Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Journal of Sound and Vibration 296 (2006) 935–948

JOURNAL OF SOUND AND VIBRATION

www.elsevier.com/locate/jsvi

Adaptive recurrent fuzzy neural networks for active noise control

Qi-Zhi Zhang[a,*], Woon-Seng Gan[b], Ya-li Zhou[a]

[a]*Department of Computer Science and Automation, Beijing Institute of Machinery, P. O. Box 2865, Beijing 100085, People's Republic of China*
[b]*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore*

Received 26 February 2004; received in revised form 3 November 2005; accepted 17 March 2006
Available online 2 June 2006

Abstract

This paper discussed nonlinear active noise control (ANC). Some adaptive nonlinear noise control approaches using recurrent fuzzy neural networks (RFNNs) were derived. The proposed RFNNs were feed-forward fuzzy neural networks (NNs) with different local feedback connections that are used to construct dynamic fuzzy rules. Different recurrent connection strategies, diagonal recurrent and full connected recurrent ones, were considered. In addition, different fuzzy operation strategies, product (multiply) inference and "summation" (addition) inference, were proposed. Because RFNN-based ANC systems can capture the dynamic behavior of a system through the feedback links, the exact lag of the input variables need not be known in advance. Online dynamic back-propagation learning algorithms based on the error gradient descent method were proposed, and the local convergence of a closed-loop system was proven using the discrete Lyapunov function. A nonlinear simulation example showed that an adaptive ANC system based on an RFNN with summation inference is superior to a system based on other fuzzy NNs.

1. Introduction

Active noise control (ANC) has received much attention in recent years. In an ANC system, a secondary source is introduced to generate antinoise of equal amplitude but of opposite phase with reference to the primary noise. ANC techniques can be utilized to extract a signal buried in noise or to cancel unwanted noise. The filter-x least mean square (FXLMS) algorithm is a popular adaptive filtering algorithm using a finite impulse response (FIR) filters [1,2]. Its popularity is because of its simplicity, robustness, and relatively low computational load. With the development of digital signal processing (DSP) hardware, adaptive algorithms may be implemented in real-time practical systems [3]. Linear ANC systems have been successfully used to cancel both broadband noise and narrowband noise in air conditioning duct systems, handsets, and others [1–3]. However, in a practical ANC system, the secondary path and primary path of the ANC system may exhibit nonlinear behaviors. Thus, the development of nonlinear filters is necessary. Adaptive Volterra filters

[*]Corresponding author. Tel.: +86010 82426920.
E-mail address: zhang_qz@sohu.com (Q.-Z. Zhang).

0022-460X/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jsv.2006.03.020

have been introduced for active control of nonlinear noise processes [4]. The drawback is that the filter size increases exponentially when the inputs of the filter increases. Neural networks (NNs) have been introduced to control nonlinear noise, and multilayer perceptron networks are used to control nonlinear plants [5–8]. The multilayer perceptron network is a global approximate NN, and the major problem with an NN-based ANC is its relatively slow learning (or convergence) process. A fast NN learning algorithm has been proposed for ANC systems [8], but it is too complex to develop in practice. A local approximate NN, known as a radial basis function (RBF) network [9], can be introduced to improve convergence performance. Recently, a fuzzy modeling and controlling technique [10,11] has been used in a nonlinear filter. A fuzzy NN is a local approximate model, and the adaptive process can be accelerated. Gaussian functions are often chosen for local approximate models. The parameters of the network are separated into linear and nonlinear parameter sets [12]. Given the input data set, the nonlinear parameters, the mean and variance of the Gaussian functions can be estimated using a clustering algorithm. Only the linear parameters are updated online [11]. Both an acoustic model of the primary path and an inverse model of the secondary path have been identified using the Takagi–Sugeno (TS) model [10], and results have been obtained experimentally for a multichannel ANC system. Because an open-loop strategy is utilized in an ANC system, the performance of such a system will be degraded by disturbances and model mismatches. A feed-forward NN [5–11] is a static mapping. With tapped delays, a feed-forward NN can be used to represent a dynamic mapping, but a large number of neurons are required for representing dynamic responses in the time domain [13]. On the other hand, recurrent NNs (RNNs) may be used to deal with time-varying input or output through their natural temporal operation itself. Thus, an RNN is a dynamic mapping and is better suited for a dynamic system than a feed-forward NN [13,14]. A recurrent adaptive fuzzy filter has been proposed to resolve speech processing problems involving noise [15]. Good performances have been obtained, and the exact order of the inputs need not be known. In Ref. [15], because the secondary path is not considered, the noise-canceling process is a system identification problem and not a control problem. This paper focuses on the ANC problem involved in the nonlinear response of an unknown primary acoustic path. An example of a nonlinear response of the primary path is the nonlinear distortion that occurs when the primary noise propagating in a duct has a high sound pressure [4]. This paper is organized as follows. Section 2 describes a nonlinear ANC system and a general adaptive control algorithm. Section 3 describes the architecture of several recurrent fuzzy NNs (RFNNs) and their functions. Section 4 describes adaptive control algorithms using different RFNNs. In Section 5, the convergence of closed-loop systems is proven using the discrete Lyapunov function. In Section 6, proposed nonlinear ANC systems with different RFNNs are compared using computer simulations. The conclusions are given in Section 7.

2. System descriptions

An ANC system with a nonlinear primary noise path is shown in Fig. 1. The secondary path is modeled with an FIR filter. The ANC system can be described using the following equation [11]:

$$e(k) = d(k) + y(k) = g(X(k)) + \sum_{j=0}^{m} h(j)u(k-j), \tag{1}$$

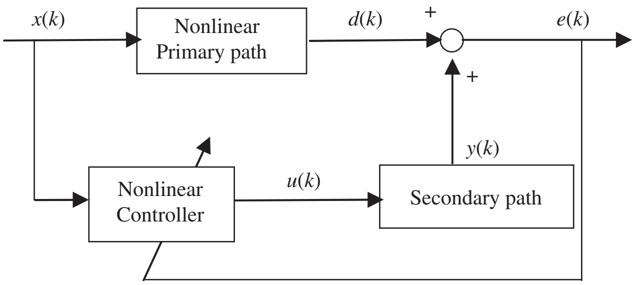

Fig. 1. Block diagram of a nonlinear control system.

