Applied Acoustics 156 (2019) 158–165

Contents lists available at ScienceDirect

Applied Acoustics

journal homepage: www.elsevier.com/locate/apacoust

M-max partial update leaky bilinear filter-error least mean square algorithm for nonlinear active noise control

Dinh Cong Le [a,b], Defang Li [a], Jiashu Zhang [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:
Received 2 March 2019
Received in revised form 14 June 2019
Accepted 1 July 2019

Keywords:
Active noise control
Filter-error algorithm
Partial update
Leaky algorithm
Bilinear FLANN

ABSTRACT

To reduce the computational burden of the bilinear FLANN (BFLANN) filter for active noise control (ANC), an M-max partial update leaky bilinear filtered-error least mean square (MmLBFE-LMS) algorithm is proposed in this paper. Unlike the algorithm based on filtered-reference technique in BFLANN-based ANC system, the proposed MmLBFE-LMS algorithm uses the filtered-error method and data-dependence partial update strategy to reduce computational complexity, and employs a leaky technique to mitigate the instability problem as in bilinear filters. The simulation results and computational complexity analysis indicate that the proposed algorithm can significantly reduce the computational burden of the BFLANN-based ANC system without suffering from noise-canceling performance degradation.

1. Introduction

Active noise control (ANC) has been recognized as one of the most effective solutions in noise reduction systems, especially in the low-frequency domain (less than or equal to 500 Hz). It, however, only really emerges when adaptive filtering technology develops and becomes popular. The ANC system using the linear filter with filtered-x least mean square (Fx-LMS) algorithm has exhibited superior performance over previous ANC systems [1,2]. During the past decades, many variants of the Fx-LMS algorithm have been developed for both the single channel and multi-channel ANC problems [3–5]. However, in actual ANC systems, the plant may be nonlinear, the reference noise may also be nonlinear, and the components of the ANC system may also be affected by the nonlinearity [6,7]. As a result, the ANC system using a linear controller may suffer from performance degradation when applied to real systems. To overcome this disadvantage, various ANC systems based on different nonlinear controllers have been proposed in the literature [7–10] (such as radial basis functions (RBF) [8], recurrent neural network (RNN) [9], multilayer artificial neural networks (MLANN) [10], truncated Volterra filter (VF) [7], Bilinear filter [11–14], Function link artificial neural networks (FLANN) [15–17], etc).

Among these nonlinear controllers, the FLANN-based system has been widely used because of low computational complexity, linear coefficients-output relationship and simple implementation structure. An FLANN-based ANC system trained using a filtered-s LMS (FsLMS) algorithm was first introduced by Das and Panda [15]. To increase the convergence rate, some methods have been presented in the literature including fast FsLMS [18] and normalized FsLMS algorithms [19]. With the objective to further improve the performance of nonlinear ANC systems, numerous modifiers of FLANN have been developed such as recursive FLANN [20], reduced feedback FLANN [21], and convex combination-based FLANN [22,23]. But because of the nature of FLANN, most of these methods provide relatively high residual noise power in the presence of strong nonlinearity.

As indicated in [24], the main reason for problem mentioned above is that the ANC system based on the FLANN cannot compensate for strong nonlinear distortions existing in the secondary path and the primary path. By introducing suitable cross-terms into a traditional FLANN structure, a generalized FLANN (GFLANN) has been presented for NANC system in [25]. The report revealed that the GFLANN performs better than FLANN and can be equivalent to the high-order Volterra filter, in the presence of strong nonlinearity. To improve the convergence performance of the GFLANN, a combination of the cross-terms and exponential nonlinearity has been developed by D.C Le *et al.* [26]. In a recent paper, bilinear FLANN (BFLANN) filter has been introduced for the NANC to overcome the strong nonlinear distortion in the system [27]. The BFLANN exploits cross-terms based on both feedback and feedforward polynomials, and thus it can accurately model nonlinear systems with shorter

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

https://doi.org/10.1016/j.apacoust.2019.07.006
0003-682X/© 2019 Elsevier Ltd. All rights reserved.

filter length than that of the GFLANN. However, introducing the cross-terms into the trigonometric functional expansion can improve noise-canceling performance, but this also increases the computational complexity of the system. Furthermore, since BFLANN using algorithm based on the filtered-reference method, the computational cost of filtering reference signal through secondary path is also a cause of increasing computational complexity, especially when the secondary path has nonlinear behavior. Besides, as in the bilinear filter case, the BFLANN-based ANC system suffers from an instability problem if the selected parameters for the algorithm are not carefully designed.

With the objective of reducing the computational burden and increasing the stability for the BFLANN-based ANC system, a novel algorithm called M-max partial update leaky bilinear filtered-error least mean square (MmLBFE-LMS) is proposed in this paper. Based on the filter-error method [24,28] and partial update strategy [29–32], the algorithm can avoid the computational cost of filtering reference signals through the secondary path and greatly reduce the cost to update the filtering weights. Moreover, the proposed algorithm uses leaky technique [33,34], so it is possible to mitigate the instability problem of BFLANN filter.

