An online adaptive PSS based on RBF neural network identifier

Kai Xu, Shanchao Liu
College of Information Science and Engineering
University of Chongqing Jiaotong
No. 66, XueFu Rd., Nan'an District, Chonqing
China
xkxjxwx@hotmail.com, lscxihuan@hotmail.com, 838490042@qq.com

Abstract: The design of a conventional power system stabilizer (PSS) based on linearized model cannot guarantee its performance in practical operating, so some intelligent techniques have been used. However, the parameters cannot update online in most of these mehods, thus the performance cannot be further improved. This paper adopts the design method of adaptive neural-fuzzy based power system stabilizer (ANFPSS). It consists of two separate neural networks, namely radial basis function neural network (RBFNN) identifier and adaptive neuro-fuzzy controller (ANFC). Meanwhile, particle swarm optimization (PSO) algorithm is used to obtain appropriate initial parameters of the RBFNN identifier. Based on the optimized initial parameters, the RBFNN online identifier provides a dynamic model of controlled plant and updates the adaptive link weights of the ANFC. Simulation results on a single machine infinite bus power system demonstrate that the proposed stabilizer is effective in damping low-frequency oscillations as well as improving system dynamic stability. In addition, the proposed approach provides superior performance when compared to IEEE PSS2B.

Key-Words: power system stabilizers, particle swarm optimization, neural network identifier, neuro-fuzzy controller.

1 Introduction

The application of a supplementary control signal in the excitation system of a generating unit can provide extra damping for the system and thus improve the unit’s dynamic performance. The power system stabilizer (PSS) can generate the supplementary control signal for the excitation system in result to damp the low frequency oscillations and help improve power system stability.

The conventional power system stabilizers (CPSS) are designed very extensively, using phase compensation techniques. The parameters of CPSS are calculated based on linearized model of the power system [1]. The application of CPSS for the improvement of small signal oscillation and the dynamic stability of a power system has been explained in the literature [2]. Different types of arrangement of lag-lead compensator based PSS are discussed in details [3]. However, power systems are dynamic and highly nonlinear. Therefore, CPSS performance may deteriorate under variations which result from nonlinear and time-variant characteristics of a controlled plant. Thus, the adaptive PSS with the nonlinear nature of the plant is required.

To improve the performance of CPSS, different intelligent techniques have been proposed for designing, such as fuzzy logic, artificial neural network, intelligent optimization and hybrid artificial intelligent techniques, which have been proposed in the literature [4]-[8]. The gain and time constants of PSS are optimized through different computational optimization techniques such as genetic algorithm, particle swarm optimization and bacteria foraging optimization, which have been explained in [9]-[14]. Usually, it is difficult to obtain fuzzy rules and adjust parameters online in fuzzy logic control. The tuning of CPSS parameters does not update the weights of neural networks online, so their performances highly depend on the quality of offline training samples. The intelligent optimization algorithm is used to determine the optimal parameters for CPSS by optimizing a cost function in an offline mode. Since above methods are based on a linearized model and the parameters are not updated online, therefore they lack satisfactory performance during practical operations.

The radial basis function neural network (RBFNN) has fast convergence, strong generalization, and simple structure, which can achieve higher accuracy than BP network. In this paper, a RBFNN identifier is used. The values of output weight w_j, center vector c_{ji} and basis width b_j in RBFNN have great influence on the identification performance. Currently, gradient descent method is widely used, which particularly depends on the selection of initial values of w_j、c_{ji} and b_j. The inappropriate initial values of the RBFNN will lead to a relatively slow convergent speed and the falling of local optimum. Finally, it will cause the deterioration in identification performance.

The commonly used learning methods are orthogonal least square, clustering and self-organizing mapping and so on. However, the traditional learning strategies of RBFNN have some shortcomings: they can only find the optimal solution in the local space to determine the parameters of the network structure. At present, it is difficult to obtain optimum values of the network structure parameters theoritically. Simultaneously, the process of learning is complexd. Currently, the papers in [15]-[16] optimize parameters in RBFNN by genetic algorithm. However, the reproduction, crossover and mutation bring problems of local optimization. With the increase of scale and complex of the problem, the training time in neutral network shows exponential growth.

In this thesis, an online adaptive PSS based on RBF neural network identifier is proposed. Firstly, we use particle swarm optimization (PSO) to obtain appropriate initial parameters of the RBFNN identifier. Secondly, based on the optimized initial parameters, the RBFNN identifier provides a dynamic model of the controlled plant and updates the adaptive link weights of the adaptive neuro-fuzzy controller (ANFC) online. Finally, the effectiveness of this method is proved by simulation experiments.

2 Power system model

The single machine infinite bus power system (SMIB) model used to evaluate the adaptive neural-fuzzy based power system stabilizer (ANFPSS) is shown in Fig. 1.