where $X(k)=[x(k)x(k-1)\ldots x(k-n)]^{\mathrm{T}}$ is the reference signal vector, $u(k)$ is the output of the nonlinear controller, and $h(j)$ $(j=0,1\ldots,m)$ is the FIR filter coefficient of the secondary path model. d is the disturbance signal received at the error microphone, and g(.) is a smoothing nonlinear function. The output of the feed-forward nonlinear controller can be expressed as

$$u(k)=f(X(k),\ W), \tag{2}$$

where $f(.)$ is a smoothing nonlinear function and W is a parameter vector. W is the weights vector when an NN is used as the nonlinear controller. The performance index can be described as

$$J(k)=\frac{1}{2}\,e^2(k)=\frac{1}{2}[y(k)+d(k)]^2. \tag{3}$$

The unknown parameters can be adjusted according to the gradient descent method,

$$W(k+1)=W(k)-\mu\,\frac{\partial J(k)}{\partial W(k)}=W(k)-\mu e(k)\,\frac{\partial y(k)}{\partial W(k)}, \tag{4}$$

where μ is the learning rate. Applying the chain rule to Eq. (4),

$$\frac{\partial y(k)}{\partial W(k)}=\sum_{j=0}^{m}\frac{\partial y(k)}{\partial u(k-j)}\,\frac{\partial u(k-j)}{\partial W(k)}=\sum_{j=0}^{m}h(j)\,\frac{\partial u(k-j)}{\partial W(k)}. \tag{5}$$

If the parameters $W(k)$ are made to adapt slowly enough with time, the gradients of u in Eq. (5) can be approximately written as

$$\frac{\partial u(k-j)}{\partial W(k)}\approx\frac{\partial u(k-j)}{\partial W(k-j)}=\left.\frac{\partial f(X,W)}{\partial W}\right|_{X=X(k-j),W=W(k-j)}. \tag{6}$$

The parameters of the nonlinear controller can be adjusted online using the update rule (Eq. (4)), with the gradients calculated in Eqs. (5) and (6).

3. Structures of RFNNs

A general feed-forward nonlinear controller was proposed in Section 2. The nonlinear controller $f(X(k),W)$ can be approximated using an NN because of its universal approximation ability. The vector W represents the weights of the NN. Several NNs may be used, for example, multilayer perceptrons (MLPs), RBF networks, and FNNs. In this paper, RFNNs are used as nonlinear filters.

A diagonal RFNN (DRFNN) structure is shown in Fig. 2. The system has five layers as proposed in Ref. [13]. A model with two inputs and a single output is considered here for convenience. The nodes in Layer 1 are input nodes that directly transmit input signals to the next layer. Layer 5 is the output layer.

The nodes in Layer 2 are "term nodes" (G), and they act as membership functions expressing the input fuzzy linguistic variables. A Gaussian function is used for the membership function, in which the mean value, m, and the variance, σ, can be adjusted during the learning process. There are two advantages

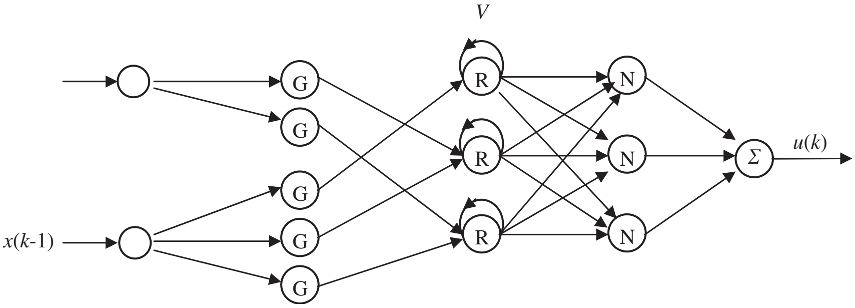

Fig. 2. Structure of five-layered DRFNN.

in employing a Gaussian membership function. The first is the continuity of the Gaussian function, which is usually required for most conventional gradient-based optimization techniques. The other advantage is that a multidimensional Gaussian membership function can be decomposed into multiple one-dimensional Gaussian membership functions for the corresponding number of input variables [12]. The two fuzzy sets of the first and the second input variables consist of $n_1$and n_2 linguistic terms, respectively. Each node in Layer 3 is called a "rule node" (R) and represents a single fuzzy rule. A diagonal feedback connection is introduced to give the feed-forward fuzzy NN a temporal processing capability. In total, there are $n_1 \times n_2$ nodes in Layer 3 forming a fuzzy rule base for two linguistic input variables. The nodes in Layer 4 (N) perform the normalization of firing strengths from Layer 3, and the input links are fully connected. The normalization of firing strengths is helpful in improving the convergence performance of the linear adaptive process. The number of nodes in this layer is equal to that of the nodes in Layer 3. In the following descriptions, the symbol $v_i^{(k)}$ denotes the ith input of a node in the kth layer, and the symbol $a^{(k)}$ denotes the output of a node in the kth layer. To provide a clear understanding of an RFNN, the functions of Layer 1 to Layer 5 are defined as follows:

Layer 1: The nodes in this layer only transmit input values to the nodes of the next layer directly,

$$a_i^{(1)}(k) = v_i^{(1)}(k). \tag{7}$$

Layer 2: The nodes in this layer represent Gaussian membership functions. The functions of the nodes are defined as

$$a_j^{(2)}(k) = \exp\left\{ -\frac{(v_i^{(2)}(k) - m_{ij})^2}{\sigma_{ij}^2} \right\}, \tag{8}$$

where m_{ij} and σ_{ij} are the mean and the width of the Gaussian membership function of the jth term of the ith input variable, $x(i)$, respectively.

Layer 3: The nodes in this layer are rule nodes, and a diagonal recurrent architecture is selected. The rule nodes perform a fuzzy AND operation (or product inference) to calculate the firing strength,

$$a_i^{(3)}(k) = \prod_j v_j^{(3)}(k) V_i a_i^{(3)}(k-1), \tag{9}$$

where V_i and $a_i(k-1)$ are the recurrent link weight and the output in the last steps of the ith node in Layer 3, respectively.

Layer 4: Nodes in Layer 4 perform the normalization of firing strengths from Layer 3,

$$a_i^{(4)}(k) = \frac{v_i^{(4)}(k)}{\sum_j v_j^{(4)}(k)}. \tag{10}$$

Layer 5: This layer is the output layer. The link weights in this layer represent the singleton constituents (W_i) of the output variable. The output node integrates all the normalized firing strengths from Layer 4 with the corresponding singleton constituents and acts as a defuzzifier,

$$u(k) = a^{(5)}(k) = \sum_i v_i^{(5)}(k) W_i. \tag{11}$$

Remark 1. The architecture of a DRFNN as shown in Fig. 2 possesses the advantage of a simple structure with dynamic characteristics. The purpose of the recurrent is to counter the past firing strength of its corresponding rule in Layer 3. Because the feedback terms contain the firing history of the rules, the recurrent fuzzy network has dynamic characteristics [13]. The fuzzy NN is a local approximate model, and the most firing strengths of the rules in Layer 3 are zeros (or near zeros) for arbitrary input. Thus, once an output of a node in Layer 3 is zero, it will be zero forever for the DRFNN. Therefore, a DRFNN is difficult to use in nonlinear ANC systems.

