Digital Signal Processing 93 (2019) 160–171

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Hierarchical partial update generalized functional link artificial neural network filter for nonlinear active noise control

Dinh Cong Le [a,b], Jiashu Zhang [a,*], Defang Li [a]

[a] Sichuan Province Key Lab of Signal and Information Processing, Southwest Jiaotong University, Chengdu 610031, PR China
[b] Institute of Engineering and Technology, Vinh University, Viet Nam

ARTICLE INFO

Article history:
Available online 23 July 2019

Keywords:
Nonlinear adaptive filter
Active noise control
Generalized FLANN
Hierarchical partial update
Pipelined architecture

ABSTRACT

To reduce the computational burden of the generalized FLANN (GFLANN) filter for nonlinear active noise control (NANC), a hierarchical partial update GFLANN (HPU-GFLANN) filter is presented in this paper. Based on the principle of divide and conquer, the proposed HPU-GFLANN divides the complex GFLANN filter (i.e., long memory length and large cross-terms selection parameter) into simple small-scale GFLANN modules and then interconnected in a pipelined form. Since those modules are simultaneously performed in a parallelism fashion, there is a significant improvement in computational efficiency. Besides, a hierarchical learning strategy is used to avoid the coupling effect between the nonlinear and linear part of the pipelined architecture. Data-dependent hierarchical M-Max filtered-error LMS algorithm is derived to selectively update coefficients of the HPU-GFLANN filter, which can further reduce the computational complexity. Moreover, the convergence analysis of the NANC system indicates that the proposed algorithm is stable. Computer simulation results verify that the proposed adaptive HPU-GFLANN filter is more effective in nonlinear ANC systems than the FLANN and GFLANN filters.

1. Introduction

It is well known that the adaptive finite impulse response (FIR) filter has been widely applied in the linear ANC system due to its simple structure and low computational complexity [1,2]. But since the actual ANC system may contain some nonlinear distortions and the reference noise may also be a nonlinear or deterministic noise process, the performance of the linear ANC system is degraded, even failed [3,4]. Thus the nonlinearity must be considered in the design of the controllers for the NANC system. In the past decades, various nonlinear adaptive filters for the NANC system have been proposed, including the truncated Volterra filter (VF) [5,6], multi-layer perceptron (MLP) [7,8], recurrent neural networks [9,10], fuzzy neural network [11], spline filter [12], functional link artificial neural networks (FLANN) [13–15], just to mention a few. Especially, the FLANN has received much attention in the literature on ANC due to its simple structure for practical implementation.

In order to improve the noise-canceling performance, various FLANN structures for the NANC system have been proposed in recent years. Two structures based on feedback FLANN (FFLANN) and recursive FLANN with a bounded-input bounded-output (BIBO) stability condition have been introduced in [16,17]. Based on the exploitation of both feedback and feedforward polynomial, a bilinear FLANN (BFLANN) filter was proposed in [18]. To further improve the performance of the pure sinusoids-based FLANN filter, the nonlinear filters using FLANN with exponentially varying amplitudes have been presented in [19,20]. In addition, many efficient structures based on the convex/cascade combinations of the nonlinear adaptive FLANN filter have been proposed [21,22]. By introducing cross-terms in a traditional FLANN structure and arranging them in a similar way to the diagonal representation of the Volterra filter, a generalized FLANN (GFLANN) has been developed in [23]. It is shown that NANC applications with the presence of strong nonlinearity, the GFLANN filter can achieve better performance than high-order volterra filters. Using long memory length and large enough cross-terms, however, in order to achieve a good noise cancellation may increase its computational complexity and negatively influence on the use of system resources.

According to the study in [24], to reduce the computational burden of the recurrent neural networks (RNN), a computationally efficient pipelined RNN (PRNN) structure was developed. The idea of the design is based on the engineering principle of divide and conquer and the biological principle of modules. Following this work, a variety of nonlinear filters based on pipelined structures have been proposed to reduce the computational complexity, such as the pipelined feedforward second-order Volterra

* Corresponding author.
E-mail address: jszhang@home.swjtu.edu.cn (J. Zhang).

https://doi.org/10.1016/j.dsp.2019.07.006
1051-2004/

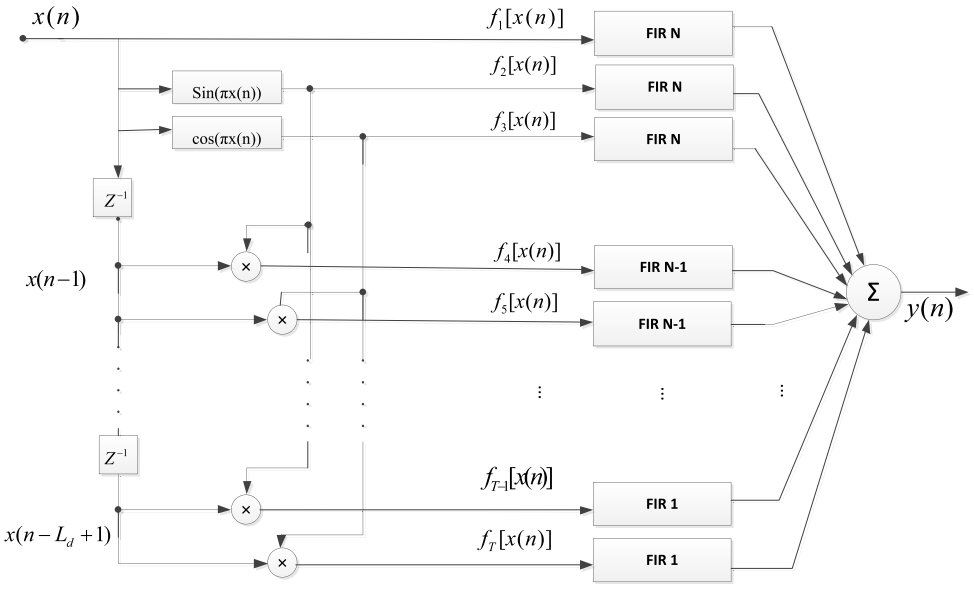

Fig. 1. Implementation of GFLANN structure.

[25], adaptive bilinear filter based on pipelined architecture [26], Pipelined second-order Volterra filter against impulsive noise [27, 28], pipelined recurrent fuzzy neural networks filter [29], hierarchical pipelined alternative update adaptive Volterra filter (HPAVF) [30], and so on. In the HPAVF architecture, the authors proposed a hierarchical learning strategy to eliminate the coupling effect between linear and nonlinear parts of the pipelined architecture.

Inspired by the HPAVF, a hierarchical partial update GFLANN (HPU-GFLANN) filter for NANC is proposed to reduce the computational complexity of the GFLANN in this paper. The proposed HPU-GFLANN filter inherits many advantages of pipelined architecture, and uses a computationally efficient hierarchical M-max filtered-error least mean square (HMmFE-LMS) algorithm to partially update the weights of the HPU-GFLANN. In order to evaluate the performance as well as the computational complexity of the proposed HPU-GFLANN filter in NANC applications, a series of the computer simulations are conducted.

The rest of this paper is organized as follows. Section 2 and 3 present the brief of GFLANN structure and hierarchical update GFLANN architecture, respectively. In section 4, the HMmFE-LMS algorithm is designed. Sections 5 and 6 present the analysis of computational complexity and convergence condition, respectively. The effectiveness of the proposed nonlinear filter is illustrated by comparing with GFLANN and FLANN filters for nonlinear active noise control system in Section 7. Finally, Section 8 is devoted to a brief summary and discussion

2. Brief of GFLANN structure

From the analysis of the Taylor series expansion, a traditional FLANN can only directly model the power and sinusoidal nonlinearity. Therefore, with the aim of improving nonlinear modeling capability of FLANN structure-based filter, G.L. Sicuranza et al. proposed a generalized FLANN (GFLANN) structure, which achieved by introducing cross-terms in a FLANN structure and exploiting them in a suitable strategy [23]. Research also has shown that in some applications of NANC, the GFLANN filter can offer better performance than the FLANN and even as good as the high-order Volterra filters with a reduced complexity. Fig. 1 illustrates the implementation of the GFLANN structure.

Let $X(n)$ denotes the L input signal vector of the filter

$$X(n) = \left[x(n), x(n-1), ..., x(n-L+1)\right]^T \tag{1}$$

where L is the number of input samples at each time step n. Consequently, the expanded input signal vector of GFLANN is given by

$$\begin{aligned} Xg(n) = \big[&x(n)x(n-1)\dots x(n-L+1) \\ &\sin\big(\pi x(n)\big)\sin\big(\pi x(n-1)\big)\dots\sin\big(\pi x(n-L+1)\big) \\ &\cos\big(\pi x(n)\big)\cos\big(\pi x(n-1)\big)\dots\cos\big(\pi x(n-L+1)\big) \\ &x(n-1)\sin\big(\pi x(n)\big)x(n-2)\sin\big(\pi x(n-1)\big)\dots \\ &x(n-L+1)\sin\big(\pi x(n-L+2)\big) \\ &x(n-1)\cos\big(\pi x(n)\big)x(n-2)\cos\big(\pi x(n-1)\big)\dots \\ &x(n-L+1)\cos\big(\pi x(n-L+2)\big) \\ &x(n-L+1)\sin\big(\pi x(n)\big)x(n-L+1)\cos\big(\pi x(n)\big)\big]^T \end{aligned} \tag{2}$$

Note that in this method to reduce computational complexity, the GFLANN only exploits the cross-terms in a similar way to the diagonal representation of the Volterra filter. As shown clearly in Fig. 1, the FIR filters for the input signal $x(n)$ and $\cos(\pi x(n))$, $\sin(\pi x(n))$ expansions have lengths of L samples; the cross-terms generated by products $x(n-1)\cos(\pi x(n))$, $x(n-1)\sin(\pi x(n))$ have lengths of $L-1$ samples; and so on until the last cross-terms $x(n-L+1)\cos(\pi x(n))$, $x(n-L+1)\sin(\pi x(n))$ have lengths of one sample.