The rest of this paper is organized as follows. In Section 2, the MmLBFE-LMS algorithm is presented. Section 3 investigates the stability and the convergence condition of the algorithm. Comparison of the computational complexity of MmLBFE-LMS algorithm is also presented in Section 4. Section 5 provides computer simulation experiments of the proposed algorithm. Finally, conclusions are drawn in Section 6.

2. M-max partial update leaky bilinear filter error-least mean square (MmLBFE-LMS) algorithm

Many researchers have shown that filtered-error technique-based algorithms in [24,28] can be applied to nonlinear ANC system to reduce the computational burden. In contrast to the filtered-reference algorithm, the filtered-error algorithm only needs to filter error signals through the error filter, thus achieving considerable savings in computational requirements. In addition, the solution using partial update strategy to reduce computational complexity has been also studied and successfully applied in various engineering areas, as shown in the literature [29–32]. Among many possible partial update methods, the M-max partial update method is selected in this study due to its effectiveness and simplicity. Furthermore, as suggested in [11], the instability problem of bilinear filter can be mitigated by setting the step-size parameter carefully and employing the leaky adaptive algorithm.

Therefore, by combining the filtered-error method, M-max partial update strategy and leaky technique, a novel MmLBFE-LMS algorithm is proposed to reduce the computational complexity of the BFLANN controller-based ANC system, as illustrated in Fig. 1.

The relationship between input $x(n)$ and output $y(n)$ of the BFLANN structure with a memory length of N can be expressed as

$$\begin{aligned} y(n) = {} & \sum_{j=0}^{N-1} a_j(n)x(n-j) + \sum_{j=0}^{N-1} b1_j(n)\sin(\pi x(n-j)) \\ & + \sum_{j=0}^{N-1} b2_j(n)\cos(\pi x(n-j)) + \sum_{j=1}^{N-1} c_j(n)y(n-j) \\ & + \sum_{i=0}^{N-1}\sum_{j=1}^{N-1} d1_{i,j}(n)x(n-i)y(n-j) \\ & + \sum_{i=0}^{N-1}\sum_{j=1}^{N-1} d2_{i,j}(n)\sin(\pi x(n-i))y(n-j) \\ & + \sum_{i=0}^{N-1}\sum_{j=1}^{N-1} d3_{i,j}(n)\cos(\pi x(n-i))y(n-j) \end{aligned} \tag{1}$$

where $a_j(n)$, $b1_j(n)$ and $b2_j(n)$ are feedforward coefficients extended by FLANN with the order $P = 1$; $c_j(n)$ are feedback coefficients; $d1_{i,j}(n)$, $d2_{i,j}(n)$ and $d3_{i,j}(n)$ are the coefficients of cross-terms.

To derive the adaptive algorithm for BFLANN filter, the model in (1) is rewritten under the vector form as follows