Several techniques can be used to improve the performance of the DRFNN. First, as in Ref. [15], a global membership function, $f(x) = 1/(1 + \mathrm{e}^{-x})$, can be used with the feedback term node. The structure of the improved DRFNN is shown in Fig. 3. For simplicity, only the rule layer is illustrated, and the other layers are the same as in Fig. 2. The firing strength of a rule term in Layer 3 can take a nonzero value, even if it is zero in the previous iteration. The function of Layer 3 can be defined as follows:

$$\begin{aligned} a_i^{(3)}(k) &= S_i(k) f(\text{net}_i(k)), \\ S_i(k) &= \prod_j v_j^{(3)}(k), \text{net}_i(k) = V_i a_i^{(3)}(k-1). \end{aligned} \tag{9a}$$

The fully connected recurrent NN has interlinked weights, and it can capture more complex dynamic systems. A fully connected RFNN is shown in Fig. 4. As in Fig. 3, only the rule layer is illustrated. Σ and z^{-1} denote summation and one-sample delay, respectively. The function of Layer 3 can be defined as follows:

$$\begin{aligned} a_i^{(3)}(k) &= S_i(k) f(\text{net}_i(k)), \\ S_i(k) &= \prod_j v_j^{(3)}(k), \text{net}_i(k) = \sum_j V_{ij} a_j^{(3)}(k-1). \end{aligned} \tag{9b}$$

Three feedback structures of the rule layer are shown in Figs. 2–4. We can also consider other fuzzy operations for calculating the firing strength. For example, one can replace the "product" (multiply) operation with the "summation" (addition) operation and obtain two new RFNNs with the operation "summation (addition)" (denoted "RFNNA"). In contrast, the RFNNs with the product operation are simply denoted by RFNNM. For the diagonal RFNNA (DRFNNA), the function of Layer 3 can be defined as in Fig. 4.

$$\begin{aligned} a_i^{(3)}(k) &= S_i(k) + f(\text{net}_i(k)), \\ S_i(k) &= \prod_j v_j^{(3)}(k), \text{net}_i(k) = V_i a_i^{(3)}(k-1). \end{aligned} \tag{9c}$$

For the fully connected RFNNA, the function of Layer 3 can be defined as follows:

$$\begin{aligned} a_i^{(3)}(k) &= S_i(k) + f(\text{net}_i(k)) \\ S_i(k) &= \prod_j v_j^{(3)}(k), \text{net}_i(k) = \sum_j V_{ij} a_j^{(3)}(k-1). \end{aligned} \tag{9d}$$

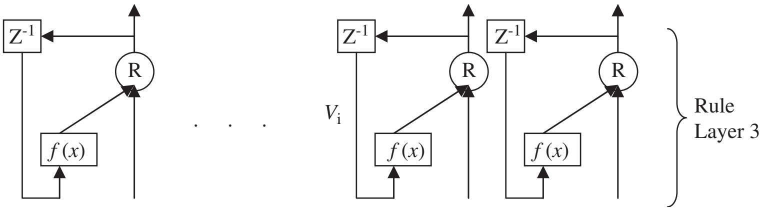

Fig. 3. Rule layer of the improved DRFNN.

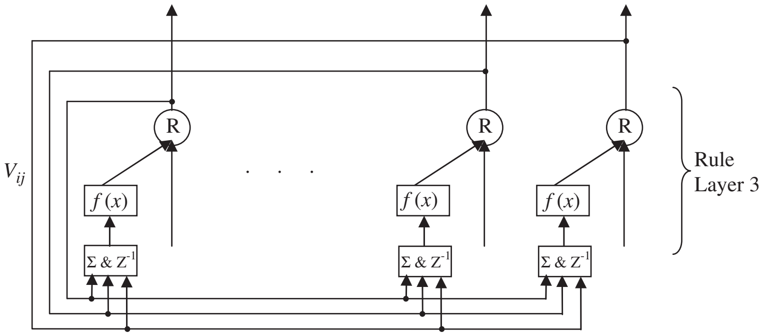

Fig. 4. Fully connected RFNN.

In the next section, we shall discuss the learning algorithm of RFNNs and use these NNs to control a nonlinear ANC system.

4. The adaptive control approach using RFNNS

Generally, the learning algorithm of an RFNN consists of two major components:

(1) Input/output space partitioning and construction of fuzzy rules.
(2) Identification of parameters.

In this paper, the input space is partitioned using *a priori* knowledge. The gradient descent method is used to adjust the parameters of an RFNN. Fig. 5 shows the block diagram of an RFNN-based ANC system. The RFNN controller is selected to replace the nonlinear controller (compared with Fig. 1). The RFNN controller is a nonlinear tap-delay filter, and the input of the RFNN is the reference $x(k)$, and its delays are $x(k-1)$, $x(k-2),\ldots,x(k-n)$.

It is presumed that the input space is partitioned by a *priori* knowledge. Only the singleton constituents of the output variable and the recurrent weights are adaptively adjusted when the ANC system is running. The rule of adaptive learning can be obtained using the gradient descent technique according to Eqs. (3)–(6). The gradients with respect to the weight vectors, W and V, can be computed using the chain rule as follows:

$$\frac{\partial J(k)}{\partial W} = e(k)\,\frac{\partial y(k)}{\partial W} = e(k)\sum_{j=0}^{m} h(j)\,\frac{\partial u(k-j)}{\partial W}, \tag{12}$$

$$\frac{\partial J(k)}{\partial V} = e(k)\,\frac{\partial y(k)}{\partial V} = e(k)\sum_{j=0}^{m} h(j)\,\frac{\partial u(k-j)}{\partial V}, \tag{13}$$

$$\frac{\partial u(k)}{\partial W_i} = v_i^{(5)}(k) = a_i^{(4)}(k). \tag{14}$$

For a DRFNN, the gradient can be computed as

$$\frac{\partial u(k)}{\partial V_i} = \sum_{j=1} W_j\,\frac{\partial v_j^{(5)}(k)}{\partial V_i} = \left\{[W_i - u(k)]/\sum_{j} a_j^{(3)}(k)\right\}\frac{\partial a_i^{(3)}(k)}{\partial V_i}. \tag{15a}$$