Thus, the expanded $X_g(n)$ signal can be represented as follows

$$Xg(n) = \left[Xg_1^T(n)Xg_2^T(n)\dots Xg_T^T(n)\right]^T \tag{3}$$

where $Xg_r(n)$ denotes sub-vector of $Xg(n)$, $r = 1, 2, \dots T$, and its elements are

$$Xg_r(n) = \left[f_r\big[x_r(n)\big]f_r\big[x_r(n-1)\dots f_r\big[x_r(n-L+1)\big]\right]^T \tag{4}$$

where the nonlinear expansions of $f_r[x_r(n)]$ are defined as

$$
\begin{aligned}
&f_1[x_1(n)] = x(n)\\
&f_2[x_2(n)] = \cos(\pi x(n))\\
&f_3[x_3(n)] = \sin(\pi x(n))\\
&f_4[x_4(n)] = x(n-1)\cos(\pi x(n))\\
&f_5[x_5(n)] = x(n-1)\sin(\pi x(n))\\
&\qquad\vdots\\
&f_{T-1}\big[x_{T-1}(n)\big] = x(n-L_d+1)\cos\big(\pi x(n)\big)\\
&f_T\big[x_T(n)\big] = x(n-L_d+1)\sin\big(\pi x(n)\big)
\end{aligned}
\tag{5}
$$

Note that we can also adjust the compromise between the complexity and performance of this filter based on the cross-terms selection parameter L_d to limit the number of active channels of Fig. 1. In particular, $1 \le L_d \le L-1$ and $T = 3+2L_d$ is the channel number of the GFLANN.

3. Hierarchical update generalized FLANN (HU-GFLANN) architecture

By combining the efficient pipelined architecture and hierarchical learning strategy, the HU-GFLANN architecture is proposed as illustrated in Fig. 2. The design of the HU-GFLANN architecture consists of two subsections. The nonlinear subsection comprises a number of simple small-scale GFLANN modules (i.e., the short memory length and small cross-term selection parameter) which are interconnected in a pipelined fashion. It performs a nonlinear mapping from input space to an intermediate space. The linear subsection is a conventional transversal filter that performs a linear mapping from the intermediate space to the output space. The output of the HU-GFLANN is a linear combination of these two subsections.

3.1. Nonlinear subsection

The nonlinear subsection of the HU-GFLANN filter consists of the M identical small-scale GFLANN modules, i.e. all of them have exactly the same number of external inputs and cross-term elements. Inputs of each module are composed of L delayed external input signals and one delayed signal of previous module output. In the case of module M, the output of the previous module is nonexistent and it is replaced by its own feedback signal. In this study, the weight vector of each module is different, i.e., each module updates its weight independently.

Let $X_i(n)$ denotes the $L+1$ input signal vector of the i-th module and is defined by

$$
\begin{aligned}
X_i(n) &= \big[\hat{X}_i^T(n), U_i(n)\big]^T\\
&= \big[x(n-i), x(n-i-1), ..., x(n-i-L+1), U_i(n)\big]^T,\\
&\quad i = 1, 2..., M
\end{aligned}
\tag{6}
$$

where $\hat{X}_i(n) = [x(n-i), x(n-i-1), ..., x(n-i-L+1)]^T$ is the external input vector at the n-th time point. This external input vector is delayed by $Z^{-i}I$ at the input of the i-th module, where Z^{-i} is the delay operator and I denotes the $(L \times L)$-dimensional identity matrix. $U_i(n)$ is other input signal vector of the i-th module. $U_i(n) = y_{i+1}(n)$ when $1 \le i < M$, which is the output of adjacent module $i+1$; $U_i(n) = y_M(n-1)$, when $i = M$, which is the output of module M after a delay of a one-time unit.

As mentioned above, each module is a small-scale GFLANN. Hence, at the n-th time point, input signal vector $X_i(n)$ is expanded to $Xg_i(n)$ by the GFLANN series and is expressed as

$$
Xg_i(n) = \big[Xg_{1,i}^T(n), Xg_{2,i}^T(n), \cdots, Xg_{T,i}^T(n)\big]^T \tag{7}
$$

where $T = (3+2(L_d+1))$ is considered as the number of channels in the implementation of the GFLANN filter; $1 \le L_d \le L-1$.

More specifically, the subvectors $Xg_{1,i}(n)$, $Xg_{2,i}(n), \ldots, Xg_{T,i}(n)$ are explicitly given by

$$
Xg_{1,i}(n) = \big[x(n-i), x(n-i-1), ...x(n-i-L+1), U_i(n)\big]^T \tag{8}
$$

$$
\begin{aligned}
Xg_{2,i}(n) = \big[&\sin\big(\pi x(n-i)\big), \sin\big(\pi x(n-i-1)\big), ...\\
&\sin\big(\pi x(n-i-L+1)\big), sin\big(\pi U_i(n)\big)\big]^T
\end{aligned}
\tag{9}
$$

$$
\begin{aligned}
Xg_{3,i}(n) = \big[&\cos\big(\pi x(n-i)\big), \cos\big(\pi x(n-i-1)\big), ...\\
&\cos\big(\pi x(n-i-L+1)\big), \cos\big(\pi U_i(n)\big)\big]^T
\end{aligned}
\tag{10}
$$

$$
\begin{aligned}
Xg_{4,i}(n) = \big[&x(n-i-1)\sin\big(\pi x(n-i)\big), x(n-i-2)\\
&\times \sin\big(\pi x(n-i-1)\big), ..., U_i(n)\\
&\times \sin\big(\pi x(n-i-L_d+1)\big)\big]^T
\end{aligned}
\tag{11}
$$

$$
\begin{aligned}
Xg_{5,i}(n) = \big[&x(n-i-1)\cos\big(\pi x(n-i)\big), x(n-i-2)\\
&\times \cos\big(\pi x(n-i-1)\big), ..., U_i(n)\\
&\times \cos\big(\pi x(n-i-L_d+1)\big)\big]^T
\end{aligned}
\tag{12}
$$

$$
\vdots \qquad = \qquad \vdots
$$

$$
\begin{aligned}
Xg_{(T-3),i}(n) = \big[&x(n-i-L_d+1)\sin\big(\pi x(n-i)\big),\\
&U_i(n)\sin\big(\pi x(n-i-1)\big)\big]^T
\end{aligned}
\tag{13}
$$

$$
\begin{aligned}
Xg_{(T-2),i}(n) = \big[&x(n-i-L_d+1)\cos\big(\pi x(n-i)\big),\\
&U_i(n)\cos\big(\pi x(n-i-1)\big)\big]^T
\end{aligned}
\tag{14}
$$

$$
Xg_{(T-1),i}(n) = \big[U_i(n)\sin\big(\pi x(n-i)\big)\big]^T \tag{15}
$$

$$
Xg_{T,i}(n) = \big[U_i(n)\cos\big(\pi x(n-i)\big)\big]^T \tag{16}
$$

where the length L_h of $Xg_i(n)$ is defined by $L_h = 3(L+1) + (L_d+1)(L_d+2)$.

As mentioned above, the weight vector $H_i(n)$ of each module is different $(H_1(n) \ne \ldots H_i(n) \ne \ldots H_M(n))$, which is defined by

$$
H_i(n) = \big[h_{1,i}(n), h_{2,i}(n), ..., h_{L_h,i}(n)\big]^T \tag{17}
$$

Hence, the output $y_i(n)$ of the i-th module is written by

$$
y_i(n) = H_i^T(n) Xg_i(n) \tag{18}
$$

3.2. Linear section

As illustrated in Fig. 2, each module in the nonlinear subsection provides a local interpolation for M time-series points. Then, thanks to a conventional transversal filter we can obtain global interpolation with good localization properties. Here the weight vector of the linear transversal filter is defined as

$$
W(n) = \big[w_1(n), w_2(n), ..., w_M(n)\big]^T \tag{19}
$$

where the weights of the linear subsection are equal to the number of designed modules. The input of the transversal filter includes the present outputs of each module, and is written as

$$
Y(n) = \big[y_1(n), y_2(n), ..., y_M(n)\big]^T \tag{20}
$$

Thus the output of the linear filter is

$$
y_L(n) = W^T(n) Y(n) \tag{21}
$$

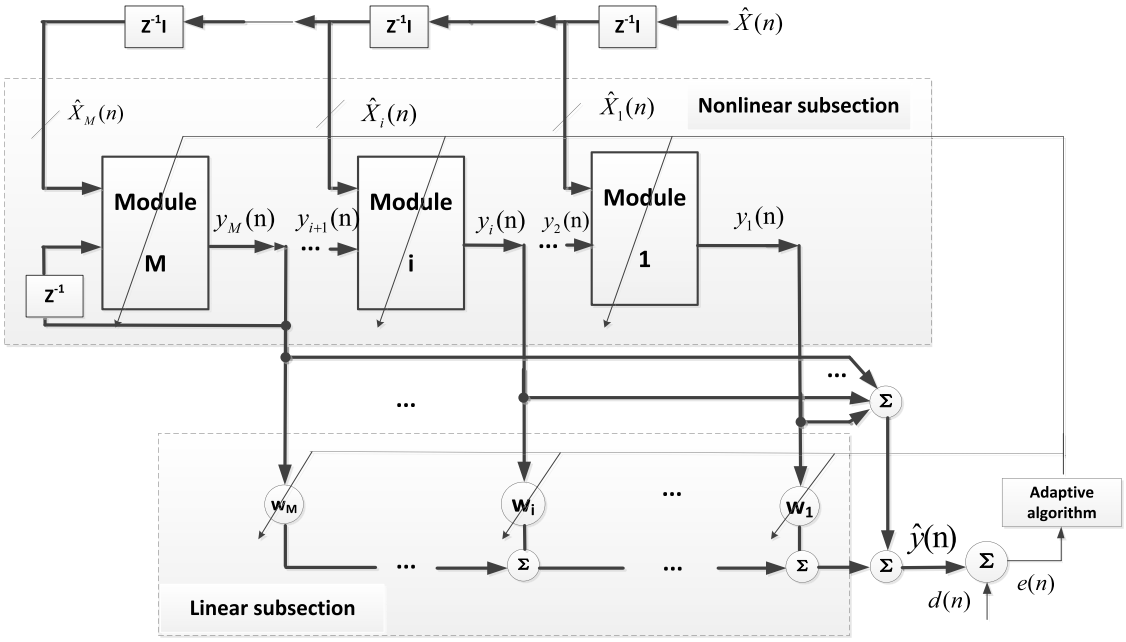

Fig. 2. The HU-GFLANN architecture.

