

Robust Neural Network Tracking Controller Using Simultaneous Perturbation Stochastic Approximation

Qing Song, *Member, IEEE*, James C. Spall, *Fellow, IEEE*, Yeng Chai Soh, *Member, IEEE*, and Jie Ni, *Student Member, IEEE*

Abstract—This paper considers the design of robust neural network tracking controllers for nonlinear systems. The neural network is used in the closed-loop system to estimate the nonlinear system function. We introduce the conic sector theory to establish a robust neural control system, with guaranteed boundedness for both the input/output (I/O) signals and the weights of the neural network. The neural network is trained by the simultaneous perturbation stochastic approximation (SPSA) method instead of the standard backpropagation (BP) algorithm. The proposed neural control system guarantees closed-loop stability of the estimation system, 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.

Index Terms—Conic sector, dead zone, neural network, simultaneous perturbation stochastic approximation (SPSA).

I. MAIN NOTATION LIST

k	Discrete time step.
m	Dimension of the plant input/output (I/O) variables.
l	Number of time-delayed plant outputs.
n	Number of time-delayed plant inputs.
n_I	$= (l + n - 1) \times m$. Number of input layer neurons.
n_h	Number of hidden layer neurons.
p^v	$= m \times n_h$. Dimension of the output layer weight vector.
p^w	$= n_I \times n_h$. Dimension of the hidden layer weight vector.
p	$= p^w + p^v$. Dimension of the total weight vector.
u_k	$\in R^m$. Plant input vector.

y_k^*, y_k	$\in R^m$. Desired and plant output vectors, respectively.
ε_k	$\in R^m$. Total disturbance vector.
x_{k-1}	$\in R^{n_I}$. Neural network input vector.
\hat{f}_{k-1}, f_{k-1}	$\in R^m$. Estimated and plant nonlinear functions, respectively.
$\hat{\theta}_{k-1}^w, \theta^{w*}, \tilde{\theta}_{k-1}^w$	$\in R^{p^w}$. Estimated weight, ideal weight, and weight estimate error vectors of the hidden layer, respectively.
$\hat{\theta}_{k-1,i}^{w_I}, \theta_i^{w_I*}, \tilde{\theta}_{k-1,i}^{w_I}$	$\in R^{n_I}$; $1 \leq i \leq n_h$. Estimated weight, ideal weight, and weight estimate error vectors of the i th hidden layer neuron, respectively.
$\hat{\theta}_{k-1}^v, \theta^{v*}, \tilde{\theta}_{k-1}^v$	$\in R^{p^v}$. Estimated weight, ideal weight, and weight estimate error vectors of the output layer, respectively.
$\hat{\theta}_{k-1,i}^v$	$\in R$; $1 \leq i \leq p^v$. Scalar component of the estimated weight vector $\hat{\theta}_{k-1}^v$.
$\hat{\theta}_{k-1,i}^{v_m}$	$\in R^m$; $1 \leq i \leq n_h$. Estimated weight vector of the output layer with inputs from the i th hidden layer neuron.
s_k	$\in R^m$. Tracking error vector of the control system.
e_k	$\in R^m$. Estimation error vector.
\bar{e}_k^v, \bar{e}_k^w	$\in R^m$. Normalized estimation error vectors of output and hidden layers, respectively.
$\tilde{e}_k^v, \tilde{e}_k^w$	$\in R^m$. Equivalent disturbances of the conic sector analysis for output and hidden layers, respectively.
δ_k^v, δ_k^w	$\in R$. Equivalent approximation errors of output and hidden layers, respectively.
k_v	Gain parameter of the proportional controller.
c_k	Perturbation gain parameter of the SPSA.
Δ_k^v, r_k^v	$\in R^{p^v}$. Perturbation vectors of the output layer.
Δ_k^w, r_k^w	$\in R^{p^w}$. Perturbation vectors of the hidden layer.

Manuscript received June 8, 2006; revised March 8, 2007 and October 17, 2007; accepted October 17, 2007. The theoretical work was initialized by Q. Song, on leave from Johns Hopkins University, Applied Physics Laboratory, in 2002.

Q. Song, Y. C. Soh, and J. Ni are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: eqsong@ntu.edu.sg; eycsoh@ntu.edu.sg).

J. C. Spall is with the Applied Physics Laboratory, Johns Hopkins University, Laurel, MD 20723-6099 USA (e-mail: james.spall@jhuapl.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNN.2007.912315

$H(\hat{\theta}_{k-1}^w, x_{k-1})$	$\in R^{m \times p^v}$. Nonlinear activation function matrix.
$h_{k-1,i}$	$= h(x_{k-1}^T \hat{\theta}_{k-1}^{wi})$. The nonlinear activation function and the scalar element of $H(\hat{\theta}_{k-1}^w, x_{k-1})$.
$\tilde{\Omega}_{k-1}$	$\in R^{m \times p^w}$. Error function approximation matrix for the hidden layer analysis.
$\tilde{\mu}_{k-1,i}$	$\in R; 1 \leq i \leq n_h$. Mean value of the i th hidden layer neuron.
λ	Constant of the sigma activation function.
H_1^v, H_1^w	Nonlinear operators of the conic sector for the output and hidden layers, respectively.
H_2	Linear operators of the conic sector.
Φ_k^v, Φ_k^w	$\in R^m$. Regression vectors of the output and hidden layers, respectively.
$\bar{\Phi}_k^v, \bar{\Phi}_k^w$	$\in R^m$. Normalized regression vectors of the output and hidden layers, respectively.
α_k^v, α_k^w	Learning rates of the output and hidden layers, respectively.
ρ_k^v, ρ_k^w	Normalization factors of the output and hidden layers, respectively.

II. INTRODUCTION

RECENTLY, there have been significant progresses in the area of robust discrete-time neural controller design for nonlinear systems with specific nonlinear functions [1]–[6], [14], [23], [24]. For example, a first-order approximation is applied in the convergence proof of [1] and [2] to deal with the nonlinear activation function. Variable structure and dead zone schemes have also been introduced to design robust adaptive algorithms of neural control systems so as to achieve improved tracking performance [3]–[5]. An important achievement is that the well-known persistent exciting (PE) condition has been removed in the presence of disturbance [6]. More recently, the idea of simultaneous perturbation stochastic approximation (SPSA) has been introduced as a model-free control method for dynamical systems [7], [8], [14], [17].

In this paper, we will propose an SPSA-based neural control structure and derive a general stability proof when it is applied to a nonlinear input/output (I/O) dynamical plant. The plant under consideration is nonlinear and the neural network in the system is used to estimate the nonlinear function in closed loop. The conic sector theory [9]–[11] is introduced to design the robust neural control system. We aim to achieve guaranteed weight convergence and boundedness for both the I/O signals and the weights of the neural network via a deterministic approach.

To the best of our knowledge, we cannot mix the deterministic approach of convergence and stability analyses adopted in this paper, where we treat disturbances as bounded signals, and the stochastic approach presented in the original SPSA papers

[7], [8], where statistical properties of disturbances are considered and well established under certain regularity conditions [7], [8]. However, we will provide an in-depth analysis to reveal the inherent relationships between the two different approaches to a certain extent, especially, the gain parameter of perturbations and its effect on weight convergence in the framework of deterministic systems.

One of the main advantages of the conic sector approach is that it provides a model-free analysis. The neural controller is superior as compared to its conventional adaptive control counterpart in that the latter requires linear-in-parameters representation for system estimation. Our neural network is trained by the SPSA algorithm in closed loop to provide an improved training performance over the standard methods, such as the backpropagation (BP) algorithm, in terms of guaranteed stability of the weights. This in turn will yield a good tracking performance for the dynamical control system. The main motivation for using the SPSA instead of the popular BP algorithm is its excellent convergence property. The SPSA algorithm which was proposed by Spall [7] provides a good parameter estimate through simultaneous perturbation of the weights.

In addition to the general stability proof, one of the most interesting contributions of this paper is the revelation of the relationship between the conventional adaptive control system and the generalization theory, which is mainly developed for neural-network-based pattern recognition systems [21] and, to the best of our knowledge, is not widely acknowledged by the control community. That is, a relatively large learning rate with a reasonable number of neurons will contribute to a faster convergence speed of the SPSA training algorithm and in turn will yield a good adaptive learning capability. This is closely linked to the concept of generalization of neural network theory [21]. The early neural control system approaches, including the classical BP-based training algorithm, tend to emphasize the approximation property of a large neural network. It is only recently that good generalization property has attracted more interests, for it suggests that a network with a reasonable number of neurons may be the best way to approximate a nonlinear system rather than an overfitted large network. As will be shown later, in the theoretical analysis about the learning rate, and in the simulation results, an optimal number of neurons (not necessary the biggest network) can be derived based on the maximum learning rates calculated by the conic sector condition. Thus, it allows one to achieve good generalization performance in terms of reduced control signal error and fast tracking speed. This idea can also be further developed into an adaptive pruning-based SPSA algorithm to automatically search for an optimal neural network structure [20].

Because stability is the primary issue in a closed-loop system, instead of a stochastic convergence analysis in [7], we will follow the traditional approach of adaptive control system to provide a robust I/O stability design and weight convergence analysis for our proposed SPSA-based neural control algorithm in the framework of a deterministic system. We do not require the weights to converge to the ideal values exactly and we use the dead zone concept [5], [9]–[11] to provide an insight to the important role of the perturbation gain parameter and its relationship with the stochastic approach in [7] and [8]. 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 pretrained neural network [16], we provide an online scheme for the robustness analysis of the neural control system. A special normalized learning law is designed for 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 to achieve guaranteed I/O stability using the conic sector condition. The performance improvement of the proposed algorithm can be measured in terms of preventing weight shifting, fast convergence, and robustness against system disturbance.

III. CONTROL SYSTEMS AND SPSA TRAINING ALGORITHM

A class of dynamical nonlinear plants, which has wide applications in robotics and tracking control systems [5], [6], [15], [18], can be represented as an I/O form as follows:

$$y_k = f_{k-1} + u_{k-1} + \varepsilon_k \quad (1)$$

where $y_k \in R^m$ is the output, $f_{k-1} = f_{k-1}(y_{k-1}, \dots, y_{k-l}, u_{k-2}, \dots, u_{k-n}, \theta^*) \in R^m$ ($l \geq 1, n \geq 2$) is a dynamic nonlinear function ($\theta^* \in R^p$ is an ideal weight vector), $\varepsilon_k \in R^m$ denotes the overall bounded noise vector of the control system, and $u_{k-1} \in R^m$ is the control signal vector with a unit time delay (which can also be extended to a d -step delay system by using a linear predictor [18]).

The tracking error of the control system is defined as

$$s_k = y_k - y_k^* \quad (2)$$

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

$$u_{k-1} = -\hat{f}_{k-1} + y_k^* + k_v s_{k-1} \quad (3)$$

where k_v is the gain parameter of the fixed proportional controller and $\hat{f}_{k-1} = \hat{f}_{k-1}(\hat{\theta}_{k-1})$ is the estimate of the nonlinear function f_{k-1} by the neural network to be defined later, and $\hat{\theta}_{k-1} \in R^p$ is the estimated weight vector of the neural network. Then, the estimation error vector of the neural network can be presented as

$$e_k = f_{k-1} - \hat{f}_{k-1} + \varepsilon_k. \quad (4)$$

This error is to be used to train the neural network as shown in Fig. 1. However, this estimation error e_k may not be directly measurable, so we will use the tracking error to generate it by using the closed-loop relationship via (1)(4) (see Fig. 1 and summary of the SPSA algorithm in Section V), so that

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

where z^{-1} is backward shift operator, i.e., $z^{-1}s_k = s_{k-1}$.

The loss function for SPSA is defined as

$$L(\hat{\theta}_{k-1}) = \frac{1}{2} \|f_{k-1} - \hat{f}_{k-1}\|^2. \quad (6)$$

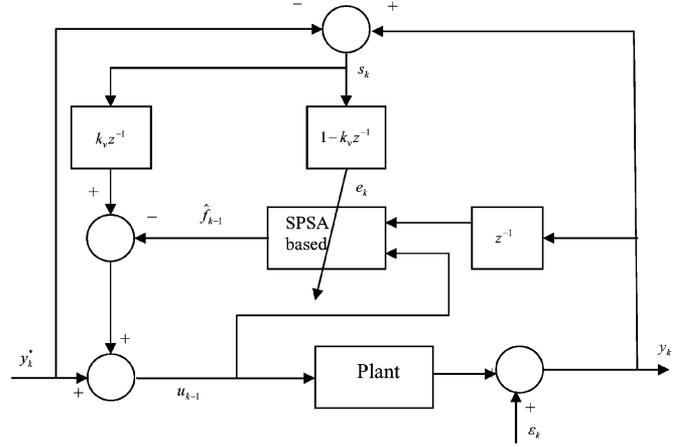


Fig. 1. Structure of the control scheme.

Because f_{k-1} is a function of θ^* and \hat{f}_{k-1} is a function of $\hat{\theta}_{k-1}$, the SPSA algorithm seeks to find an optimal weight estimate to approximate the ideal weight vector θ^* of the gradient equation (the ideal weight may present a local minimum point rather than a global one; see discussion in Sections IV–VI), i.e., the weight vector that minimizes differentiable $L(\hat{\theta}_{k-1})$, through

$$\min_{\hat{\theta}_{k-1} \rightarrow \theta^* |_{k \rightarrow \infty}} g_k(\hat{\theta}_{k-1}) = \min_{\hat{\theta}_{k-1} \rightarrow \theta^* |_{k \rightarrow \infty}} \frac{\partial L(\hat{\theta}_{k-1})}{\partial \hat{\theta}_{k-1}}.$$

That is, our proposed SPSA algorithm for updating $\hat{\theta}_{k-1} \in R^p$ as an estimate of the ideal weight vector θ^* is of the form

$$\hat{\theta}_k = \hat{\theta}_{k-1} - \alpha_k \hat{g}_k(\hat{\theta}_{k-1}) \quad (7)$$

where α_k is an adaptive learning rate and $\hat{g}_k(\hat{\theta}_{k-1})$ is an approximation of the gradient of the loss function. Note that we do not require the weights to converge to the ideal values exactly because the adaptive learning rate is based on the traditional dead zone approach [9]–[11]. This approximated gradient function (normalized) is of the form

$$\hat{g}_k(\hat{\theta}_{k-1}) = \frac{L(\hat{\theta}_{k-1} + c_k \Delta_k) - L(\hat{\theta}_{k-1} - c_k \Delta_k) + \delta_k}{2c_k \rho_k} r_k \quad (8)$$

where δ_k can be interpreted as an equivalent approximation error of the loss function, which is also called the measurement noise in [7], and $\Delta_k \in R^p$, $r_k \in R^p$, and $c_k > 0$ are the controlling parameters of the algorithm, and ρ_k is the normalization factor to be defined later for the output and hidden layers, respectively. These parameters are defined as follows.

