01 Oct 2007

Comparisons of an Adaptive Neural Network Based Controller and an Optimized Conventional Power System Stabilizer

Wenxin Liu

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

Jagannathan Sarangapani
Missouri University of Science and Technology, sarangap@mst.edu

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

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/ele_comeng_facwork/1509

Recommended Citation

W. Liu et al., "Comparisons of an Adaptive Neural Network Based Controller and an Optimized Conventional Power System Stabilizer," *Proceedings of the 16th IEEE International Conference on Control Applications (2007, Singapore)*, pp. 922-927, Institute of Electrical and Electronics Engineers (IEEE), Oct 2007.
The definitive version is available at https://doi.org/10.1109/CCA.2007.4389351

16th IEEE International Conference on Control Applications
Part of IEEE Multi-conference on Systems and Control
Singapore, 1-3 October 2007

Comparisons Of An Adaptive Neural Network Based Controller And An Optimized Conventional Power System Stabilizer

Wenxin Liu[1], Ganesh K. Venayagamoorthy[2], Jagannathan Sarangapani[2]
Donald C. Wunsch II[2], Mariesa L. Crow[2], Li Liu[1], and David A. Cartes[1]

Abstract—Power system stabilizers are widely used to damp out the low frequency oscillations in power systems. In power system control literature, there is a lack of stability analysis for proposed controller designs. This paper proposes a Neural Network (NN) based stabilizing controller design based on a sixth order single machine infinite bus power system model. The NN is used to compensate the complex nonlinear dynamics of power system. To speed up the learning process, an adaptive signal is introduced to the NN's weights updating rule. The NN can be directly used online without offline training process. Magnitude constraint of the activators is modeled as saturation nonlinearities and is included in the stability analysis. The proposed controller design is compared with Conventional Power System Stabilizers whose parameters are optimized by Particle Swarm Optimization. Simulation results demonstrate the effectiveness of the proposed controller design.

I. INTRODUCTION

IN power systems, changes of operating conditions will result in small-magnitude low-frequency oscillations that may persist for long periods of time. In some cases, the oscillations will limit the power transfer capability. The issue of power system stabilizing control has received a lot of attention since 1960's. Power system stabilizer (PSS) is designed to generate supplementary control signal in the excitation system to damp out low frequency oscillations.

Earlier studies on stabilizing controller designs are based on linearized models. For example, the widely used conventional power system stabilizer (CPSS) is designed based on the theory of phase compensation and introduced as a lead-lag compensator. CPSS is simple in structure and easier to implement. But to make CPSS provide good damping over wide operating conditions, its parameters need to be fine tuned, which is a time-consuming job. To simplify this process, intelligent optimization algorithms have been applied to offline determining the "optimal parameters" of CPSS by optimizing an eigenvalue based cost function [1~4]. In the past decade, fuzzy logic and NN were applied to this area to adjust the parameters of CPSS online based on the knowledge gained from offline training. Since power systems are highly nonlinear systems, with configurations and parameters changing with time, adaptive controller designs based on nonlinear models are more promising.

In the past decades, nonlinear control and intelligent techniques (such as neural networks, fuzzy logic, etc.) have been applied to the designs of adaptive stabilizing controllers [5-7]. Most publications in power system control demonstrate the effectiveness of the proposed controller designs via simulations or even experiments. But there is still a lack of controller designs based on stability analyses. To address this problem, some controller designs based on feedback linearization have appeared [8-10]. It is well known that feedback linearization requires the system model to be known exactly. Thus, imprecise model will degrade the performance of this type of controller designs tremendously. Since it is very difficult to get precise models for complex systems, there are some limitations with this type of controller designs. To overcome this problem, different kinds of algorithm have been applied [11-16].