3.3. The HU-GFLANN output

As shown in Fig. 2, the output of the HU-GFLANN is the combination of the outputs of the nonlinear and linear subsections. Thus, the output of the PU-GFLANN architecture at time n can be written as

$$\hat{y}(n) = W^T(n)Y(n) + \sum_{i=1}^{M} H_i^T(n)Xg_i(n) \tag{22}$$

4. Hierarchical M-max filtered-error least mean square (HMmFE-LMS) algorithm

As pointed out in [31], the filtered-error least mean square (FE-LMS) algorithm significantly reduces the computational burden of the filtered-x least mean square (Fx-LMS) algorithm since it can reduce the computational cost for filtering the signal through the secondary path. Especially in the case of the secondary path is nonlinear. In addition, in order to control the computational complexity of the adaptive algorithm, the strategies of the partial coefficient update have been proposed in the literature [32–34]. Among the many possible partial update methods, the M-max partial update method is selected and studied here [32–38]. Complexity reduction by M-max partial update technique is a data-dependent update algorithm which is based on finding the M_m largest values of the input vector. Therefore, a hierarchical M-max update filtered-error least mean square (HMmFE-LMS) algorithm for the HU-GFLANN architecture has proposed in this section. Due to the use of partial update filtered-error technique and hierarchical learning strategy, the proposed algorithm may considerably reduce the computational complexity.

Fig. 3 shows the schematic diagram of the NANC system based on HPU-GFLANN filter. In the figure, $S(z)$ is the transfer function of the secondary path (represent path from the output of the HPU-GFLANN controller to the output of the error sensor $e(n)$), $P(z)$ is the transfer function of the primary path (represent path from reference sensor to error sensor $e(n)$), $x(n)$ is the reference signal which is sensed by a reference microphone and applied to the HPU-GFLANN controller to generate the anti-noise $y_s(n)$, $d(n)$ is primary noise at the noise cancellation point and $ef(n)$ is a filtered version of the error.

According to [31], to achieve a unified structure for both the NANC with a linear secondary path (NANC/LSP) and the NANC with a nonlinear secondary path (NANC/NSP), we use a concept of a virtual secondary path filter $\tilde{g}(n)$ with coefficients as

$$\begin{aligned}\tilde{g}(n) &= \left[\tilde{g}(n,0)\tilde{g}(n,1)\ldots\tilde{g}(n,Ps)\right]^T \\ &= \left[\frac{\partial y_s(n)}{\partial \hat{y}(n)}\frac{\partial y_s(n)}{\partial \hat{y}(n-1)}\cdots\frac{\partial y_s(n)}{\partial \hat{y}(n-Ps)}\right]^T\end{aligned} \tag{23}$$

where Ps is the length of the virtual secondary path. When the secondary path is linear, the coefficient vector of the virtual secondary path filter is equal to the estimated coefficient vector of the linear secondary path. In contrast, when the secondary path is nonlinear, the coefficient vector of the virtual secondary path filter is considered as a time-varying filter.

The goal of the HPU-GFLANN filter is to minimize the instantaneous square error $J(n)$. Herein, we define the cost function $J(n)$ as follows

$$J(n) = e^2(n) = \left(d(n-1) - y_s(n)\right)^2 \tag{24}$$

where $y_s(n)$ is indirectly generated signal at the noise cancellation point.

4.1. Nonlinear subsection

The coefficient vector $H_i(n)$ of the nonlinear subsection will be adjusted according to the steepest descent algorithm to minimize the cost function $J(n)$ as follows

$$H_i(n+1) = H_i(n) - \frac{1}{2}\mu\nabla_{H_i(n)}J(n) \tag{25}$$

where μ is the learning rate; $\nabla_{Hi(n)}J(n)$ is the gradient of cost function $J(n)$ with respect to the weight vector $H_i(n)$ and is deduced by

$$\nabla_{H_i(n)}J(n) = \frac{\partial J(n)}{\partial H_i(n)} \cong -2e(n)\frac{\partial y_s(n)}{\partial H_i(n)} \tag{26}$$

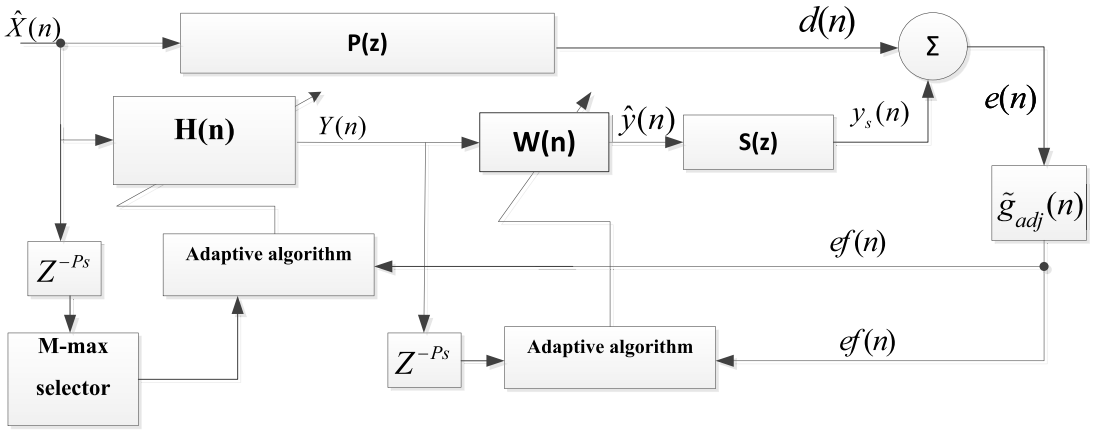

Fig. 3. Schematic diagram of NANC systems using the HPU-GFLANN filter.

Note that

$$\frac{\partial y_s(n)}{\partial H_i(n)} = \sum_{p=0}^{Ps} \frac{\partial y_s(n)}{\partial \hat{y}(n-p)} \frac{\partial \hat{y}(n-p)}{\partial H_i(n)} \tag{27}$$

We assume that when the step size is small, the weight $H_i(n)$ is a slowly varying, thus (27) can be written by

$$\begin{aligned} \frac{\partial \hat{y}(n-p)}{\partial H_i(n)} &\approx \frac{\partial \hat{y}(n-p)}{\partial H_i(n-p)} \\ &= \frac{\partial(\sum_{i=1}^{M} w_i^T(n-p) y_i(n-p) + \sum_{i=1}^{M} H_i^T(n-p) X g_i(n-p))}{\partial H_i(n-p)} \\ &= X g_i(n-p) \end{aligned} \tag{28}$$

from (26), (27) and (28) we can obtain the update equation of $H_i(n)$ as

$$\begin{aligned} H_i(n+1) &= H_i(n) + \mu e(n) \sum_{p=0}^{Ps} \frac{\partial y_s(n)}{\partial \hat{y}(n-p)} X g_i(n-p) \\ &= H_i(n) + \mu e(n) \sum_{p=0}^{Ps} \tilde{g}(n,p) X g_i(n-p) \end{aligned} \tag{29}$$

where $\tilde{g}(n,p)$ is the $(p+1)$th component of the virtual secondary path vector $\tilde{g}(n)$, as defined in (23).

Let $t = n - p + Ps$, so that $n = t + p - Ps$. Thus, the last term in (29), $e(n)\sum_{p=0}^{Ps} \tilde{g}(n,p) X g_i(n-p)$ is expressed by

$$\begin{aligned} & e(n) \sum_{p=0}^{Ps} \tilde{g}(n,p) X g_i(n-p) \\ &= \left[\sum_{p=0}^{Ps} e(t+p-Ps) \tilde{g}(t+p-Ps,p) \right] X g_i(t-Ps) \end{aligned} \tag{30}$$

Let us define vector $\tilde{g}_{adj}(n)$ as an adjoint virtual secondary path filter with coefficients vector as

$$\tilde{g}_{adj}(n) = \left[\tilde{g}(n,Ps) \tilde{g}(n-1,Ps-1) \ldots \tilde{g}(n-Ps,0) \right]^T \tag{31}$$

Note that the vector $\tilde{g}_{adj}(n)$ contrasts with the vector $\tilde{g}(n)$, it not only requires reversing the order of the coefficients of the vector $\tilde{g}(n)$ but also requires delaying the time-varying coefficients.

Obviously, the term $[\sum_{p=0}^{Ps} e(t+p-Ps)\tilde{g}(t+p-Ps,p)]$ in Equation (30) can be considered as the result of filtering the error signal through the adjoint virtual secondary path filter. Thus, we define the filtered error as

$$\begin{aligned} ef(n) &= \sum_{p=0}^{Ps} e(n+p-Ps) \tilde{g}(n+p-Ps,p) \\ &= \sum_{p=0}^{Ps} e(n-(Ps-p) \tilde{g}_{adj}(n, Ps-p) \end{aligned} \tag{32}$$

where $\tilde{g}_{adj}(n,p)$ is the $(p+1)$th component of the vector $\tilde{g}_{adj}(n)$.

Note that in the case of the secondary paths that are linear, the filtered-error can be easily derived as $ef(n) = e(n) * a(n)$, where $a(n)$ is the transfer function of the estimated secondary path.

Consequently, by combining (29), (30) and (32) we yield the weigh update equation of the nonlinear subsection:

$$H_i(n+1) = H_i(n) + \mu ef(n) X g_i(n - P_s) \tag{33}$$

where the term $Xg_i(n-Ps)$ denotes the extended input signal and is delayed by a time unit equaling the length of the secondary path.