- 1) $\Delta_k = [\Delta_{k,1} \ \dots \ \Delta_{k,p}]^T \in R^p$ is a bounded random directional vector that is used to perturb the weight vector simultaneously and can be generated randomly.
- 2) The sequence of $r_k \in R^p$ is defined as

$$r_k = \left[\frac{1}{\Delta_{k,1}} \quad \dots \quad \frac{1}{\Delta_{k,p}} \right]^T. \quad (9)$$

- 3) $c_k > 0$, a constant $k \geq 1$ is a sequence of positive numbers in [7], [8], and [17], which can be chosen as a small constant number for a slow time-varying system targeted in this paper as pointed out in [17].

We will show later that the choice of $c_k > 0$ does not affect the stability of the system in the sense of bounded system I/O signals, but it will affect the level of system noise in terms of the equivalent disturbance, and in turn, the weight convergence property (see Section IV and Remark 2).

In a multilayered neural network, it may not be possible to update all the estimated weight vectors with a single gradient approximation function (8) to meet the stability requirement. We will partition the weight estimate $\hat{\theta}_k \in R^p$ into two components: $\hat{\theta}_k^v \in R^{p^v}$ as the estimated weight vector of the output layer and $\hat{\theta}_k^w \in R^{p^w}$ as the estimated weight vector of the hidden layer of the neural network with $p^v = m \times n_h$ and $p^w = n_h \times n_I$, where n_I and n_h denote the numbers of neurons in the input and hidden layers of the network, respectively. Then, the estimated weight vectors $\hat{\theta}_k^v$ and $\hat{\theta}_k^w$ are to be updated separately in the SPSA algorithm using different gradient approximation functions as in the standard BP training algorithm. This point will be examined in the robustness analysis of Section IV and V. In what follows, the notations with superscripts v and w will be used to denote the variables/parameters associated with the output and the hidden layers, respectively.

The output of a three-layer neural network is the estimate of the nonlinear function f_{k-1} , and is further presented as the following:

$$\hat{f}_{k-1} = H \left(\hat{\theta}_{k-1}^w, x_{k-1} \right) \hat{\theta}_{k-1}^v \quad (10)$$

where $x_{k-1} \in R^{n_I}$ ($n_I = (l + n - 1) \times m$) is the input vector of the neural network defined by

$$\begin{aligned} x_{k-1} &= [x_{k-1,1}, x_{k-1,2}, \dots, x_{k-1,n_I}]^T \\ &= [y_{k-1}^T, \dots, y_{k-1}^T, u_{k-2}^T, \dots, u_{k-n}^T]^T \end{aligned} \quad (11)$$

and $H(\hat{\theta}_{k-1}^w, x_{k-1}) \in R^{m \times p^w}$ is the nonlinear activation block-diagonal matrix given by (12), shown at the bottom of the page, where $h_{k-1,i}$ with $1 \leq i \leq n_h$ is one of the most popular nonlinear activation functions, especially for neural control systems [1], [2], [5], i.e.,

$$\begin{aligned} h_{k-1,i} &= h_{k-1,i} \left(x_{k-1}^T \hat{\theta}_{k-1}^{w_I} \right) \\ &= \frac{1}{1 + \exp \left(-4\lambda x_{k-1}^T \hat{\theta}_{k-1}^{w_I} \right)} \end{aligned} \quad (13)$$

with $\hat{\theta}_{k-1}^{w_I} = [\hat{\theta}_{k-1,i}^w, \hat{\theta}_{k-1,i+1}^w, \dots, \hat{\theta}_{k-1,i+n_I-1}^w]^T \in R^{n_I}$ being the estimated weight vector linked to the i th hidden layer neuron, and $4\lambda > 0$ the gain parameter of the threshold function that is defined specifically for ease of use later when

deriving the sector condition of the hidden layer. Note that $\hat{\theta}_{k-1}^w$ and $\hat{\theta}_{k-1,i}^{w_I}$ are related by $\hat{\theta}_{k-1}^w = [(\hat{\theta}_{k-1,1}^{w_I})^T \dots (\hat{\theta}_{k-1,n_h}^{w_I})^T]^T$.

IV. CONIC SECTOR CONDITION FOR ROBUSTNESS ANALYSIS OF THE OUTPUT LAYER

Because we cannot mix the deterministic and stochastic approaches for convergence and stability analyses [7], [8], the general idea of this paper is to address the convergence property and stability of the weight estimate of the proposed SPSA training algorithm by strictly following the deterministic approach. We will prove that the estimation error (and the tracking error) of the output layer of the neural network is bounded by using the conic sector theory under some mild assumptions. This agrees with the boundedness of the weight estimate error vector of the output layer in terms of the convergent L_2 weight error norm. The boundedness condition for the weight estimate error vector of the hidden layer will be derived in Section V. In our proposed SPSA training method, the estimated weights are perturbed simultaneously to solve the weight drifting problem of a normal BP training while minimizing a loss function, which is similar to the original SPSA algorithm in [7] and [8]. Moreover, we will justify why a small constant gain parameter c_k for the perturbation signal is preferred in our deterministic convergence analysis of the proposed SPSA algorithm. This is similar, to a certain extent, to the stochastic approach in [7], [8], and [17]. The adaptive scheme also provides the guidelines for the selection of the proposed SPSA learning rate and the number of neurons to obtain an improved performance.

The purpose of the SPSA training algorithm in (7) is to make the estimated weight vector $\hat{\theta}_{k-1}$ approximate the ideal one, and in turn produce an optimal tracking error for the control system. To achieve this, one important condition is that the slow time-varying gradient approximation function $\hat{g}_k(\hat{\theta}_{k-1})$ in (8), the estimated parameter vector $\hat{\theta}_{k-1}$, and the input x_{k-1} should all be bounded as required for adaptive control systems [11]. To guarantee the boundedness condition, the robust neural controller shown in Fig. 1 uses a normalized SPSA training algorithm, which is isolated from the rest of the system, and a deterministic analysis is applied. Interestingly, a different deterministic treatment of the SPSA algorithm can also be found in [12]. In our paper, the robustness of the system is analyzed by using the conic sector theory. A two-stage normalized training strategy is then proposed for the SPSA training with guaranteed I/O stability using the conic sector condition.

In this paper, except for some specified cases, we use $\|\cdot\|$ to denote both the ‘‘row sum’’ norm of a matrix and the Euclidean norm of a vector [22]. Our main concern is with the discrete time estimation error vector e_k , which is an infinite sequence of real vectors. Following [13], we consider the extended space

$$H \left(\hat{\theta}_{k-1}^w, x_{k-1} \right) = \begin{bmatrix} h_{k-1,1}, & h_{k-1,2}, & \dots & h_{k-1,n_h}, & 0, \dots & & 0, \dots \\ & 0, \dots & & & & & \vdots \\ & & & & \ddots & & \\ & & & & & & 0, \dots \\ & 0, \dots & & & & h_{k-1,1}, & h_{k-1,2}, & \dots & h_{k-1,n_h} \end{bmatrix} \quad (12)$$

L_{2e} , consisting of those elements whose truncations lie in L_2 , e.g., e belongs to the extended space L_{2e} if

$$\|e\|_{2,N} = \left\{ \sum_{k=1}^N e_k^T e_k \right\}^{\frac{1}{2}} < \infty \quad (14)$$

$\forall N \in Z_+$ (the set of positive integer).

The following theory is an extension of the conic sector stability of Safanov [13] for discrete time control systems, like the one given in Fig. 1.

Theorem 1: Consider the following error feedback system:

$$\begin{aligned} e_k &= \tilde{e}_k - P_k \\ \Phi_k &= H_1 e_k \\ P_k &= H_2 \Phi_k \end{aligned}$$

with operators $H_1, H_2 : L_{2e} \rightarrow L_{2e}$, and discrete time signals $e_k, P_k, \Phi_k \in L_{2e}$ and $\tilde{e}_k \in L_2$. If

1) $H_1 : e_k \rightarrow \Phi_k$ satisfies

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

and

2) $H_2 : \Phi_k \rightarrow P_k$ satisfies

$$\begin{aligned} \sum_{k=1}^N [\sigma P_k^T P_k / 2 - P_k^T \Phi_k] \\ \leq -\eta \| (P_k, \Phi_k) \|_{2,N}^2 \\ \triangleq -\eta \sum_{k=1}^N (P_k^T P_k + 2P_k^T \Phi_k + \Phi_k^T \Phi_k) \end{aligned}$$

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

Proof: See [11, Corollary 2.1]. $\triangle\triangle\triangle$

Note that operator H_1 in Theorem 1 represents the SPSA training algorithm, where the input error signal is the estimation error e_k and the output is Φ_k , which will be defined later, and it is related to the weight estimate error vectors. The estimation error e_k and the tracking error s_k are related by a stable first-order controller (i.e., $|k_v| < 1$) in (5). Thus, a bounded estimation error e_k will yield a bounded tracking error s_k (see [5, Corollary 1] for more details). The operator H_2 usually represents the mismatched linear model uncertainty in a typical adaptive linear control system [9], but here it is just a simple operator $H_2 = 1$.

The first step to use the conic sector stability of Theorem 1 is to restructure the control system in Fig. 1 into an equivalent error feedback system as shown in Fig. 2. Then, the weight estimate error vector could be derived and referred to as the output signal Φ_k . For this purpose, the desired output of the neural network is defined as the plant nonlinear function given in (1) and is assumed to be given by

$$f_{k-1} = H(\theta^{w*}, x_{k-1}) \theta^{v*} \quad (15)$$

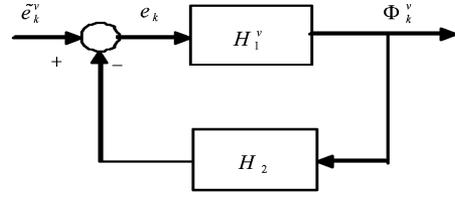


Fig. 2. Equivalent error feedback system using the conic sector condition for output layer.

where $\theta^{v*} \in R^{p^v}$ and $\theta^{w*} \in R^{p^w}$ are, respectively, the ideal weight vectors in the output layer and the hidden layer of the neural network. Then, the weight estimate error vectors of the output and hidden layers are defined, respectively, as

$$\tilde{\theta}_k^v = \theta^{v*} - \hat{\theta}_k^v \in R^{p^v} \quad (16)$$

and

$$\tilde{\theta}_k^w = \theta^{w*} - \hat{\theta}_k^w \in R^{p^w}. \quad (17)$$

We further assume that the system under consideration satisfies the following assumptions.

Assumptions:

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

These can be justified as follow. The external system disturbance ε_k is an uncontrollable variable, which must be bounded in a normal control system. The bounds for the ideal weight θ^* imply that the nonlinear function f_k itself is bounded if the input variables x_k are bounded.

To establish the relationship between the estimation error signal e_k and the estimated weight vectors of the neural network, which is referred to as the operator H_1 in Theorem 1, i.e., the SPSA algorithm, we need to express the error signal in suitable forms.

Define the variables

$$\Phi_k^v = -H(\hat{\theta}_{k-1}^w, x_{k-1}) \tilde{\theta}_{k-1}^v \quad (18)$$

$$\tilde{e}_k^v = H_2(\tilde{H}(\hat{\theta}_{k-1}^w, \theta^{w*}, x_{k-1}) + \varepsilon_k) \quad (19)$$

$$\tilde{H}(\hat{\theta}_{k-1}^w, \theta^{w*}, x_{k-1}) = (H(\theta^{w*}, x_{k-1}) - H(\hat{\theta}_{k-1}^w, x_{k-1})) \quad (20)$$

where $\tilde{\theta}_{k-1}^v$ is as defined in (16).

According to (4) and $H_2 = 1$, the estimate error signal can be expressed as

$$\begin{aligned} e_k &= H_2 e_k = H_2 (f_{k-1} - \hat{f}_{k-1} + \varepsilon_k) \\ &= H_2 \left(H(\theta^{w*}, x_{k-1}) \theta^{v*} - H(\hat{\theta}_{k-1}^w, x_{k-1}) \hat{\theta}_{k-1}^v + \varepsilon_k \right) \\ &= H_2 (H(\theta^{w*}, x_{k-1}) \theta^{v*} - H(\hat{\theta}_{k-1}^w, x_{k-1}) \theta^{v*} \\ &\quad + H(\hat{\theta}_{k-1}^w, x_{k-1}) \theta^{v*} - H(\hat{\theta}_{k-1}^w, x_{k-1}) \hat{\theta}_{k-1}^v + \varepsilon_k) \\ &= H_2 (H(\hat{\theta}_{k-1}^w, x_{k-1}) \tilde{\theta}_{k-1}^v + \tilde{H}(\hat{\theta}_{k-1}^w, \theta^{w*}, x_{k-1}) \theta^{v*} + \varepsilon_k) \\ &= H_2 H(\hat{\theta}_{k-1}^w, x_{k-1}) \tilde{\theta}_{k-1}^v + \tilde{e}_k^v \\ &= -H_2 \Phi_k^v + \tilde{e}_k^v. \end{aligned} \quad (21)$$

From the previous derivations, an equivalent error feedback system of Fig. 1 can, therefore, be represented as in Fig. 2, where the operator H_1^v represents the SPSA training algorithm of the output layer.