For an RFNN, the gradient can be computed as

$$\frac{\partial u(k)}{\partial V_{ij}} = \left\{[W_i - u(k)]/\sum_{l} a_l^{(3)}(k)\right\}\frac{\partial a_i^{(3)}(k)}{\partial V_{ij}}. \tag{15b}$$

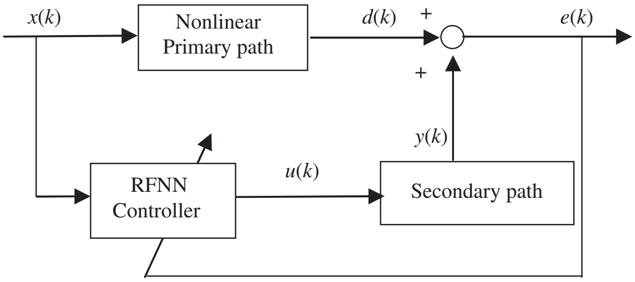

Fig. 5. Block diagram of an RFNN-based ANC system.

Because the diagonal recurrent NN whose firing strength is calculated using Eq. (9) cannot be used with the nonlinear ANC system, the gradient with respect to the recurrent weight is not given in this paper. For the DRFNNM, a diagonal recurrent NN for which the firing strength is calculated using Eq. (9a), the gradient with respect to recurrent link weight can be calculated as follows:

$$\frac{\partial a_i^{(3)}(k)}{\partial V_i} = S_i(k) f'(\mathrm{net}_i(k)) \left\{ a_i^{(3)}(k-1) + V_i \frac{\partial a_i^{(3)}(k-1)}{\partial V_i} \right\},$$
$$S_i(k) = \prod_j v_j^{(3)}(k), \quad \mathrm{net}_i(k) = V_i a_i^{(3)}(k-1). \tag{16a}$$

From Eq. (9b), the gradient with respect to the recurrent link weight of the RFNNM is found as

$$\frac{\partial a_i^{(3)}(k)}{\partial V_{ij}} = S_i(k) f'(\mathrm{net}_i(k)) \left\{ a_j^{(3)}(k-1) + V_{ii} \frac{\partial a_i^{(3)}(k-1)}{\partial V_{ij}} \right\},$$
$$S_i(k) = \prod_j v_j^{(3)}(k), \quad \mathrm{net}_i(k) = \sum_j V_{ij} a_j^{(3)}(k-1). \tag{16b}$$

From Eq. (9c), the gradient with respect to the recurrent link weight of the DRFNNA is

$$\frac{\partial a_i^{(3)}(k)}{\partial V_i} = f'(\mathrm{net}_i(k)) \left\{ a_i^{(3)}(k-1) + V_i \frac{\partial a_i^{(3)}(k-1)}{\partial V_i} \right\},$$
$$\mathrm{net}_i(k) = V_i a_i^{(3)}(k-1). \tag{16c}$$

Again, from Eq. (9d), the gradient with respect to the recurrent link weight of the RFNNA is found as follows:

$$\frac{\partial a_i^{(3)}(k)}{\partial V_{ij}} = f'(\mathrm{net}_i(k)) \left\{ a_j^{(3)}(k-1) + V_{ii} \frac{\partial a_i^{(3)}(k-1)}{\partial V_{ij}} \right\},$$
$$\mathrm{net}_i(k) = \sum_j V_{ij} a_j^{(3)}(k-1). \tag{16d}$$

From Eq. (16a) to Eq. (16d), one can find that the gradient with respect to the recurrent link weight is a dynamic equation. Using symbols similar to those in Ref. [14], the gradient for DRFNNM in Eqs. (15a) and (16a) is given by

$$\frac{\partial u(k)}{\partial V_i} = B_i(k) P_i(k), \tag{15a$'$}$$

where $B_i(k) = [W_i - u(k)]/\sum_j a_j^{(3)}(k)$ and $P_i(k) = \partial a_i^{(3)}(k)/\partial V_i$. $P_i(k)$ satisfies

$$P_i(k) = S_i(k) f'(\mathrm{net}_i(k)) \left\{ a_i^{(3)}(k-1) + V_i P_i(k-1) \right\}, P_i(0) = 0. \tag{16a$'$}$$

Eqs. (15a′) and (16a′), are dynamic recursive equations for the gradient, and it can be solved with given initial conditions recursively.

For the other RFNNs, the procedures of deriving the dynamic recursive equations are similar to the above derivation, and they are not given in this paper.

5. The convergence of the ANC system

An RFNN-based ANC system uses an error gradient descent algorithm to adjust the weight vector of the NN. As in Ref. [14], a discrete-type Lyapunov function can be given by

$$V(k) = \frac{1}{2} e^2(k). \tag{17}$$

Because of the training process, the change in the Lyapunov function can be obtained using

$$\Delta V(k) = V(k+1) - V(k) = \frac{1}{2}\left[e^2(k+1) - e^2(k)\right]. \tag{18}$$

The error difference resulting from the learning can be represented by

$$e(k+1) = e(k) + \Delta e(k) = e(k) + \left[\frac{\partial e(k)}{\partial W(k)}\right]^{\mathrm{T}} \Delta W(k). \tag{19}$$

According to the update rule of the weights, we can obtain

$$\Delta W(k) = -\mu e(k) \sum_{j=0}^{m} h(j) \frac{\partial u(k-j)}{\partial W(k)} = -\mu e(k) H A(k), \tag{20}$$

where $H = [h(0)h(1)\ldots h(m)]$ is the vector made up of the pulse responses of the secondary path and $A(k) = [\partial u(k)/\partial W \partial u(k-1)/\partial W \ldots \partial u(k-m)/\partial W]$ is the gradient matrix with respect to the general weight vector. A general convergence theorem can be presented as follows.

