**Proceedings of the 42nd IEEE
Conference on Decision and Control
Maui, Hawaii USA, December 2003**

FrE03-5

Robust Neural Network Tracking Controller Using Simultaneous Perturbation Stochastic Approximation

Q. Song
School of Electrical and Electronic Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798
eqsong@ntu.edu.sg

J. C. Spall
The Johns Hopkins University
Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, Maryland 20723-6099 USA
james.spall@jhuapl.edu

Y.C. Soh
School of Electrical and Electronic Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798
eycsoh@ntu.edu.sg

Keywords: SPSA, Conic Sector, Neural Network, Robust Control.

Abstract: This paper considers the problem of robust tracking controller design for a nonlinear plant in which the neural network is used in the closed-loop system to estimate the nonlinear system function. We introduce the conic sector theory to the design of the robust neural control system, with the aim of providing guaranteed boundedness for both the input-output signals and the weights of the neural network. The neural network is trained by the SPSA method instead of the standard back-propagation algorithm. The proposed neural control system guarantees the closed-loop stability of the estimation, and a good tracking performance. The performance improvement of the proposed system over existing systems can be quantified in terms of preventing weight shifts, fast convergence and robustness against system disturbance.

1. Introduction

Recently, there have been extensive research and significant progresses in the area of robust discrete time neural controller design for a class of nonlinear systems with specific nonlinear functions [1-6]. For example, a first-order approximation is applied in the convergence proof of [1-2] to deal with the nonlinear activation function. Variable structure and dead zone schemes have been introduced to design robust adaptive algorithms of neural network control systems to improve the tracking performance [3, 4, 5]. The well-known Persistent Exciting (PE) condition has been removed in the presence of disturbance [6]. On the other hand, Simultaneous Perturbation Stochastic Algorithm (SPSA) has been used recently as a model free control method for dynamical systems [7,8, 12,17].

In this paper, we shall propose a SPSA based neural control structure and a general stability proof for a nonlinear input-output dynamical plant. The plant under consideration is nonlinear and the neural network is used in the system to estimate the nonlinear function in closed-loop. Conic sector theory [10, 11,13] is introduced to design the robust control system, which aims to provide guaranteed boundedness for both the input-output signals and the weights of the neural network. One of the main advantages of the conic sector approach is that it is model free.. The neural controller is superior to its conventional adaptive control counterpart in the sense that the later requires linear in parameters for system estimation. The neural network is trained by the SPSA algorithm in the closed-loop to provide an improved training performance over the standard methods, such as the back-propagation algorithm, in terms of guaranteed stability of the weights, which in turn will yield good tracking performance for the dynamical control system. The main motivation for using the SPSA instead of the popular back-propagation algorithm is its excellent convergence property. The SPSA algorithm which was proposed by Spall [7] provides inherent drift-free parameter estimate through simultaneous perturbation of the weights.

Because stability is the primary concern of a closed-loop system, instead of a direct convergence analysis in [8], we follow the traditional approach of adaptive control systems to provide a robust input-output (I/O) stability design and analysis for the SPSA-based neural control system, and it does not require the weights to converge to the ideal values [9-11,14,19]. We apply the conic sector theory to isolate the SPSA algorithm from the rest of the closed-loop system. Unlike the robust conic sector analysis for a pre-trained neural network [16], we provide an on-line scheme for the robustness analysis of the neural control system. A special normalized cost function is provided to the SPSA algorithm to reject disturbance and solve the so-called vanished cone problem [11]. A two-stage normalized training strategy is proposed for the SPSA training with guaranteed I/O stability using the conic sector condition. The performance improvement of the proposed algorithm can be described in terms of preventing weight shifting, fast convergence and robustness against system disturbance.

2. Control system and SPSA training algorithm

A class of dynamical nonlinear plant, which has wide applications in robotics and variable air volume control systems [15,18], can be represented as an input-output form as follows:

$$y_k = f_{k-1} + u_{k-1} + \varepsilon_k, \tag{1}$$

where $y_k \in R^m$ is the output, $f_{k-1} \in R^m$ is a dynamic nonlinear function, $\varepsilon_k \in R^m$ denotes a bounded overall noise vector of the control system, and $u_{k-1} \in R^m$ is the control signal vector with unit time delay. The tracking error of the control system can be defined as:

$$s_k = y_k - y_k^*, \tag{2}$$

where $y_k^* \in R^m$ is the command input signal. Define the control signal as

$$u_{k-1} = -\hat{f}_{k-1} + y_k^* + k_v s_{k-1}, \tag{3}$$

where k_v is the gain parameter of the fixed controller and $\hat{f}_{k-1}$ is estimate of the nonlinear function f_{k-1} by the neural network to be defined in section 3.