In order to further release the requirement of precise model, this paper proposes an adaptive neural network stabilizing controller based on a nonlinear sixth-order single machine power system model. Since complex nonlinearities are approximated with a NN, the requirement for precise system model is released. A randomly initialized neural network can be directly used online and tuned according to the weight updating rules, thus the preliminary offline training phase is unnecessary. In order to speed up NN's learning process, an auxiliary adaptive signal is introduced to NN's weight updating rule. The bounds on the filtered error and NN estimation error under actuator magnitude constraint are also given. Comparisons with a particle swarm optimization optimized CPSS under different operating conditions demonstrate the effectiveness of the proposed algorithm.

Considering the complexities in power system model and operating conditions, there is still a lot of work to do in the field of stable power system controller design. Stability analysis during controller design is helpful to understand system behavior and can provide more guidance to the implementation process. Furthermore, the IEEE standard CPSS in this paper is simpler compared to practical sophisticate CPSS products, which may generate much better performance.

II. BACKGROUND

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

[1] Wenxin Liu, Li Liu, and Dave A. Cartes are with the Center for Advanced Power Systems (CAPS), Florida State University, Tallahassee, FL 32310, USA. Emails: {wliu, lliu, dave}@caps.fsu.edu
[2] Ganesh K. Venayagamoorthy, Jagannathan Sarangapani, Donald C. Wunsch II, and Mariesa L. Crow are with the Department of Electrical and Computer Engineering, University of Missouri at Rolla, MO 65401, USA. Emails: {sarangap, ganeshv, dwunsch, crow}@umr.edu.

1-4244-0443-6/07/$20.00 ©2007 IEEE.

A. Approximation Property of NN

The commonly used property of NNs for control is its function approximation and adaptation capacities. Let f(x) be a smooth function from Rn → Rm, then it can be shown that, as long as x is restricted to a compact set $S \in R^n$, for any given positive number εN, there exist weights and thresholds such that

$$f(x) = W^T \varphi(x) + \varepsilon(x) \tag{1}$$

where *x* is the input vector, *φ(.)* is the activation function, *W* is the weight matrix of the output layer and *ε(x)* is the approximation error that satisfies $\varepsilon(x) \le \varepsilon_N$.

For the above function approximation, φ(x) must form a basis [17]. For two layer neural networks, φ(x)=σ(VTx), where V is the weight matrix of the first layer and σ(x) is a sigmoid function. If V is fixed, then W becomes the only design parameter. It has been shown in [18] that φ(x) can form a basis if V is chosen randomly. The larger the number of the hidden layer neurons Nh, the smaller the approximation error *ε(x)*.

B. Stability of Systems

To formulate the controller, the following stability notion is needed. Consider the nonlinear system given by

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

where *x(t)* is a state vector, *u(t)* is the input vector and *y(t)* is the output vector. 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 $t \ge t_0 + T$.

III. Model of Single Machine Power System

The dynamics of a single machine power system can be represented with a two-axis model [19] as in (3).

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

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

$$\begin{cases} 0 = R_e I_d - (X_q^{'} + X_{ep}) I_q - E_d^{'} + V_s \sin(\delta - \theta_{vs}) \\ 0 = R_e I_q + (X_d^{'} + X_{ep}) I_d - E_q^{'} + V_s \cos(\delta - \theta_{vs}) \end{cases} \tag{4}$$

$$V_t = \sqrt{V_d^2 + V_q^2} \tag{5}$$

with

$$\begin{cases} V_d = R_e I_d - X_{ep} I_q + V_s \sin(\delta - \theta_{vs}) \\ V_q = R_e I_q + X_{ep} I_d + V_s \cos(\delta - \theta_{vs}) \end{cases} \tag{6}$$

In above equations, T_m is the mechanical torque, E_{fd} is the field voltage, V_r is the output of 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, 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.

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, the error dynamics of the system is transformed into the following Brunovsky Canonical Form.

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= e_3 \\ \dot{e}_3 &= e_4 \\ \dot{e}_4 &= f(\bar{x}) + g(\bar{x})u + d \end{aligned} \tag{7}$$

where $\bar{e} = [e_1 \quad e_2 \quad e_3 \quad e_4]^T = [\omega - \omega_s \quad \dot{\omega} \quad \ddot{\omega} \quad \dddot{\omega}]^T$, u stands for the control signal, and d stands for a bounded disturbance with $|d| \le d_M$, $\bar{x}$ stands for δ, ω, $E_q^{'}$, E_{fd}, V_r. The definitions of $k_1 \sim k_{28}$, and $f(\bar{x})$ and $g(\bar{x})$ can be found in [20].