Theorem 1. *Let μ be the learning rate for the general weights of the DRFNN or the RFNN. We define $g_0 = ||H||$ and $g_{\max} = \max_k ||A(k)||$, where $\|\bullet\|$ is the usual Euclidean norm of a matrix or a vector. If the learning rate, μ, is chosen as $0 < \mu < 2/(g_0 g_{\max})^2$, then the local convergence of a closed-loop control system based on NNs is guaranteed.*

Proof. Define $Q(k) = HA(k)$; according to Eqs. (17)–(20), $\Delta V(k)$ can be represented as

$$\begin{aligned}\Delta V(k) &= \Delta e(k)[2e(k) + \Delta e(k)]/2 \\ &= -\frac{1}{2}\left[\frac{\partial e(k)}{\partial W(k)}\right]^{\mathrm{T}} \mu e(k) H A(k) \left\{ 2e(k) - \left[\frac{\partial e(k)}{\partial W(k)}\right]^{\mathrm{T}} \mu e(k) H A(k) \right\} \\ &= -\frac{1}{2}\mu e(k) ||Q(k)||^2 \{2e(k) - e(k)||Q(k)||^2\} = -\frac{1}{2}\mu e^2(k) \left\|Q(k)\right\|^2 \left\{2 - \mu \left\|Q(k)\right\|^2\right\} \\ &= -\frac{1}{2}\lambda e^2(k).\end{aligned}$$

Because $||Q(k)|| \leqslant ||H|| ||A(k)|| \leqslant g_0 g_{\max}$, if the learning rate, μ, is chosen as $0 < \mu < 2/(g_0 g_{\max})^2$, then $0 < \mu < 2/||Q(k)||^2$, which implies that $\lambda = \mu ||Q(k)||^2 \{2 - \mu ||Q(k)||^2\} > 0$ and $\Delta V(k) < 0$. Therefore, the control system is locally convergent.

The general convergence theorem can be used to find the specific convergence criterion for each type of weight of different RFNNs. □

Theorem 2. *Let μ_W and μ_V be the learning rates for the feed-forward weight vector and the recurrent weight vector of the DRFNNM, respectively. The dynamic back-propagation algorithm converges if the recurrent weights satisfy $|V_i| < 1$ and the learning rates are chosen as*

$$0 < \mu_W < 2/[(m+1)g_0^2], \tag{21a}$$

$$0 < \mu_V < S_{\min}^2 \big/ \left[2n_r(m+1)g_0^2 W_{\max}^2\right], \tag{21b}$$

where n_r is the number of rule nodes in Layer 3, $g_0 = ||H||$, and $W_{\max} = \max_k ||W(k)||$. $W(k)$ is the link weight vector between Layers 4 and 5, $S_{\min} = \min_k[\mathrm{Sum}(k)]$, and $\mathrm{Sum}(k) = \left|\sum_j v_j^{(4)}(k)\right|$.

Proof. (a) From Eq. (14),

$$A(k) = [A_4(k) A_4(k-1) \ldots A_4(k-m)], \tag{22}$$

where $A_4(k) = [a_1^{(4)}(k)a_2^{(4)}(k)\dots a_{n_r}^{(4)}(k)]^{\mathrm{T}}$ is the output vector of Layer 4. Because $a_i^{(4)}(k) \geqslant 0$ and $\sum_i a_i^{(4)}(k) = 1$, $||A_4(k)||^2 = \sum_i [a_i^{(4)}(k)]^2 \leqslant 1$ follows. From this we can obtain

$$||A(k)|| \leqslant \mathrm{sqrt}\left(\sum_{j=0}^{m} \left\|A_4(k-j)\right\|^2\right) \leqslant \sqrt{m+1}. \tag{23}$$

Hence, from Theorem 1 and Eq. (23), Eq. (21a) follows.

(b) From Eq. (16a′),

$$P_i(k) = S_i(k)f'(k)\{a_i^{(3)}(k-1) + V_i P_i(k-1)\},$$

where $f'(k) = f'(\mathrm{net}_i(k))$. From Eqs. (9a) and (8), we obtain $|S_i(k)| \leqslant 1$ and $|a_i^{(3)}(k)| = |S_i(k)||f(\mathrm{net}_i(k))| \leqslant 1$. Because $0 < f'(\mathrm{net}_i(k)) < 0.5$ and $|V_i| < 1$, the above equation can be estimated as follows:

$$|P_i(k)| \leqslant |S_i(k)|\left|f'(k)\right|\{|a_i^{(3)}(k-1)| + |V_i||P_i(k-1)|\} \leqslant 0.5 + 0.5|P_i(k-1)|. \tag{24}$$

Using Eq. (24) recurrently and consider the fact that $P_i(0) = 0$, it follows

$$|P_i(k)| \leqslant 0.5 + 0.5^2 + \cdots + 0.5^{k-1} + 0.5^k|P_i(0)| = \sum_{t=1}^{k-1} 0.5^t \leqslant \sum_{t=1}^{\infty} 0.5^t = 1, \tag{25}$$

denote $A(k) = [A_I(k)A_I(k-1)\dots A_I(k-m)]$, $A_I(k) = [A_1(k)A_2(k)\cdots A_{n_r}(k)]^{\mathrm{T}}$, and $A_i(k) = \partial u(k)/\partial V_i$. From Eq. (15a′), we obtain

$$|A_i(k)| = |B_i(k)||P_i(k)| \leqslant |[W_i - u(k)]|/\left|\sum\nolimits_j a_j^{(3)}(k)\right| \leqslant \{|W_i| + |u(k)|\}/S_{\min}. \tag{26}$$

From Eqs. (10) and (11) and the condition that $|W_i \leqslant W_{\max}$, we obtain

$$|u(k)| \leqslant \sum_i |v_i^{(5)}||W_i| \leqslant W_{\max}\sum_i |v_i^{(5)}| = W_{\max}\sum_i v_i^{(5)} = W_{\max}\sum_i v_i^{(4)}\Big/\sum_j v_j^{(4)} = W_{\max}. \tag{27}$$

Thus $|A_i(k)| \leqslant \{|W_i| + |u(k)|\}/S_{\min} \leqslant 2W_{\max}/S_{\min}$ and

$$|A_I(k)| \leqslant 2\sqrt{n_r}W_{\max}/S_{\min}. \tag{28}$$

Therefore

$$||A(k)|| \leqslant \mathrm{sqrt}\left(\sum_{j=0}^{m} ||A_I(k-j)||^2\right) \leqslant 2\sqrt{n_r(m+1)}W_{\max}/S_{\min}. \tag{29}$$

Hence, from Theorem 1 and Eq. (29), Eq. (21b) follows.

Remark 2. In Theorem 2, only the convergence of the DRFNN with the product (multiply) inference (DRFNNM)-based nonlinear ANC system is considered. The convergence of other RFNN-based nonlinear ANC systems can be obtained by following the same steps. For the fully connected RFNN shown in Fig. 4, the absolute value of the diagonal elements of the recurrent weight matrix must be selected as $|V_{ii}| < 1$ to ensure the convergence of an adaptive nonlinear ANC system.