In order to further reduce the computational burden, an M-max update technique is applied to the update equation (33). In this algorithm, only a fraction of the total weights is updated during every iterations. Then, the HMmFE-LMS algorithm is given by

$$H_i(n+1) = H_i(n) + \mu ef(n) \Phi(n) X g_i(n - Ps) \tag{34}$$

where the coefficient selection matrix $\Phi(n)$ is defined by

$$\Phi(n) = \begin{bmatrix} \lambda_1(n) & 0 & \cdots & 0 \\ 0 & \lambda_2(n) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_{L_h}(n) \end{bmatrix} \tag{35}$$

With

$$\lambda_j(n) = \begin{cases} 1 & \text{if } |xg_{i,j}(n-Ps)| \in \max_{1 \le k \le L_h} (|xg_{i,k}(n-Ps)|, M_m) \\ 0 & \text{otherwise} \end{cases} \tag{36}$$

where $L_h = 3(L+1) + (L_d+1)(L_d+2)$ is the length of the weight vector $H_i(n)$; $xg_{i,j}(n-Ps)$ is the jth element of the extended input vector $Xg_i(n-Ps)$ in (7); and $1 \le M_m \le L_h$ is the pre-selected M-max parameter. The parameters M_m need to appropriately choose to achieve a good compromise between computational complexity and performance.

4.2. Linear subsection

For the linear subsection, using the steepest descent method, the update equation $W(n)$ is expressed as

$$W(n+1) = W(n) - \frac{1}{2}\eta \nabla_{W(n)} J(n) \tag{37}$$

where η is the learning rate; $\nabla_{W(n)} J(n)$ is the gradient of cost function $J(n)$ with respect to the weight vector $W(n)$.

Analysis is similar to the nonlinear subsection, we easily obtain the update equation of $W(n)$ as follows

$$W(n+1) = W(n) + \eta e f(n) Y(n - Ps) \tag{38}$$

where $Y(n-Ps)$ is the output vector of the M module and delayed Ps time units

5. The convergence of the HPU-GFLANN-based NANC system

According to the adaptive filter theory [39], we know that the update equations (34) and (38) will not ensure stability unless a strong condition is imposed on the learning rate μ and η. In this section, we give the stability conditions for nonlinear and linear subsections of the HPU-GFLANN-based NANC system.

5.1. The nonlinear subsection

According to the gradient descent algorithm, the update equation in (34) can be written equivalent to the following

$$H_i(n+1) = H_i(n) + \mu e(n) \Phi(n) Xfg_i(n) \tag{39}$$

where $Xfg_i(n) = \sum_{p=0}^{Ps} \tilde{g}(n,p) Xg_i(n)$ is the filtered signal of the $Xg_i(n)$ through the virtual secondary path $\tilde{g}(n)$.

To avoid confusion in the analysis, equation (39) is represented in the form of the following partial update rule:

$$H_{ij}(n+1) = \begin{cases} H_{ij}(n) + \mu e(n) Xfg_{ij}, & \text{if vector } Xfg_{ij} \text{ is associated} \\ & \text{with the } M_m \text{ largest values} \\ & \text{of } |xfg_i(n-j+1)|,\ j = 1, \ldots L_h, \\ & i = 1 \ldots M \\ H_{ij}(n) & \text{otherwise} \end{cases} \tag{40}$$

where $xfg_i(n-j+1)$ is the jth element of $Xfg_i(n)$.

The proposed algorithm uses a gradient descent method to adjust the weight vector, as in [11], a discrete-type Lyapunov function can be defined by

$$J_L(n) = \frac{e^2(n)}{2} \tag{41}$$

During the training process, the change of $J_L(n)$ can be expressed as

$$\Delta J_L(n) = J_L(n+1) - J_L(n) = \frac{e^2(n+1) - e^2(n)}{2} \tag{42}$$

The difference of instantaneous output error resulting from the learning can be represented in the Taylor series expansion as follows

$$e(n+1) = e(n) + \frac{\partial e(n)}{\partial H_{ij}(n)} \Delta H_{ij}(n) + h.o.t \tag{43}$$

where $h.o.t$ denotes the higher order terms of the rest of Taylor series expansion and can be ignored. Hence, we have

$$\begin{aligned} \Delta J_L(n) &= \frac{e^2(n+1) - e^2(n)}{2} \\ &= \frac{1}{2}\left[\frac{\partial e(n)}{\partial H_{ij}(n)}\right]^T \Delta H_{ij}(n) \left\{ \left[\frac{\partial e(n)}{\partial H_{ij}(n)}\right]^T \Delta H_{ij}(n) + 2e(n) \right\} \end{aligned} \tag{44}$$

From the update rule in (40), we can obtain

$$\Delta H_{ij}(n) = \mu e(n) Xfg_{ij}(n) \tag{45}$$

Substitute (45) in (44), and set $\Delta J_L(n) < 0$, we have

$$\begin{aligned} \Delta J_L(n) &= \frac{1}{2}\left[\frac{\partial e(n)}{\partial H_{ij}(n)}\right]^T \mu e(n) Xfg_{ij}(n) \\ &\quad \times \left\{ \left[\frac{\partial e(n)}{\partial H_{ij}(n)}\right]^T \mu e(n) Xfg_{ij}(n) + 2e(n) \right\} \\ &= -\frac{1}{2} \left\| Xfg_{ij}(n) \right\|^2 \mu e^2(n) \left\{ 2 - \mu \left\| Xfg_{ij}(n) \right\|^2 \right\} < 0 \end{aligned} \tag{46}$$

where $\frac{\partial e(n)}{\partial H_{ij}(n)} = \frac{\partial [d(n) - \tilde{g}(n) * \{\sum_{i=1}^{M} H_i^T(n) Xg_i(n) + W^T(n) Y(n))\}]}{\partial H_{ij}(n)} = -Xfg_{ij}(n)$, the symbol "*" and $\|.\|$ are the convolution and the Euclidean, respectively.

Therefore, the local convergence of the system is guaranteed provided that $0 < \mu < \frac{2}{\|Xfg_{ij}(n)\|^2}$.

5.2. Linear subsection

Using the methods of analyzing stability condition and convergence performance of LMS algorithm under small step size assumptions [39], we can propose the range of learning rate to guarantee a convergence condition of linear subsection for HPU-GFLANN-based NANC system as follows

$$0 < \eta < \frac{2}{\gamma_{\max}(R_{YY})} \tag{47}$$

$$R_{YY} = E\left[Y(n - P_s) Y^T(n - P_s) \right] \tag{48}$$

where $\gamma_{\max}$ denotes the max eigenvalue of matrix R_{YY}.

6. Computational complexity analysis

In this section, an analysis of the computational complexity of the NANC system based on the proposed HPU-GFLANN filter is presented. As has been reported in [23], the GFLANN uses the Fx-LMS algorithm. Thus, to make a fair comparison, we will compare it to the HFx-GFLANN filter (using the HU-GFLANN architecture with Fx-LMS algorithm), HFE-GFLANN filter (using the HU-GFLANN architecture with FE-LMS algorithm) and HPU-GFLANN filter.

Note that in the ANC system using the Fx-LMS algorithm, the secondary path has a significant effect on computational complexity. When the secondary path is linear, it can take advantage of the delay relationship in the nonlinear state to reduce the computational burden. In contrast, when the secondary path is nonlinear, the Fx-LMS algorithm cannot take advantage of the delay relationship.

Assuming Ps is the memory size of the secondary path; L is external input signals of each module; L_d is the cross-term selection parameter; M is the number of modules, the computational complexity of the HPU-GFLANN filter-based NANC system is required the following major operations

- Generating the cross-terms require $2(L_d + 1)$ multiplications.
- The output of each module $y_i(n)$ requires L_h multiplications and $L_h - 1$ additions, where $L_h = 3(L+1) + (L_d+1)(L_d+2)$ is the length of the expanded input signal of the ith module.

Table 1
Total computational requirements for NANC/LSP case.

Filters for NANC system	Multiplications	Additions
FLANN	$14L_f + 7Ps + 1$	$14L_f + 7(Ps-1) - 1$
GFLANN	$2L_{gd}(L_{gd}+2) + (3+2L_{gd})Ps + 6L_g$	$2L_{gd}(L_{gd}+1) + (3+2L_{gd})(Ps-1) + 6L_g - 1$
HFx-GFLANN	$2N_d + 2L_h + 2M + Ps + 2L_d Ps + ML_h + 4$	$2L_h + 2M + Ps + 2L_d Ps - 2L_d + ML_h - 4$
HFE-GFLANN	$2L_d + 2L_h + 2M + Ps + ML_h + 4$	$2L_h + 2M + Ps - + ML_h - 4$
HPU-GFLANN	$2L_d + L_h + 2M + Ps + M_m + ML_h + 3L + 7$	$L_h + 2M + Ps + M_m + ML_h + 3L - 1$

Table 2
Total computational requirements for NANC/NSP case.

Filters for NANC system	Multiplications	Additions
FLANN	$14L_f + 7L_f Ps + 1$	$14L_f + 7L_f(Ps-1) - 1$
GFLANN	$2L_{gd}(L_{gd}+2) + (3L_g + L_{gd}(L_{gd}+1))Ps + 6L_g$	$2L_{gd}(L_{gd}+1) + 6L_g + (3L_g + L_{gd}(L_{gd}+1))(Ps-1) - 2$
HFx-GFLANN	$2L_d + 2L_h + 2M + 5Ps + L_h Ps + ML_h + 4$	$2L_h + M + MPs + L_h(Ps-1) + ML_h - 2$
HFE-GFLANN	$2L_d + 2L_h + 2M + Ps + ML_h + 4$	$2L_h + 2M + Ps + ML_h - 4$
HPU-GFLANN	$2L_d + L_h + 2M + Ps + M_m + ML_h + 3L + 7$	$L_h + 2M + Ps + M_m + ML_h + 3L - 1$

Table 3
Total computational requirements for specific case in experiment 1 and experiment 2.