IV. NN Based 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 $|\dot{g}(\bar{x})| \le g_{dM}$.

Considering the range of the variables in above equations, assumptions 1 and 2 hold for single machine power system. Simulation studies under different kinds of operating conditions confirm the above analysis.

B. Neural Network Based Controller Design

Define the filtered error *r* as

$$r = [\Lambda^T \quad 1]\bar{e} \tag{8}$$

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

Differentiating (8) and substituting (7) to get

$$\dot{r} = [0 \quad \Lambda^T]\bar{e} + f(\bar{x}) + g(\bar{x})u + d \tag{9}$$

According to the theory of feedback linearization, the ideal control signal can be chosen as

$$u^* = -K_v r - \frac{1}{g(\bar{x})}\{f(\bar{x}) + [0 \quad \Lambda^T]\bar{e}\} \tag{10}$$

where K_v is a selected positive constant.

According to NN approximation theory, the second part in (10) can be approximated using a NN, such that

$$\hat{W}^T\Phi(\bar{x},\bar{e}) \approx -\frac{1}{g(\bar{x})}\{f(\bar{x}) + [0 \quad \Lambda^T]\bar{e}\} = f_n(\bar{x},\bar{e}) \tag{11}$$

To speed up the NN learning, an auxiliary signal R is introduced according to (12).

$$R = e^{\rho^3 + r^2} \tag{12}$$

where ρ is an intermediate signal that will be defined later.

Define the desired control signal v (without magnitude constraint) as

$$v = -K_v r + \hat{W}^T\Phi(\bar{x},\bar{e}) \tag{13}$$

Then the actual control signal applied to the power system is given by

$$u = \begin{cases} v & while\,|v| \le u_{\max} \\ u_{\max}\, sign(v) & while\,|v| > u_{\max} \end{cases} \tag{14}$$

where $u_{\max}$ is the maximum allowed control signal magnitude.

The structure of the controller is shown in Fig. 1. The next step is to determine appropriate weight updating rules so that the closed-loop stability of control system can be guaranteed. The performance of the proposed adaptive neural network controller is described by Theorem I.

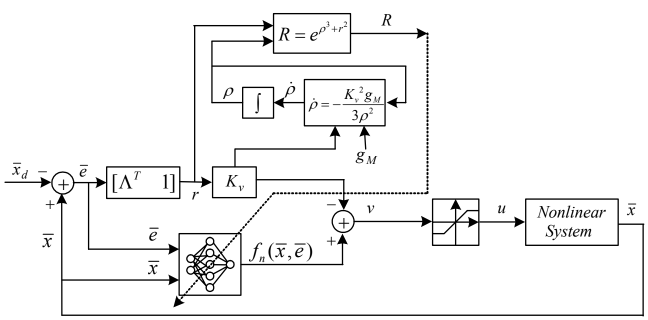

Fig. 1. Structure of the NN-based power system stabilizing controller

Assume there is a constant weight W, that can approximate (11) with designated precision, such that

$$W^T\Phi(\bar{x},\bar{e}) = -\frac{1}{g(\bar{x})}[f(\bar{x}) + [0 \quad \Lambda^T]\bar{e} + \varepsilon \tag{15}$$

where the bounded approximation error ε satisfies $|\varepsilon| \le \varepsilon_N$.

