

The number of hidden neurons is empirically selected to be 10 based on controller performance. The weights of the input layers are initially set to random numbers between 0 and 1 and held fixed thereafter. The activation function of the hidden layer is *hyperbolic tangent function*. The initial weight of the output layer W is set to zero and updated over time. Other parameters used are as follows: $K_v=0.1$, $A=[10000, 4000, 600, 40]^T$, $u_{max}=0.5$, $\Gamma=5$, and $\alpha=5$.

The comparisons of simulation results for the three cases are shown in Figs 6~8, Figs 9~11, and Figs 12~14 respectively. In these figure, the “magenta dotted line” represents the simulation results without stabilizing control – “no cpss”, the “blue dash-dot line” represents the simulation results for CPSS optimized only for one of the three cases – “cpss i”, the “red dashed line” represents the simulations results for CPSS optimized for all of the three cases – “cpss all”, and “black solid line” for the proposed NN based power system stabilizer – “nn pss”.

A. Comparison of system response for Case 1

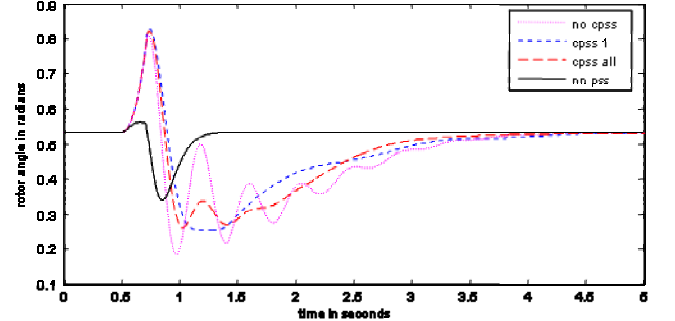


Fig. 6. Rotor angle response to 200ms 3-phase short circuit fault ($P=0.5pu$, $Q=0.1pu$)

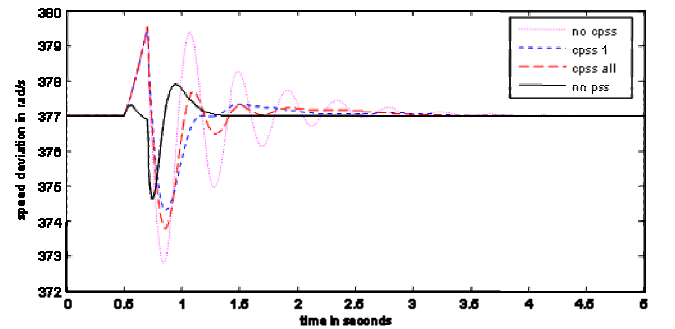


Fig. 7. Speed deviation response to 200ms 3-phase short circuit fault ($P=0.5pu$, $Q=0.1pu$)

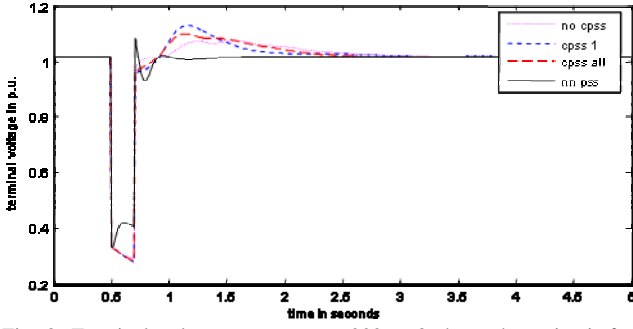


Fig. 8. Terminal voltage responses to 200ms 3-phase short circuit fault ($P=0.5pu$, $Q=0.1pu$)

B. Comparison of system response for Case 2

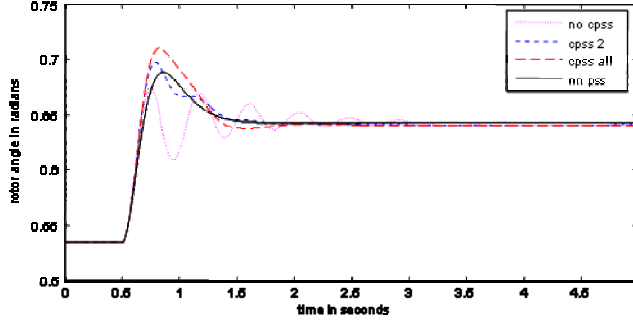


Fig. 9. Rotor angle responses to a change of operating points ($P=0.5pu$, $Q=0.1pu$ to $P=0.7pu$, $Q=0.2pu$).