The SMIB consists of a synchronous generator, a turbine, a governor, an excitation system and a transmission line connected to an infinite bus. The model is built in Matlab/simulink environment using the power system blockset. In Fig. 1, P_{REF} is the mechanical power reference, P_{SV} is the feedback through the governor, T_{M} is the turbine output torque. V_{TREF} is terminal voltage reference, V_{T} is terminal voltage, V_{A} is the AVR output. P is the active power and Q is the reactive power at the generator terminal. $\triangle\omega$ is the speed deviation, V_{PSS} is the PSS output signal, V_{inf} is the infinite bus voltage.

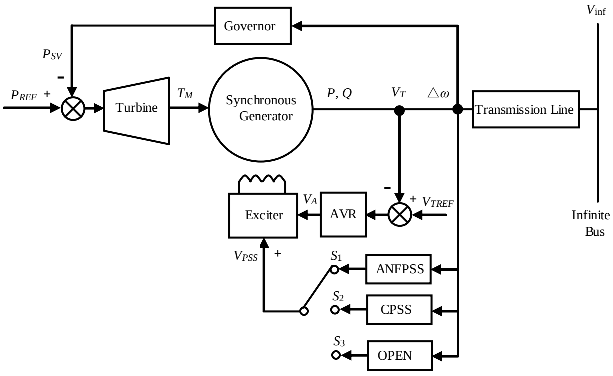


Fig. 1 System model configuration

In Fig. 1, the switch S_1、S_2、S_3 is used to carry out tests on the power system with ANFPSS, CPSS and without PSS respectively.

Parameters of the synchronous generator, hydraulic turbine and governor, automatic voltage regulator (AVR) and excitation system, equivalent circuit of the transformer used in the SMIB model are given in Appendix A.

3 Design of the ANFPSS

The online adaptive PSS design consists of two separate neural networks, namely radial basis function neural network (RBFNN) identifier and adaptive neuro-fuzzy controller (ANFC). The structure for the training of the RBFNN identifier and the ANFC is shown in Fig. 2. The dash lines show the backpropagation paths to update the parameters of the RBFNN identifier and the ANFC.

In this control architecture, the RBFNN identifier is used to track the dynamic characteristics of the plant, and the ANFC to damp the oscillations of the power system. The RBFNN identifier provided a dynamic model of the plant to evaluate and help in tuning the ANFC link weights online. The ANFC is used to generate the stabilizing supplementary control signal to the plant.

The RBFNN identifier, first trained offline by PSO. Then it is used for online object identification. In this way, the RBFNN identifier provides a dynamic model of the controlled plant. After that, the link weights of ANFC updating were performed in an online learning environment. Since both the RBFNN identifier and the ANFC are updated online, the ANFPSS can adapt to changes in the system and perform well under severe disturbances. These two phases are carried out in cascade once the ANFPSS is connected to the plant by placing the switch S_1 in Fig. 1.

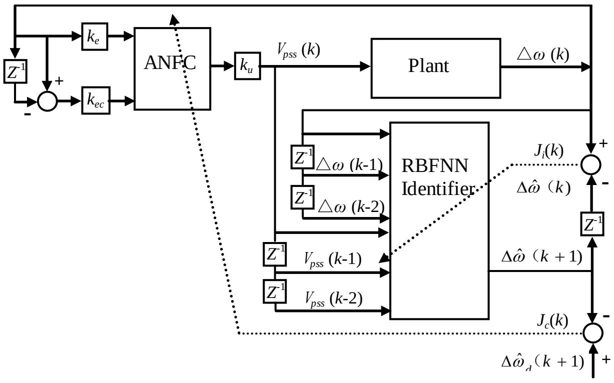


Fig. 2 Structure of the control system

3.1 RBFNN identifier

The RBFNN identifier is developed using the series-parallel nonlinear auto regressive moving average (NARMA) model. The model output $\hat{y}$ at time k+1 depends on both past *n* values of output and *m* past values of input. The neuro-identifier output equation takes the form given by the following formula:

$$\hat{y}(k+1) = f\begin{bmatrix} y(k), y(k-1),..., y(k-n+1) \\ u(k), u(k-1),..., u(k-m+1) \end{bmatrix} \quad (1)$$

Where $y(k)$ and $u(k)$ represent the output and input of the plant to be controlled at time *k*. For this particular system, *y*, *u* and $\hat{y}$ are the speed deviation $\Delta\omega$ of the plant, the output V_{pss} of the ANFC and the estimated plant output $\Delta\hat{\omega}$ by the RBFNN identifier respectively. Here both *m* and *n* are chosen to be 3. One reason for choosing three time step values is because a third order model of the system is sufficient for the study of transient stability. The other reason is that more time delays means more computation and one author's previous work verified that three time delays is enough for this kind of problem [17].