Assume W is bounded by W_{max}, that is, $||W|| \le W_{max}$. Rearrange (15) as an expression of $f(\bar{x})$ and substitute which into (9) to get

$$\dot{r} = -K_v g(\bar{x})r - g(\bar{x})\tilde{W}^T\Phi(\bar{x},\bar{e}) + g(\bar{x})\varepsilon + d \tag{16}$$

where $\tilde{W}$ is the weights approximation error and is defined as:

$$\tilde{W} = W - \hat{W} \tag{17}$$

Theorem I: Assume the unknown disturbance d, and the weight approximation error ε are bounded by known constants such that $|d| \le d_N$, $|\varepsilon| \le \varepsilon_N$ respectively. Select the weight updating rule as

$$\dot{\hat{W}} = -\Gamma r\Phi - 2\Gamma rR^2\Phi - \alpha\Gamma\|r\|\hat{W} \tag{18}$$

where α, $\Gamma > 0$ are the adaptation gains and the gain K_v satisfying

$$K_v > \frac{g_{dM}}{2g_m^2} \tag{19}$$

and the auxiliary signal ρ is updating according to

$$\dot{\rho} = -\frac{K_v{}^2 g_M}{3\rho^2} \tag{20}$$

then the filtered error $r(t)$ and the weight estimation error $\tilde{W}$ are uniformly ultimately bounded.

Proof: The proof is done in two cases.

Case 1: $|v| \le u_{\max}, u = v$

- *Filtered Error Bound*

Choose the Lyapunov function V as [21]

$$V = \frac{1}{2}\tilde{W}^T\Gamma^{-1}\tilde{W} + \frac{r^2}{2g(x)} + \frac{R^2}{2g(x)} \tag{21}$$

Evaluating the first derivative of V to get

$$\dot{V} = \tilde{W}^T\Gamma^{-1}\dot{\tilde{W}} + \frac{r\dot{r}}{g(x)} - \frac{\dot{g}(x)r^2}{2g(x)^2} + \frac{R\dot{R}}{g(x)} - \frac{\dot{g}(x)R^2}{2g(x)^2} \tag{22}$$

Substitute $\dot{R} = 3\rho^2 R\dot{\rho} + 2rR\dot{r}$ and the error dynamics (16) into (22), we get

$$\begin{aligned} \dot{V} &= \tilde{W}^T\Gamma^{-1}\dot{\tilde{W}} - K_v r^2 - r\tilde{W}^T\Phi - \frac{\dot{g}(x)r^2}{2g(x)^2} + \\ &\quad \frac{R}{g(x)}(3\rho^2 R\dot{\rho} + 2rR\dot{r}) - \frac{\dot{g}(x)R^2}{2g(x)^2} - r\varepsilon + \frac{dr}{g(x)} \\ &\le \tilde{W}^T\Gamma^{-1}\dot{\tilde{W}} - (K_v - \frac{g_{dM}}{2g_m^2})r^2 - r\tilde{W}^T\Phi + \frac{3\rho^2R^2\dot{\rho}}{g(x)} \\ &\quad - 2rR^2(K_v r + \tilde{W}^T\Phi + \varepsilon - \frac{d}{g(x)}) + \frac{g_{dM}}{2g_m^2}R^2 - r\varepsilon + \frac{dr}{g(x)} \end{aligned} \tag{23}$$

Define $K = K_v - \frac{g_{dM}}{2g_m^2} > 0$. According to (23), we have

$$\begin{aligned} \dot{V} &\le \tilde{W}^T[\Gamma^{-1}\dot{\tilde{W}} - r\Phi - 2rR^2\Phi] - Kr^2 + [\frac{3\rho^2R^2\dot{\rho}}{g(x)} + \frac{g_{dM}}{2g_m^2}R^2] \\ &\quad - 2K_v r^2R^2 - r\varepsilon - 2rR^2\varepsilon + \frac{dr}{g(x)} - \frac{2rR^2d}{g(x)} \\ &\le -Kr^2 + \alpha\|r\|\tilde{W}^T(W - \tilde{W}) - r\varepsilon + \frac{dr}{g(x)} - 2K_v r^2R^2 - 2rR^2\varepsilon - \frac{2rR^2d}{g(x)} \\ &\le -\|r\|[K\|r\| - (\alpha\frac{W_{\max}^2}{4} + \varepsilon_N + \frac{d_M}{g_m})] - 2\|r\|R^2[K_v\|r\| - (\varepsilon_N + \frac{d_M}{g_m})] \end{aligned} \tag{24}$$

$\dot{V}$ is negative as long as $\|r\| > \dfrac{\varepsilon_N + \frac{d_M}{g_m}}{K_v}$ and $\|r\| > \dfrac{\alpha\frac{W_{\max}^2}{4} + \varepsilon_N + \frac{d_M}{g_m}}{K}$.