6. Simulation results

Some illustrative results are presented to compare the performances of the different RFNN-based ANC systems. A personal computer with a Pentium 1.8 GHz processor and 256 MB of DRAM is used to implement the simulations. The sampling frequency used for this simulation is 1000 Hz. The disturbance signal chosen is a 100 Hz pure tone signal with additional Gaussian white noise signal. There is only one input node and one output node in all the RFNNs used. Given the input data set, the mean and variance of the Gaussian functions can be estimated using a clustering algorithm. Because only one input node is selected in the simulation, the input space is uniformly partitioned to eight fuzzy sets, and the means and widths of the Gaussian membership functions are selected as [11]:

$$m = [-0.65, -5/8, -3/8, -1/8, 1/8, 3/8, 5/8, 0.65],$$
$$\sigma = [-20, 0.14, 0.14, 0.14, 0.14, 0.14, 0.14, 20].$$

Only the weight vectors W and V are adjusted online, and the means and widths of the Gaussian membership functions remain fixed when the ANC system is running. In the simulation, only the nonlinearity in the primary path is considered.

Case 1: The acoustic model is selected as in Ref. [4]. There is one sample delay in the secondary path, and it has a root outside the unit circle. The secondary path simulated is a nonminimum phase system.

The primary acoustic path is selected as

$$d(k) = x(k-3) - 0.3x(k-4) + 0.2x(k-5) + 0.8x^2(k-4).$$

The secondary acoustic path from the secondary source to the error microphone is

$$y(k) = u(k-1) + 1.5u(k-2) - u(k-3).$$

Simulations are carried out for the RFNNs, whose firing strengths are calculated using "multiply". Both diagonal recurrent and fully recurrent NNs are selected as single-input and single-output networks. The variance of the Gaussian white noise signal is 0.04.

Fig. 6 shows the mean square error (MSE) in the error microphone versus the number of iterations. The results of the DRFNNM-based and RFNNM-based ANC systems are shown as thin and thick lines, respectively. The MSE of the RFNNM control system is approximately 4 dB below that of the DRFNNM control system.

Fig. 7 shows the results of the simulation of the canceling errors between the 3000th and 5000th iterations in the frequency domain. Similarly, the results for the DRFNNM and RFNNM are shown by the thin and thick lines, respectively. The dashed line shows the sound pressure level of the disturbance signal when the ANC system is turned off. There is a 200 Hz peak in the noise spectrum, which is produced by the nonlinear square term in the primary acoustic path. The DRFNNM-based ANC system can reduce the 200 Hz noise by 8 dB, but it cannot reduce the 100 Hz noise. The RFNNM-based ANC system can reduce the 200 Hz noise by 18 dB

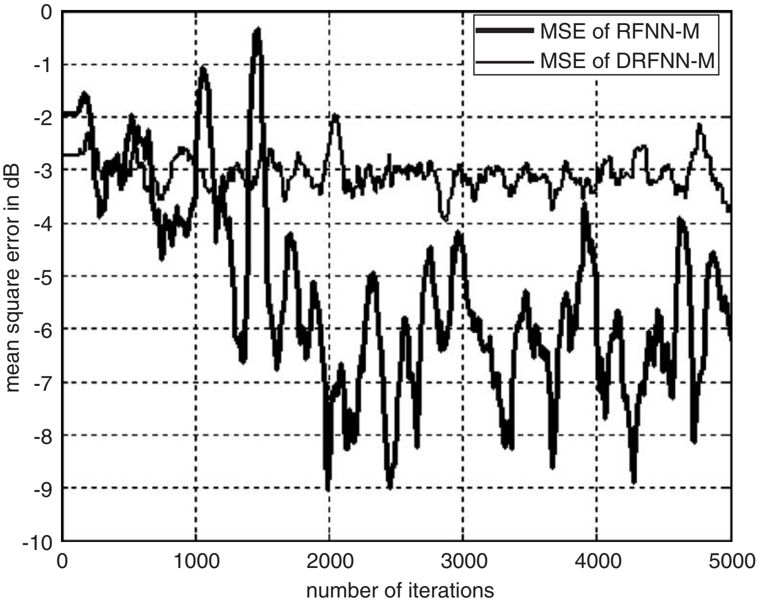

Fig. 6. Performance comparison: DRFNNM controller (thin line); and RFNNM controller (thick line).

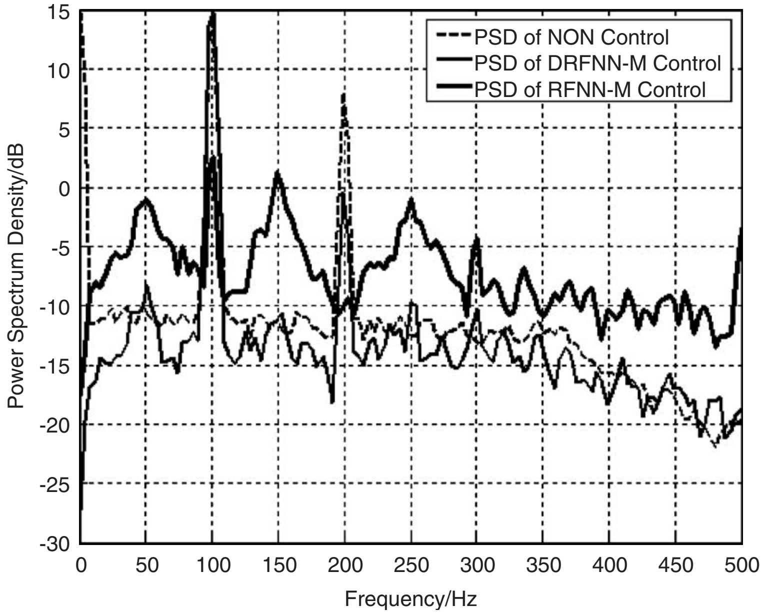

Fig. 7. Power spectrum of canceling errors. The thin line is for DRFNNM controller, the thick line is for the RFNNM controller, and the dashed line represents the ANC turning off.

and can also reduce the 100 Hz noise by 12 dB. However, the broadband noise level of other frequencies has been increased.

Case 2: Simulations are presented for the RFNNs whose firing strength is calculated by "summation." Both diagonal recurrent and fully recurrent NNs are selected as single-input and single-output networks. The acoustic model is similar to that of Case 1, and the variance of the Gaussian white noise signal is set to 0.04.