The RBFNN is a three-layer structural network, with an input layer, a hidden layer and an output layer. It has been proved that it can approximate any continuous network with arbitrary precision. The RBFNN mapping from input to output is nonlinear while the mapping from hidden layer to output is linear, which can improve learning speed greatly. The RBFNN learning algorithm is faster in convergence than the BP network and it can avoid local minimum problems. In the same conditions of training time, steps of implementation and required parameters, RBFNN can achieve higher accuracy than BP network. RBFNN has fast convergence, strong generalization, simple structure and other characteristics, so it is adopted in this paper. Fig. 3 shows the structure of RBFNN. The numbers of neurons in the input, hidden and output layers are six, ten and one respectively. All the inputs and outputs signals of the RBFNN identifier are normalized to the range of [-1,1].

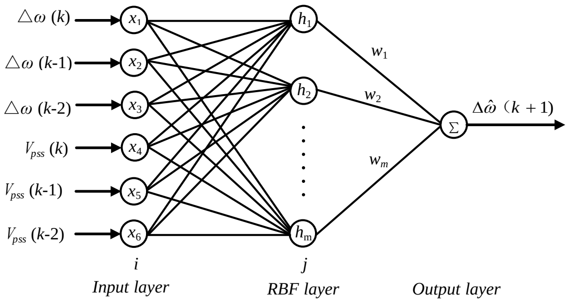


Fig. 3 Structure of RBFNN identifier

In the RBFNN identifier, $X = [x_1, x_2,...x_n]^T$ is the input vector, $H = [h_1, h_2,...h_m]^T$ is the radial basis vector, h_j is Gaussian function, which can be described as:

$$h_j = \exp\left(-\frac{\left\|X - C_j\right\|^2}{2b_j^2}\right), j = 1,2,...m \quad (2)$$

$C_j = [c_{j1}, c_{j2},...c_{ji}...c_{jn}]^T$ is the center vector of the j th node. $B = [b_1, b_2,...b_m]^T$ is the basis width vector. The weight vector of the network is given by

$$W = [w_1, w_2,...w_j...w_m]^T \quad (3)$$

The output vector of the network is given by

$$\Delta\hat{\omega}(k+1) = w_1h_1 + w_2h_2 + ... + w_mh_m \quad (4)$$

The RBFNN is used as an identifier to obtain the Jacobian information of controlled object. Training of the RBFNN identifier has two steps, offline training by PSO and online update.

3.2 RBFNN identifier offline optimization with PSO

PSO is a new method of intelligent group optimization, which has strong global search ability and is easy to be operated. In order to select appropriate initial parameters of w_j、c_{ji} and b_j in the RBFNN, PSO algorithm is applied here.

3.2.1 Particle swarm optimization algorithm

The PSO is an optimization algorithms based on collective intelligent theory, famous for its strong points as follows: quick convergence rate, strong global search ability, omitting the process of gradient computing. The basic PSO algorithms are as follows:

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 \times rand_1^k (pbest_{id}^k - x_{id}^k) + c_2 \times rand_2^k (gbest_d^k - x_{id}^k) \quad (5)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \quad (6)$$

w ---- inertia weight, usually 0.4~1.2;

v_{id}^k ---- the d dimensional velocity of the particle *i* for the *k* time;

c_1、c_2 ---- acceleration factor, which adjust the longest step width toward the best global particle and the best individual particle. Proper c_1、c_2 can accelerate the convergence rate and not get into the local optimum. Usually $c_1 = c_2 = 2$;

$rand_{1、2}^k$ ----any value among [0,1];

x_{id}^k ----the d dimensional position of the particle *i* for the *k* time;

$pbest_{id}^k$ ----the *d* dimensional position of the individual extreme point of the particle *i* for the *k* time;

$gbest_d^k$ ----the *d* dimensional position of global extreme point of the whole swarm for the *k* time;

v_d value range is $-v_{d\max}$ to $+v_{d\max}$ in order to prevent the particle from staying away from the search space.

For practical optimization problems, we often consider the global search first, and make the search space fast enough to converge to an area. Then, we make a careful local search to obtain the high accuracy solution. The study finds that a larger w can strengthen the global search ability, while a smaller w strengthen the local search ability. In this paper, a linear descending algorithms is adopted. w reduces linearly along with the increasing number of iteration. In this way the global search ability is greatly improved. The formula for w is as follows:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \quad (7)$$

$w_{\max}$ $w_{\min}$ ---- maximal and minimal inertia weight, usually $w_{\max} = 0.8 \sim 1.2$, $w_{\min} = 0.4$;

$iter_{\max}$ -----the maximum number of the iteration set;

$iter$ ---- the current number of iteration.

The experiment result shows that PSO has better convergence rate when w value is 0.8~1.2.

3.2.2 Offline optimization of RBFNN parameters with PSO

In the process of optimizing RBFNN identifier, the position vector x is defined as the output weight w_j, center vector c_{ji} and basis width b_j, and finally to construct particle swarm. Initialize x and search for the optimal position to get the minimal error of mean square:

$$J = E = \frac{1}{N}\sum_{i}^{N}(y_i^* - y_i)^2 \quad (8)$$

y_i^* ---- the expected output value of the particle *i*;

y_j ---- the real output value of the particle *i*;

N ---- the number of sample in the training set.