Then the estimation error vector of the neural network can be presented as

$$e_k = f_{k-1} - \hat{f}_{k-1} + \varepsilon_k \tag{4}$$

to train the neural network as shown in Figure 1.

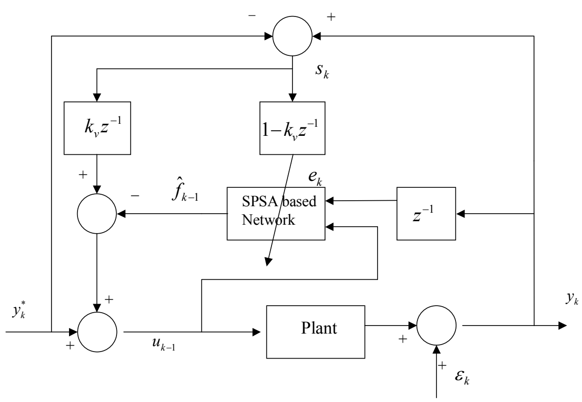

Figure 1. Structure of the control scheme.

Note however that the estimation error e_k may not be directly measurable, so we should use the tracking error to generate it by using the closed-loop relationship via (1), (3) and (4), i.e.

$$e_k = (1 - z^{-1} k_v) s_k. \tag{5}$$

The loss function for SPSA is defined as $L(\theta): R^p \to R^1$, where $\theta \in R^p$ is the parameter vector of the neural network. Consider the problem of finding the optimal parameter of the gradient equation

$$g(\theta) = \left.\frac{\partial L(\theta)}{\partial \theta}\right|_{\theta=\theta^*} = 0.$$

for the differentiable loss function $L(\theta)$.

Now we can define the SPSA algorithm to update $\hat{\theta}_k \in R^p$, which is an estimate of an optimal parameter vector θ^* as

$$\hat{\theta}_k = \hat{\theta}_{k-1} - a\hat{g}_k(\hat{\theta}_{k-1}), \tag{6}$$

where α is the learning rate and $\hat{g}(\hat{\theta}_{k-1})$ is the approximation of the gradient function with

$$\hat{g}(\hat{\theta}_{k-1}) = \frac{L(\hat{\theta}_{k-1} + c_k \Delta_k) - L(\hat{\theta}_{k-1} - c_k \Delta_k) + \varepsilon_k^\theta}{2c_k} r_k, \tag{7}$$

and where ε_k^θ is the measurement disturbance as defined in [7]. In the above equation, $\Delta_k \in R^p$ is a random directional vector, that is used to stimulate the weight vector simultaneously, $c_k > 0$ is a sequence of positive number satisfying certain regularity conditions [7-8]. The random vector Δ_k is generated via Monte Carlo according to conditions specified in [7] or [8]. If the *ith* element of Δ_k is denoted as Δ_{ki}, then the sequence of $r_k \in R^p$ is defined as

$$r_k = \left[\frac{1}{\Delta_{k1}} \quad ,..., \quad \frac{1}{\Delta_{kp}}\right].$$

The output of a three-layer neural network can be presented as

$$\hat{f}_{k-1} = H(\hat{\theta}_{k-1}^w, x_{k-1})\hat{\theta}_{k-1}^v, \tag{9}$$

where the input vector $x_{k-1} \in R^{n_i}$ of the neural network is

$$x_{k-1} = [x_{k-1,1} \; x_{k-1,2} \; ....x_{k-1,n_i}]^T = [y_{k-1}^T \; y_{k-2}^T \; ....]^T, \tag{10}$$

$\hat{\theta}_{k-1}^v \in R^{p_v}$ is the weight vector of the output layer, and $\hat{\theta}_{k-1}^w \in R^{p_w}$ is the weight vector of the hidden layer of the neural network with $p_v = m \times n_h$ and $p_w = n_h \times n_i$, and where n_i and n_h are the numbers of neurons in the input and hidden layers of the network, respectively. $H(\hat{\theta}_{k-1}^w, x_{k-1}) \in R^{m \times p_v}$ is the nonlinear activation function matrix

$$H(\hat{\theta}_{k-1}^w, x_{k-1}) = \begin{bmatrix} h_{k-1,1} \; h_{k-1,2} \; ... \; h_{k-1,n_h} & 0... & ...0 \\ 0... & h_{k-1,1} \; h_{k-1,2} \; ... \; h_{k-1,n_h} & ...0 \\ 0... & ..0.. & h_{k-1,1} \; h_{k-1,2} \; ... \; h_{k-1,n_h} \end{bmatrix},$$