Fig. 8 shows the MSE at the error microphone versus the number of iterations. The results of the DRFNNA-based and RFNNA-based ANC systems are shown as the solid thin line and the thick line, respectively. The MSE of the RFNNA control system is approximately 20 dB below that of the DRFNNM control system. A static fuzzy NN (SFNN)-based ANC system has also been tested [11]. For comparison, two input nodes, which are obtained by delaying the outputs of Layer 2 in the SFNN, are used in the simulation such that the number of parameters in the SFNN is close to the number of parameters in the RFNNA. The results are shown as the dashed line. Its performance is similar to the performance of the RFNNA control system, but the SFNN-based ANC system takes more CPU times than does the RFNNA-based ANC system (for example, a Matlab simulation took 13.9400 s for the RFNNA system and 20.3590 s for the SFNN system).

Fig. 9 shows simulation results of the canceling errors between the 3000th and 5000th iterations in the frequency domain. The result for the DRFNNA system is shown as the solid thin line, the result of the RFNNA system is shown as the solid thick line, and the dashed line shows the sound pressure level of the disturbance signal when the ANC system is turned off. The DRFNNA-based ANC system can reduce the 200 Hz noise by 8 dB, but it increases the 100 Hz noise by 10 dB. The RFNNA-based ANC system can reduce the 200 Hz noise by 18 dB and can also reduce the 100 Hz noise by 20 dB. The broadband noise level of other frequencies has not been increased.

Case 3: A more realistic acoustic model is selected as in Ref. [16]. There is a five-sample delay in the secondary path, and has a root outside the unit circle. The secondary path simulated is a nonminimum phase system.

The primary acoustic path is selected as

$$\begin{aligned} d(k) = {} & 0.8x(k-9) + 0.6x(k-10) - 0.2x(k-11) - 0.5x(k-12) - 0.1x(k-13) \\ & + 0.4x(k-14) - 0.05x(k-15) + 0.8x^2(k-9). \end{aligned}$$

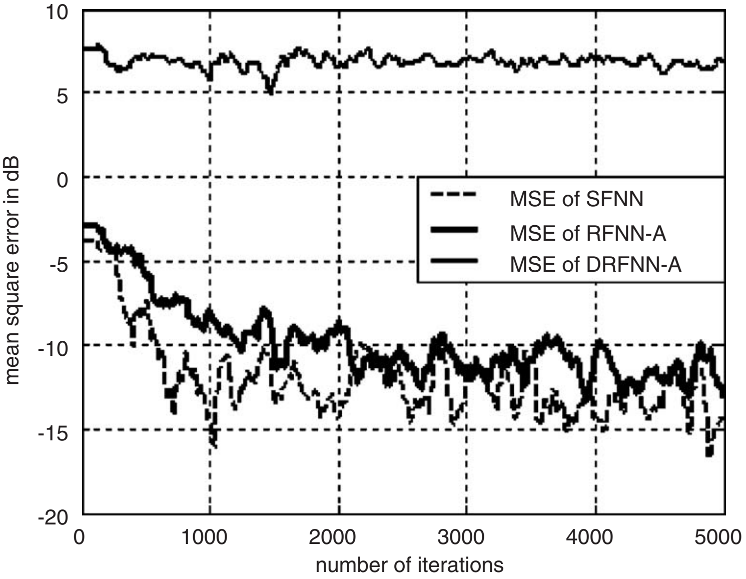

Fig. 8. Performance comparison, the thin line for DRFNNA controller, the thick line for RFNNA controller, and the dashed line for SFNN controller.

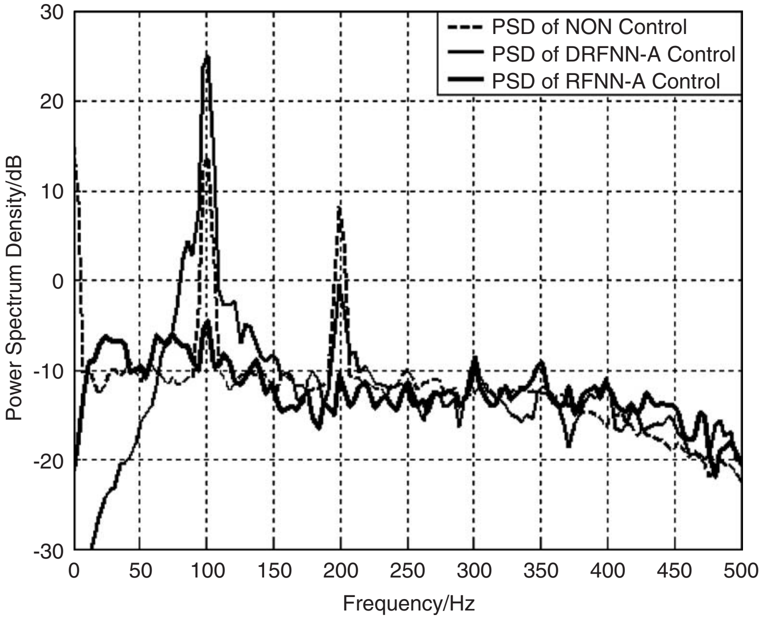

Fig. 9. Power spectrum of canceling errors: thin line, DRFNNA controller; thick line, RFNNA controller; and dished line, ANC turning off.

The secondary acoustic path from the secondary source to the error microphone is

$$\begin{aligned} y(k) = {} & u(k-5) + 2.5u(k-6) + 1.76u(k-7) + 0.15u(k-8) \\ & - 0.4825u(k-9) - 0.18625u(k-10) - 0.005u(k-11) \\ & - 0.001875u(k-12). \end{aligned}$$

Simulations are carried out for the RFNNs, whose firing strengths are calculated by ‘‘summation’’. The variance of the Gaussian white noise signal is set as 1.

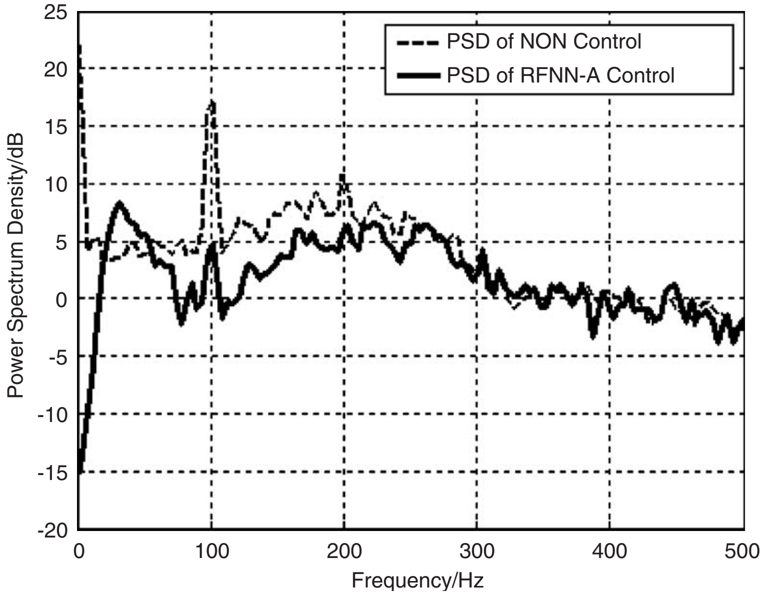

Fig. 10. Power spectrum of active noise canceling errors: solid line, RFNNA controller; and dashed line, ANC turning off.