Therefore r is bounded according to

$$\|r\| \le \max\left(\frac{\varepsilon_N + \frac{d_M}{g_m}}{K_v}, \frac{\alpha \frac{W_{\max}^2}{4} + \varepsilon_N + \frac{d_M}{g_m}}{K}\right) \tag{25}$$

- *Weight Estimation Error Bound*

Choose the Lyapunov function candidate the same as (21), according to (24) and (25)

$$\dot{V} \le -\alpha \|r\| \left(\|\tilde{W}\|^2 - \|\tilde{W}\| W_{\max} - \frac{\varepsilon_N + \frac{d_M}{g_m}}{\alpha}\right) \tag{26}$$

It can be seen that $\dot{V} < 0$ as long as $\|\tilde{W}\|^2 - W_{\max}\|\tilde{W}\| - \frac{\varepsilon_N + \frac{d_M}{g_m}}{\alpha} > 0$, so the weights estimation error bound is given by

$$\|\tilde{W}\| \le \frac{W_{\max} + \sqrt{W_{\max}^2 + \frac{4(\varepsilon_N + \frac{d_M}{g_m})}{\alpha}}}{2} \tag{27}$$

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

Define $\Delta u = u - v$, with Δu satisfying $|\Delta u| \le \Delta u_{\max}$. Substitute $u = v + \Delta u$ into (9), similarly, we can get

$$\dot{r} = -K_v g(\bar{x}) r - g(\bar{x}) \tilde{W}^T \Phi(\bar{x}, \bar{e}) + g(\bar{x})\varepsilon + d + g(\bar{x})\Delta u \tag{28}$$

- *Filtered Error Bound*

Choose the Lyapunov function candidate the same as (21), similar as (24)

$$\begin{aligned} \dot{V} &= \tilde{W}^T \Gamma^{-1} \dot{\tilde{W}} - K_v r^2 - r\tilde{W}^T \Phi - \frac{\dot{g}(x) r^2}{2g(x)^2} \\ &\quad + \frac{R}{g(x)}(3\rho^2 R\dot{\rho} + 2rR\dot{r}) - \frac{\dot{g}(x)R^2}{2g(x)^2} - r\varepsilon + \frac{dr}{g(x)} + r\Delta u \qquad (29) \\ &\le -\|r\|[K\|r\| - (\alpha \frac{W_{\max}^2}{4} + \varepsilon_N + \frac{d_M}{g_m} + \Delta u_{\max})] \\ &\quad - 2\|r\| R^2 [K_v \|r\| - (\varepsilon_N + \frac{d_M}{g_m})] \end{aligned}$$

Now the boundary for the filtered error becomes

$$\|r\| > \max\left(\frac{\alpha \frac{W_{\max}^2}{4} + \varepsilon_N + \frac{d_M}{g_m} + \Delta u_{\max}}{K}, \frac{\varepsilon_N + \frac{d_M}{g_m}}{K_v}\right) \tag{30}$$

- *Weight Estimation Error Bound*

Similarly, we can get the bound for the weight estimation

$$\|\tilde{W}\| > \frac{W_{\max} + \sqrt{W_{\max}^2 + \frac{4(\varepsilon_N + \frac{d_M}{g_m} + \Delta u_{\max})}{\alpha}}}{2} \tag{31}$$

Remark 1: The weights of the hidden layer are randomly initialized between 0 and 1 and held constant thereafter. The initial weights of the output layer are set to zero and then tuned online according to (18). There is no preliminary off-line training phase. This is a significant improvement over other NN control techniques where one must find some initial stabilizing weights, which is generally difficult for complex nonlinear systems.