So, the fitness function is defined as:

$$f = 1/J \quad (9)$$

The calculation process of RBFNN identifier trained with PSO is given in Fig.4, which is described as follows:

Step 1: Collecting the training samples.

Step 2: Initialize the topological structure of RBFNN. It is supposed that the dimension of input vector X is n, and the radial basis vector H is m. Then the dimension of w_j、c_{ji} and b_j is m, $m*n$ and m, respectively, thus the total dimension of all parameters is $m(n+2)$.

Step 3: Generate initial condition of each agent. Each particle is constituted by parameters w_j、c_{ji} and b_j, then generate the initial location and velocity of a set of particles. Initialize inertia weight w, acceleration factor c_1、c_2, max iteration $N_{\max}$.The current searching point is set to pbest for each agent. The best-evaluated value of pbest is set to gbest and the agent number with the best value is stored.

Step 4: Calculate the fitness function value of each particle according to the formula (8)、(9).

Step 5: For each particle, compare the current fitness value with its corresponding the *pbest* fitness value, if better, update the *pbest.*

Step 6: For each particle, compare the current fitness value with the global optimum *gbest* fitness value, if better, update the *gbest*.

Step 7: Make iterative update of speed and position for all particles of population according to the formula (5)、(6).

Step 8: Checking the exit condition. If reach to the maximum iteration number $N_{\max}$ or the expected precision, then exit. Otherwise, continue iteration. Go to *step 4.*

Step 9: The individual that generates the latest is the optimal parameters of w_j、c_{ji} and b_j.

Step 10**:** Input the test samples to verify whether the RBFNN is able to meet the required functionality.

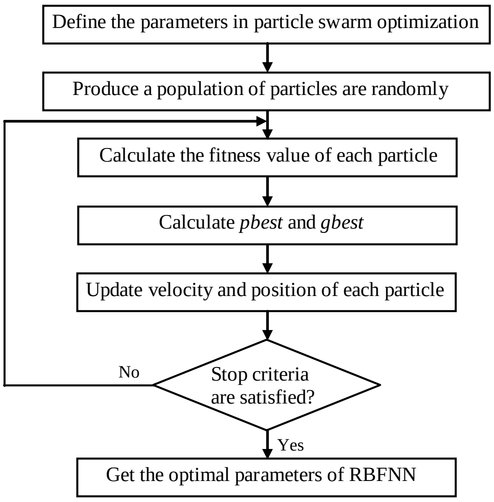

Define the parameters in particle swarm optimization
Produce a population of particles are randomly
Calculate the fitness value of each particle
Calculate *pbest* and *gbest*
Update velocity and position of each particle
Stop criteria are satisfied?
No
Yes
Get the optimal parameters of RBFNN

Fig. 4 The parameters optimization of RBFNN identifier based on PSO

In step1, training data for the RBFNN identifier is acquired from the system controlled by the CPSS, which is tune for each operating condition. The RBFNN identifier is trained offline over a wide working range of the generator operating conditions i.e. output ranging from 0.1 to 1.0 pu. Similarly, different kinds of disturbances are also included in the training, like change in reference voltage, governor input torque variation and three phase fault on one circuit of the double circuit transmission line.

3.3 RBFNN online identification of controlled object

The RBFNN for online system identification during control is shown in Fig. 2. The inputs to the RBFNN identifier during this phase are ($\Delta\omega(k)$, $\Delta\omega(k-1)$, $\Delta\omega(k-2)$, $V_{pss}(k)$, $V_{pss}(k-1)$, $V_{pss}(k-2)$) and its output is $\Delta\hat{\omega}(k+1)$. The desired output is the output of the plant $\Delta\omega(k)$. The cost function for training the RBFNN identifier is given by:

$$J_i(k) = \frac{1}{2}e_i(k)^2 = \frac{1}{2}[\Delta\omega(k) - \Delta\hat{\omega}(k)]^2 \qquad (10)$$

According to the gradient descent method, the output weight w_j, basis width b_j, center vector c_{ji} can be calculated from:

$$w_j(k) = w_j(k-1) + \eta(\Delta\omega(k) - \Delta\hat{\omega}(k))h_j \\ + \alpha(w_j(k-1) - w_j(k-2)) \qquad (11)$$

$$\Delta b_j = (\Delta\omega(k) - \Delta\hat{\omega}(k))w_j h_j \frac{\|X - C_j\|^2}{b_j^3} \qquad (12)$$

$$b_j(k) = b_j(k-1) + \eta\Delta b_j + \alpha(b_j(k-1) - b_j(k-2)) \qquad (13)$$

$$\Delta c_{ji} = (\Delta\omega(k) - \Delta\hat{\omega}(k))w_j \frac{x_j - c_{ji}}{b_j^2} \qquad (14)$$

$$c_{ji}(k) = c_{ji}(k-1) + \eta\Delta c_{ji} + \alpha(c_{ji}(k-1) - c_{ji}(k-2)) \qquad (15)$$

Where η is the learning speed, α is the momentum factor.