Filters for NANC system	Nonlinear secondary path ($P_s = 3$) $L_f = 10$; $L_g = 10$, $L_{gd} = 9$; $L = 4$, $L_d = 3$, $M = 5$, $M_m = 5$		Linear secondary path ($P_s = 5$) $L_f = 10$; $L_g = 10$, $L_{gd} = 9$; $L = 4$, $L_d = 3$, $M = 5$, $M_m = 5$	
	Multiplications	Additions	Multiplications	Additions
FLANN [13]	351	279	176	167
GFLANN [23]	618	496	363	341
HFx-GFLANN	385	333	300	280
HFE-GFLANN	268	254	270	256
HPU-GFLANN	253	239	255	241

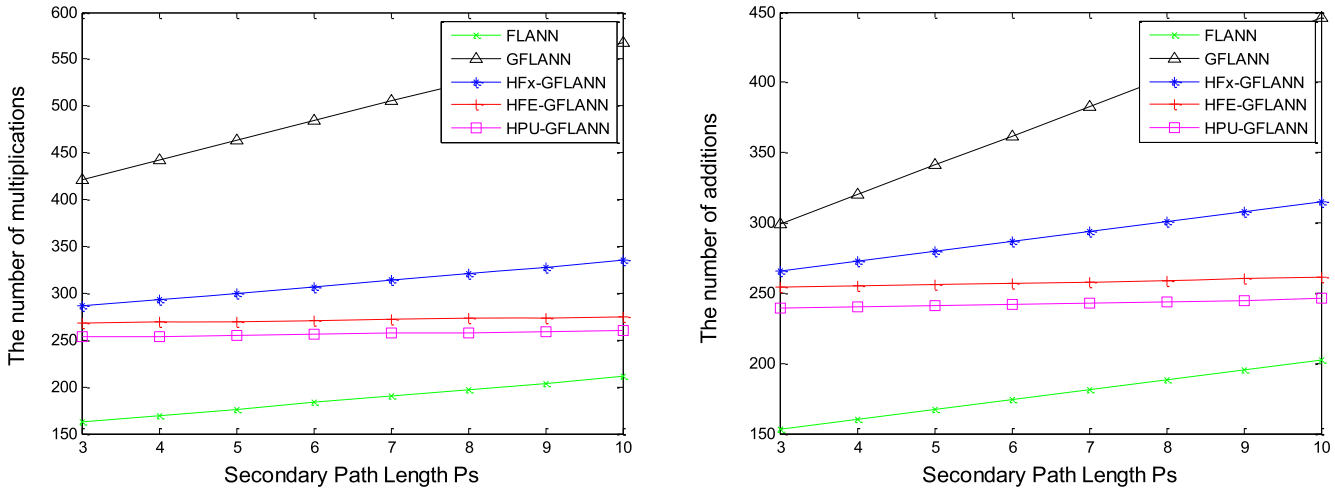

Fig. 4. The number of multiplications and additions required in each of the controller for NANC/LSP.

- The output of the HPU-GFLANN filter $\hat{y}(n)$ requires $M + ML_h$ multiplications and $2(M-1) + M(L_h - 1)$ additions
- Calculate $ef(n)$, the filtered error of the adjoint virtual secondary path filter, which require Ps multiplications and $Ps - 1$ additions.
- Update linear filter coefficient $W(n)$ in (38) requires $M + 1$ multiplications and M additions.
- Update filter coefficient $H_i(n)$ in (34) requires $M_m + 1$ multiplications and M_m additions, where M_m is the parameter of the M-max partial update algorithm.

Therefore, the total calculation requirements of the NANC with a linear secondary path (NANC/LSP) based on FLANN, GFLANN, HFx-GFLANN, HFE-GFLANN and HPU-GFLANN filters are summarized in Table 1. Similarly, the total calculation requirements for the NANC with a nonlinear secondary path (NANC/NSP) case are summarized in Table 2, where L_g and L_{gd} are memory length and the cross-term selection parameter of the GFLANN, respectively. In addition, the computational cost comparison in specific cases is also shown in Table 3.

For a more general view, we assume the length of the secondary path increases from 3 to 10. Thus, the number of multiplications and additions required in each of the controller for the NANC/LSP case is plotted in Fig. 4 as a function of P_s, respectively. It is clear that the HPU-GFLANN, HFE-GFLANN and HFx-GFLANN controllers are more computationally efficient than the GFLANN controller. The FLANN controller has the lowest computational complexity but its noise reduction capability is too poor compared to other controllers. Similarly, the number of additions and multiplications required in each of the controllers for the NANC/NSP

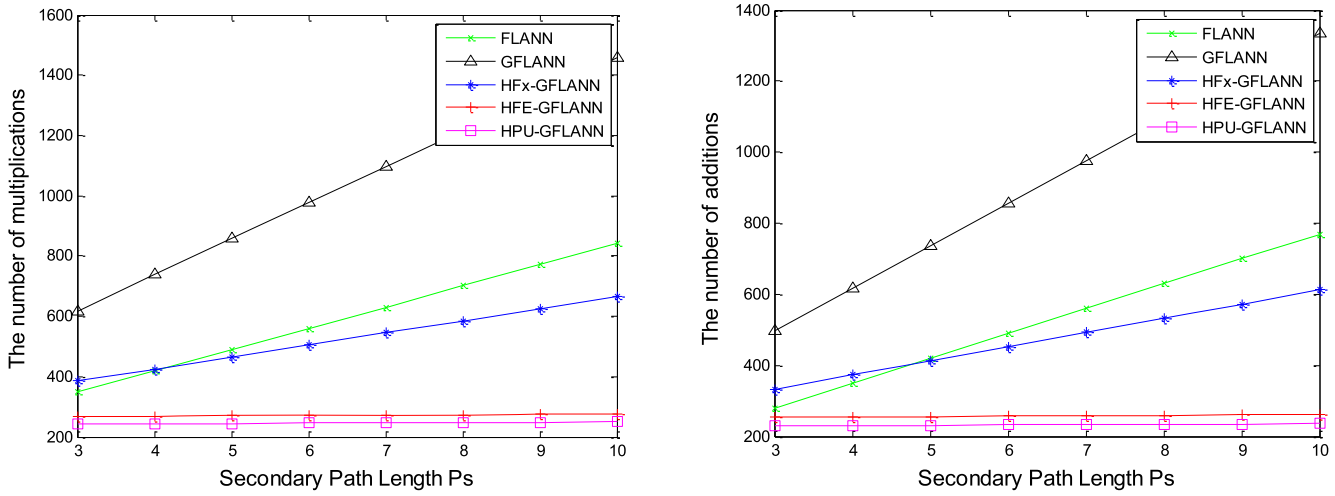

Fig. 5. The number of multiplications and additions required in each of the controller for NANC/NSP.

is also illustrated in Fig. 5, respectively. From these figures, it is evident that the HPU-GFLANN and HFE-GFLANN controllers are computationally superior to the GFLANN and FLANN controllers, and the HFx-GFLANN are more computationally efficient than the FLANN controller when the secondary path length $P_s \geq 5$. This can be explained since the HU-GFLANN architecture is simpler than the GFLANN structure and the proposed algorithms are more efficient than the Fx-LMS algorithm.

7. Simulation

In order to prove the effectiveness of proposed HPU-GFLANN based-NANC system, several simulation experiments are presented in this section. The noise cancellation performance and the computational complexity of the HPU-GFLANN based-NANC system are compared with the FLANN and GFLANN based-NANC systems for both NANC/LSP and NANC/NSP.

In these simulations, the memory length of the FLANN and GFLANN filters are chosen as $L_g = L_f = 10$; and of the HPU-GFLANN denoted as $L = 4$. The parameter for selecting the cross-term of GFLANN is $L_{gd} = 9$, that of HPU-GFLANN as $L_d = 3$. The function expansion of the input signal is third-order type ($B = 3$) for the FLANN and first-order type ($B = 1$) for HPU-GFLANN and GFLANN. Without losing generality, the coefficients of the nonlinear subsection of the HPU-GFLANN can be divided into three groups for increased flexibility during the update process. The first group updates the weights of the linear part, the second group updates the weights of the sin(.), cos(.), and the third group updates the weights of the cross-term elements. In fact, the weight group of the cross-terms has the greatest influence on the computational complexity of the system, thus we only apply the HMmFE-LMS algorithm to this group. The learning curves are ensemble averages on 100 independent runs and are smoothed with a rectangular window of length equal to 100 samples in order to better discern the curves behavior.

7.1. Experiment 1

In this experiment, we assume that the primary path exhibiting strong nonlinear behavior. Its primary noise that generated at the canceling point can be considered as a following third-order polynomial model

$$d(n) = u(n-2) + g_1 u^2(n-2) - g_2 u^3(n-2) \tag{49}$$

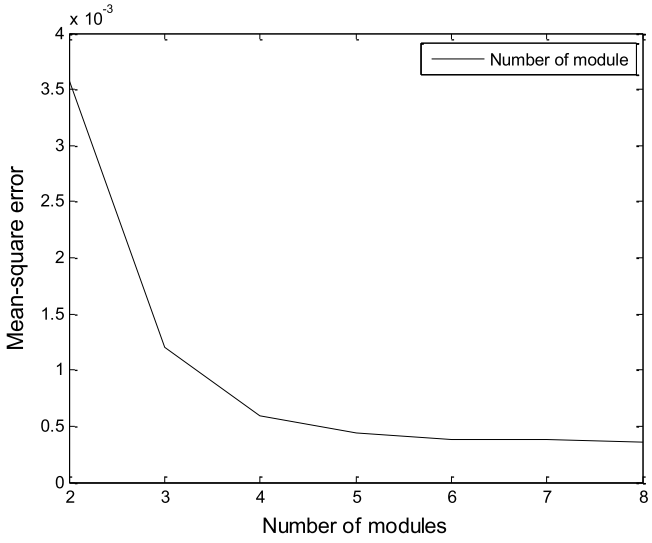

Fig. 6. MSE versus the number of modules M.

where $g_1 = 0.8$, $g_2 = 0.4$ are a measure of the strength of the primary path nonlinearity, $u(n) = x(n) * f(n)$ and $f(n)$ is the impulsive response of the transfer function $f(z) = z^{-3} - 0.3z^{-4} + 0.2z^{-5}$.