Remark 1: There is an important implication in (21). The estimation error signal e_k is directly linked to the output signal Φ_k^v , and in turn via (18) the weight estimate error vector $\hat{\theta}_k^v$ of the output layer of the neural network. This implies that the training procedure of the output layer of the neural network should be treated separately from the hidden layer of the network in order to obtain a bounded disturbance term \tilde{e}_k^v as defined in (19), i.e., $\tilde{e}_k^v \in L_2$ as required by Theorem 1 (where it is defined as \tilde{e}_k). Therefore, using (21), we are able to form an equivalent error feedback system of Fig. 2 for Theorem 1. Note that H_2 usually represents the mismatched linear model uncertainty in a typical adaptive linear control system [9]. Because the neural network has powerful approximation ability to match the nonlinear function, there is no need to worry about any linear model mismatch with the simple operator $H_2 = 1$. Furthermore, the condition 2) of Theorem 1 can be treated as a positive real function, i.e., the plot of H_2 should be in the positive half of a complex plane as discussed in [10]. \square

Now, our algorithm for updating the estimated weight vector $\hat{\theta}_k^v$ of the output layer is

$$\begin{aligned}\hat{\theta}_k^v &= \hat{\theta}_{k-1}^v - \alpha_k^v \hat{g}_k(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^v, \Delta_k^v) \\ &= \hat{\theta}_{k-1}^v + \alpha_k^v \hat{g}_k(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^v, \Delta_k^v)\end{aligned}\quad (22)$$

where the adaptive learning rate of the output layer is defined as (23), shown at the bottom of the page, with $0 < \alpha^v \leq 1$ being a constant and $\tilde{e}_{\max}^v = \max\{\tilde{e}_k^v\} \forall k$ being the maximum value of the bounded equivalent disturbance \tilde{e}_k^v in (19); $\Delta_k^v \in R^{p^v}$ and $r_k^v \in R^{p^v}$ are, respectively, the first p^v components of perturbation vectors Δ_k and r_k defined in (8) and (9); $\hat{g}_k(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^v, \Delta_k^v)$ is the normalized gradient approximation that uses the simultaneous perturbation vectors Δ_k^v and r_k^v to stimulate the weight of the output layer, i.e.,

$$\hat{g}_k(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^v, \Delta_k^v) = -\frac{e_k^T H(\hat{\theta}_{k-1}^w, x_{k-1}) \Delta_k^v}{\rho_k^v} r_k^v \quad (24)$$

and the bounded normalization factor ρ_k^v , which is traditionally used in adaptive control system to bound the signals in learning algorithms [11], [23], [24], is updated via

$$\rho_k^v = \mu \rho_{k-1}^v + \max \left\{ \frac{\alpha^v}{p^v} \left\| H(\hat{\theta}_{k-1}^w, x_{k-1}) \right\|^2 \|\Delta_k^v\|^2 \|r_k^v\|^2, \bar{\rho} \right\} \quad (25)$$

with $\bar{\rho} > 0$ and $\mu \in (0, 1)$ being two arbitrary small positive constants. The previous implementation of ρ_k^v ensures that it satisfies

$$\rho_k^v > \frac{\alpha^v}{p^v} \left\| H(\hat{\theta}_{k-1}^w, x_{k-1}) \right\|^2 \|\Delta_k^v\|^2 \|r_k^v\|^2.$$

The nonzero learning rate value α^v in (23) has an upper bound as defined in the following:

$$\alpha^v < \frac{\rho_{\max}^v}{p^v |h_{\min}|^2 |\Delta_{\min}^v|^2 |r_{\min}^v|^2} \quad (26)$$

where $h_{\min} = \min_{k,i} \{|h_{k-1,i}|\}$, $r_{\min}^v = \min_k \{\|r_k^v\|\}$ and $\Delta_{\min}^v = \min_k \{\|\Delta_k^v\|\}$ are the nonzero minimum values of the activation function defined in (13) and perturbation vectors of the output layer, respectively, and $\rho_{\max}^v = \max_k \{\rho_k^v\}$.

Lemma 1: It can be shown that (22) is a realization of the SPSA algorithm of the form (8) and the SPSA measurement noise can be defined as $\delta_k^v = -2c_k(H(\hat{\theta}_{k-1}^w, x_{k-1})\Delta_k^v)^T \varepsilon_k$.

Proof: To show this, we will use the definition of the loss function (6) and rewrite it as

$$L(\hat{\theta}_{k-1}) = L(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^v) = \frac{1}{2} \|f_{k-1} - \hat{f}_{k-1}\|^2. \quad (27)$$

Further, we use $\hat{f}_{k-1} = H(\hat{\theta}_{k-1}^w, x_{k-1})\hat{\theta}_{k-1}^v$ in (10) to define

$$\hat{f}_{k-1}^{v+} = H(\hat{\theta}_{k-1}^w, x_{k-1})(\hat{\theta}_{k-1}^v + c_k \Delta_k^v) \quad (28)$$

$$\hat{f}_{k-1}^{v-} = H(\hat{\theta}_{k-1}^w, x_{k-1})(\hat{\theta}_{k-1}^v - c_k \Delta_k^v) \quad (29)$$

so that

$$\hat{f}_{k-1}^{v+} - \hat{f}_{k-1}^{v-} = H(\hat{\theta}_{k-1}^w, x_{k-1})2c_k \Delta_k^v. \quad (30)$$

Then, with (6), (4), and (30), we can derive (24) from the general form of the gradient function (8) for the specific output layer case as shown in (31), at the bottom of the page, where, except the normalization factor ρ_k^v , the fourth equality is in an identical

$$\alpha_k^v = \begin{cases} \alpha^v, & \text{if } \|e_k\| \geq \tilde{e}_{\max}^v / \sqrt{1 - \left\| H(\hat{\theta}_{k-1}^w, x_{k-1}) \right\|^2 \|\Delta_k^v\|^2 \|r_k^v\|^2 \alpha^v (p^v \rho_k^v)^{-1}} \\ 0, & \text{if } \|e_k\| < \tilde{e}_{\max}^v / \sqrt{1 - \left\| H(\hat{\theta}_{k-1}^w, x_{k-1}) \right\|^2 \|\Delta_k^v\|^2 \|r_k^v\|^2 \alpha^v (p^v \rho_k^v)^{-1}} \end{cases} \quad (23)$$

$$\begin{aligned} & \frac{L(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^v + c_k \Delta_k^v) - L(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^v - c_k \Delta_k^v) + \delta_k^v}{2c_k \rho_k^v} r_k^v \\ &= \frac{\left\| f_{k-1} - \hat{f}_{k-1}^{v+} \right\|^2 - \left\| f_{k-1} - \hat{f}_{k-1}^{v-} \right\|^2 + 2\delta_k^v}{4c_k \rho_k^v} r_k^v = \frac{(f_{k-1} - \hat{f}_{k-1}^{v+} + f_{k-1} - \hat{f}_{k-1}^{v-})^T (\hat{f}_{k-1}^{v-} - \hat{f}_{k-1}^{v+}) + 2\delta_k^v}{4c_k \rho_k^v} r_k^v \\ &= \frac{(f_{k-1} - \hat{f}_{k-1} + \varepsilon_k)^T (\hat{f}_{k-1}^{v-} - \hat{f}_{k-1}^{v+})}{2c_k \rho_k^v} r_k^v = \frac{e_k^T (\hat{f}_{k-1}^{v-} - \hat{f}_{k-1}^{v+})}{2c_k \rho_k^v} r_k^v = -\frac{e_k^T H(\hat{\theta}_{k-1}^w, x_{k-1}) 2\Delta_k^v}{2\rho_k^v} r_k^v = \hat{g}_k(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^v, \Delta_k^v) \end{aligned} \quad (31)$$

format as the original SPSA algorithm in [7, eq. (2.2)], and the third equality is derived by defining the measurement noise δ_k^v , which is related to the overall system disturbance ε_k according to

$$\begin{aligned} \delta_k^v &= \left(\hat{f}_{k-1}^v - \hat{f}_{k-1}^+ \right)^T \\ \varepsilon_k &= -2c_k \left(H(\hat{\theta}_{k-1}^w, x_{k-1}) \Delta_k^v \right)^T \varepsilon_k. \end{aligned} \quad (32)$$

(Otherwise, no disturbance is introduced for the output layer of the SPSA learning algorithm.) $\triangle\triangle\triangle$

We are now ready to establish the convergence and stability of the robust neural controller via the conic sector condition. This requires the feedback system in Fig. 2 to meet certain dissipative condition as in Theorem 1. This is established in the following theorem.

Theorem 2: The operator $H_1^v : e_k \rightarrow \Phi_k^v$, which represents the SPSA learning algorithm for the output layer (see Fig. 2), satisfies the conditions 1) and 2) of Theorem 1, i.e., $e_k, \Phi_k^v \in L_2$ and the weight $\hat{\theta}_{k-1}^v$ is convergent in the sense that the L_2 -norm $\|\hat{\theta}_{k-1}^v\|$ is a nonincreasing sequence as $k \rightarrow \infty$.

Proof: To establish the weight convergence and conic sector condition of the estimation error e_k , we use a similar SPSA approach in terms of simultaneous perturbation (except for the noise properties) as in [7] and note that (22) gives

$$\tilde{\theta}_k^v = \tilde{\theta}_{k-1}^v - \alpha_k^v \frac{e_k^T H(\hat{\theta}_{k-1}^w, x_{k-1}) 2\Delta_k^v r_k^v}{2\rho_k^v} \quad (33)$$

where $\tilde{\theta}_k^v$ is defined in (16). Then, rearranging (33) by using the trace property of $\tilde{\theta}_{k-1}^{vT} (\tilde{\theta}_{k-1}^v \tilde{\theta}_{k-1}^{vT})^{-1} \Delta_k^v r_k^{vT} \tilde{\theta}_{k-1}^v = \text{tr}(\tilde{\theta}_{k-1}^v \tilde{\theta}_{k-1}^{vT} (\tilde{\theta}_{k-1}^v \tilde{\theta}_{k-1}^{vT})^{-1} \Delta_k^v r_k^{vT}) = p^v$, and (18), we get

$$\begin{aligned} & \left\| \tilde{\theta}_k^v \right\|^2 - \left\| \tilde{\theta}_{k-1}^v \right\|^2 \\ &= -2\alpha_k^v \left\{ e_k^T H_k(\hat{\theta}_{k-1}^w, x_{k-1}) \Delta_k^v (r_k^v)^T \tilde{\theta}_{k-1}^v \right\} (\rho_k^v)^{-1} \\ & \quad + \left\| \left(\alpha_k^v e_k^T H(\hat{\theta}_{k-1}^w, x_{k-1}) \Delta_k^v r_k^v (\rho_k^v)^{-1} \right) \right\|^2 \\ &= -2\alpha_k^v \left\{ e_k^T H_k(\hat{\theta}_{k-1}^w, x_{k-1}) \tilde{\theta}_{k-1}^v \right\} \\ & \quad \times \left\{ \tilde{\theta}_{k-1}^{vT} \left(\tilde{\theta}_{k-1}^v \tilde{\theta}_{k-1}^{vT} \right)^{-1} \Delta_k^v r_k^{vT} \tilde{\theta}_{k-1}^v \right\} (\rho_k^v)^{-1} \\ & \quad + \left\| \left(\alpha_k^v e_k^T H(\hat{\theta}_{k-1}^w, x_{k-1}) \Delta_k^v r_k^v (\rho_k^v)^{-1} \right) \right\|^2 \\ &= 2\alpha_k^v p^v \left\{ e_k^T \Phi_k^v \right\} (\rho_k^v)^{-1} \\ & \quad + \left\| \left(\alpha_k^v e_k^T H(\hat{\theta}_{k-1}^w, x_{k-1}) \Delta_k^v r_k^v (\rho_k^v)^{-1} \right) \right\|^2 \\ &\leq 2\alpha_k^v p^v \left\{ e_k^T \Phi_k^v \right\} (\rho_k^v)^{-1} \\ & \quad + \|e_k\|^2 \left\| \left(\alpha_k^v H(\hat{\theta}_{k-1}^w, x_{k-1}) \Delta_k^v r_k^v (\rho_k^v)^{-1} \right) \right\|^2 \\ &\leq 2\alpha_k^v p^v \left\{ e_k^T \Phi_k^v \right\} (\rho_k^v)^{-1} + \|e_k\|^2 \left(\alpha_k^v (\rho_k^v)^{-1} \right)^2 \\ & \quad \times \left\| H(\hat{\theta}_{k-1}^w, x_{k-1}) \right\|^2 \left\| \Delta_k^v \right\|^2 \|r_k^v\|^2. \end{aligned} \quad (34)$$

Further, considering the definition of the equivalent error system $\Phi_k^v = \tilde{e}_k^v - e_k$ in (21) with $H_2^v = 1$, we have the first inequality of following Lyapunov function extended from the previous equation for weight convergence in the sense of a

nonincreasing sequence of weight error vector L_2 -norm

$$\begin{aligned} & \left\| \tilde{\theta}_k^v \right\|^2 - \left\| \tilde{\theta}_{k-1}^v \right\|^2 \\ & \leq 2\alpha_k^v p^v \left\{ e_k^T (\tilde{e}_k^v - e_k) \right\} (\rho_k^v)^{-1} + \|e_k\|^2 \left(\alpha_k^v (\rho_k^v)^{-1} \right)^2 \\ & \quad \times \left\| H(\hat{\theta}_{k-1}^w, x_{k-1}) \right\|^2 \left\| \Delta_k^v \right\|^2 \|r_k^v\|^2 \\ & \leq \alpha_k^v p^v \left\{ \|\tilde{e}_k^v\|^2 + \|e_k\|^2 - 2\|e_k\|^2 \right\} (\rho_k^v)^{-1} \\ & \quad + \|e_k\|^2 \left(\alpha_k^v (\rho_k^v)^{-1} \right)^2 \\ & \quad \times \left\| H(\hat{\theta}_{k-1}^w, x_{k-1}) \right\|^2 \left\| \Delta_k^v \right\|^2 \|r_k^v\|^2 \\ & \leq \alpha_k^v p^v (\rho_k^v)^{-1} \left\{ \|\tilde{e}_k^v\|^2 - \left[1 - \alpha^v (p^v \rho_k^v)^{-1} \right. \right. \\ & \quad \times \left\| H(\hat{\theta}_{k-1}^w, x_{k-1}) \right\|^2 \\ & \quad \left. \left. \times \left\| \Delta_k^v \right\|^2 \|r_k^v\|^2 \right] \right\} \\ & \leq 0. \end{aligned} \quad (35)$$

Note also that the second inequality is from the fact that $2\tilde{e}_k^v e_k \leq \|\tilde{e}_k^v\|^2 + \|e_k\|^2$. The last two inequalities are based on the definition of α_k^v in (23). In particular, the normalized factor ρ_k^v is designed as in (25) such that we can guarantee

$$1 - (\rho_k^v)^{-1} \left[\frac{\alpha^v}{p^v} \left\| H(\hat{\theta}_{k-1}^w, x_{k-1}) \right\|^2 \left\| \Delta_k^v \right\|^2 \|r_k^v\|^2 \right] > 0. \quad (36)$$

Furthermore, due to the definition of the adaptive learning rate α_k^v in (23), we only need to consider the case of $\alpha_k^v = \alpha^v$ in the rest of proof of Theorem 2; otherwise, the weight is kept constant from the previous iteration. Defining the normalized estimation error $\bar{e}_k^v = e_k (\sqrt{\rho_k^v})^{-1}$ and the normalized error system output $\bar{\Phi}_k^v = \Phi_k^v (\sqrt{\rho_k^v})^{-1} = -H(\hat{\theta}_{k-1}^w, x_{k-1}) \tilde{\theta}_k^v (\sqrt{\rho_k^v})^{-1}$ in Fig. 2, we can rewrite the inequality (34) into

$$\begin{aligned} & \frac{\left\| \tilde{\theta}_k^v \right\|^2 - \left\| \tilde{\theta}_{k-1}^v \right\|^2}{2\alpha^v p^v} \\ & \leq \left\{ e_k^T \Phi_k^v (\rho_k^v)^{-1} \right\} \\ & \quad + \frac{\|e_k\|^2 (\rho_k^v)^{-2} \left\{ \alpha^v \left\| H(\hat{\theta}_{k-1}^w, x_{k-1}) \right\|^2 \left\| \Delta_k^v \right\|^2 \|r_k^v\|^2 \right\}}{2p^v} \\ & \leq \left\{ (\bar{e}_k^v)^T \bar{\Phi}_k^v \right\} + \frac{\|\bar{e}_k^v\|^2 \sigma^v}{2} \end{aligned} \quad (37)$$

where the normalized factor ρ_k^v in (25) is chosen to guarantee that there exists a constant σ^v such that

$$\begin{aligned} 0 & \leq \frac{\alpha^v}{p^v \rho_k^v} \left\| H(\hat{\theta}_{k-1}^w, x_{k-1}) \right\|^2 \left\| \Delta_k^v \right\|^2 \|r_k^v\|^2 \leq \sigma^v < 1 \\ & \Rightarrow \rho_k^v > \frac{\alpha^v}{p^v} \left\| H(\hat{\theta}_{k-1}^w, x_{k-1}) \right\|^2 \left\| \Delta_k^v \right\|^2 \|r_k^v\|^2. \end{aligned} \quad (38)$$

Summing (37) to N steps, we have

$$\sum_{k=1}^N \left\{ (\bar{e}_k^v)^T \bar{\Phi}_k^v + (\bar{e}_k^v)^T \bar{e}_k^v \frac{\sigma^v}{2} \right\} \geq -\frac{\|\tilde{\theta}_0^v\|^2}{(2\alpha^v p^v)}. \quad (39)$$

Note that we can also simply present the sum of $(\{\|\tilde{\theta}_k^v\|^2 - \|\tilde{\theta}_{k-1}^v\|^2\})/(2\alpha^v p^v)$ on the right-hand side of the previous equation because the item is smaller or equal to zero for every k according to the convergence property (35).