where $h_{k-1,i}$ is the nonlinear activation function

$$h_{k-1,i} = h(x_{k-1}^T \hat{\theta}_{k-1,i}^w) = \frac{1}{1 + \exp(-4\lambda x_{k-1}^T \hat{\theta}_{k-1,i}^w)} \tag{12}$$

with $\hat{\theta}_{k-1,i}^w \in R^{n_i}$, $\hat{\theta}_{k-1}^w = [\hat{\theta}_{k-1,1}^{wT} ... \hat{\theta}_{k-1,n_h}^{wT}]^T$ and $4\lambda > 0$, which is the gain parameter of the threshold function that is required to derive the sector condition of the hidden layer.

3. Conic sector condition for robustness analysis and learning laws

The following theory is a necessary extension to the conic sector stability of Safanov [13] for discrete time control systems, like the one in Figure 1.

Theorem 1: Consider the following error feedback system

$$s_k = e_k^* - P_k$$
$$\Phi_k = H_1 s_k$$
$$P_k = H_2 \Phi_k$$

with operators H_1, H_2: $L_{2e} \to L_{2e}$ and discrete time signal $s_k, P_k, \Phi_k \in L_{2e}$ and $e_k^* \in L_2$.

If

(a) $H_1 : s_k \to \Phi_k$ satisfies

$$\sum_{k=1}^{N} [s_k^T \Phi_k + \sigma s_k^T s_k / 2] > -\gamma,$$

(b) $H_2 : \Phi_k \to P_k$ satisfies

$$\sum_{k=1}^{N}[\sigma P_k^T P_k / 2 - P_k^T \Phi_k] \le -\eta \left\|(P_k, \Phi_k)\right\|_N^2$$

for some $\sigma, \gamma, \eta > 0$, then the above feedback system is stable with $s_k, \Phi_k \in L_2$.

Proof: See Corollary 2.1 [11].

Note that operator H_1 represents the SPSA training algorithm, the input error signal is the tracking error s_k defined in (2) and the output is Φ_k, which will be defined later and is related to the weight error vectors, and in turn, the estimation parameter error vector e_k and tracking error s_k through (5). H_2 usually represents the mismatched linear model uncertainty in a typical adaptive linear control system and will be defined later in the next section [9].

The first step to use the conic sector stability theorem 1 is to restructure the control system in Figure 1 into an equivalent error feedback system as shown in Figure 2. Then the parameter estimation error vector should be derived and referred to the output signal Φ_k. For this purpose, define the desired output of the neural network as the plant nonlinear function in (1), i.e.

$$f_{k-1} = H(\hat{\theta}_{k-1}^{w*}, x_{k-1})\theta^{v*}, \tag{13}$$

where $\theta^{v*} \in R^{p_v}$ is an ideal weight vector in the output layer of the neural network, $\theta^{w*} \in R^{p_w}$ is the desired weight vector of the hidden layer. Therefore, the parameter estimate error vectors can be defined as $\tilde{\theta}_k^v = \theta^{v*} - \hat{\theta}_k^v \in R^{p_v}$ and $\tilde{\theta}_k^w = \theta^{w*} - \hat{\theta}_k^w \in R^{p_w}$ for the output and hidden layers, respectively.

Assumptions:

a) The system disturbance ε_k defined in (1) is bounded;
b) The ideal weight vectors θ^{v*} and θ^{w*} are bounded above.

Now, we are ready to establish the relationship between the tracking error signal s_k and the parameter estimate vectors of the neural network, which is referred to as the operator H_1, in Theorem 1, i.e. the SPSA algorithm. According to equations (4) and (5) the error signals can be extended as

$$\begin{aligned} s_k &= H_2 e_k = H_2(f_{k-1} - \hat{f}_{k-1} + \varepsilon_k) \\ &= H_2(\frac{1}{2}(f_{k-1} - \hat{f}_{k-1}^{v+}) + \frac{1}{2}(f_{k-1} - \hat{f}_{k-1}^{v-}) + \varepsilon_k) \\ &= H_2 H(\hat{\theta}_{k-1}^w, x_{k-1})\tilde{\theta}_{k-1}^v + \tilde{e}_k^v \\ &= -H_2 \Phi_k^v + \tilde{e}_k^v, \end{aligned} \tag{14}$$

where the operator $H_2 = \dfrac{1}{1-k_v z^{-1}}$, and

$$\Phi_k^v = -H(\hat{\theta}_{k-1}^w, x_{k-1})\tilde{\theta}_{k-1}^v, \tag{15}$$

$$\tilde{e}_k^v = H_2(\tilde{H}(\hat{\theta}_{k-1}^w, \theta^{w*}, x_{k-1}) + \varepsilon_k), \tag{16}$$