The secondary path is the non-minimum phase and its estimate is shown as $S(z) = \hat{S}(z) = z^{-2} + 1.5z^{-3} - z^{-4}$. The reference noise $x(n)$ is a sinusoidal wave of 500 Hz sampled at the rate of 8000 samples/s, which is obtained by

$$x(n) = \sqrt{2}\sin\left(\frac{2\pi 500n}{8000}\right) + v(n) \tag{50}$$

where $v(n)$ is a white noise process with the Gaussian distribution. The signal power-to-noise power ratio is set to 40 dB.

a) Choice of the number of modules The number of modules of the HPU-GFLANN based-ANC system is an important parameter. It directly affects the performance and the computational cost of the system. Hence, to make the best choice, in this experiment, we will evaluate the noise-canceling performance of the system by giving M increases from 2 to 8. Learning rate of the nonlinear subsection of the HPU-GFLANN are set to $\mu_1 = 0.0007$, $\mu_2 = 0.0005$, $\mu_3 = 0.0003$ for linear part, sin(.) cos(.) part and cross-terms part, respectively. Learning rate of the linear subsection of the HPU-GFLANN is $\eta = 0.13$. Fig. 6 shows the influence of the number

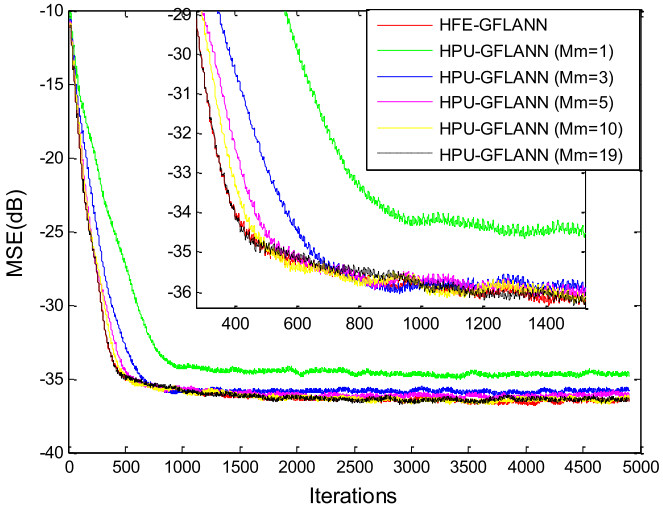

Fig. 7. The HPU-GFLANN versus HFE-GFLANN for different values of M_m. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

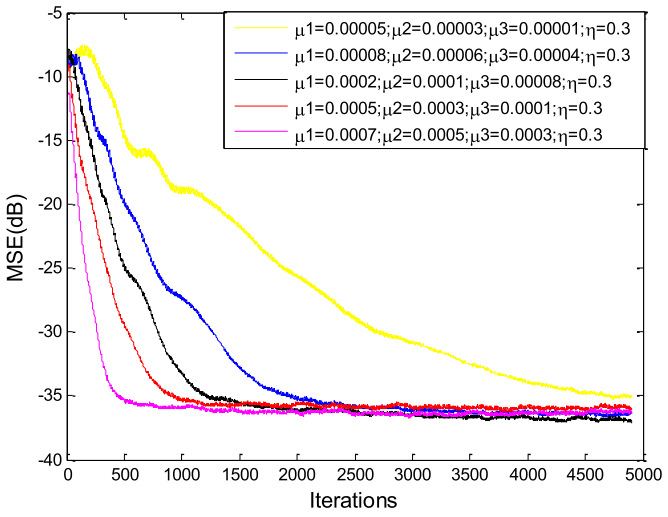

Fig. 8. MSE versus step-size of nonlinear part of HPU-GFLANN.

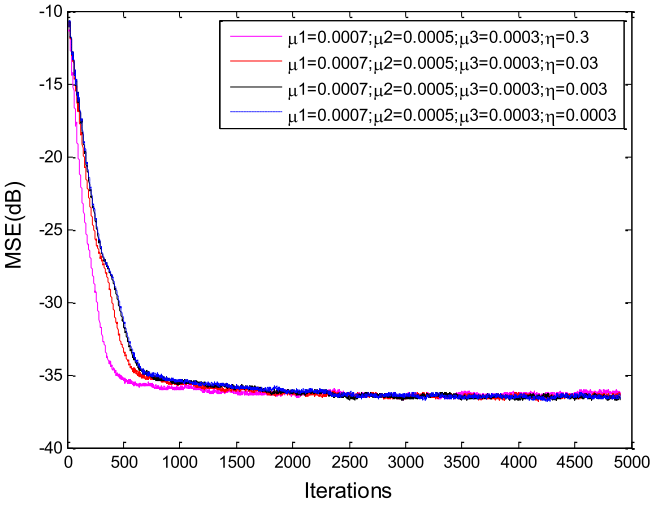

Fig. 9. MSE versus step-size of linear part of HPU-GFLANN.

of module on the mean-square error (MSE). Clearly, the MSE decreased slightly as M increased from 5 to 8. Therefore, we here choose M to be equal to 5.

b) Choice of the M-max parameter To select the appropriate parameter M_m, we carried out the evaluation of the performance of HPU-GFLANN versus HFE-GFLANN for different values of $1 < M_m < 20$ (length of the cross-term vector). The suitable M_m value is a good compromise between computational complexity and performance. In this experiment, we keep the number of modules $M = 5$; Learning rate of the HPU-GFLANN and HFE-GFLANN is set as part (a). Fig. 7 illustrates the performance MSE of HPU-GFLANN versus HFE-GFLANN for different values of M_m. It is clear that when $M_m = 5$, the performance of the HPU-GFLANN can be equivalent to that of HFE-GFLANN.

c) Choice of the step-sizes As shown above, the HPU-GFLANN filter consists of two step-sizes groups that are nonlinear step-sizes group ($\mu 1$, $\mu 2$, and $\mu 3$ for linear input signal part, sin(.) cos(.) input signal part and cross-terms input signal part, respectively) and linear sep-size η. To obtain the appropriate step-sizes parameter, we first keep one of them as constants and change the other. Figs. 8 and 9 show the performance of HPU-GFLANN when one step-size changes. From these figures, it is easy to see that the convergence performance of the filter changes rapidly when the step-size of the nonlinear part increases from ($\mu 1 = 0.00005$, $\mu 2 = 0.00003$, and $\mu 3 = 0.00001$) to ($\mu 1 = 0.0007$, $\mu 2 = 0.0005$, and $\mu 3 = 0.0003$), and changes slowly when the step-size of the linear part increases from 0.0003 to 0.3.

Moreover, based on the analysis in section 5, we also easily calculate the theoretical convergence condition of step-size of the nonlinear subsection is $0 < \mu < 2/\|Xfg_{ij}(n)\|^2 = 0.0637$ and step-size of linear subsection is $0 < \eta < 2$ (linear subsection of HPU-GFLANN filter can be considered as a conventional LMS filter). Here, $Xfg_{ij}(n)$ is the filtered version of the expanded input signal $Xg_i(n)$ through the secondary path. Clearly, the step-sizes selected in this experiment satisfy the stable conditions.

d) Comparing performance Next, comparing the noise-canceling performance of the proposed HPU-GFLANN with FLANN and GFLANN is presented. In this experiment, the learning rate of the nonlinear subsection of the HPU-GFLANN and HFE-GFLANN are set to $\mu_1 = 0.0007$, $\mu_2 = 0.0005$, $\mu_3 = 0.0003$ for linear part, sin(.) cos(.) part, cross-terms part, respectively. Learning rate of the linear subsection is $\eta = 0.13$. The learning rate of the GFLANN are set to $\mu_{G1} = 0.005$, $\mu_{G2} = 0.0015$ and $\mu_{G3} = 0.001$ for linear part, sin(.) cos(.) part and cross-terms part, respectively. The learning rate of the FLANN are set to $\mu_{F1} = 0.007$ and $\mu_{F2} = 0.002$ for linear part and sin(.) cos(.) part, respectively. Fig. 10 shows a comparative plot of the MSE achieved by the NANC/LSP systems with the HPU-GFLANN, HFE-GFLANN, GFLANN and FLANN controllers. It is clear that the HPU-GFLANN and HFE-GFLANN controllers can achieve better noise-canceling performance than FLANN and GFLANN controllers when the primary noise at the canceling point exhibits high nonlinear behavior.

7.2. Experiment 2

To evaluate the noise-canceling performance of the HPU-GFLANN filter for the NANC/NSP system, we refer here to the situation described in Example 2 in [23]. In this simulation, the secondary path is modeled by a Volterra series give as

$$y_s(n) = y(n) + 0.35y(n-1) + 0.09y(n-2) - 0.5y(n)y(n-1) + 0.4y(n)y(n-2) \quad (51)$$

The primary path is assumed to be exhibiting high nonlinear behavior and modeled by a Volterra series as follows

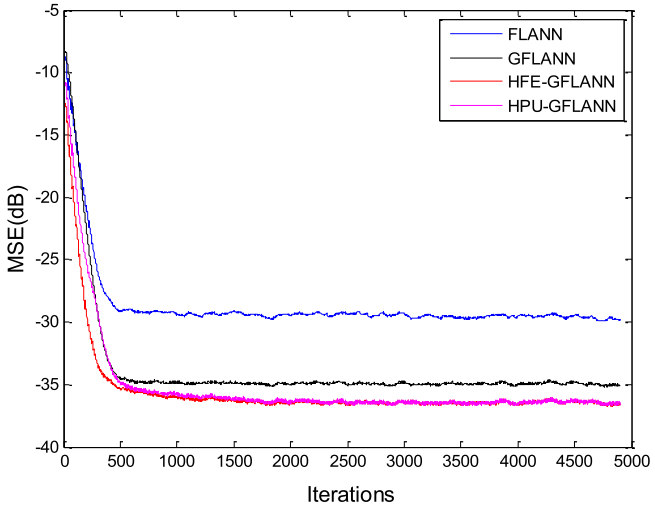

Fig. 10. Learning curves of controllers use the linear secondary path.