This implies that the normalization factor ρ_k^v guarantees the satisfaction of the conic sector condition in Theorem 1. Furthermore, the specified normalized factor ρ_k^v , defined in (25), plays two important roles in relation to the conic theory. First, it guarantees $\sigma^v < 1$ to avoid the so-called vanished cone problem [11]. Second, it guarantees that the sector conditions of Theorem 1 are satisfied simultaneously by both the original equivalent feedback system in Fig. 2 and the normalized equivalent feedback system [9]–[11]. Therefore, both conditions 1) and 2) of Theorem 1 are fulfilled. Hence, $\tilde{e}_k^v, \tilde{\Phi}_k^v \in L_2$ and using the results of [11], we have $e_k^v, \Phi_k^v \in L_2$.

With the specific normalization factor ρ_k^v , and applying the results of [11] and Theorem 1 on the original $e_k, \Phi_k^v = -H(\hat{\theta}_{k-1}^w, x_{k-1})\tilde{\theta}_{k-1}^v \in L_2$, we obtained a bounded weight estimate error vector $\tilde{\theta}_{k-1}^v$ with the training law (22) for the output layer.

Furthermore, from (38), we have the nonzero learning rate value in (23) meeting the stability condition

$$\begin{aligned} \alpha^v &< \frac{\rho_k^v p^v}{\left\|H(\hat{\theta}_{k-1}^w, x_{k-1})\right\|^2 \|\Delta_k^v\|^2 \|r_k^v\|^2} \\ &\leq \frac{\rho_{\max}^v}{p^v |h_{\min}|^2 |\Delta_{\min}^v|^2 |r_{\min}^v|^2}. \end{aligned} \quad (40)$$

Note that the matrix ‘‘row sum’’ norm [22] is used in (40). As for the number p^v of neurons of the output layer of the SPSA training algorithm, clearly, a larger bound on the nonzero learning rate value α^v can be obtained with a smaller number of neurons. $\triangle\triangle\triangle$

Remark 2: The stability condition (38) agreed with the weight convergence condition (36) because both the conic sector and the Lyapunov convergence conditions provide similar L_2 stability results. Furthermore, according to [9] and [11] with the specific selection of the normalization factor (25), it is easy to extend the L_2 convergence and stability results into L_∞ stability conditions with the bounded supremum according to (38).

However, the weight convergence property (35) is a bit more demanding than (38) in the sense that an adaptive dead zone learning rate α_k^v is required. This, in turn, it reveals the real reason why a small constant gain parameter c_k for the perturbation is preferred in our deterministic approach, similar to the suggestion in [17].

An interesting point is that c_k does not appear explicitly in the proposed SPSA learning law (22). The basic reason is that we deal with the output layer first, which is linear in parameters. Therefore, there is a direct c_k cancellation between the denominator and numerator, which is revealed by the derivation in (31). On the fourth equality of (31), the gradient approximation presentation appears exactly the same as the original one in [7] (see [7, eq. (2.2)]). However, there is no linear layer in the original SPSA papers [7], [8], therefore, the cancellation is not explicit (we have the similar case for the hidden layer; see Section IV for more details). Then, a natural question is whether c_k will affect

the learning procedure as discussed in the original SPSA papers [7], [8]. The answer is yes. This can be seen by expanding the error signal $e_k = \Phi_k^v + \tilde{e}_k^v$ as in (21) and using (42) as presented in the following to replace e_k in (22), then the gain parameter c_k will appear explicitly in one of the additive terms of the SPSA learning law (22) as follows.

Using (32), we have the presentation of the system disturbance in terms of the measurement noise δ_k^v of the SPSA algorithm as

$$\begin{aligned} \varepsilon_k &= - \left\{ H(\hat{\theta}_{k-1}^w, x_{k-1}) \Delta_k^v (H(\hat{\theta}_{k-1}^w, x_{k-1}) \Delta_k^v)^T \right\}^{-1} \\ &\quad \times H(\hat{\theta}_{k-1}^w, x_{k-1}) \Delta_k^v \delta_k^v / (2c_k). \end{aligned} \quad (41)$$

Furthermore, according to the equivalent disturbance (19) of the system with $H_2 = 1$, we have

$$\begin{aligned} \tilde{e}_k^v &= H_2 \left(\tilde{H}(\hat{\theta}_{k-1}^w, \theta^{w*}, x_{k-1}) + \varepsilon_k \right) \\ &= \tilde{H}(\hat{\theta}_{k-1}^w, \theta^{w*}, x_{k-1}) \\ &\quad - \left\{ H(\hat{\theta}_{k-1}^w, x_{k-1}) \Delta_k^v (\Delta_k^v)^T H^T(\hat{\theta}_{k-1}^w, x_{k-1}) \right\}^{-1} \\ &\quad \times H(\hat{\theta}_{k-1}^w, x_{k-1}) \Delta_k^v \delta_k^v / (2c_k). \end{aligned} \quad (42)$$

The equivalent disturbance \tilde{e}_k^v is important in two ways. First, it can be presented in the SPSA learning law (22) as additive terms of (42) by expanding e_k as discussed above. Second, its maximum magnitude \tilde{e}_{\max}^v decides the range of the dead zone for α_k^v in (23). Note also that the measurement noise δ_k^v is defined the same way as in [7], which measures the difference between the true gradient and the gradient approximation as derived in (31). Because the system disturbance ε_k is decided only by external conditions, therefore, a relatively smaller gain parameter c_k will imply a smaller measurement noise δ_k^v according to (41). This allows us to choose the dead zone range mainly according to the external system disturbance ε_k . Furthermore, if the external disturbance is also small, then we can choose a small dead zone range, which ensures that the adaptive learning rate α_k^v has the nonzero value for extended learning capability and better tracking performance.

Interestingly, this also agrees with, to some extent, the original stochastic approach in [7], where c_k is presented as an additive term in the original SPSA learning law (see the equation below Section III-B in [7]). This explains why the gain parameter c_k is indeed presented implicitly in our SPSA learning law (22). Moreover, the randomness of the perturbation also guarantees the nonsingularity of $\{\Delta_k^v (\Delta_k^v)^T\}^{-1}$ in (42), which coincided with the discussion in [7]. \square

V. ROBUST CONIC SECTOR CONDITION FOR THE HIDDEN LAYER TRAINING

We will now derive the weight convergence and boundedness condition for the estimation error of the hidden layer. Similar to the result for the output layer, the hidden layer parameter of the network should also be estimated separately. A conic sector condition and weight convergence will be established for the hidden layer training so that we can prove stability of the whole

system. The development is similar to that in Section III but with significant differences in the detail mathematical derivations.

First, from the mean value theorem and that the activation function in (13) is a nondecreasing function, it can be readily shown that there exist unique positive mean values $\tilde{\mu}_{k-1,i}$, such that

$$\begin{aligned} & h_{k-1,i}(x_{k-1}^T \theta_{k-1,i}^{wI*}) - h_{k-1,i}(x_{k-1}^T \hat{\theta}_{k-1,i}^{wI}) \\ &= \tilde{\mu}_{k-1,i} x_{k-1}^T (\theta_{k-1,i}^{wI*} - \hat{\theta}_{k-1,i}^{wI}) \\ &= \tilde{\mu}_{k-1,i} x_{k-1}^T \tilde{\theta}_{k-1,i}^{wI} \end{aligned} \quad (43)$$

where $\hat{\theta}_{k-1,i}^{wI}$, $\theta_{k-1,i}^{wI*}$, $\tilde{\theta}_{k-1,i}^{wI} \in R^{m_I}$ are the estimated weight [also see the definition below (13)], ideal weight, and weight estimate error vectors linked to the i th hidden layer neurons, respectively. From (13), the maximum value of the derivative of $h_{k-1,i}$, $\dot{h}_{k-1,i} = (\partial h_{k-1,i}(x_{k-1}^T \hat{\theta}_{k-1,i}^{wI})) / (\partial (x_{k-1}^T \hat{\theta}_{k-1,i}^{wI}))$ is λ , and hence

$$\lambda \geq \tilde{\mu}_{k-1,i} \geq 0 (1 \leq i \leq n_h). \quad (44)$$

Then, from (43) and the matrix $H(\hat{\theta}_{k-1}^w, x_{k-1})$ in (12), it can be shown that

$$(H(\theta_{k-1}^{w*}, x_{k-1}) - H(\hat{\theta}_{k-1}^w, x_{k-1})) \hat{\theta}_{k-1}^v = \tilde{\Omega}_{k-1} \tilde{\theta}_{k-1}^w \quad (45)$$

where the matrix $\tilde{\Omega}_{k-1} \in R^{m \times p^w}$ is defined as (46), shown at the bottom of the page, with $\hat{\theta}_{k-1,i}^{v_m} = [\hat{\theta}_{k-1,i}^v \ \hat{\theta}_{k-1,n_h+i}^v \ \cdots \ \hat{\theta}_{k-1,(m-1)n_h+i}^v]^T \in R^m (1 \leq i \leq n_h)$ being the estimated weight vector of the output layer with inputs from the i th hidden layer neuron.

Now, similar to the error (21), we can rewrite the estimation error as

$$\begin{aligned} e_k &= H_2 e_k = H_2 [f_{k-1} - \hat{f}_{k-1} + \varepsilon_k] \\ &= H_2 \left[H(\theta^{w*}, x_{k-1}) \theta^{v*} - H(\hat{\theta}_{k-1}^w, x_{k-1}) \hat{\theta}_{k-1}^v + \varepsilon_k \right] \\ &= H_2 \left[H(\theta^{w*}, x_{k-1}) \theta^{v*} - H(\theta^{w*}, x_{k-1}) \hat{\theta}_{k-1}^v \right. \\ &\quad \left. + H(\theta^{w*}, x_{k-1}) \hat{\theta}_{k-1}^v - H(\hat{\theta}_{k-1}^w, x_{k-1}) \hat{\theta}_{k-1}^v + \varepsilon_k \right] \\ &= H_2 \left[\tilde{H}(\hat{\theta}_{k-1}^w, x_{k-1}) \hat{\theta}_{k-1}^v + H(\theta^{w*}, x_{k-1}) \tilde{\theta}_{k-1}^v + \varepsilon_k \right] \\ &= -H_2 \Phi_k^w + \tilde{e}_k^w \end{aligned} \quad (47)$$

where

$$\Phi_k^w = -\tilde{\Omega}_{k-1} \tilde{\theta}_{k-1}^w \quad (48)$$

$$\tilde{e}_k^w = H_2 (H(\theta^{w*}, x_{k-1}) \tilde{\theta}_{k-1}^v + \varepsilon_k) \quad (49)$$

$$\tilde{H}(\hat{\theta}_{k-1}^w, x_{k-1}) = \left(H(\theta_{k-1}^{w*}, x_{k-1}) - H(\hat{\theta}_{k-1}^w, x_{k-1}) \right). \quad (50)$$

Note that \tilde{e}_k^w is a bounded signal because $\tilde{\theta}_{k-1}^v$ has been proven to be bounded in Section III. Further, similar to the definitions of \hat{f}_{k-1}^+ and \hat{f}_{k-1}^- in (28) and (29), we define the following two perturbations functions to \hat{f}_{k-1} , i.e.:

$$\hat{f}_{k-1}^+ = H(\hat{\theta}_{k-1}^w + c_k \Delta_k^w, x_{k-1}) \hat{\theta}_{k-1}^v \quad (51)$$

$$\hat{f}_{k-1}^- = H(\hat{\theta}_{k-1}^w - c_k \Delta_k^w, x_{k-1}) \hat{\theta}_{k-1}^v \quad (52)$$

where $\Delta_k^w \in R^{p^w}$ is the last p^w components of Δ_k used in (8). From (51) and (52), we have (53), shown at the bottom of the page, with the matrix $\Omega_{k-1} \in R^{m \times p^w}$ being similar to $\tilde{\Omega}_{k-1}$ in (46) except with different mean values μ .

Note also that the constant c_k will be cancelled between the denominator and numerator of (53) if the mean value theorem is applied, similar to the case of output layer (see Remark 2 for more details).