$$\tilde{H}(\hat{\theta}_{k-1}^w, \theta^{w*}, x_{k-1}) = (H(\theta^{w*}, x_{k-1}) - H(\hat{\theta}_{k-1}^w x_{k-1}))\theta^{v*}, \tag{17}$$

$$\hat{f}_{k-1}^{v+} = H(\hat{\theta}_{k-1}^w, x_{k-1})(\hat{\theta}_{k-1}^v + c_k \Delta_k^v), \tag{18}$$

$$\hat{f}_{k-1}^{v-} = H(\hat{\theta}_{k-1}^w, x_{k-1})(\hat{\theta}_{k-1}^v - c_k \Delta_k^v), \tag{19}$$

with Δ_k^v and $r_k^v \in R^{p_v}$, which can be viewed as the first p_v components of Δ_k and r_k defined in (7) and (8) for the SPSA, respectively.

Remark 1: There is an important implication in equation (14). The tracking error signal s_k is directly linked to the output signal Φ_k^v in equation (15), and in turn, the parameter estimation error vector $\tilde{\theta}_k^v$ of the output layer of the neural network, which implies that the training procedure of the output layer of the neural network should be treated separately from the hidden layer of the network to obtain a bounded disturbance term $\tilde{e}_k^v$ as defined in (16), i.e. $\tilde{e}_k^v \in L_2$ as required by Theorem 1 (defined as e_k^*). Therefore, using equation (14), we are able to form an equivalent error feedback system Figure 2 as the one in Theorem 1. Note that H_2 usually represents the mismatched linear model uncertainty in a typical adaptive linear control system [9]. Since the neural network has powerful approximation ability to match the nonlinear function without the need to worry about the linear model mismatch, therefore, the operator $H_2 = \dfrac{1}{1-k_v z^{-1}}$ represents only the fixed controller and is always stable as $|k_v| < 1$. Furthermore, the condition (b) of Theorem 1 can be treated as positive real function, i.e. the plot of H_2 should be in the positive half of a complex plane as shown in [10].

We define the operator H_1^v, which represents the SPSA training algorithm of the output layer, and the loss function as

$$L(\hat{\theta}_{k-1}^v, \hat{\theta}_{k-1}^w) = \left\| f_k - \hat{f}_k \right\|^2,$$

and with definitions in equation (7), (18) and (19), we have a normalized gradient approximation using the simultaneous perturbation vector $\Delta_k^v \in R^{p_v}$ to stimulate the weight of the output layer:

$$\hat{g}(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^v, \Delta_k^v) = -\frac{s_k^T (1 - z^{-1}k_v) H(\hat{\theta}_{k-1}^w, x_{k-1}) 2\Delta_k^v}{\rho_k^v} r_k^v, \tag{20}$$

where the bounded normalization factor, which is traditionally used in adaptive control system to bound the signals in learning algorithms [11], is

$$\rho_k^v = \mu\rho_{k-1}^v + \max\left(\frac{\bar{\Phi}^v}{k_v} + 2(1+\bar{\Phi}^v)\frac{a_k}{p_v}\left\|H(\hat{\theta}_{k-1}^w, x_{k-1})\right\|^2 \left\|\Delta_k^v\right\|^2 \left\|r_k^v\right\|^2, \bar{\rho}\right)$$

with $\bar{\rho} > 0, \mu \in (0,1)$.

ε_k^v is the bounded measurement error and has a relationship with the overall system disturbance ε_k in (1) as

$$\begin{aligned} \varepsilon_k &= \left\|\hat{f}_{k-1}^{v+} - \hat{f}_{k-1}^{v-}\right\|^{-2} (\hat{f}_{k-1}^{v+} - \hat{f}_{k-1}^{v-})\varepsilon_k^v \\ &= \frac{(H(\hat{\theta}_{k-1}^w, x_{k-1})\Delta_k^v)\varepsilon_k^v}{\left\|H(\hat{\theta}_{k-1}^w, x_{k-1})\Delta_k^v\right\|^2 2c_k}, \end{aligned} \tag{21}$$

which is also bounded.