Fig. 10 gives the simulation results of the canceling errors between the 3000th and 5000th iterations in the frequency domain. Similarly, the results of the RFNNA are shown as the solid line. The dashed line shows the sound pressure level of the disturbance signal when the ANC system is turned off. There is a 200 Hz peak in the noise spectrum, which is produced by the nonlinear square term in the primary acoustic path. The RFNNA-based ANC system can reduce the 100 Hz noise by 12 dB and reduce the 200 Hz noise by 5 dB. The broadband noise between 100 and 200 Hz is reduced by about 5 dB.

From the simulation results, one can find that the convergence of the RFNNA control is superior to that of the other recurrent NN control.

Using the original DRFNN, we will give a brief analysis to explain why the RFNN with the "summation" operation is better than the RFNN with the "product" operation. The output in Layer 3 with the "product" operation (Eq. (9)) is as follows:

$$a_i^{(3)}(k) = S(k)V_i a_i^{(3)}(k-1) = S(k)S(k-1)\ldots S(0)V_i^k. \tag{30}$$

The output in Layer 3 with the "summation" operation is

$$a_i^{(3)}(k) = S(k) + V_i a_i^{(3)}(k-1) = S(k) + V_i S(k-1) + \cdots + V_i^k S(0). \tag{31}$$

According to Theorem 2 the recurrent weights satisfy $|V_i < 1$. If k is a large number, the output in Eq. (30) will tend to zero. But Eq. (31) gives a better memory structure; the weight for the past output decreases with memory lengths.

7. Conclusions

Nonlinear ANC was studied. In particular, some RFNNs were proposed for solving the nonlinear effect in the primary acoustic path of an ANC system. Using RFNNs, only one input is required, and fewer parameters are used. The convergence rate of an RFNN is faster than that of an SFNN of the same size. An online back-propagation learning algorithm based on the error gradient descent method is proposed, and the local convergence of the closed-loop system is proven using the discrete Lyapunov function. Some simulation results were given to compare different RFNN-based methods. The results showed that the adaptive ANC method based on an RFNN with "summation" (addition) inference (RFNNA) is very effective in nonlinear noise control, and the convergence of the RFNNA control is superior to that of the other fuzzy NN controls.

This is because of the fact that a diagonal recurrent NN only has self-feedback links and, therefore, cannot approximate a complex dynamic system. The direct self-feedback link in Layer 3 results in all outputs of this layer being zeros, and the improvements proposed in this paper were necessary to overcome this drawback. The simulations assumed that the secondary path, $H(z)$, is estimated off-line prior to the operation of the ANC system. Online modeling of the secondary path can also be carried out as described in Refs. [2,16]. The proposed algorithms can also be applied to the problem of nonlinear secondary paths as described in Refs. [6–8]. Our current work is focused on designing an ANC system based on RFNNs to meet the requirements of practical applications.

Acknowledgments

This research is supported by Science Foundations of Educational Committee of Beijing City (KM200311232137, KM200511232008) and Training Foundations for Elitist of Beijing (20051A0500603).

References

[1] S.J. Elliott, P.A. Nelson, Active noise control, *IEEE Signal Process Magazine* 10 (1993) 12–35.

[2] S.M. Kuo, D.R. Morgan, Active noise control: a tutorial review, *IEEE Proceedings* 87 (1999) 973–993.

[3] S.M. Kuo, W.S. Gan, *Digital Signal Processors: Architecture, Implementation, and Applications*, Prentice-Hall, Englewood Cliffs, 2004.

[4] L. Tan, J. Jiang, Adaptive volterra filters for active control of nonlinear noise processes, *IEEE Transactions on Signal Processing* 49 (2001) 1667–1676.

[5] L. Yin, J. Astola, Y. Neuvo, A new class of nonlinear filters-Neural filters, *IEEE Transactions on Signal Processing* 41 (1993) 1201–1222.

[6] M. Bouchard, B. Pailard, C.T.L. Dinh, Improved training of neural networks for the nonlinear active control of sound and vibration, *IEEE Transactions on Neural Networks* 10 (1999) 391–401.

[7] M. Bouchard, New recursive-least-squares algorithms for non-linear active control of sound and vibrations using neural networks, *IEEE Transactions on Neural Networks* 12 (2001) 135–147.

[8] Q.Z. Zhang, Y.L. Jia, Active noise hybrid feedforward /feedback control using neural network compensation, *ASME Journal of Vibration and Acoustics* 124 (2002) 100–104.

[9] P. Strauch, B. Mulgrew, Active control of nonlinear noise processes in a linear duct, *IEEE Transactions on Signal Processing* 46 (1998) 2404–2412.

[10] J.M. Sousa, C.A. Silva, J.M.G. Costa, Fuzzy active noise modeling and control, *International Journal of Approximate Reasoning* 33 (2003) 51–70.

[11] Q.Z. Zhang, W.S. Gan, Active noise control using a simplified fuzzy neural network, *Journal of Sound and Vibration* 272 (2004) 437–449.

[12] J.S. Wang, C.S. Lee, Self-adaptive neuro-fuzzy inference systems for classification applications, *IEEE Transactions on Fuzzy Systems* 10 (2002) 790–802.

[13] F.J. Lin, R.J. Wai, Hybrid control using recurrent fuzzy neural network for linear-induction motor servo drive, *IEEE Transactions on Fuzzy Systems* 9 (2001) 102–115.

[14] C.C. Ku, K.Y. Lee, Diagonal recurrent neural networks for dynamic systems control, *IEEE Transactions on Neural Networks* 6 (1995) 144–156.

[15] C.F. Juang, C.T. Lin, Noisy speech processing by recurrently adaptive fuzzy filters, *IEEE Transactions on Fuzzy Systems* 9 (2001) 139–152.

[16] Q.Z. Zhang, W.S. Gan, A model predictive algorithm for active noise control with online secondary path modelling, *Journal of Sound and Vibration* 270 (2004) 1056–1066.