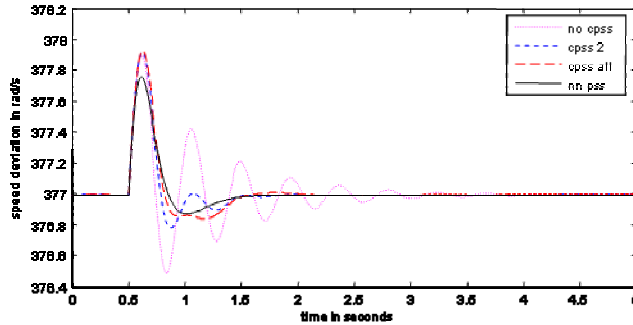


Fig. 10. Speed deviation responses to a change of operating points ($P=0.5pu$, $Q=0.1pu$ to $P=0.7pu$, $Q=0.2pu$).

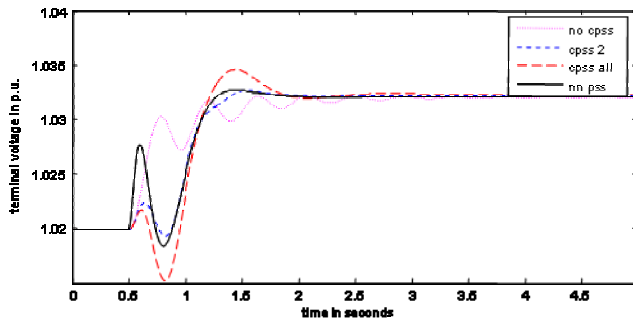


Fig. 11. Terminal voltage responses to a change of operating points ($P=0.5pu$, $Q=0.1pu$ to $P=0.7pu$, $Q=0.2pu$).

C. Comparison of system response for Case 3

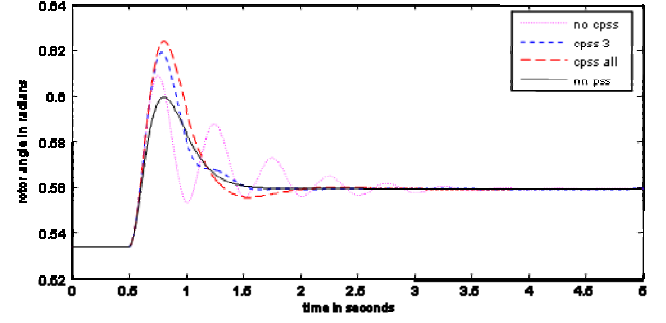


Fig. 12. Rotor angle responses to a change in line impedance connected to the infinite bus from $R_e=0.025$, $X_{ep}=0.085$ to $R_e=0.05$, $X_{ep}=0.17$.

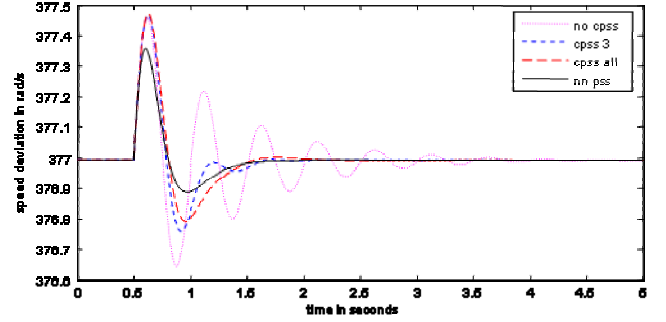


Fig. 13. Speed deviation response to a change in the line impedance connected to the infinite bus from $R_e=0.025$, $X_{ep}=0.085$ to $R_e=0.05$, $X_{ep}=0.17$.