Remark 2: The weight updating rule (18) can be seen as an unsupervised version of backpropagation in that the ideal plant output is not needed; instead the filtered error, which is easily measurable in the closed-loop system, is used in tuning NN weights. It should also be realized that this is a version of the backpropagation through time algorithm, as the weights are continuously tuned as a function of time.

Remark 3: Although an auxiliary signal is introduced to speed up the learning of NN, the proposed control scheme cannot be seen as reinforcement learning or adaptive critic design based controller design. The main reason is that there is no optimization process with the auxiliary signal.

V. Optimization Of CPSS Parameters Using Particle Swarm Optimization

Fig. 2 shows the typical block diagram of CPSS recommended by IEEE [22]. 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 design a CPSS with good damping performance, the above parameters need to be fine-tuned, which is a time-consuming job. To compare the proposed NN controller design with the best possible performance of CPSS, inspired by [4], Particle Swarm Optimization (PSO) is used in this paper to find a good CPSS design.

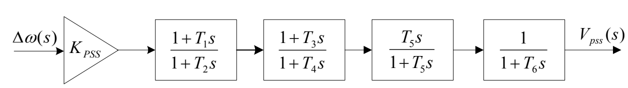

Fig. 2. Structure of CPSS suggested by IEEE Std. 421.5

PSO is one of the recent evolutionary computation techniques that is simple in concept, easy to implement and computational efficient [23]. The updating rules of PSO are given in (35).

$$V_{\text{new}} = w \times V_{old} + c_1 \times rand_1 \times (P_{best} - P_{old}) + c_2 \times rand_2 \times (L_{best} - P_{old})$$
$$P_{new} = P_{old} + V_{new} \tag{32}$$

where V_{new} is the new velocity vector calculated for each particle, V_{old} is the velocity vector of the robot from the previous iteration, P_{new} is the new position vector calculated for each robot, P_{old} is the position vector of the robot from the previous iteration, W is the inertia weight constant, c_1 and c_2 is the acceleration constant, and *rand* is the generates a uniform random value between [0 1].

For the optimization of CPSS, there are five parameters, thus the dimension of x_i is 5. The range of the five parameters are set as follows, $T_1 \in [0.1 \quad 1]$, $T_2 \in [0.01 \quad 0.1]$, $T_5 \in [1 \quad 10]$, $T_6 \in [0.001 \quad 0.01]$, and $K_{PSS} \in [0.01 \quad 0.1]$. 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 with the set of parameters for some kind of fault. Before the fault is applied, the system is running stable. For evaluation of dynamic performance, only the response after fault is considered. The sampling time is 0.01 second. 500 samples data are collected for each candidate solutions, which means 5-second performance after the fault is applied. Then the cost is calculated from the simulation data according to (33). The details of the PSO optimization process can be found in [23].

$$\cos t = \sum_{i=1}^{n} t(i) \cdot |\Delta\omega(i)| \tag{33}$$

where n is the number of samples, $t(i)$ is the time of the *ith* 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.

Table I shows the value of the parameters during simulation.

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

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=0.5*p.u.* and Q_g=0.1*p.u.* to P_g=0.7*p.u.* and Q_g=0.2*p.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.

Table II shows the obtained optimal sets of CPSS parameters.

TABLE II
OPTIMAL CPSS PARAMETERS TUNED BY PSO

	K_{PSS}	T_1	T_2	T_5	T_6
Case 1	0.0403	0.7827	0.0651	5.7049	0.0069
Case 2	0.0396	0.5226	0.0590	2.8453	0.0048
Case 3	0.0418	0.5047	0.0592	6.9329	0.0024

From Table II, it can be seen that the three sets of CPSS parameters optimized for three different operating conditions are different. Simulation studies also show that a CPSS optimized for some operating condition may not work well for another operating condition. This is the reason why the tuning of CPSS parameters is difficult.