Then, similar to the SPSA training for the output layer, we propose the following normalized hidden layer SPSA training algorithm:

$$\begin{aligned} \hat{\theta}_k^w &= \hat{\theta}_{k-1}^w - \alpha_k^w \hat{g}_k \left(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^v, \Delta_k^w \right) \\ &= \hat{\theta}_{k-1}^w + \alpha_k^w \frac{e_k^T (f_{k-1}^{w+} - f_{k-1}^{w-})}{2c_k \rho_k^w} r_k^w \end{aligned} \quad (54)$$

where the adaptive learning rate of the hidden layer is defined as (55), shown at the bottom of the page, with $1 \geq \alpha^w > 0$ and $\tilde{e}_{\max}^w = \max\{\tilde{e}_k^w\} \forall k$; and the normalized factor is updated as

$$\begin{aligned} \rho_k^w &= \mu \rho_{k-1}^w \\ &+ \max \left\{ \frac{\alpha^w}{p^w} \left(\frac{\lambda}{\lambda_{\min}} \right) \left\| \frac{\hat{f}_{k-1}^+ - \hat{f}_{k-1}^-}{2c_k} \right\|^2, \bar{\rho} \right\} \end{aligned} \quad (56)$$

$$\tilde{\Omega}_{k-1} = \begin{bmatrix} \tilde{\mu}_{k-1,1} \hat{\theta}_{k-1,1}^v x_{k-1}^T & \cdots & \tilde{\mu}_{k-1,n_h} \hat{\theta}_{k-1,n_h}^v x_{k-1}^T \\ \tilde{\mu}_{k-1,1} \hat{\theta}_{k-1,n_h+1}^v x_{k-1}^T & \cdots & \tilde{\mu}_{k-1,n_h} \hat{\theta}_{k-1,2n_h}^v x_{k-1}^T \\ \vdots & & \vdots \\ \tilde{\mu}_{k-1,1} \hat{\theta}_{k-1,(m-1)n_h+1}^v x_{k-1}^T & & \tilde{\mu}_{k-1,n_h} \hat{\theta}_{k-1,p^v}^v x_{k-1}^T \end{bmatrix} = [\tilde{\mu}_{k-1,1} \hat{\theta}_{k-1,1}^{v_m} x_{k-1}^T \ \cdots \ \tilde{\mu}_{k-1,n_h} \hat{\theta}_{k-1,n_h}^{v_m} x_{k-1}^T] \quad (46)$$

$$\frac{\hat{f}_{k-1}^+ - \hat{f}_{k-1}^-}{2c_k} = \left(\frac{H(\hat{\theta}_{k-1}^w + c_k \Delta_k^w, x_{k-1}) - H(\hat{\theta}_{k-1}^w - c_k \Delta_k^w, x_{k-1})}{2c_k} \right) \hat{\theta}_{k-1}^v = \Omega_{k-1} \Delta_k^w \quad (53)$$

$$\alpha_k^w = \begin{cases} \alpha^w, & \text{if } \|e_k\| \geq \tilde{e}_{\max}^w / \sqrt{1 - \left\| \frac{\hat{f}_{k-1}^+ - \hat{f}_{k-1}^-}{2c_k} \right\|^2 \left(\frac{\lambda}{\lambda_{\min}} \right) \frac{\alpha^w \|r_k^w\|^2}{p^w \rho_k^w}} \\ 0, & \text{if } \|e_k\| < \tilde{e}_{\max}^w / \sqrt{1 - \left\| \frac{\hat{f}_{k-1}^+ - \hat{f}_{k-1}^-}{2c_k} \right\|^2 \left(\frac{\lambda}{\lambda_{\min}} \right) \frac{\alpha^w \|r_k^w\|^2}{p^w \rho_k^w}} \end{cases} \quad (55)$$

with λ being the constant of the activation function [which is also the maximum of the mean values $\mu_{k-1,i}$ in Ω_{k-1} of (53)], and λ_{\min} is the minimum of the mean values $\tilde{\mu}_{k-1,i}$ in $\tilde{\Omega}_{k-1}$ of (46). In practice, λ_{\min} can be a small fraction of λ .

The nonzero learning rate value α^w has an upper bound as defined in the following:

$$\alpha^w < \left(\frac{\lambda_{\min}}{\lambda} \right) \frac{\rho_{\max}^w}{p^w |\omega_{\min}|^2 |\Delta_{\min}^w|^2 |r_{\min}^w|^2} \quad (57)$$

where $\omega_{\min} = \min_{k,\ell,j} \{|\hat{\theta}_{k-1,\ell}^v x_{k-1,j}|\}$ is the minimum nonzero value of the elements of Ω defined in (53), while $r_{\min}^w = \min_k \{||r_k^w||\}$, $\Delta_{\min}^w = \min_k \{||\Delta_k^w||\}$ are the nonzero minimum values of the perturbation vectors of the hidden layer, and $\rho_{\max}^w = \max_k \{|\rho_k^w|\}$.

Lemma 2: We will also show that (54) is derived from the SPSA algorithm of the general form (8).

Proof: Indeed, the normalized gradient approximation (against the hidden layer weight vector $\hat{\theta}_{k-1}^w$) can be derived as (58), shown at the bottom of the page, where the equivalent disturbance δ_k^w is defined as

$$\begin{aligned} \delta_k^w &= (\hat{f}_{k-1}^{w-} - \hat{f}_{k-1}^{w+})^T \varepsilon_k \\ &+ \frac{1}{2} (\hat{f}_{k-1}^{w-} - \hat{f}_{k-1}^{w+})^T (\hat{f}_{k-1}^{w+} + \hat{f}_{k-1}^{w-} - 2\hat{f}_{k-1}) \cdot \quad (59) \\ &\quad \Delta \Delta \Delta \end{aligned}$$

The main result concerning the hidden layer of the neural network can now be stated as the following.

Theorem 3: The operator $H_1^w : e_k \rightarrow \Phi_k^w$, which represents the SPSA learning algorithm of the hidden layer, and H_2 satisfying the conditions 1) and 2) of Theorem 1 provide guaranteed stability under the conic sector framework, if the nonzero learning rate value α^w satisfies (57). Furthermore, the hidden layer weight is convergent in the sense that the L_2 -norm of $\tilde{\theta}_k^w$ is a nonincreasing sequence.

Proof: Consider that the SPSA is an approximation of the gradient algorithm [7], and using the property of local minimum points of the gradient $-(\partial(e_k^T e_k))/(\partial(\hat{\theta}_{k-1}^w)^T) \tilde{\theta}_{k-1}^w \geq 0$ [5], we have

$$\begin{aligned} 0 &\leq -\frac{\partial(e_k^T e_k)}{\partial(\hat{\theta}_{k-1}^w)^T} \tilde{\theta}_{k-1}^w = \sum_{i=1}^{n_h} \left\{ -\frac{\partial(e_k^T e_k)}{\partial(\hat{\theta}_{k-1,i}^w)^T} \tilde{\theta}_{k-1,i}^w \right\} \\ &= \sum_{i=1}^{n_h} \left\{ \dot{h}_{k-1,i} e_k^T \hat{\theta}_{k-1,i}^{v_m} x_{k-1,i}^T \tilde{\theta}_{k-1,i}^w \right\} \end{aligned}$$

$$\begin{aligned} &= \sum_{i=1}^{n_h} \left\{ \frac{\dot{h}_{k-1,i}}{\tilde{\mu}_{k-1,i}} \tilde{\mu}_{k-1,i} e_k^T \hat{\theta}_{k-1,i}^{v_m} x_{k-1,i}^T \tilde{\theta}_{k-1,i}^w \right\} \\ &\leq \frac{\lambda}{\lambda_{\min}} \sum_{i=1}^{n_h} \left\{ \tilde{\mu}_{k-1,i} e_k^T \hat{\theta}_{k-1,i}^{v_m} x_{k-1,i}^T \tilde{\theta}_{k-1,i}^w \right\} \\ &= \frac{\lambda}{\lambda_{\min}} e_k^T (\tilde{\Omega}_{k-1} \tilde{\theta}_{k-1}^w) = -\frac{\lambda}{\lambda_{\min}} e_k^T \Phi_k^w \quad (60) \end{aligned}$$

where λ is the maximum value of the derivative $\dot{h}_{k-1,i}$ of the activation function in (13), and $\lambda_{\min} \neq 0$ is the minimum nonzero value of the mean values $\tilde{\mu}_{k-1,i}$ defined in (44). (Note that the inequality is always true if $\lambda_{\min} = 0$, which implies $\dot{h}_{k-1,i} = \tilde{\mu}_{k-1,i} = 0$.)

Now, from (54), we have

$$\tilde{\theta}_k^w = \tilde{\theta}_{k-1}^w - \alpha_k^w \frac{e_k^T (f_{k-1}^{w+} - f_{k-1}^{w-})}{2c_k \rho_k^w} r_k^w \quad (61)$$

where $\tilde{\theta}_k^w$ is as defined in (17). Then, using (61), (53), and (60), and similar to the proof of Theorem 2, we get

$$\begin{aligned} &||\tilde{\theta}_k^w||^2 - ||\tilde{\theta}_{k-1}^w||^2 \\ &= - \left\{ 2\alpha_k^w \frac{e_k^T (\hat{f}_{k-1}^{w+} - \hat{f}_{k-1}^{w-})}{2c_k \rho_k^w} (r_k^w)^T \tilde{\theta}_{k-1}^w \right\} \\ &+ \left\| \left(\alpha_k^w \frac{e_k^T (\hat{f}_{k-1}^{w+} - \hat{f}_{k-1}^{w-})}{2c_k \rho_k^w} r_k^w \right) \right\|^2 \\ &= \sum_{i=1}^{n_h} \left\{ -\mu_{k-1,i} e_k^T \hat{\theta}_{k-1,i}^{v_m} x_{k-1,i}^T \tilde{\theta}_{k-1,i}^w \right\} \\ &\times \left\{ \tilde{\theta}_{k-1}^{wT} (\tilde{\theta}_{k-1}^w \tilde{\theta}_{k-1}^{wT})^{-1} \Delta_k^w (r_k^w)^T \tilde{\theta}_{k-1}^w \right\} \frac{2\alpha_k^w}{\rho_k^w} \\ &+ \left\| \left(\alpha_k^w \frac{e_k^T (\hat{f}_{k-1}^{w+} - \hat{f}_{k-1}^{w-})}{2c_k \rho_k^w} r_k^w \right) \right\|^2 \\ &= \sum_{i=1}^{n_h} \left\{ \frac{\mu_{k-1,i}}{\tilde{\mu}_{k-1,i}} (-\tilde{\mu}_{k-1,i} e_k^T \hat{\theta}_{k-1,i}^{v_m} x_{k-1,i}^T \tilde{\theta}_{k-1,i}^w) \right\} \\ &\times \left(\frac{2p^w \alpha_k^w}{\rho_k^w} \right) + \left\| \alpha_k^w \frac{e_k^T (\hat{f}_{k-1}^{w+} - \hat{f}_{k-1}^{w-})}{2c_k \rho_k^w} r_k^w \right\|^2 \end{aligned}$$

$$\begin{aligned} &\frac{L(\theta_{k-1}^v, \theta_{k-1}^w + c_k \Delta_k^w) - L(\theta_{k-1}^v, \theta_{k-1}^w - c_k \Delta_k^w) + \delta_k^w r_k^w}{2c_k \rho_k} \\ &= \frac{||f_{k-1} - \hat{f}_{k-1}^{w+}||^2 - ||f_{k-1} - \hat{f}_{k-1}^{w-}||^2 + 2\delta_k^w}{4c_k \rho_k^w} r_k^w = \frac{(f_{k-1} - \hat{f}_{k-1}^{w+} + f_{k-1} - \hat{f}_{k-1}^{w-})^T (\hat{f}_{k-1}^{w-} - \hat{f}_{k-1}^{w+}) + 2\delta_k^w}{4c_k \rho_k^w} r_k^w \\ &= \frac{[2f_{k-1} - \hat{f}_{k-1}^{w+} - \hat{f}_{k-1}^{w-} + \hat{f}_{k-1}^{w+} + \hat{f}_{k-1}^{w-} - 2\hat{f}_{k-1} + 2\varepsilon_k]^T (\hat{f}_{k-1}^{w-} - \hat{f}_{k-1}^{w+})}{4c_k \rho_k^w} r_k^w \\ &= \frac{[f_{k-1} - \hat{f}_{k-1} + \varepsilon_k]^T (\hat{f}_{k-1}^{w-} - \hat{f}_{k-1}^{w+})}{2c_k \rho_k^w} r_k^w = \frac{e_k^T (\hat{f}_{k-1}^{w-} - \hat{f}_{k-1}^{w+})}{2c_k \rho_k^w} r_k^w = \hat{g}_k(\hat{\theta}_{k-1}^w, \hat{\theta}_{k-1}^w, \Delta_k^w) \quad (58) \end{aligned}$$

$$\begin{aligned}
&\leq \frac{\lambda_{\min}}{\lambda} \sum_{i=1}^{n_h} \left\{ -\tilde{\mu}_{k-1,i} e_k^T \hat{\theta}_{k-1,i}^w x_{k-1}^T \tilde{\theta}_{k-1,i}^w \right\} \\
&\quad \times \left(\frac{2p^w \alpha_k^w}{\rho_k^w} \right) + \left\| \alpha_k^w \frac{e_k^T (\hat{f}_{k-1}^{w+} - \hat{f}_{k-1}^{w-})}{2c_k \rho_k^w} r_k^w \right\|^2 \\
&\leq 2\alpha_k^w \left(\frac{\lambda_{\min}}{\lambda} \right) \frac{p^w}{\rho_k^w} e_k^T \Phi_k^w \\
&\quad + \|e_k\|^2 \left\| \frac{\hat{f}_{k-1}^{w+} - \hat{f}_{k-1}^{w-}}{2c_k} \right\|^2 \|r_k^w\|^2 \left(\frac{\alpha_k^w}{\rho_k^w} \right)^2. \quad (62)
\end{aligned}$$

Similar to the output layer case, we have the following Lyapunov function for convergence analysis:

$$\begin{aligned}
&\left\| \tilde{\theta}_k^w \right\|^2 - \left\| \tilde{\theta}_{k-1}^w \right\|^2 \\
&\leq 2\alpha_k^w \left(\frac{\lambda_{\min}}{\lambda} \right) \frac{p^w}{\rho_k^w} e_k^T (\tilde{e}_k^w - e_k) \\
&\quad + \|e_k\|^2 \left\| \frac{\hat{f}_{k-1}^{w+} - \hat{f}_{k-1}^{w-}}{2c_k} \right\|^2 \|r_k^w\|^2 \left(\frac{\alpha_k^w}{\rho_k^w} \right)^2 \\
&\leq \alpha_k^w \left(\frac{\lambda_{\min}}{\lambda} \right) \frac{p^w}{\rho_k^w} \left[\|\tilde{e}_k^w\|^2 \right. \\
&\quad \left. - \left(1 - \left\| \frac{\hat{f}_{k-1}^{w+} - \hat{f}_{k-1}^{w-}}{2c_k} \right\|^2 \right) \right. \\
&\quad \left. \times \|r_k^w\|^2 \left(\frac{\alpha_k^w}{p^w \rho_k^w} \right) \left(\frac{\lambda}{\lambda_{\min}} \right) \|e_k\|^2 \right] \\
&\leq 0. \quad (63)
\end{aligned}$$

Now, similar to the case of output layer, for the nonzero learning rate value α^w by defining the normalized estimation error $\tilde{e}_k^w = e_k (\sqrt{\rho_k^w})^{-1}$ and the normalized output $\tilde{\Phi}_k^w = \Phi_k^w (\sqrt{\rho_k^w})^{-1} = -\tilde{\Omega}_{k-1} \tilde{\theta}_{k-1}^w (\sqrt{\rho_k^w})^{-1}$, we have

$$\begin{aligned}
&\frac{\left\{ \left\| \tilde{\theta}_k^w \right\|^2 - \left\| \tilde{\theta}_{k-1}^w \right\|^2 \right\} \left(\frac{\lambda}{\lambda_{\min}} \right)}{2\alpha^w p^w} \\
&\leq \left\{ e_k^T \tilde{\Phi}_k^w (\rho_k^w)^{-1} \right\} \\
&\quad + \frac{\|e_k\|^2 (\rho_k^w)^{-2} \left\{ \frac{\alpha^w}{p^w} \left(\frac{\lambda}{\lambda_{\min}} \right) \left\| \frac{\hat{f}_{k-1}^{w+} - \hat{f}_{k-1}^{w-}}{2c_k} \right\|^2 \|r_k^w\|^2 \right\}}{2} \\
&\leq \left\{ (\tilde{e}_k^w)^T \tilde{\Phi}_k^w \right\} + \frac{\|\tilde{e}_k^w\|^2 \sigma^w}{2} \quad (64)
\end{aligned}$$

if the normalized factor ρ_k^w is chosen to guarantee the existence of a constant σ^w such that

$$\begin{aligned}
0 &\leq \left\{ \frac{\alpha^w}{p^w \rho_k^w} \left(\frac{\lambda}{\lambda_{\min}} \right) \left\| \frac{\hat{f}_{k-1}^{w+} - \hat{f}_{k-1}^{w-}}{2c_k} \right\|^2 \|r_k^w\|^2 \right\} \\
&\leq \sigma^w < 1. \quad (65)
\end{aligned}$$

Summing (64) to N steps, we have

$$\sum_{k=1}^N (\tilde{e}_k^w)^T \tilde{\Phi}_k^w + \frac{1}{2} \sigma^w (\tilde{e}_k^w)^T \tilde{e}_k^w \geq -\frac{1}{2\alpha^w p^w} \left(\frac{\lambda}{\lambda_{\min}} \right) \left\| \tilde{\theta}_0^w \right\|^2. \quad (66)$$

Therefore, the conic sector conditions of Theorem 1 are satisfied and hence $\tilde{e}_k^w, \tilde{\Phi}_k^w \in L_2$, and with the results of [11] we have $e_k^w, \Phi_k^w \in L_2$.