Jacobian matrix (sensitivity of plant output to controlled input) algorithm is as follows:

$$\frac{\partial\Delta\omega(k+1)}{\partial V_{pss}(k)} \approx \frac{\partial\Delta\hat{\omega}(k+1)}{\partial V_{pss}(k)} = \sum_{j=1}^{m} w_j h_j \frac{c_{ji} - V_{pss}(k)}{b_j^2} \qquad (16)$$

In the above formula, it is necessary to clarify the time step definition. Both $V_{pss}(k)$ and $\Delta\omega(k)$ signals are sampled at time step k, but $\Delta\omega(k)$ is not the response for the control signal $V_{pss}(k)$. Due to the time lag property of the plant, the impact of the control signal $V_{pss}(k)$ is reflected in the next time sample of the output signal $\Delta\omega(k+1)$.

3.4 Adaptive neuro-fuzzy controller (ANFC)

The structure of ANFC is shown in Fig. 5, it is a zero-order sugeno-type fuzzy controller. The inputs of the ANFC are the speed deviation $\Delta\omega$ and its derivative $\Delta\dot{\omega}$. The output is the control signal $V_{pss}(k)$. k_e and k_{ec} are the input scaling factors, and k_u is the proportion factor for the output. λ_1 and λ_2 are adaptive link weights between the input scaling factors and the input layer. The products $k_e\lambda_1$ and $k_{ec}\lambda_2$ represent the sensitivity of the controller for each assigned input. In this paper, it is used to make them adaptive through the online modification of the weights λ_1 and λ_2. It is need to work with only a small number of tuning parameters, regardless of the number and shape of input membership functions. The initial values for both λ_1 and λ_2 are set to 1.0.

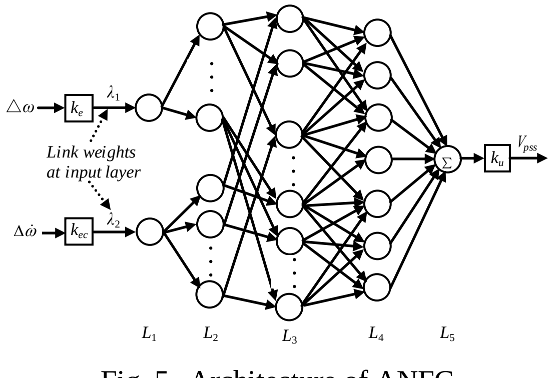


Fig. 5 Architecture of ANFC

As for the ANFC, the network has 2, 14, 49, 7 and 1 neurons for layers L_1, L_2, L_3, L_4 and L_5, respectively. Seven triangular membership functions are used with the following fuzzy sets for each input: NB (negative big), NM (negative medium), NS (negative small), ZO (zero), PS (positive small), PM (positive medium) and PB (positive big). The output variable also uses seven fuzzy sets to determine its value. The firing strength of each rule is calculated using the "product" operation. The rule base for the ANFC is given in Table 1. The fuzzy sets are normalized to the range of [-1,1].

Table 1 Control rules of ANFC.

$\Delta\dot{\omega}$ \ $\Delta\omega$	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZO
NM	NB	NB	NB	NM	NS	ZO	PS
NS	NB	NB	NM	NS	ZO	PS	PM
ZO	NB	NM	NS	ZO	PS	PM	PB
PS	NM	NS	ZO	PS	PM	PB	PB
PM	NS	ZO	PS	PM	PB	PB	PB
PB	ZO	PS	PM	PB	PB	PB	PB

Gradient descent method is employed to update these parameters according to the following cost function:

$$J_c(k)=\frac{1}{2}e_c(k+1)^2=\frac{1}{2}[\Delta\hat{\omega}(k+1)]^2 \qquad (17)$$

For the parameter λ_1, the gradient $\nabla_{\lambda 1}J_c(k)$ is

$$\nabla_{\lambda 1}J_c(k)=\left[\Delta\hat{\omega}(k+1)\frac{\partial\Delta\hat{\omega}(k+1)}{\partial V_{pss}(k)}\right]\left[\frac{\partial V_{pss}(k)}{\partial\lambda_1}\right] \qquad (18)$$

Where $\partial\Delta\hat{\omega}(k+1)/\partial V_{pss}(k)$, the Jacobian matrix of the plant, is obtained through the RBFNN identifier. It is the same with parameter λ_2.

4 Simulation results

In this section, the simulations are carried out in Matlab and Simulink environments for training the system introduced by Fig. 1. The simulation results of the proposed ANFPSS for SIMB system will be evaluated and compared with those of IEEE PSS2B and without PSS. For evaluating the robustness of the designed ANFPSS, performance of these PSS are simulated for different types of the disturbances.