VI. SIMULATION RESULTS

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

$$[\delta, \omega, E_d', E_q', E_{fd}, V_r, \dot{\omega}, \ddot{\omega}, \dddot{\omega}, 1]^T \tag{34}$$

The number of hidden neurons is empirically selected to be 10 based on controller performance. The activation function of the hidden layer is *hyperbolic tangent function*. Other parameters used are as follows: K_v=0.1, Λ=[10000, 4000, 600, 40]^T, u_{max}=0.5, Γ=5, and α=5.

The comparisons of simulation results for the three cases are shown in Figs 3~8. 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", and "red solid line" for the proposed NN based power system stabilizer – "nn pss".

A. Comparison of system response for Case 1

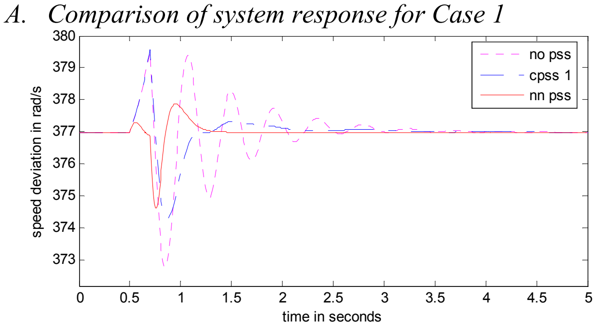

Fig. 3. Speed deviation responses comparison for Case 1

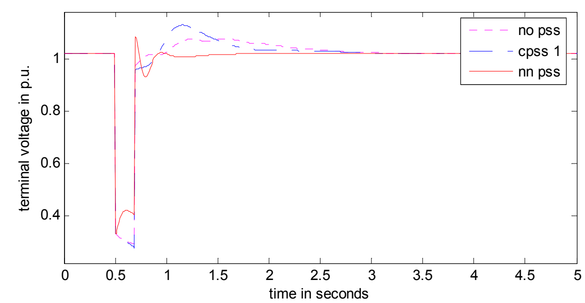

Fig. 4. Terminal voltage responses comparison for Case 1

B. Comparison of system response for Case 2

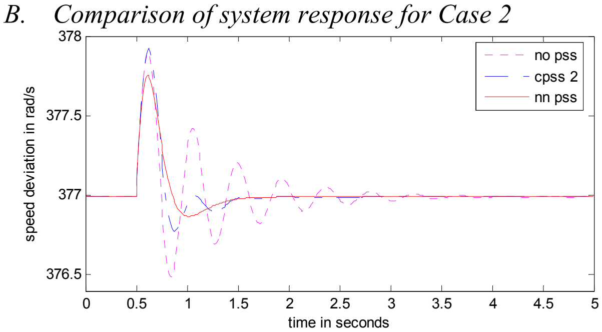

Fig. 5. Speed deviation responses comparison for Case 2

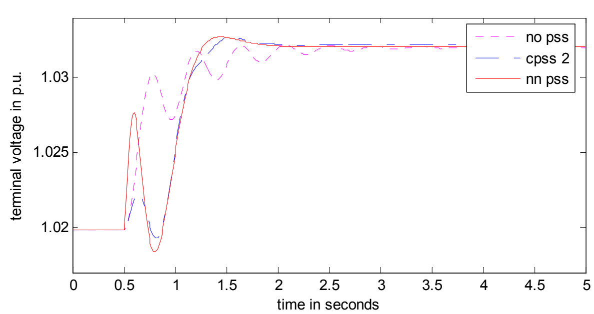

Fig. 6. Terminal voltage responses comparison for Case 2

C. Comparison of system response for Case 3

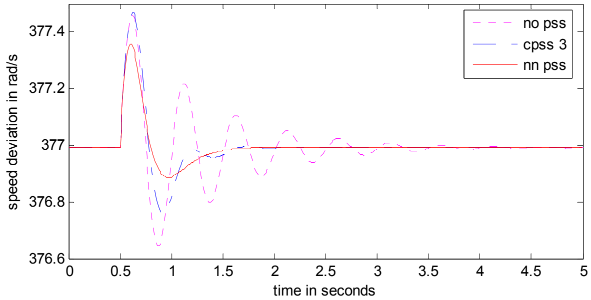

Fig. 7. Speed deviation responses comparison for Case 3

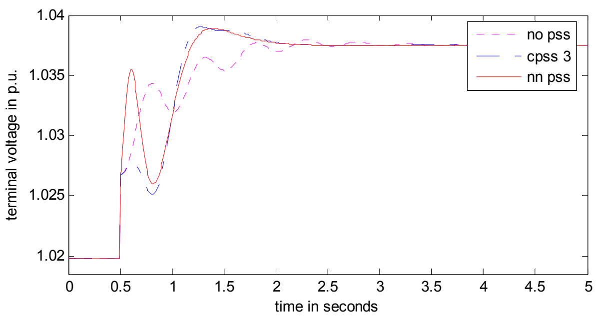

Fig. 8. Terminal voltage responses comparison for Case 3

As can be observed from the above Figs that the proposed NN controller always provides better damping. Furthermore, the proposed stabilizing controller does not have detrimental impact on the existing AVR.

VII. Conclusion

This paper proposed a neural network based stabilizing controller for 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. Comparisons with PSO optimized CPSSs under different operating conditions demonstrate the effectiveness of the proposed controller. The proposed control scheme can also be applied to the control of similar nonlinear systems.

Reference

[1] 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.

[2] 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.

[3] 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.

[4] M.A. Abido, "Particle swarm optimization for multimachine power system stabilizer design," IEEE Power Engineering Society Summer Meeting, Vol. 3, 15-19, pp. 1346-1351, July 2001.