To derive the theoretical upper bound of the nonzero learning rate value α^w , we note from (53)

$$\begin{aligned}
\left\| \frac{\hat{f}_{k-1}^{w+} - \hat{f}_{k-1}^{w-}}{2c_k} \right\|^2 &= \Delta_k^{wT} \Omega_{k-1}^T \Omega_{k-1} \Delta_k^w \\
&\geq (p^w)^2 |\omega_{\min}|^2 |\Delta_{\min}^w|^2.
\end{aligned}$$

Hence, we have

$$\begin{aligned}
\alpha^w &< \left(\frac{\lambda_{\min}}{\lambda} \right) \frac{\rho_k^w p^w}{\|\Omega_{k-1} \Delta_k^w\|^2 \|r_k^w\|^2} \\
&< \left(\frac{\lambda_{\min}}{\lambda} \right) \frac{\rho_{\max}^w}{p^w |\omega_{\min}|^2 |\Delta_{\min}^w|^2 \|r_{\min}^w\|^2}.
\end{aligned}$$

Note that the matrix ‘‘row sum’’ norm [22] is used to derive the last inequality. $\triangle\triangle\triangle$

VI. DISCUSSIONS AND IMPLEMENTATION OF THE SPSA ALGORITHM

We can draw the following remarks about our SPSA controller.

- 1) In addition to the stability proof, one of the most interesting contributions of this paper is the revelation of the relationship between the conventional adaptive control system and generalization theory. We know that a relatively larger learning rate will contribute to a faster convergence speed of the SPSA training algorithm, and our (57) and (40) reveal that a relatively small number of neurons, i.e., small parameters p^w and p^v will yield relatively larger bounds for the nonzero learning rate values α^w and α^v . This concept is closely linked to the generalization property of neural network theory. Although the theoretical upper bounds for α^w and α^v in (57) and (40) may not be computed, they serve the purpose of illuminating the generalization property of neural network theory. As illustrated in the simulation studies later, a reasonable number of neurons (not necessary the biggest network) with maximum learning rates bounded by (57) and (40), respectively, can indeed achieve a good generalization performance in terms of reduced control signal error and fast tracking performance. This can be further developed into an adaptive pruning based algorithm in future research, following the ideas in [20].
- 2) The estimation error vector e_k and the tracking error vector s_k , which is linked to the former by (5) through a stable first-order filter ($k_v < 1$), are bounded as in Theorem 2 for the output layer training without any influence from the hidden layer. The important role of Theorem 3 is that the weight estimate error vector $\tilde{\theta}_k^w$ should be bounded to meet

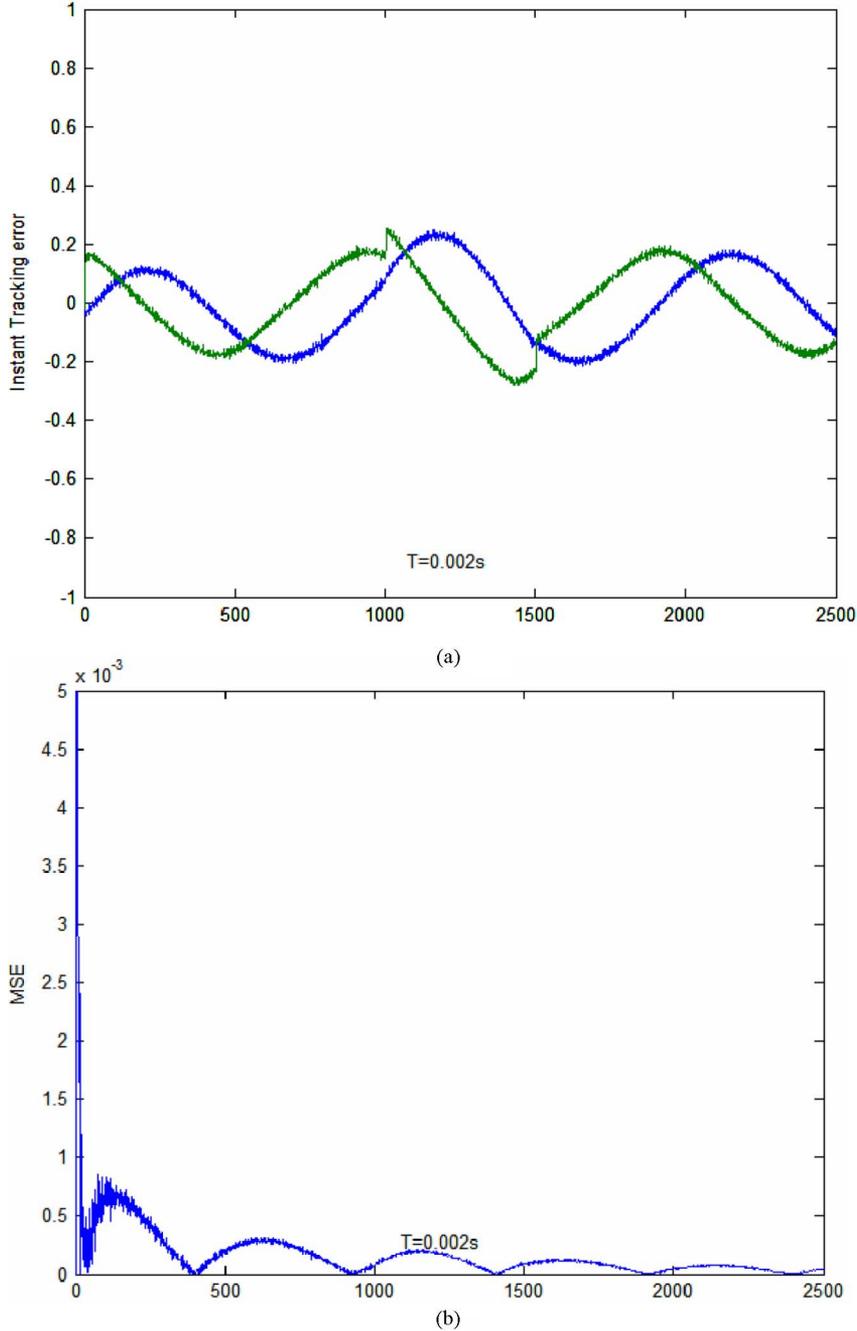


Fig. 3. (a) Tracking error s_k using the standard BP algorithm. (b) MSE of s_k using the standard BP algorithm.

the stability condition, and in turn yields small estimation error e_k and tracking error s_k .

- 3) The SPSA measurement noises δ_k^v and δ_k^w are linked dynamically and proportionally to the overall control system noise ε_k via (32) and (59), i.e.,

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

and

$$\begin{aligned} \delta_k^w &= \left(\hat{f}_{k-1}^{w-} - \hat{f}_{k-1}^{w+} \right)^T \varepsilon_k \\ &+ \frac{1}{2} \left(\hat{f}_{k-1}^{w-} - \hat{f}_{k-1}^{w+} \right)^T \left(\hat{f}_{k-1}^{w+} + \hat{f}_{k-1}^{w-} - 2\hat{f}_{k-1} \right). \end{aligned}$$

The previous equations imply that the gain parameter c_k is embedded in the training error signal e_k , which can be expanded by the equivalent disturbances \tilde{e}_k^v and \tilde{e}_k^w , in the training laws (22) and (54) (see Remark 2 in Section III and note that c_k is also implicitly inside of the second equation), respectively. These require a small constant c_k to reduce the perturbation level of the SPSA training.

- 4) To the best of our knowledge, it is not possible to mix the stochastic and deterministic approaches as adopted in the original papers [7], [8] and this paper, respectively. However, there are differences and similar relationships, to a certain extent, as revealed by this paper. For example, the stochastic approach in [7] and [8] relied on a number of as-

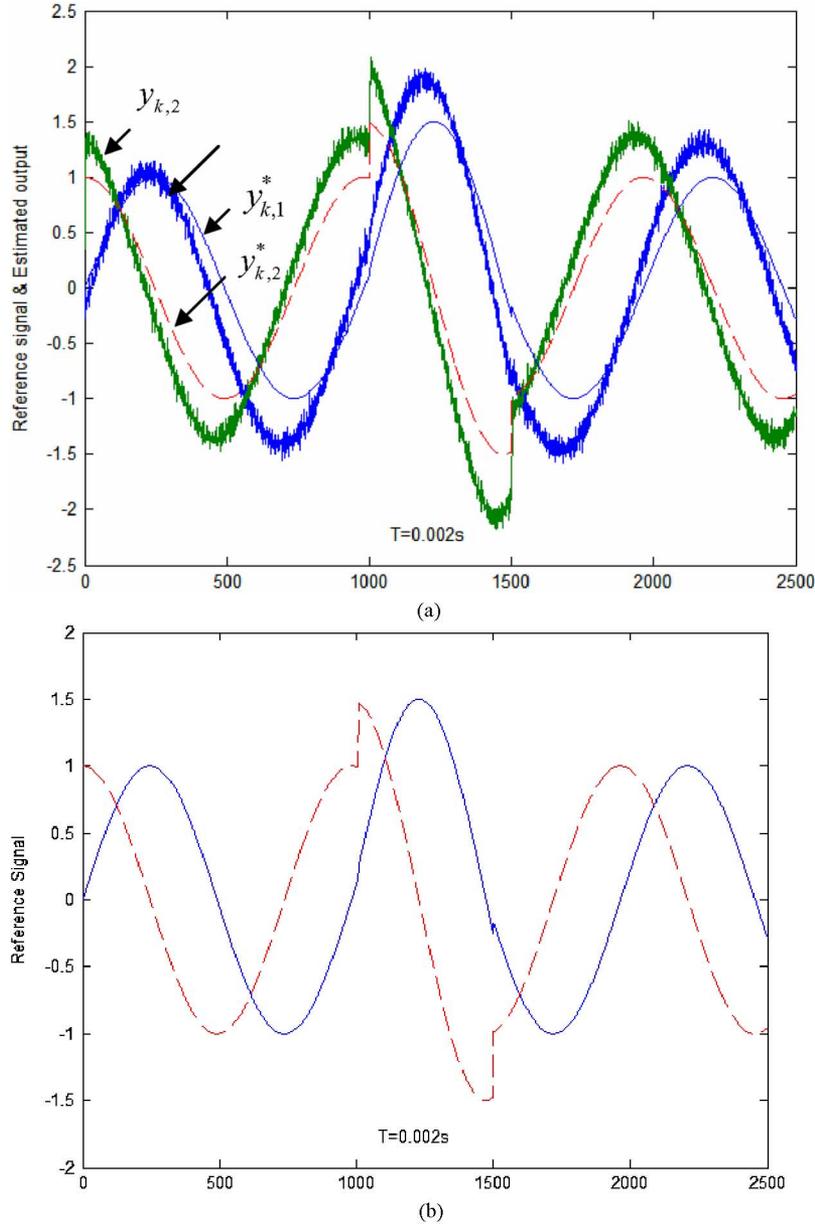


Fig. 4. (a) Output y_k and reference signal y_k^* using the standard BP algorithm. (b) Reference signal y_k^* used for all the algorithms in the simulation studies.

assumptions and regularity conditions to adjust the gain parameter c_k and achieves zero loss function as $k \rightarrow \infty$. This is interesting and can be interpreted by our deterministic analysis [as in (31) and (32)] as follow (see also Remark 2 in Section III): the original SPSA algorithm tends to drive the loss function to approximate zero if the perturbation gain parameter $c_k \rightarrow 0$ under a number of assumptions, which is clearly indicated by our (31) and (32) (similarly for the hidden layer in Section IV), which in turn implies that the optimal parameter estimate is reached by the stochastic SPSA approach. However, we take a slightly different route, as in the traditional deterministic study for adaptive control systems [5], [9]–[11], which does not require the system estimate error to be zero. In other words, we do not necessarily have the SPSA loss function (6) approaching zero and neither the gradient approximation;

we minimize them instead. There is a good reason to do this because we take the traditional adaptive control approach by introducing the system disturbance ε_k in (1), which is uncontrollable under the traditional adaptive control system concept [9]–[11]. Then, we follow the traditional adaptive control design to minimize the learning error e_k rather than driving it to zero by implementing the dead zone scheme (stop leaning when the error e_k is too small to prevent weigh drifting), because of the uncontrollable disturbance ε_k in (1). This makes things a little bit easier for our deterministic approach to train the SPSA because we will not be restricted by the assumptions and regularity conditions in the original stochastic approach [7], [8] (of course, as a tradeoff, we will lose some of the nice properties of the stochastic convergence analysis because we ignore the statistical properties of distur-

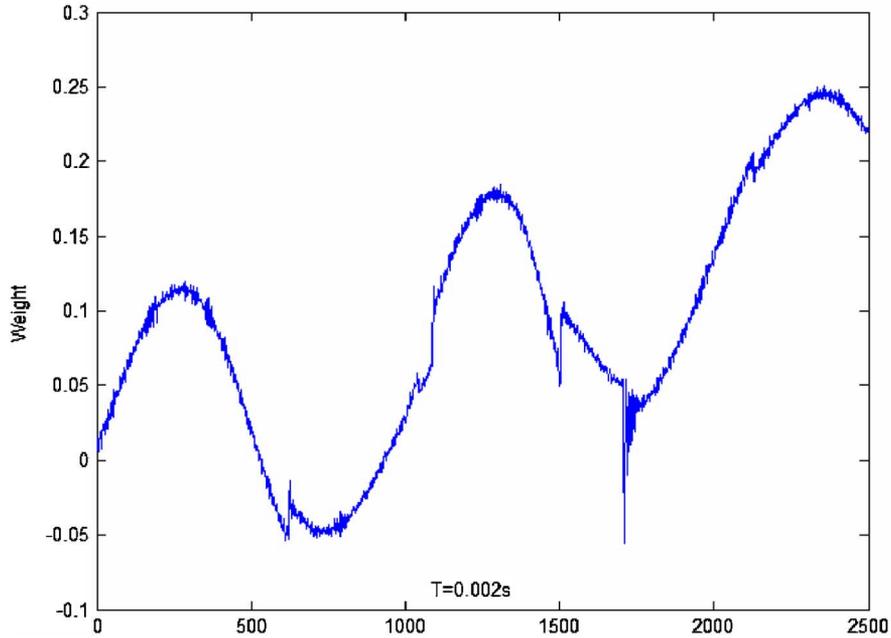


Fig. 5. Estimated parameter $\hat{\theta}_{k,1}^w$ of the hidden layer using the standard BP algorithm.