4.1 Input mechanical power change

To verify the robustness of the proposed ANFPSS, a small disturbance was applied to the SIMB system.

A 5% increase in mechanical input is applied at 5 second and removed at 5.2 second, the response of rotor speed deviation and the line power variation are shown in Fig. 6.

As seen in Fig. 6, the open loop response without PSS is highly oscillatory. The speed deviation with PSS2B having large overshoots and settling time is 3.1 seconds. The ANFPSS having small overshoots and settling time is less than 2.2 seconds. From the comparison of three methods, it is indicated that the system oscillations of the proposed strategy are damped faster than those of the IEEE PSS2B and no PSS.

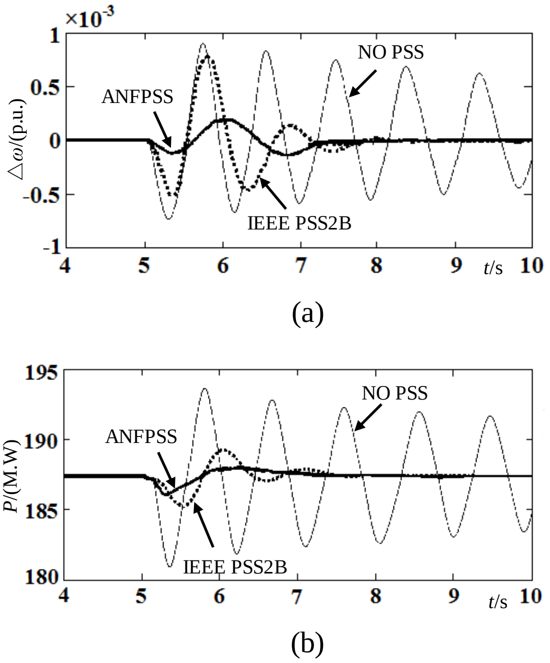
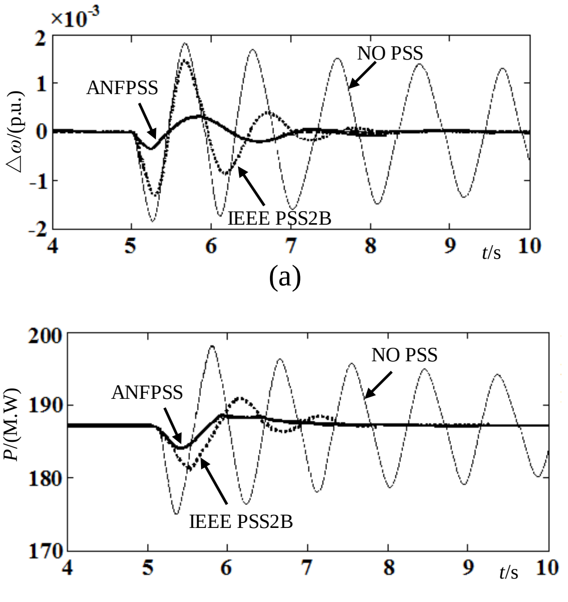


Fig. 6 Response to 5% increase in mechanical input a) Rotor speed deviation b) Line power variation

4.2 Input reference voltage change

A 5% increase in reference input voltage is applied at 5 second and removed at 5.2 second, the response of rotor speed deviation and the line power variation are shown in Fig. 7.

As seen in Fig. 7, the system without stabilizer is highly oscillatory. Although the IEEE PSS2B is effective in damping the oscillations, the overshoot is still large. The ANFPSS having less peak overshoot and quicker response, it settles the oscillations smoothly and quickly.



(a)

(b)

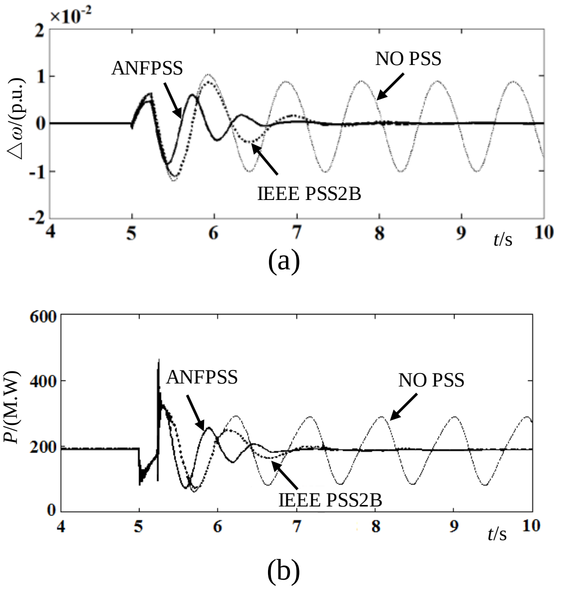
Fig. 7 Response to 5% increase in reference voltage
a) Rotor speed deviation b) Line power variation

4.3 Single phase to earth fault

In the next simulation, the responses of speed deviations and the line power variation when a large disturbance was applied were investigated.