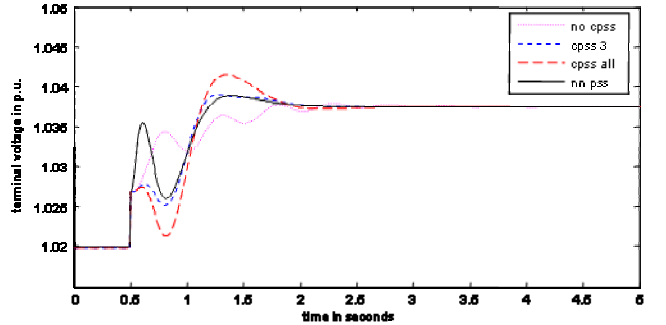


Fig. 14. Terminal voltage response to a change in line impedance connected to the infinite bus from $R_e=0.025$, $X_{ep}=0.085$ to $R_e=0.05$, $X_{ep}=0.17$.

From the simulation results, it can be seen that the proposed controller damps out the oscillations very well. It also can be seen that the neural network controller can adapt to changes in the operating condition in a fast manner despite no offline training was carried out.

VII. CONCLUSION

The design of stabilizing controller is a necessary for power systems. This paper proposed a stabilizing neural network controller for a single machine infinite bus power system. The weight updating rule does not require the persistently excitation condition and can guarantee the stability of the closed loop system when the control signal is subject to magnitude constraints. The proposed NN

controller performance is compared with that PSO optimized CPSS. Simulations under different operating conditions and disturbances demonstrate the effectiveness of the proposed controller. The proposed control scheme can also be applied to control similar class of nonlinear systems. Future research will include the design of stable decentralized controllers for multi-machine power systems.

REFERENCE

- [1] E.V. Larsen and D.A. Swann, "Applying power system stabilizers, Part I, II, III", *IEEE Transaction on Power Apparatus and Systems*, vol. PAS-100, no. 6, pp. 3017-3041, 1981.
- [2] M. A. Abido, "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] A. L. B. Do Bomfim, G. N. Taranto, and D. M. Falcao, "Simultaneous tuning of power system damping controllers using genetic algorithms," *IEEE Transaction on Power Systems*, vol. 15, no. 1, pp. 163-169, 2000.
- [4] Y. L. Abdel-Magid, M. A. Abido, and A. H. Mantaway, "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] W. Liu, G.K. Venayagamoorthy, and D.C. Wunsch, "Design of an adaptive neural network based power system stabilizer", *Neural Networks*, vol. 16, no. 5-6, pp. 891-898, 2003.
- [6] Q Lu and Y Sun, "Nonlinear stabilizing control of multimachine system," *IEEE Transactions on Power Systems*, Vol. 4, No. 1, pp. 236-241, February 1989.
- [7] J W Chapman, M D Illic, C A King, L Eng and H Kaufman, "Stabilizing a multimachine power system via decentralized feedback linearizing excitation control," *IEEE Transactions on Power Systems*, Vol. 8, No. 3, pp. 830-839, August 1993.
- [8] M Nambu and Y Ohsawa, "Development of an advanced power system stabilizer using a strict linearization approach," *IEEE Transactions on Power Systems*, Vol. 11, No. 2, pp. 813-818, May 1996.
- [9] A. R. Baron, "Universal approximation bounds for superposition of a sigmoid function", *IEEE Transaction on Information Theory*, vol. 39, no. 3, pp. 930-945, 1993.
- [10] N. Sadeh, "A perceptron network for functional identification and control of nonlinear systems", *IEEE Transaction on Neural Networks*, vol. 4, pp. 1823-1836, 1992.
- [11] B. Igelnik and Y-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net", *IEEE Transaction on Neural Networks*, vol. 6, no. 6, pp. 1320-1329, November 1995.
- [12] F. L. Lewis, S. Jagannathan, and A. Yesildirek, "Neural network control of robot manipulators and nonlinear systems", Taylor and Francis 1999.
- [13] P. W. Sauer and M. M. Pai, *Power System Dynamics and Stability*, Upper Saddle River, N.J, Prentice Hall, 1998.
- [14] W. Liu, J. Sarangapani, G. K. Venayagamoorthy and D. C. Wunsch II, "Feedback linearization based power system stabilizer design with control limits," *Proceedings of The 36th North American Power Symposium*, pp. 343-348, August 9-10, 2004, Moscow, Idaho, USA
- [15] A. Isidori, "Nonlinear control systems", Springer-Verlag, 1989.
- [16] S. S. Ge and C. Wang, "Direct adaptive NN control of a class of nonlinear systems", *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 214-221, January, 2002.
- [17] S. P. Karason and A. M. Annaswamy, "Adaptive control in the presence of input constraints", *IEEE Transactions on Automatic Control*, vol. 39, no. 11, pp. 2325-2330, November, 1994.
- [18] IEEE, "Recommended practice for excitation system models for power system stability studies", IEEE Std. 421.5-1992, 1992.
- [19] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceeding of IEEE International Conference on Neural Networks (ICNN)*, Vol. IV, pp. 1942-1948, 1995.
- [20] Y. Shi and R. Eberhart, "A modified particle swarm optimize," *IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, USA, pp. 69-73, May 1998.
- [21] G. K. Venayagamoorthy, "Adaptive critics for dynamic particle swarm optimization," *Proceedings of the 2004 IEEE International Symposium on Intelligent Control*, Taipei, China, pp.380-384, September 2-4, 2004..