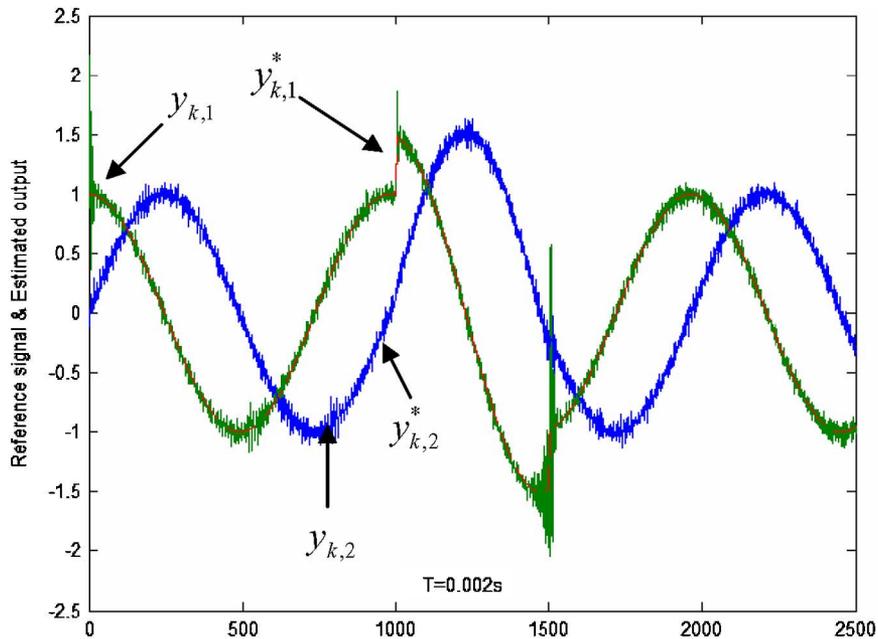


Fig. 6. Output y_k and reference signal y_k^* using the robust SPSA-based neural controller.

bances and simply treat them as bounded signals). Thus, the simultaneous perturbation in our deterministic SPSA approach makes sense in terms of preventing the weight drifting behavior rather than convergence in the stochastic sense. Our deterministic convergence proof in Theorems 2 and 3 shows that the perturbation level is indeed related to the gain parameter c_k , and in turn, the range of dead zone. The advantage of our SPSA approach over the pure dead zone study is that we can choose a suitably small c_k (any value between 0.05 and 0.1 as suggested by [17]) to control the range of dead zone to make the adaptive learning rate nonzero for most times and keeping the system adaptive for slow time varying systems.

5) We derived the weight convergence analysis based on the Lyapunov function as in Theorems 2 and 3. Therefore, we need to use the dead zone to switch off training if the error e_k is too small to prevent the weight drifting behavior. In fact, for SPSA, this is not necessary because simultaneous perturbations prevent the error signal from being “too small” if a suitable gain parameter c_k is chosen. Therefore, the perturbation and the dead zone have a somewhat overlapped role. However, the theoretical addition of the weight convergence analysis brings out an extra benefit for us to understand the role of c_k as discussed in Remark 2. Because there is a c_k cancellation in (31), it means that c_k does not contribute to the stability and convergence condi-

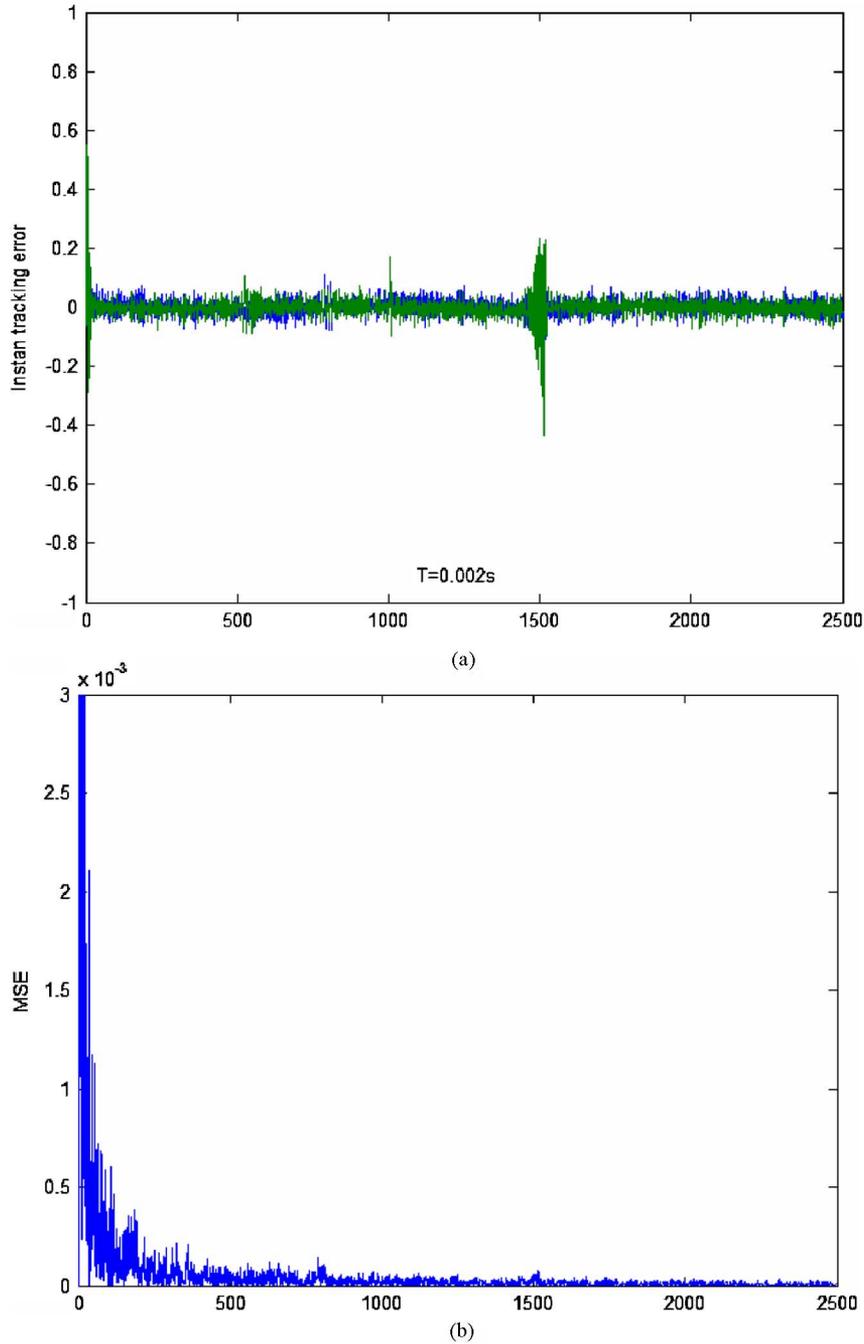


Fig. 7. (a) Tracking error s_k of the robust SPSA-based algorithm. (b) MSE of the tracking error s_b of the robust SPSA-based algorithm.

tion (38) (as well as the upper bound of the learning rate as they are from the same source) directly. However, through further analysis of the relationship between the measurement noise δ_k^v and the system disturbance ε_k as well as the error signal e_k , we find that a smaller c_k implies a smaller measurement noise δ_k^v , which, in turn, will result in a better convergence performance because we can choose a suitable dead zone range according to the external system disturbance ε_k . Therefore, the choice of c_k does not affect the convergence and stability condition (36) [similarly for (38)] directly. However, it does affect the weight convergence property in the sense that we can choose a smaller

dead zone range to obtain more accurate tracking performance and adaptive capability without stop learning, i.e., set the adaptive learning rate to zero. There is also a similar analysis of c_k in the original stochastic approach in terms of an additive bias in the learning law [7], [8].

In summary, our proposed robust neural controller algorithm can be implemented as follows (refer to Fig. 1).

- Step 1) Form the new input vector x_{k-1} of the neural network defined in (11).
- Step 2) Calculate the neural network output \hat{f}_{k-1} using the input state x_{k-1} and the existing or initial weights of the network in the first iteration.

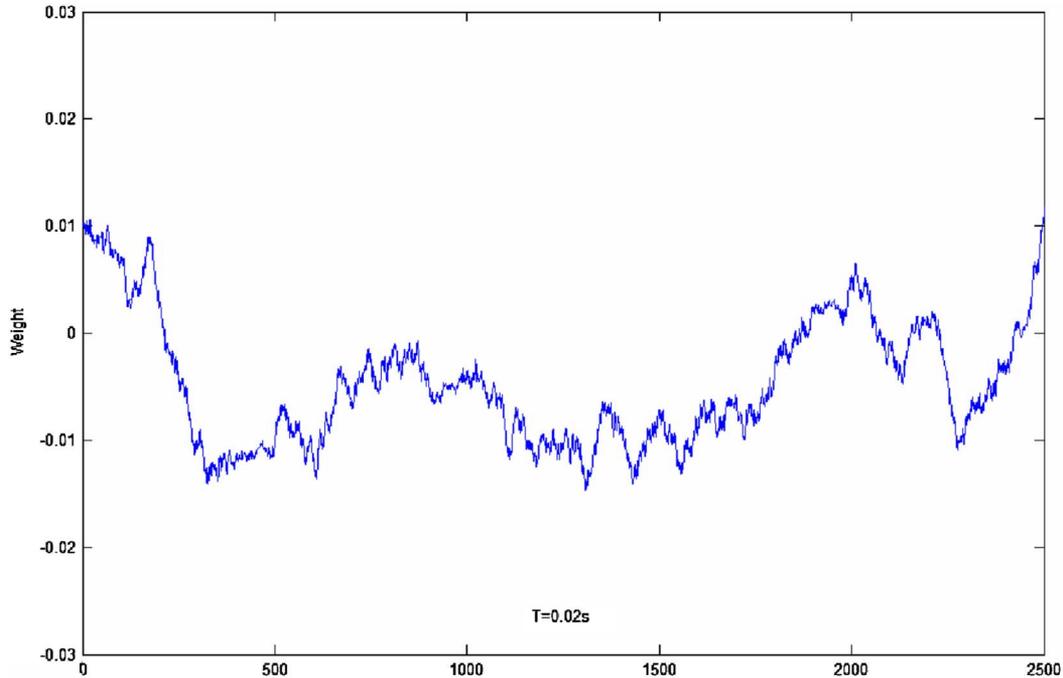


Fig. 8. Estimated parameter $\hat{\theta}_{k,1}^w$ of the hidden layer using the robust SPSA-based algorithm.

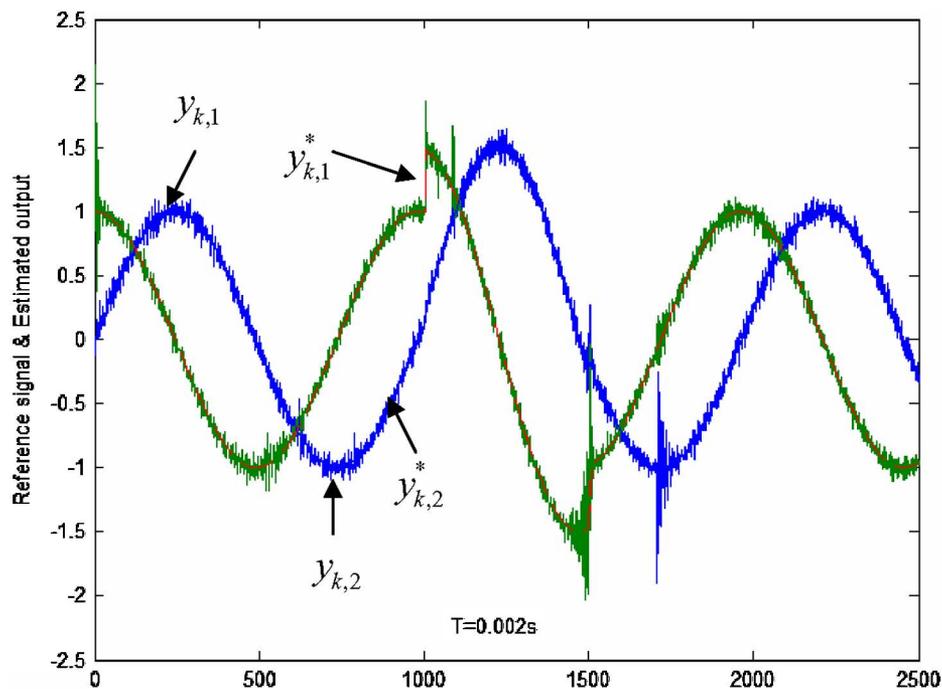


Fig. 9. Output y_k and reference signal y_k^* using a fixed PID controller with changes in the input signal.

- Step 3) The control input u_{k-1} is calculated based on (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 (5).
 Step 5) The implicit estimation error signal e_k is then used to train the neural network and calculates the new weights $\hat{\theta}_k^v$ and $\hat{\theta}_k^w$ using learning laws in (22) and

(54) for the output and hidden layers, respectively (one step at each iteration).

- Step 6) Go back to Step 1) to continue the iteration.