[5] K.A. El-Metwally, G.C. Hancock, and O.P. Malik, "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] P. Shamsollahi, and O.P. Malik, "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.

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

[8] 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.

[9] J.W. Chapman, M.D.Ilic, C. 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.

[10] 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.

[11] S. Jain, F. Khorrami, and B. Fardanesh, "Adaptive nonlinear excitation control of power systems with unknown interconnections", *IEEE Transactions on Control Systems Technology*, Vol. 2, No. 4, pp. 436-446, 1994.

[12] H. Jiang, H. Cai, J.F. Dorsey, and Z. Qu, "Toward a globally robust decentralized control for large-scale power systems", *IEEE Transactions on Control Systems Technology*, Vo. 5, No. 3, pp. 309-319, May 1997.

[13] Y. Guo, D.J. Hill, and Y. Wang, "Nonlinear decentralized control of large scale power systems", *Automatica*, Vol. 36, No. 9, pp. 1275-1289, 2000.

[14] L. Jiang, Q.H. Wu, and J.Y. Wen, "Decentralized nonlinear adaptive control of multimachine power systems via high-gain perturbation observer", *IEEE Transactions on Circuit Systems-I*, Vol. 51, No. 10, pp. October 2004.

[15] S. Huang, K.K. Tan, and T.H. Lee, "Decentralized control design for large-scale systems with strong interconnections using neural networks", *IEEE Transactions on Automatic Control*, Vol. 48, No. 5, pp. 805-810, 2003.

[16] Wenxin Liu, S. Jagannathan, G.K. Venayagamoorthy, Donald C. Wunsch II, and David A. Cartes, "Neural network based decentralized excitation control of large scale power systems", the 2006 International Joint Conference on Neural Networks, July 16-21, 2006, Vancouver, BC, Canada.

[17] 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.

[18] 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.

[19] P. W. Sauer and M. M. Pai, Power System Dynamics and Stability, Upper Saddle River, N.J, Prentice Hall, 1998.

[20] Wenxin Liu, S. Jagannathan, 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, August 9-10, 2004, Moscow, Idaho, USA.

[21] 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. 214-221January, 2002.

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

[23] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceeding of IEEE International Conference on Neural Networks (ICNN), Vol. IV, pp. 1942-1948, 1995.