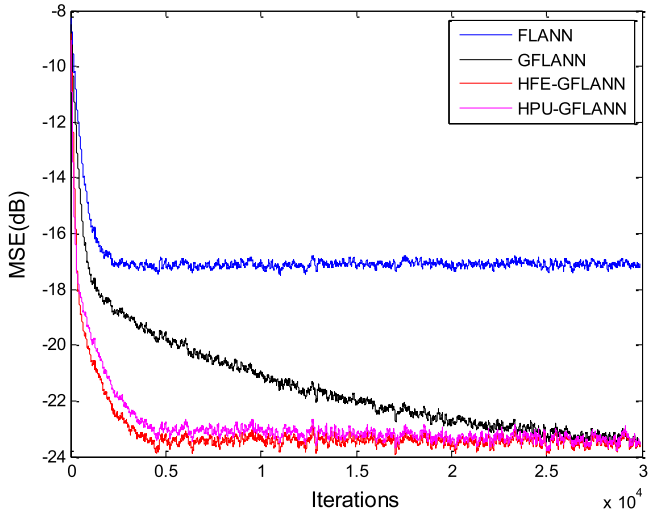

Fig. 11. Learning curves of controllers use the secondary path as the Volterra model.

$$
\begin{aligned}
d(n) &= x(n) + 0.8x(n-1) + 0.3x(n-2) + 0.4x(n-3) \\
&\quad - 0.8x(n)x(n-1) + 0.9x^2(n-2) \\
&\quad + 0.7x^2(n-3) - 3.9x^3(n-1) - 2.6x^2(n-1)x(n-3) \\
&\quad + 2.1x^2(n-2)x(n-3)
\end{aligned} \tag{52}
$$

The reference signal is random noise with a uniform distribution between -0.5 and $+0.5$. The learning rate for all the controllers are chosen as: the HPU-GFLANN and HFE-GFLANN are $\mu_1 = 0.009$, $\mu_2 = 0.003$, $\mu_3 = 0.009$ and $\eta = 0.05$; the GFLANN are $\mu_{G1} = 0.09$, $\mu_{G2} = 0.05$ and $\mu_{G3} = 0.2$; the FLANN are $\mu_{F1} = 0.5$ and $\mu_{F2} = 0.01$. The M_m parameter is set to 5. Fig. 11 shows the averaged MSE performance of the controllers when the primary path and the secondary path are time-varying nonlinear models. It is clearly observed that the proposed HPU-GFLANN and HFE-GFLANN controllers outperform both the GFLANN and the FLANN controllers.

7.3. Experiment 3

In actual ANC systems, the output signal of the controller needs to be transmitted through the main components such as the amplifier, the loudspeaker to produce an anti-noise signal. Thus the secondary path can be approximated as block-oriented nonlinear

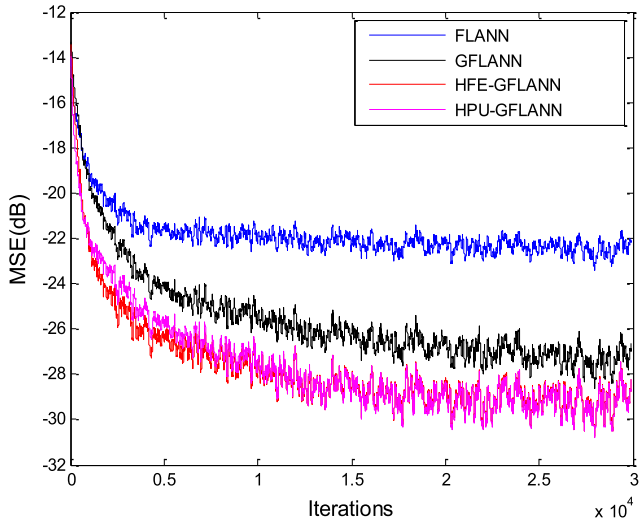

Fig. 12. Learning curves of controllers use the secondary path as the Hammerstein model.

models. There are several secondary path models shown in [31] such as linear-memoryless nonlinear-linear (LNL) cascades, Wiener and Hammerstein models. In this case, we assume that the secondary path is described as Hammerstein model with a memoryless nonlinearity and followed by a linear filter:

$$
\begin{aligned}
u(n) &= \tanh(y(n)) \\
y_s(n) &= u(n) + 0.2u(n-1) + 0.05u(n-2)
\end{aligned} \tag{53}
$$

The primary path is the same as in experiment 2. The reference signal is colored noise and is generated by

$$
\begin{aligned}
x(n) &= 0.04x(n-1) - 0.034x(n-2) + 0.0396x(n-3) \\
&\quad - 0.07565x(n-4) - 0.1v(n) - 0.01v(n-1) \\
&\quad - 0.137v(n-2) + 0.0353v(n-3) + 0.06984v(n-4)
\end{aligned} \tag{54}
$$

where $v(n)$ is mean-zero white Gaussian sequence with variance one.

Fig. 12 depicts the results of MSE's for the controllers. The learning rates of the controllers are: for the HPU-GFLANN and HFE-GFLANN ($\mu_1 = 0.02$, $\mu_2 = 0.003$, $\mu_3 = 0.009$ and $\eta = 0.2$); for the GFLANN ($\mu_{G1} = 0.15$, $\mu_{G2} = 0.05$ and $\mu_{G3} = 0.25$); for the FLANN ($\mu_{F1} = 0.5$ and $\mu_{F2} = 0.1$). The M_m parameter is set to 5. In this experiment, it is also clearly observed that the proposed HPU-GFLANN and HFE-GFLANN controllers outperform both the GFLANN and the FLANN controllers.

7.4. Experiment 4

In this experiment, the secondary path and the primary path are used similarly to those of Experiment 3. The reference noise is a logistic chaotic noise, which is generated by the following recursive equation:

$$
x(n+1) = \gamma x(n)\big[1 - x(n)\big] \tag{55}
$$

where $\gamma = 4$ and initial value $x(0) = 0.9$.

For this experiment, the amplitude of the reference signal is limited to the range $(-0.5$ to $0.5)$ and its length is equal to 20000 samples. The learning rates of the controllers are: for the HPU-GFLANN and HFE-GFLANN ($\mu_1 = 0.001$, $\mu_2 = 0.0003$, $\mu_3 = 0.001$ and $\eta = 0.05$); for the GFLANN ($\mu_{G1} = 0.06$, $\mu_{G2} = 0.05$ and $\mu_{G3} = 0.06$); for the FLANN ($\mu_{F1} = 0.05$ and $\mu_{F2} = 0.01$). The M_m parameter is set to 5.

Table 4
Noise attenuation and computational complexity for experiment 4.

Controllers for NANC system	Noise attenuate on (dB)	Mul	add
FLANN	8.212	351	279
GFLANN	15.844	618	496
HFE-GFLANN	16.285	268	254
HPU-GFLANN	16.188	253	239

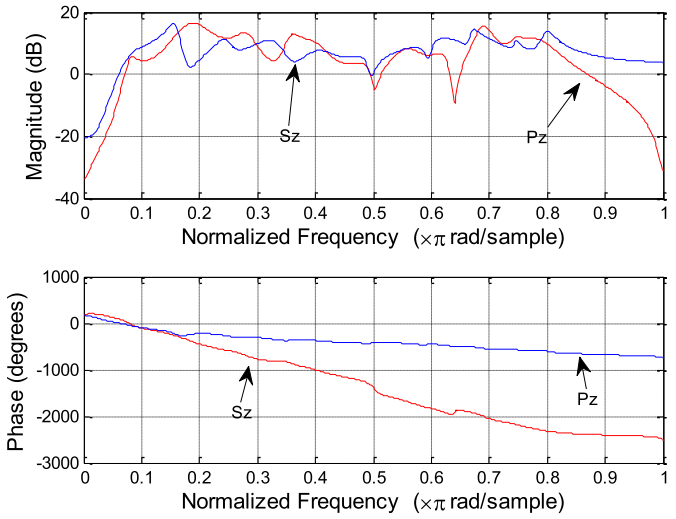

Fig. 13. The amplitude response and phase response for the measured primary path and secondary path.

Table 4 shows the noise attenuation and computational complexity of the controllers. Here, the noise attenuation at the cancellation point is equal to the ratio between the mean noise power values without and with NANC. Note that the noise power is measured in the last 10,000 samples and the controllers are selected with the same convergence rate. As it appears from Table 4, the proposed HPU-GFLANN and HFE-GFLANN controllers achieve better noise attenuation and lower computational complexity compared to the FLANN and GFLANN controller. The HPU-GFLANN reduced the computational requirement compared to the GFLANN approximately equal 60% for multiplication and 52% for addition.

7.5. Experiment 5

To demonstrate the effectiveness of the proposed controllers in an actual application, the real measured primary path and secondary path used in [2] are adopted. Fig. 13 depicts the amplitude response and phase response for the secondary and primary paths.

The reference input signal is a sinusoidal signal which consists of three normalized frequencies of 0.02, 0.04, 0.08, and is normalized to have a unit power. The SNR at the noise canceling point is set to 40 dB. The reference signal is assumed to be strongly distorted by clipping threshold at 50% of the maximum signal value.

The learning rates of the controllers are: for the HPU-GFLANN and HFE-GFLANN ($\mu_1 = 0.0000025$, $\mu_2 = 0.000002$, $\mu_3 = 0.0000005$ and $\eta = 0.013$); for the GFLANN ($\mu_{G1} = 0.000005$, $\mu_{G2} = 0.000003$ and $\mu_{G3} = 0.000002$); for the FLANN ($\mu_{F1} = 0.000007$ and $\mu_{F2} = 0.0000015$). The memory length of the HPU-GFLANN and HFE-GFLANN are chosen as $L = 9$; the parameter for selecting the cross-terms is $L_d = 3$; the M_m parameter is set to 5. Fig. 14 shows the average learning curves for the strong saturated nonlinear noise under the real measured secondary and primary paths. It is obvious that the proposed HFE-GFLANN and HPU-GFLANN controllers achieve an improvement in the noise-canceling performance compared to the GFLANN and FLANN controllers.