Note that the gradient approximation function in (20) can be used only for the output layer as justified in remark 1. Therefore, the parameter vector

$\hat{\theta}_k^v$ can be estimated by the SPSA training algorithm in equation (6) with the gradient approximation $\hat{g}(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^v, \Delta_k^v)$, i.e.

$$\begin{aligned}\hat{\theta}_k^v &= \hat{\theta}_k^v - \alpha_k \hat{g}(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^v, \Delta_k^v) \\ &= \hat{\theta}_{k-1}^v + \alpha_k \frac{s_k^T(1-z^{-1}k_v)H(\hat{\theta}_{k-1}^w, x_{k-1})2\Delta_k^v}{\rho_k^v} r_k^v.\end{aligned} \tag{22}$$

Then stability analysis of the robust neural controller can be justified by the conic sector condition, which requires the feedback system in Figure 2 to meet certain dissipative condition as in Theorem 1 and can be justified as in the following theorem.

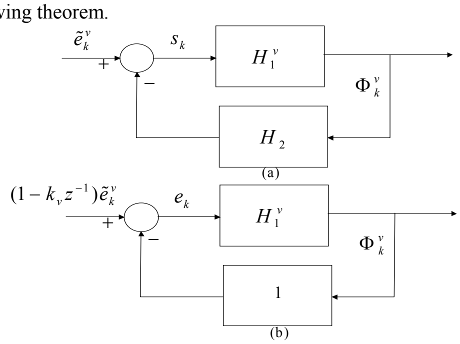

Figure 2. The equivalant error feedback systems using the conic sector condition: (a) For the tracking s_k and the output $\Phi_k^v = -H(\hat{\theta}_{k-1}^w, x_{k-1})\tilde{\theta}_k^v$; (b) For the estimate error e_k and the output $\Phi_k^v = -H(\hat{\theta}_{k-1}^w, x_{k-1})\tilde{\theta}_k^v$.

Theorem 2: The operator $H_1^v : s_k \to \Phi_k^v$, which represents the SPSA learning algorithm of for the output layer (see Figure 2 (a)), satisfies the conditions (a) and (b) of Theorem 1, i.e. $s_k, \Phi_k^v \in L_2$.

Proof: (see [20]).

Similar approach can be applied to the hidden layer for updating law and robustness analysis [20] to yield

$$\hat{\theta}_k^w = \hat{\theta}_k^w - \alpha_k \hat{g}(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^v, \Delta_k^v) = \hat{\theta}_{k-1}^w + \alpha_k \frac{s_k^T(1-k_v z^{-1})(f_{k-1}^{w+} - f_{k-1}^{w-})}{c_k \rho_k^w} r_k^w. \tag{23}$$

The robust neural controller algorithm can be summarized as (refer to Figure 1):

Step 1: Form the new input vector x_{k-1} of the neural network defined in equation (10);

Step 2: Calculate the neural network output $\hat{f}_{k-1}$ by using the input state x_{k-1} and the existing or initial weights of the network in the first iteration;

Step 3: The control input u_{k-1} is calculated based on equation (3);

Step 4: The new measurement of the system dynamics is taken and the measurable tracking error signal s_k is fed through a fixed filter to produce the implicit training error signal e_k of the network according to equation (5);

Step 5: The tracking error s_k is used directly to train the neural network and calculates the new weights $\hat{\theta}_k^v$ and $\hat{\theta}_k^w$ by using learning law in (22) and (23) for the output and hidden layers, respectively, of the next iteration;

Step 6: Go back to Step 1 to continue the iteration.

4. Simulation Results

Consider a two-link direct drive robot model with an input-output discrete-time version obtained from Euler's rule as follows [15, 19]:

$$y_k = \begin{bmatrix} y_{k,1} \\ y_{k,2} \end{bmatrix} = f_{k-1}(y_{k-1}, y_{k-2}) + u_{k-1} + \varepsilon_k ,$$

where $y_{k,1}$ and $y_{k,2}$ are the joint velocities of the two links, respectively, T is the sampling period, u_{k-1} is the toque control signal, ε_k is a normally distributed disturbance with a bound $\|\varepsilon_k\| \le 0.2$, and the nonlinear function

$$f_{k-1}(y_{k-1}, y_{k-2}) = y_{k-1} - TM^{-1}(y_{k-1,1})[V(y_{k-1}, y_{k-2}) + F(y_{k-1}, y_{k-2})]$$

with

$M(y_{k-1,1}) = \begin{bmatrix} 3.32 + 0.32\cos(y_{k-1,1}) & 0.12 + 0.16\cos(y_{k-1,1}) \\ 0.12 + 0.16\cos(y_{k-1,1}) & 0.12 \end{bmatrix}$ (configuration dependent inertia matrix),

$V(y_{k-1}, y_{k-2}) = \begin{bmatrix} -(y_{k-1,2} - y_{k-2,2})(2(y_{k-1,1} - y_{k-2,1}) + y_{k-1,2} - y_{k-2,2})0.16\sin(y_{k-2,2})/T^2 \\ 0.16(y_{k-1,1} - y_{k-2,1})^2 \sin(y_{k-2,2})/T^2 \end{bmatrix}$

(centrifugal and coriolis effect),

$F(y_{k-1}, y_{k-2}) = \begin{bmatrix} 5.3\,\mathrm{sgn}((y_{k-1,1} - y_{k-2,1})/T) \\ 1.1\,\mathrm{sgn}((y_{k-1,2} - y_{k-2,2})/T) \end{bmatrix}$ (coulomb friction).