APPENDIX

A. Definition of the constants in Equations (7) ~ (12):

$$\begin{aligned}
 k_1 &= \frac{R_e}{R_e^2 + (X'_d + X_{ep})(X'_q + X_{ep})}, \\
 k_2 &= \frac{X'_q + X_{ep}}{R_e^2 + (X'_d + X_{ep})(X'_q + X_{ep})}, \\
 k_3 &= \frac{-X'_d - X_{ep}}{R_e^2 + (X'_d + X_{ep})(X'_q + X_{ep})}, \\
 k_4 &= X'_d - X'_q, \\
 k_5 &= \frac{\omega_s}{2H}, k_6 = (-k_1 + k_1 k_3 k_4) k_5, k_7 = (-k_1 + k_1 k_2 k_4) k_5, \\
 k_8 &= [-k_2 - k_3 + (k_1^2 + k_2 k_3) k_4] k_5, k_9 = (k_1 - 2k_1 k_3 k_4) k_5 V_s, \\
 k_{10} &= [k_2 - (k_1^2 + k_2 k_3) k_4] k_5 V_s, k_{11} = [k_3 - (k_1^2 + k_2 k_3) k_4] k_5 V_s, \\
 k_{12} &= (k_1 - 2k_1 k_2 k_4) k_5 V_s, k_{13} = k_1 k_3 k_4 k_5 V_s^2, k_{14} = k_1 k_2 k_4 k_5 V_s^2, \\
 k_{15} &= (k_1^2 + k_2 k_3) k_4 k_5 V_s^2, k_{16} = k_5 T_m, k_{17} = -\frac{1}{T_{q0}} E'_d + \frac{X'_q - X_q}{T_{q0}} k_3, \\
 k_{18} &= \frac{X_q - X'_q}{T_{q0}} k_3, k_{19} = \frac{X'_q - X_q}{T_{q0}} k_3 V_s, k_{20} = \frac{X'_q - X_q}{T_{q0}} k_1 V_s, \\
 k_{21} &= \frac{X'_d - X_d}{T_{d0}} k_1, k_{22} = -\frac{1}{T_{d0}} - \frac{X_d - X'_d}{T_{d0}} k_2, k_{23} = \frac{X_d - X'_d}{T_{d0}} k_1 V_s, \\
 k_{24} &= \frac{X_d - X'_d}{T_{d0}} k_2 V_s, k_{25} = \frac{1}{T_{d0}}, k_{26} = \frac{K_a}{T_e T_a}
 \end{aligned}$$