VII. SIMULATION RESULTS

Consider a two-link direct drive robot model with its I/O discrete time version obtained from Euler's rule as follows [15],

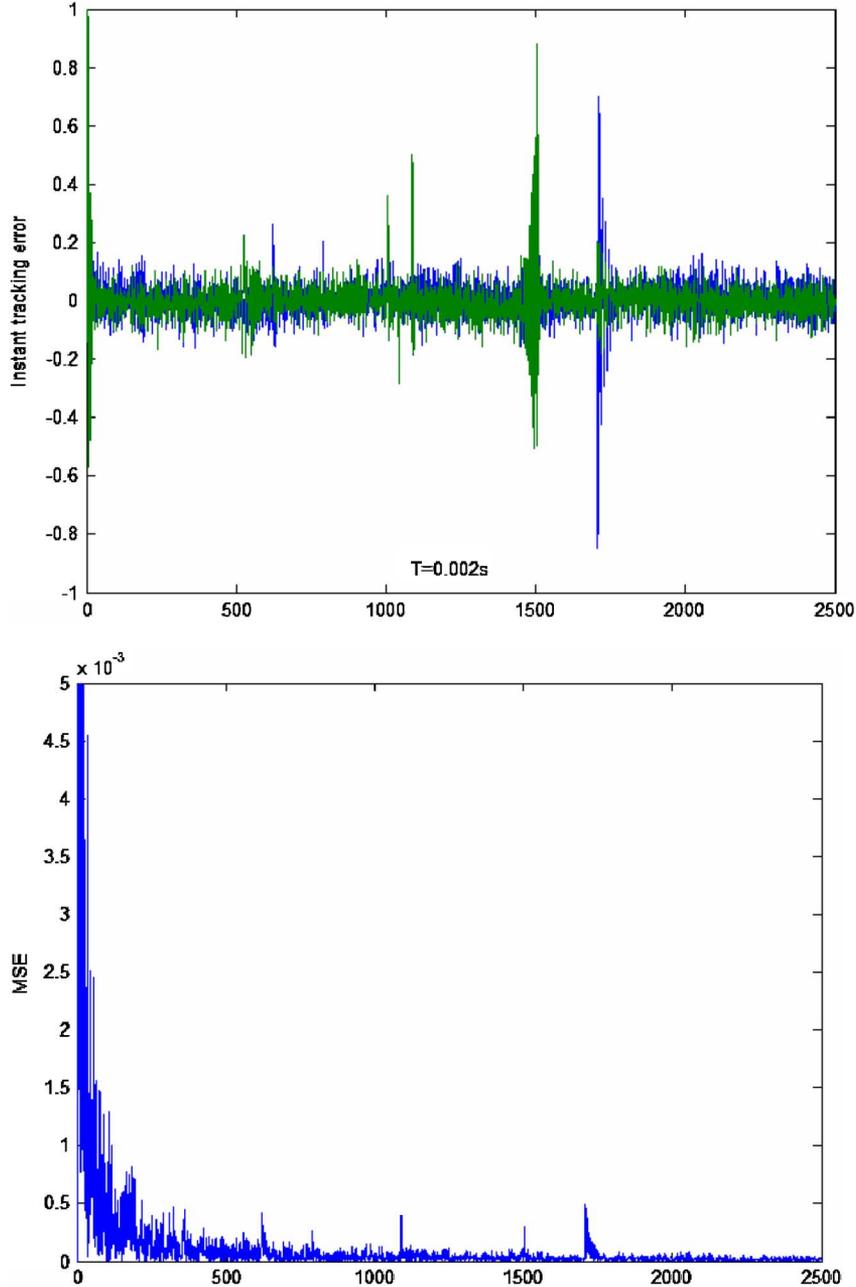


Fig. 10. (a) Tracking error signal of the fixed PID controller with changes in the reference signal. (b) MSE of the tracking error signal of the fixed PID controller with change of the reference signal.

[19]:

$$y_k = \begin{bmatrix} y_{k,1} \\ y_{k,2} \end{bmatrix} = \begin{bmatrix} y_{k-1,2} + T y_{k-2,2} \\ f_{k-1}(y_{k-1}, y_{k-2}) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u_{k-1} + \begin{bmatrix} 0 \\ I \end{bmatrix} \varepsilon_k$$

where $y_{k,1}$ and $y_{k,2}$ are the joint angle and velocity vectors, respectively, T is the sampling period, u_{k-1} is the torque control signal, ε_k is a normally distributed disturbance with a bounded magnitude $\|\varepsilon_k\| \leq 0.2$, and $f_{k-1}(y_{k-1}, y_{k-2})$ is a nonlinear function defined by

$$f_{k-1}(y_{k-1}, y_{k-2}) = y_{k-1,2} - T M^{-1}(y_{k-1,1}) \times [V(y_{k-1}, y_{k-2}) + F(y_{k-1}, y_{k-2})]$$

and where the configuration dependent inertia matrix, the centrifugal and coriolis effect, and the coulomb friction are, respectively, given as shown in the equation at the top of the next page.

A three-layered neural network as defined in Section II is used for this simulation study, which has 30 hidden layer neurons and two output neurons (to ensure a fair comparison, we choose only a reasonable number of neurons). It was trained by the standard BP and SPSA training algorithm with the same control structure shown in Fig. 1. The desired joint trajectory is selected as

$$y_k^* = \begin{bmatrix} y_{k1}^* \\ y_{k2}^* \end{bmatrix} = \begin{bmatrix} (\pi/4) \sin(\pi \cdot kT) \\ (\pi/4) \cos(\pi \cdot kT) \end{bmatrix}.$$

The sampling period is $T = 0.002$ s. The proportional gain parameter of the fixed controller is given as $k_v = 0.5$ and all the initial conditions are set to zero.

We use a variable input tracking signal, where the magnitude of the reference signal y_k^* changes from $1/4$ to $1/3$ and then back to $1/4$ at the 1000 and 1600 time instances. Note that for

$$\begin{aligned}
M(y_{k-1,1}) &= \begin{bmatrix} 3.32 + 0.32 \cos(y_{k-1,1}) & 0.12 + 0.16 \cos(y_{k-1,1}) \\ 0.12 + 0.16 \cos(y_{k-1,1}) & 0.12 \end{bmatrix} \\
V(y_{k-1}, y_{k-2}) &= \begin{bmatrix} -(y_{k-1,2} - y_{k-2,2})(2(y_{k-1,1} - y_{k-2,1}) + y_{k-1,2} - y_{k-2,2})0.16 \sin(y_{k-2,2})/T^2 \\ 0.16(y_{k-1,1} - y_{k-2,1})^2 \sin(y_{k-2,2})/T^2 \end{bmatrix} \\
F(y_{k-1}, y_{k-2}) &= \begin{bmatrix} 5.3 \operatorname{sgn}((y_{k-1,1} - y_{k-2,1})/T) \\ 1.1 \operatorname{sgn}((y_{k-1,2} - y_{k-2,2})/T) \end{bmatrix}
\end{aligned}$$

TABLE I
TRACKING ERROR COMPARISON

s_k	BP	SPSA	PID
Mean	0.135	0.0433	0.163
MSE	0.00036	0.00014	0.00041

nonlinear control systems, a change in the reference input signal magnitude is equivalent to a change in the system model setting because the superposition theory is no longer valid. From the simulation results, the neural controller using the standard BP algorithm performs the worst, with relatively larger tracking errors. This is because of the drifting of weight $\hat{\theta}_{k,1}$, which is the first element of the weight vector $\hat{\theta}_k$. The relevant simulation results with the standard BP neural controller are shown in Figs. 3–5 [where the reference signal y_k^* is given in Fig. 4(b)] to illustrate clearly the slow time-varying nature of the reference input). In comparison, Figs. 6 and 7 show the improved outputs of the plant and the error signal when the robust SPSA-based neural controller is used.

As discussed in Section IV, a suitable number of neurons (we choose 30 neurons for both the BP and SPSA controllers) with maximum learning rates calculated by (26) and (57), respectively, enable the robust SPSA controller to achieve a good generalization performance in terms of reduced control signal error and faster tracking performance. Fig. 8 shows that the robust SPSA-based neural controller has a drift-free parameter estimate. It also outperforms the proportional–integral–derivative (PID) controller as it possesses good generalization properties and has fast learning capabilities. The fixed PID controller performs quite well with optimal parameters before the change of set point, but there is a larger transient when set point change occurs, as shown in Figs. 9 and 10.

The overall performance comparison among the three systems is summarized in Table I, in terms of the mean value of mean square errors (MSE) of the tracking error s_k .

VIII. CONCLUSION

The robust neural controller based on the SPSA has been developed, and conditions for guaranteed stability with a normalized learning algorithm have been derived. Complete stability analysis is performed for the closed-loop control system by using the conic sector theory. Furthermore, an interesting contribution of this paper is the revelation of the relationship between the conventional adaptive control system and generalization theory. That is, a relatively larger learning rate will

contribute to a faster convergence speed of the SPSA training algorithm and our results showed that this can be achieved with a relatively small number of neurons. Simulation results show that the proposed robust neural controller performs better than a neural controller based on the standard BP algorithm or the PID controller.

REFERENCES

- [1] M. M. Polycarpou and P. A. Ioannou, "Learning and convergence analysis of neural-type structured networks," *IEEE Trans. Neural Netw.*, vol. 3, no. 1, pp. 39–50, Jan. 1992.
- [2] Y. L. Wu and Q. Song, "A normalized adaptive training of recurrent neural networks with augmented error gradient," *IEEE Trans. Neural Netw.*, vol. 19, no. 2, pp. 351–356, Feb. 2008.
- [3] A. Y. Alanis, E. N. Sanchez, and A. G. Loukianov, "Discrete-time adaptive backstepping nonlinear control via high-order neural networks," *IEEE Trans. Neural Netw.*, vol. 18, no. 4, pp. 1185–1195, Jul. 2007.
- [4] W. Lan and J. Huang, "Neural-network-based approximate output regulation of discrete-time nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 18, no. 4, pp. 1196–1208, Jul. 2007.
- [5] Q. Song, J. Xiao, and Y. C. Soh, "Robust back-propagation training algorithm for multi-layered neural tracking controller," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1133–1141, Sep. 1999.
- [6] S. Jagannathan and F. L. Lewis, "Discrete-time neural net controller for a class of nonlinear dynamic systems," *IEEE Trans. Autom. Control*, vol. 41, no. 11, pp. 1693–1699, Nov. 1996.
- [7] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Autom. Control*, vol. 37, no. 3, pp. 332–341, Mar. 1992.
- [8] J. C. Spall and J. A. Cristion, "Model free control of nonlinear stochastic systems with discrete-time measurements," *IEEE Trans. Autom. Control*, vol. 43, no. 9, pp. 1198–1210, Sep. 1998.
- [9] R. Ortega, L. Praly, and I. D. Landau, "Robustness of discrete-time direct adaptive controllers," *IEEE Trans. Autom. Control*, vol. AC-30, no. 12, pp. 1179–1187, Dec. 1985.
- [10] Q. Song, R. Katebi, and M. J. Grimble, "Robust multivariable implicit adaptive control," *IMA J. Math. Control Inf.*, 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," *Inst. Electr. Eng. Proc.—D*, vol. 135, no. 2, pp. 1179–1187, 1988.
- [12] I. J. Wang and E. K. P. Chong, "A deterministic analysis of stochastic approximation with randomized directions," *IEEE Trans. Autom. Control*, vol. 43, no. 12, pp. 1745–1749, Dec. 1998.
- [13] M. G. Safanov, *Stability and Robustness of Multivariable Feedback Systems*. Cambridge, MA: MIT Press, 1980.
- [14] A. B. Trunov and M. M. Polycarpou, "Automated fault diagnosis in nonlinear multivariable systems using a learning methodology," *IEEE Trans. Neural Netw.*, vol. 11, no. 1, pp. 91–101, Jan. 2000.
- [15] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators*. New York: 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. Neural Netw.*, vol. 2, no. 6, pp. 1491–1497, Nov. 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. Syst. Man Cybern. B, Cybern.*, vol. 27, no. 3, pp. 369–375, Jun. 1997.

- [18] Q. Song, W. J. Hu, and T. N. Zhao, "Robust neural controller for variable airflow volume system," in *Inst. Electr. Eng. Proc.—Control Theory Appl.*, 2003, vol. 150, no. 2, pp. 112–118.
- [19] J. Xiao, "Robust neural tracking control systems," M.S. thesis, Schl. Electr. Electron. Eng., Nanyang Technol. Univ., Singapore, 1997.
- [20] J. Ni and Q. Song, "Adaptive simultaneous perturbation based pruning algorithm for a class of nonlinear neural control systems," *Neurocomputing*, vol. 69, no. 16–18, pp. 2097–2111, 2006.
- [21] Q. Song, "A robust information clustering algorithm," *Neural Comput.*, vol. 17, no. 12, pp. 2672–2698, 2005.
- [22] E. Kreyszig, *Advanced Engineering Mathematics*. New York: Wiley, 1993.
- [23] L. Behera, S. Kumar, and A. Patnaik, "On adaptive learning rate that guarantees convergence in feedforward networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1116–1125, Sep. 2006.
- [24] J. M. Bahii and S. Contassot-Vivier, "Basins of attraction in fully asynchronous discrete-time discrete-state dynamic networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 2, pp. 397–408, Mar. 2006.



Qing Song (M'92) received the B.S. and the M.S. degrees from Harbin Shipbuilding Engineering Institute and Dalian Maritime University, China, in 1982 and 1986, respectively, and the Ph.D. degree from the Industrial Control Centre, Strathclyde University, Glasgow, U.K., in 1992.

He was an Assistant Professor at the Dalian Maritime University from 1986 to 1987. He worked as a Research Engineer at the System and Control Pte, Ltd., U.K., before joining the School of Electrical and Electronic Engineering, Nanyang Technological University, as a Lecturer, in 1992, where he is currently an Associate Professor. He is the first inventor of a few neural network related patents and authorized more than 100 of referred international journal and conference publications. He held short visiting research positions at Johns Hopkins University and Tsinghua University, respectively. He is also an active industrial consultant and the principal investigator of research programs targeted to industry in the area of computational intelligence.



James C. Spall (S'82–M'83–SM'90–F'03) is a member of the Principal Professional Staff at the Applied Physics Laboratory and a Research Professor in the Department of Applied Mathematics and Statistics (main campus), the Johns Hopkins University, Laurel, MD. He is also Chairman of the Applied and Computational Mathematics Program within the Johns Hopkins University Engineering Programs for Professionals. He has published many articles in the areas of statistics and control and holds two U.S. patents for inventions in control systems.

He is the editor and coauthor for the book *Bayesian Analysis of Time Series and Dynamic Models* (New York: Marcel Dekker, 1988) and is the author of *Introduction to Stochastic Search and Optimization* (New York: Wiley, 2003).

Dr. Spall is an Associate Editor at Large for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL and a Contributing Editor for the *Current Index to Statistics*. He is the Program Chair for the 2007 IEEE Conference on Decision and Control. He has received numerous research and publications awards.



Yeng Chai Soh (M'89) received the B.Eng. degree in electrical and electronic engineering from the University of Canterbury, New Zealand, in 1983 and the Ph.D. degree in electrical engineering from the University of Newcastle, Australia, in 1987.

From 1986 to 1987, he was a Research Assistant at the Department of Electrical and Computer Engineering, University of Newcastle. In 1987, he joined the Nanyang Technological University, Singapore, where he is currently a Professor at the School of Electrical and Electronic Engineering. From 1995 to 2005, he was concurrently the Head of the Control and Instrumentation Division. He is now the Associate Dean (Research & Graduate Studies) of the College of Engineering. Since 2003, he has been the Director of the Centre for Modeling and Control of Complex Systems. He is an Honorary Professor of Northeastern University. He is serving in several project evaluation and review committees of national research and funding agencies. His current research interests are in the areas of robust system theory and applications, estimation and filtering, information and signal processing, hybrid and switched systems, and optical field control and signal processing. He has published more than 180 international journal articles in these areas.



Jie Ni (S'00) received the B.E. degree from Huazhong University of Science & Technology, Hubei, P. R. China, in 2002. Currently, he is working towards the Ph.D. degree at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, under Singapore government scholarship.

His research interests include artificial intelligent system, neural networks, adaptive control, robust control, machine learning, etc.