A three-layered neural network is used as defined in section 2 for this simulation study with 30 hidden layer neurons and two output neurons, which was trained by the standard back-propagation and SPSA training algorithm with the same control structure as shown in Figure 1. The desired joint velocity trajectory is selected as

$$y_k^* = \begin{bmatrix} y_{k,1}^* \\ y_{k,2}^* \end{bmatrix} = \begin{bmatrix} (\pi/4)\sin(\pi \cdot kT) \\ (\pi/4)\cos(\pi \cdot kT) \end{bmatrix}.$$

The sampling period is $T = 0.002$sec. The linear gain parameter of the fixed controller was given as $k_v = 0.5$ and all the initial conditions are chosen to be zero.

Figures 3 and 4 show the outputs of the plant of the robust SPSA based neural controller, in which the tracking errors are relatively small as a result of the draft-free weight $\hat{\theta}_{k,1}$, which is the first element of the weight vector $\hat{\theta}_k$ as shown in Figure 5 (up to 50 seconds to highlight the trend). In contrast, the neural controller using the standard back-propagation algorithm performs poorer because there is parameter drift and relatively larger tracking errors are achieved (see Figures 6, 7 and 8). For the purpose of comparison, the outputs of the same plant under a finely tuned PID controller are also shown in Figure 9 and 10. Again, the robust SPSA based neural controller performs better.

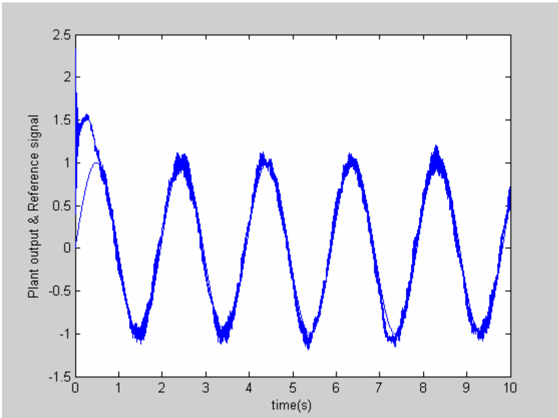

Figure 3. Output $y_{k,1}$ and reference signal $y^*_{k,1}$ using the robust SPSA based neural controller.

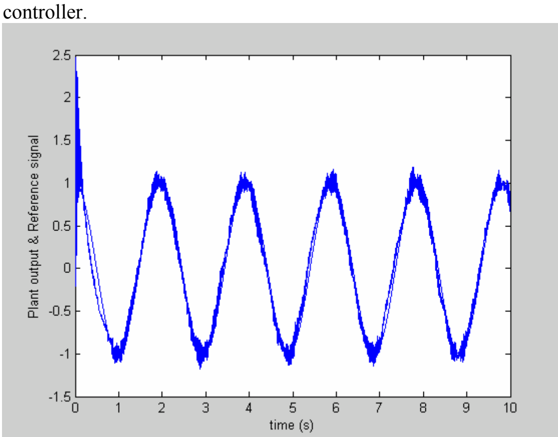

Figure 4. Output $y_{k,2}$ and reference signal $y^*_{k,2}$ using the robust SPSA based algorithm.

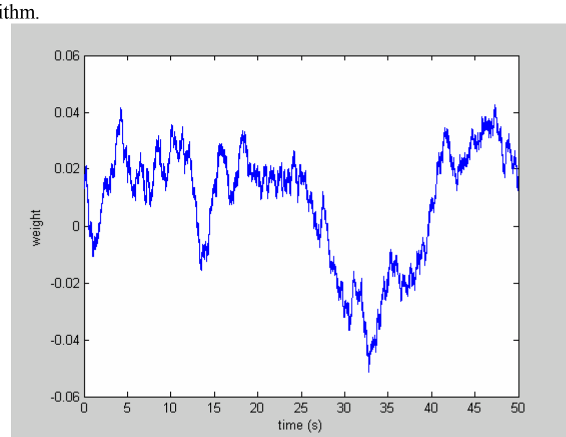

Figure 5. Estimated parameter $\hat{\theta}_{k,1}$ of the hidden layer using the robust SPSA based algorithm.