A single phase to earth fault is created at 5 second at the sending end of the circuits of transmission line and cleared after 200ms. The response of rotor speed deviation and the line power variation are shown in Fig. 8.

As seen in Fig. 8, the settling time of IEEE PSS2B is 2.4 seconds, while the ANFPSS is less than 1.7 seconds. It is clearly shown that the oscillations and the setting time of the proposed strategy is better than that of the IEEE PSS2B. The overshoots and settling time are reduced to a greater extent compared to the open loop response without PSS.



(a)

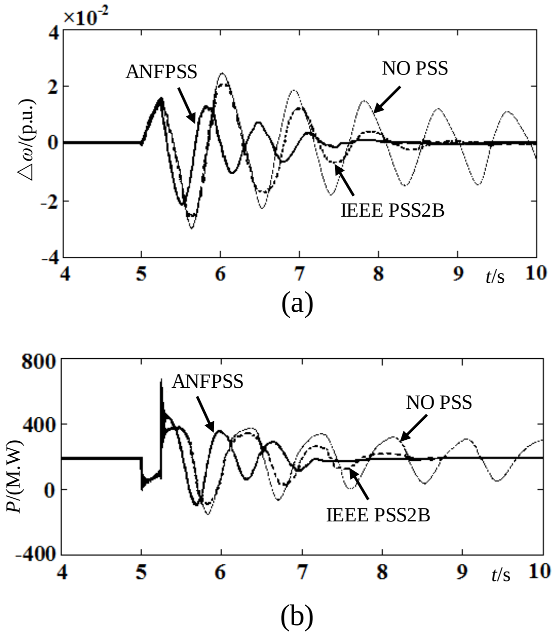
(b)

Fig. 8 Response to a single phase to earth fault
a) Rotor speed deviation b) Line power variation

4.4 Phase to phase fault

A phase to phase fault is created at 5 second at the sending end of the circuits of transmission line and cleared after 200ms. The response of rotor speed deviation and the line power variation are shown in Fig. 9.

It is clear that the proposed ANFPSS is more effective than IEEE PSS2B and no PSS both in better damping and quicker response.



(a)

(b)

Fig. 9. Response to a phase to phase fault
a) Rotor speed deviation b) Line power variation

5 Conclusion

The studies show that the ANFPSS has better performance in comparison with IEEE PSS2B. The conclusions of the paper can be summarized as follows:

1. The ANFPSS is an effective mean to enhance transient stability in SMIB. A large number of simulations for the different types of disturbances show that the ANFPSS prevents stability violations and damps oscillations in power system, which is more efficient in comparsion with IEEE PSS2B.

2. The proposed PSO algorithm provids a better solution to nonlinear system identification performance in the SMIB.

3. The design of RBFNN online identification and ANFC are quite robust and maintain a superior performance for different kinds of disturbances. This robustness is actually an inherent feature of

the RBFNN identifier and ANFC .

6 Acknowledgment

This work is supported by the Major Project of Postgraduate Education Reform of Chongqing (No. yjg131001).

Appendix A. System parameters

All data are in p.u. unless specified otherwise.
Synchronous generator
Rotor Type: Salient-pole, Pole Pairs: 32, Nominal Power: 200MVA, Line to Line Voltage: 13.8kV, *f*=50Hz, $x_d = 1.305$, $x_d^{'} = 0.296$, $x_d^{''} = 0.252$, $x_q = 0.474$, $x_q^{'} = 0.243$, $x_q^{''} = 0.18$, $T_d^{'} = 1.01s$, $T_d^{''} = 0.053s$, $T_{q0}^{''} = 0.1s$, $R_s = 2.8544 \times 10^{-3}$, Inertia Factor: 3.2, Friction Factor: 0.

Hydraulic turbine and governor (*HTG*)
Servo Motor: $K_A = 3.33$, $T_A = 0.07s$, $G_{\min} = 0.01$, $G_{\max} = 0.9752$, $V_{g\min} = -0.1pu/s$, $V_{g\max} = 0.1pu/s$; Permanent Droop and PID Regulator: $R_p = 0.05$, $K_p = 1.163$, $K_i = 0.105$, $K_d = 0$, $T_d = 0.01s$.
Hydraulic Turbine: $\beta = 0$, $T_w = 2.67s$.

AVR and excitation system
Low pass Filter Time Constant: $T_r = 0.02s$; Regulator Gain and Time Constant: $K_a = 300$, $T_a = 0.001s$; Exciter: $K_e = 1$, $T_e = 0s$; Transient Gain Reduction: $T_b = 0s$, $T_c = 0s$; Damping Filter Gain and Time Constant: $K_f = 0.001$, $T_f = 0.1s$; Regulator Output Limits and Gain: $E_{f\min} = -11.5$, $E_{f\max} = 11.5$, $K_p = 0$.