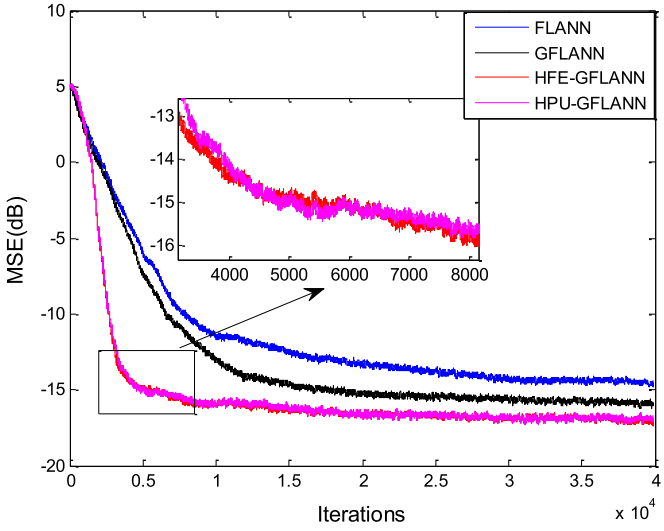

Fig. 14. Learning curves of controllers use the real measured secondary and primary paths.

8. Conclusion

This paper has proposed a nonlinear adaptive HPU-GFLANN filter to reduce the computational complexity of the GFLANN filter for NANC system. The proposed filter inherits the good properties of filters based on pipelined architecture such as low computational complexity, easy implementation in practice. The proposed HMmFE-LMS algorithm using the data-dependence partial update filtered-error technique and the hierarchical learning strategy is suitable for HU-GFLANN architecture. Simulation results have demonstrated that the proposed HPU-GFLANN filter-based NANC system can offer better performance than that of the GFLANN filter-based NANC system with a significantly reduced computational complexity.

Declaration of Competing Interest

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Acknowledgments

This work was partially supported by National Science Foundation of China (Grant: 61671392).

References

[1] N.V. George, G. Panda, Advances in ANC: a survey, with emphasis on recent nonlinear techniques, Signal Process. 93 (2013) 363–377.
[2] S.M. Kuo, D.R. Morgan, Active Noise Control Systems: Algorithm and DSP Implementations, Wiley, New York, 1996.
[3] P. Strauch, B. Mulgrew, Active control of nonlinear noise processes in a linear duct, IEEE Trans. Signal Process. 46 (9) (1998) 2404–2412.
[4] M.H. Costa, J.C.M. Bermudez, N.J. Bershad, Stochastic analysis of the filtered-X LMS algorithm in systems with nonlinear secondary paths, IEEE Trans. Signal Process. 50 (6) (2002) 1327–1342.
[5] L. Tan, J. Jiang, Adaptive Volterra filters for active control of nonlinear noise processes, IEEE Trans. Signal Process. 49 (8) (2001) 1667–1676.
[6] G.L. Sicuranza, A. Carini, Filtered-X affine projection algorithm for multichannel active noise control using second-order Volterra filters, IEEE Signal Process. Lett. 11 (11) (2004) 853–857.
[7] N. Sadegh, A perceptron network for functional identification and control of nonlinear systems, IEEE Trans. Neural Netw. 4 (1993) 982–988.

[8] T. Krukowicz, Neural fixed-parameter active noise controller for variable frequency tonal noise, Neurocomputing 121 (2013) 387–391.
[9] C.Y. Chang, Neural filtered-U algorithm for the application of active noise control system with correction terms momentum, Digit. Signal Process. 20 (4) (2010) 1019–1026.
[10] R.T. Bambang, Adjoint EKF learning in recurrent neural networks for nonlinear active noise control, Appl. Soft Comput. 8 (4) (2008) 1498–1504.
[11] Q.Z. Zhang, W.S. Gan, Y.L. Zhou, Adaptive recurrent fuzzy neural networks for active noise control, J. Sound Vib. 296 (2006) 935–948.
[12] V. Patel, N.V. George, Nonlinear active noise control using spline adaptive filters, Appl. Acoust. 93 (2015) 38–43.
[13] D.P. Das, G. Panda, Active mitigation of nonlinear noise processes using a novel filtered-s LMS algorithm, IEEE Trans. Speech Audio Process. 12 (3) (2004) 313–322.
[14] N.V. George, G. Panda, A robust filtered-s LMS algorithm for nonlinear active noise control, Appl. Acoust. 73 (2012) 836–841.
[15] S.K. Behera, D.P. Das, B. Subudhi, Adaptive nonlinear active noise control algorithm for active headrest with moving error microphones, Appl. Acoust. 123 (2017) 9–19.
[16] H. Zhao, X. Zeng, J. Zhang, Adaptive reduced feedback FLNN filter for active control of nonlinear noise processes, Signal Process. 90 (3) (2010) 834–847.
[17] G.L. Sicuranza, A. Carini, On the BIBO stability condition of adaptive recursive FLANN filters with application to nonlinear active noise control, IEEE Trans. Audio Speech Lang. Process. 20 (1) (2012) 234–245.
[18] D.C. Le, J.S. Zhang, Y.J. Pang, A bilinear functional link artificial neural network filter for nonlinear active noise control and its stability condition, Appl. Acoust. 132 (2018) 19–25.
[19] V. Patel, V. Gandhi, S. Heda, N.V. George, Design of adaptive exponential functional link network-based nonlinear filters, IEEE Trans. Circuits Syst. I 63 (9) (2016) 1434–1442.
[20] D.C. Le, J.S. Zhang, D.F. Li, S. Zhang, A generalized exponential functional link artificial neural networks filter with channel-reduced diagonal structure for nonlinear active noise control, Appl. Acoust. 139 (2018) 174–181.
[21] N.V. George, G. Panda, On the development of adaptive hybrid active noise control system for effective mitigation of nonlinear noise, Signal Process. 92 (2) (2012) 509–516.
[22] N.V. George, A. Gonzalez, Convex combination of nonlinear adaptive filters for active noise control, Appl. Acoust. 76 (2014) 157–161.
[23] G.L. Sicuranza, A. Carini, A generalized FLANN filter for nonlinear active noise control, IEEE Trans. Audio Speech Lang. Process. 19 (8) (2011) 2412–2417.
[24] S. Haykin, L. Li, Nonlinear adaptive prediction of nonstationary signals, IEEE Trans. Signal Process. 43 (2) (1995) 526–535.
[25] H.Q. Zhao, J.S. Zhang, A novel adaptive nonlinear filter-based pipelined feedforward second-order Volterra architecture, IEEE Trans. Signal Process. 57 (1) (Jan. 2009) 237–246.
[26] J.S. Zhang, H.Q. Zhao, A novel adaptive bilinear filter based on pipelined architecture, Digit. Signal Process. 20 (1) (2010) 23–38.
[27] J.S. Zhang, Y.J. Pang, Pipelined robust M-estimate adaptive second order Volterra filter against impulsive noise, Digit. Signal Process. 26 (2014) 71–80.
[28] S. Zhang, J.S. Zhang, Y.J. Pang, Pipelined set-membership approach to adaptive Volterra filtering, Signal Process. 129 (2016) 195–203.
[29] D.G. Stavrakoudis, J.B. Theocharis, Pipelined recurrent fuzzy neural networks for nonlinear adaptive speech prediction, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 37 (5) (2007) 1305–1320.
[30] Y.J. Pang, J.S. Zhang, A hierarchical alternative updated adaptive Volterra filter with pipelined architecture, Digit. Signal Process. 56 (2016) 67–78.
[31] D. Zhou, V. DeBrunner, Efficient adaptive nonlinear filters for nonlinear active noise control, IEEE Trans. Circuits Syst. I, Regul. Pap. 54 (3) (2007) 669–681.
[32] K. Dogancay, Partial-Update Adaptive Signal Processing: Design Analysis and Implementation, 2008.
[33] T. Aboulnasr, K. Mayyas, Complexity reduction of the NLMS algorithm via selective coefficient update, IEEE Trans. Signal Process. 47 (5) (1999) 1421–1424.
[34] Y. Chien, W. Tseng, A new variable step-size method for the M-max LMS algorithms, in: IEEE International Conference on Consumer Electronics-Taiwan, Taipei, 2014, pp. 21–22.
[35] M. Boudiaf, M. Benkherrat, M.A. Boudiaf, Partial-update adaptive filters for event-related potentials denoising, in: IET 3rd International Conference on Intelligent Signal Processing, London, UK, 2017, pp. 1–6.
[36] K. Mayyas, Performance analysis of the selective coefficient update NLMS algorithm in an undermodeling situation, Digit. Signal Process. 23 (2013) 1967–1973.
[37] V. Patel, N.V. George, Partial update even mirror Fourier non-linear filters for active noise control, in: Signal Processing Conference, IEEE, 2015.
[38] N.V. George, G. Panda, V. Kumar, On the development of a partial update multichannel nonlinear active noise control system, in: 7th International Conference on Signal Processing and Communication Systems, ICSPCS, 2013.
[39] S. Haykin, Adaptive Filter Theory, 4th ed., Prentice-Hall, Englewood Cliffs, NJ, 2002.

Dinh Cong Le received the B.S. degree from Hanoi National University, Hanoi, Vietnam, in 2001 and the M.S. degree from Le Quy Don Technology University, Hanoi, Vietnam, in 2011. He is currently working towards the Ph.D. degree in the field of nonlinear signal processing at the Si-Chuan Province Key Lab of Signal and Information Processing, Southwest Jiaotong University, Chengdu, China. His current research interests include nonlinear adaptive signal processing, active noise control, and nonlinear signal processing for communication.

Jiashu Zhang received the B.S. degree from University of Electronic Science and Technology of China, Chengdu, China, in 1987, the M.S. degree from Chongqing University, Chongqing, China, in 1990, and the Ph.D. degree from University of Electronic Science and Technology of China, Chengdu, China, in 2001. In 2001, he joined the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China.

Currently, he is a professor and the director of Sichuan Province Key Lab of Signal and Information Processing, Southwest Jiaotong University, Chengdu, China. He has published more than 200 journal and conference papers. His research interests include nonlinear and adaptive signal processing, biometric security and privacy, and information forensic and data hiding.

Defang Li received the B.S. degree from Southwest University, Chongqing, China, in 1988. She joined the Psychological Research and Consulting Center, Southwest Jiaotong University, Chengdu, China, in 2001. Her current research interests include cognitive psychology and facial affective recognition