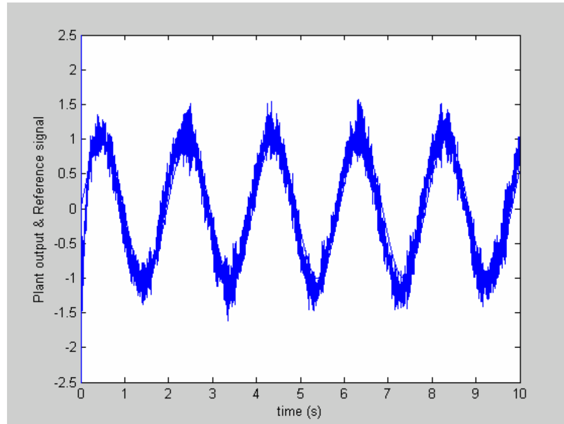

Figure 6. Output $y_{k,1}$ and reference signal $y^*_{k,1}$ using the standard back-propagation algorithm.

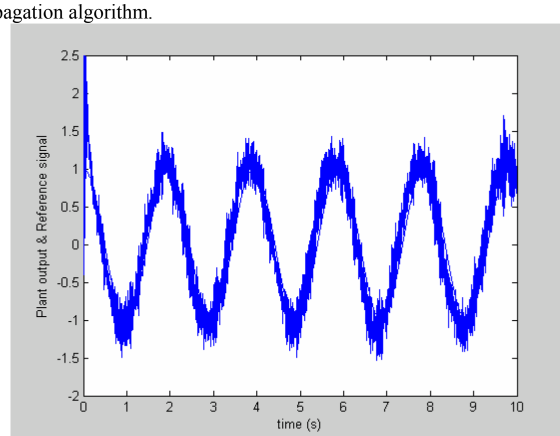

Figure 7. Output $y_{k,2}$ and reference signal $y^*_{k,2}$ using the standard back-propagation algorithm.

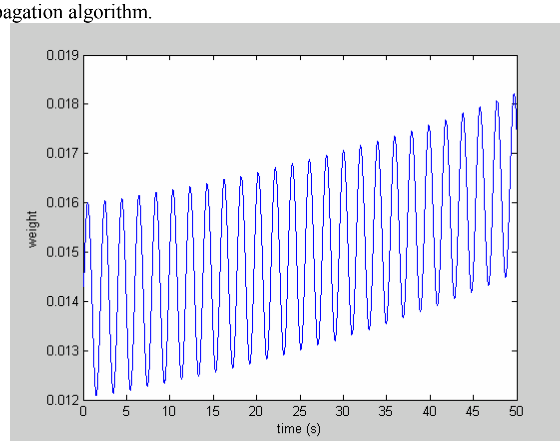

Figure 8. Estimated parameter $\hat{\theta}_{k,1}$ of the hidden layer using the standard back-propagation algorithm.

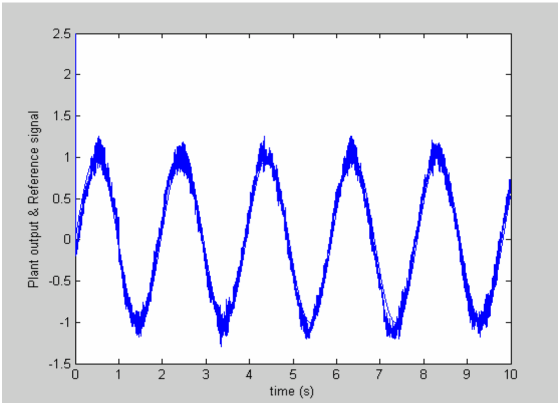

Figure 9. Output $y_{k,1}$ and reference signal $y^*_{k,1}$ using a fixed PID controller with the optimal parameter.

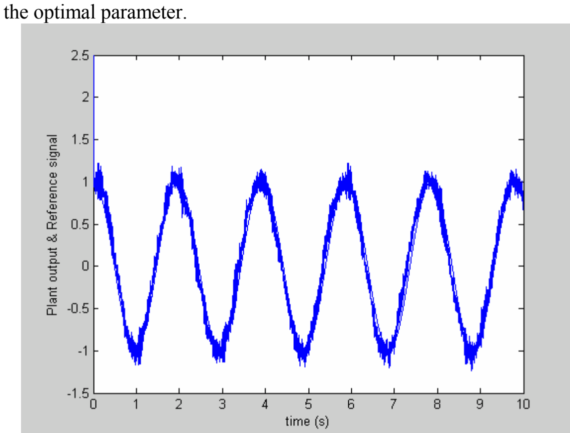

Figure 10. Output $y_{k,2}$ and reference signal $y^*_{k,2}$ using a fixed PID controller with the optimal parameter.