Power transformer
Nominal Power: 200MVA, 13.8/230kV, *f*=50Hz, D$_1$/Y$_g$ connection; Winding 1 and Winding 2 Parameters: $R_1 = 0.0027$, $L_1 = 0.08$, $R_2 = 0.0027$, $L_2 = 0.12$; Magnetizing Resistance and Reactance: $R_m = 500$, $L_m = 500$.

References:

[1] E. V. Larsen and D. A. Swann, “Applying power system stabilizers, parts I and II” , *IEEE Trans. PAS*, vol. 100, pp. 3017-3046, 1981.

[2] Michael J. Baler, Richard C. Schaefer, “Understanding power-system stability” , *IEEE Transactions on Industry Applications*, vol. 44, no. 2, pp. 37-47, April 2008.

[3] Kamwa R. Grondin, and G. Trudel, “IEEE PSS2B versus PSS4B: The limits of performance of modern power system stabilizers”, *IEEE Transactions on Power Systems*, Vol. 20, No. 2, pp. 903-915, May 2005.

[4] M. Soliman, A.L. Elshafei, F. Bendary, and W. Mansour, “LMI static output-feedback design of fuzzy power system stabilizers”, *Expert Systems with Applications*, Volume 36, pp. 6817-6825, 2009.

[5] M.Caner, N.Umurkan, S. Tokat, S.V. Ustun, “Determination of optimal hierarchical fuzzy controller parameters according to loading condition with ANN”, *Expert Systems with Applications*, Volume 34, pp. 2650-2655, 2008.

[6] Lokman H.Hassan, M.Moghavvemi, Haider AF. Almurib, K.M. Muttaqi, H.Du, “Damping of low-frequency oscillations and improving power system stability via auto-tuned PI stabilizer using Takagi-Sugeno fuzzy logic”, *International Journal of Electrical Power & Energy Systems*, Volume 38, No. 1, pp. 72-83, June 2012.

[7] S.M. Radaideh, I.M. Nejdawi, M.H. Mushtaha , “Design of power system stabilizers using two level fuzzy and adaptive neuro-fuzzy inference systems”, *International Journal of Electrical Power & Energy Systems*, Volume 35, No. 1, pp. 47-56, February 2012.

[8] A.T. Al-Awami, Y.L. Abdel-Magid, M.A. Abido, “A particle swarm-based approach of power system stability enhancement with unified power flow controller” , *Electric Power and Energy Systems*, Volume 29, pp. 251-259, 2007.

[9] Jesus Fraile-Ardanuy, P.J. Zufiria, “Design and comparison of adaptive power system stabilizers based on neural fuzzy networks and genetic algorithms”, *Neurocomputing*, Volume 70, Issues 16-18, pp. 2902-2912, October 2007.

[10] K. Sebaa, M. Boudour, “Optimal locations and tuning of robust power system stabilizer using genetic algorithms” , *Electric Power Systems Research*, Volume 79, pp. 406-416, 2009.

[11] A.M. El-Zonkoly , A.A. Khalil, and N.M. Ahmied, “Optimal tunning of lead-lag and fuzzy logic power system stabilizers using particle swarm optimization” , *Expert Systems with Applications,* Volume 36, pp. 2097-2106, 2009.

[12] A. Chatterjee, V. Mukherjee, S.P. Ghoshal, “Velocity relaxed and craziness-based swarm optimized intelligent PID and PSS controlled AVR system”, *International Journal of Electrical Power & Energy Systems*, Volume 31, Issues 7-8, pp. 323-333, September, 2009.

[13] S.P. Ghoshal, A. Chatterjee, V. Mukherjee,

"Bio-inspired fuzzy logic based tuning of power system stabilizer", *Expert Systems with Applications*, Volume 36, No. 5, pp. 9281-9292, 2009.

[14] Mishra, M.Tripathy, and J. Nanda, V. Mukherjee,"Multi-machine power system stabilizer design by rule based bacteria foraging" , *Electric Power Systems Research*, Volume 77, pp. 1595-1607, 2007.

[15] Torun, Mustafa Ugur, Gürkan Kuntalp, Damla , "Complexity reduction of RBF multiuser detector for DS-CDMA using a genetic algorithm", *Turkish journal of electrical engineering and computer sciences*, vol. 21, no. 4, pp. 1134–1150, 2013.

[16] Vizitiu, Iulian-Constantin, Ciotîrnae, Petrica, Oroian, Teofil, "Training of RFB neural networks using a full-genetic approach", *WSEAS Transactions on information science and applications*, vol. 7, no. 8, pp. 1015–1024, 2010.

[17] Venayagamoorthy, G. K. and Harley, "A continually online trained neurocontroller for excitation and turbine control of a turbogenerator", *IEEE Transactions on Energy Conversion*, vol. 16, no. 16, pp. 261–269, 2001.