5. Conclusions

The robust neural controller based on the SPSA has been developed to obtain the guaranteed stability with a normalized learning algorithm. A complete stability analysis is performed for the closed loop control system. Simulation results show that the proposed robust neural controller performs better than both the neural controller based on the standard back-propagation algorithm and the PID controller.

References

[1] M.M. Polycarpou and P.A. Ioannou, "Learning and convergence analysis of neural-type structured networks", *IEEE Trans. On Neural Networks*, Vol.3, No.1, pp.39-50, 1992.

[2] F. Sun, Z. Sun and P. Woo, "Stable neural-network-based adaptive control for sampled-data nonlinear systems," *IEEE Tans. On Neural Networks*, Vol.9, No.5, pp.956-968, 1998.

[3] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators,* McMillan, 1993.

[4] F.C. Chen and H.K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks", *IEEE Trans. On Automatic Control*, Vol.40, No.5, pp.791-801, 1995.

[5] Q. Song, J. Xiao, and Y. C. Soh, "Robust Back-propagation Training Algorithm for Multi-layered Neural Tracking Controller", *IEEE Trans. On Neural Networks,* Vol.10, No.5, pp.1133-1141, 1999.

[6] S. Jagannathan and F.L. Lewis, " Discrete-time neural net controller for a class of nonlinear dynamic systems", *IEEE Trans. On Automatic Control*, Vol. 41, No.11., pp.1693-1699, 1996.

[7] J.C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation", *IEEE Trans. On Automatic Control*, Vol. 37, pp. 332-341, 1992.

[8] J.C. Spall and J.A. Cristion, "Model Free Control of Nonlinear Stochastic Systems with Discrete-Time Measurements", *IEEE Trans. On Automatic Control*, Vol. 43, No.9, pp. 1198-1210, 1998.

[9] R. Ortega, L. Praly and I.D. Landau, "Robustness of discrete-time direct adaptive controllers," *IEEE Trans. Automatic Control*, Vol.30, No.12, pp. 1179-1187, 1985.

[10] Q. Song, R. Katebi and M.J. Grimble, "Robust multivariable implicit adaptive control," *IMA Journal of Mathematical Control & Information*, Vol.10, pp. 49-70, 1993.

[11] V.R. Cluett, L. Shah and G. Fisher, "Robustness Analysis of Discrete-Time Adaptive Control Systems Using Input-Output Stability Theory: a Tutorial', *IEE Proceedings, Part-D*, Vol. 135, No. 2, pp.1179-1187, 1988, pp.133-141.

[12] I.J. Wang and E.K.P. Chong, " A deterministic analysis of stochastic approximation with randomized directions", *IEEE Trans. Automatic Control*, Vol. 43, No.12, pp. 1745-1749, 1998.

[13] M.G. Safanov, "Stability and robustness of multivariable feedback systems", MIT Press, 1980.

[14] M. M. Polycarpou and P.A. Ioannou, "Learning and convergence analysis of neural-type structured networks", *IEEE Trans. On Neural Networks*, Vo3., No.1, pp.39-50, 1992.

[15] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, Control of Robot Manipulators*,* McMillan, 1993.

[16] J. F. D. Canete, A. Barreiro, A. G. Cerezo, and I. G. Moral, "An input-output based stabilization criterion for neural-network control of nonlinear systems", *IEEE Trans. On Neural Networks*, Vo12, No.6, pp.1491-1497, 2001.

[17] J.C. Spall and J.A. Cristion, "A neural network controller for systems with unmodeled dynamics with applications to water plant treatment", *IEEE Trans. On Systems, Man and Cybernetics, Part B*, Vol. 27, pp. 369-375, 1997.

[18] Q. Song, W.J. Hu and T.N. Zhao, "Robust neural controller for variable airflow volume system", *IEE Proc.-Control Theory Appli*., Vol. 150, No. 2, 2003, pp. 112-118.

[19] Q. Song, L. Yin, "Robust fault accommodation for a robot system using a radial basis function neural network", *International Journal of System Science*, Vol. 32, No.2, 2001, pp.195-204.

[20] Q.Song, J. C. Spall and Y.C. Soh, "Robust neural network tracking controller using simultaneous perturbation stochastic approximation ", submitted to the *IEEE Trans. On Neural Networks*, 2003.