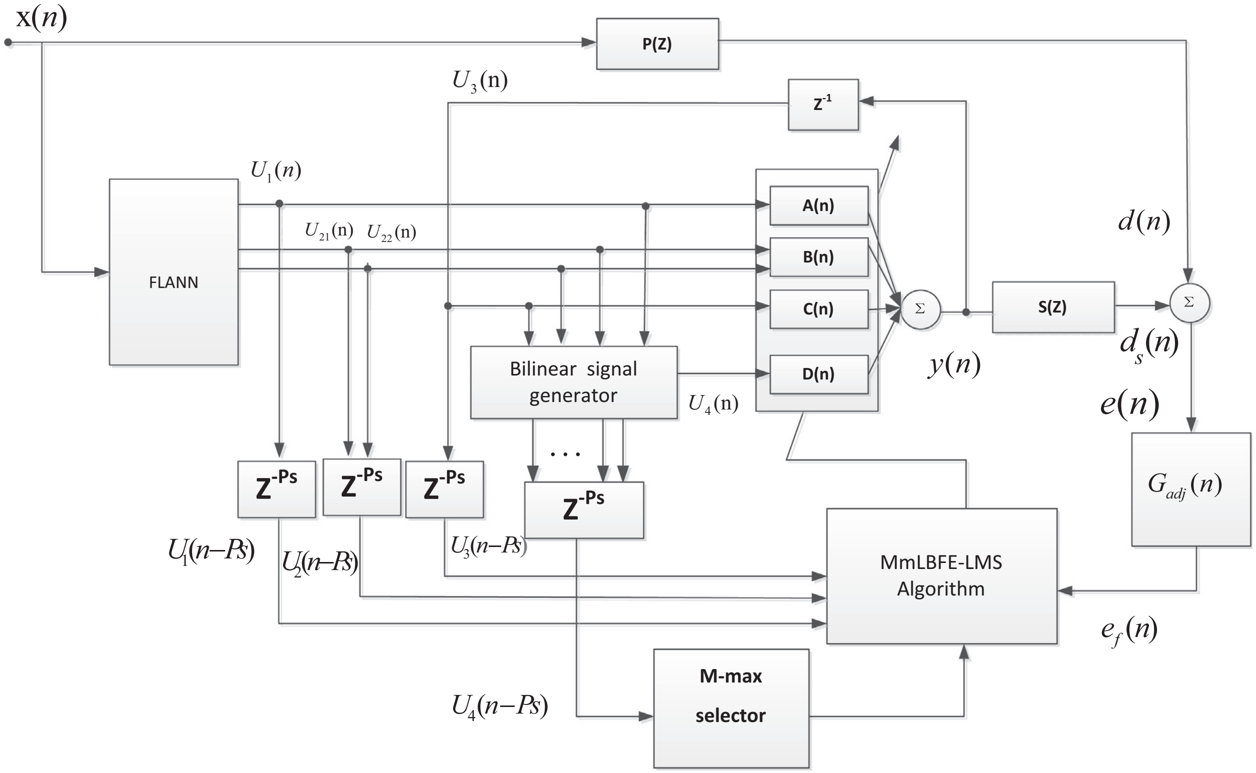


Fig. 1. The nonlinear BFLANN-based ANC system with MmLBFE-LMS algorithm.

$$y(n) = A^T(n)U_1(n) + B^T(n)U_2(n) + C^T(n)U_3(n) + D^T(n)U_4(n) \tag{2}$$

where signal vectors and their corresponding coefficient vectors are listed below

$$A(n) = [a_0(n)a_1(n)\cdots a_{N-1}(n)]^T \tag{3}$$

$$U_1(n) = [x(n)x(n-1)\cdots x(n-N+1)]^T \tag{4}$$

$$B(n) = [b1_0(n)b1_1(n)\cdots b1_{N-1}(n)b2_0(n)b2_1(n)\cdots b2_{N-1}(n)]^T \tag{5}$$

$$\begin{aligned} U_2(n) = {} & [\sin(\pi x(n))\sin(\pi x(n-1))\cdots\sin(\pi x(n-N+2))\sin(\pi x(n-N+1)) \\ & \cos(\pi x(n))\cos(\pi x(n-1))\cdots\cos(\pi x(n-N+2))\cos(\pi x(n-N+1))]^T \end{aligned} \tag{6}$$

$$C(n) = [c_1(n)c_2(n)\cdots c_{N-1}(n)]^T \tag{7}$$

$$U_3(n) = [y(n-1)y(n-2)\cdots y(n-N+1)]^T \tag{8}$$

$$\begin{aligned} D(n) = {} & [d1_{0,1}(n)\cdots d1_{0,N-1}(n)d1_{1,1}(n)\cdots d1_{1,N-1}(n)\cdots d1_{N-1,N-1}(n) \\ & d2_{0,1}(n)\cdots d2_{0,N-1}(n)d2_{1,1}(n)\cdots d2_{1,N-1}(n)\cdots d2_{N-1,N-1}(n) \\ & d3_{0,1}(n)\cdots d3_{0,N-1}(n)d3_{1,1}(n)\cdots d3_{1,N-1}(n)\cdots d3_{N-1,N-1}(n)]^T \end{aligned} \tag{9}$$

$$\begin{aligned} U_4(n) = {} & [x(n)y(n-1)\cdots x(n)y(n-N+1)x(n-1)y(n-1)\cdots x(n-N+1)y(n-N+1) \\ & \sin(\pi x(n))y(n-1)\cdots\sin(\pi x(n))y(n-N+1)\sin(\pi x(n-1))y(n-1)\cdots \\ & \cdots\sin(\pi x(n-1))y(n-N+1)\cdots\cdots\sin(\pi x(n-N+1))y(n-N+1) \\ & \cos(\pi x(n))y(n-1)\cdots\cos(\pi x(n))y(n-N+1)\cos(\pi x(n-1))y(n-1)\cdots \\ & \cdots\cos(\pi x(n-1))y(n-N+1)\cdots\cdots\cos(\pi x(n-N+1))y(n-N+1)]^T \end{aligned} \tag{10}$$

By combination of (3), (5), (7) and (9), we get the coefficient vector as an overall vector *W(n)* as follows

$$W(n) = \left[A^T(n)B^T(n)C^T(n)D^T(n)\right]^T \tag{11}$$

Similarly, we can combine (4), (6), (8) and (10) to generalize signal vector *U(n)* as follows

$$U(n) = \left[U_1^T(n)U_2^T(n)U_3^T(n)U_4^T(n)\right]^T \tag{12}$$

With the definitions in (11) and (12), the BFLANN filter output can be simplified to

$$y(n) = W^T(n)U(n) \tag{13}$$

In order to achieve a unified algorithm for both NANC/linear secondary path (LSP) and NANC/nonlinear secondary path (NSP), we use a concept of virtual secondary path$G(n)$, as in [24]. It is defined as a time-varying filter with coefficients as follows

$$\begin{aligned} G(n) &= [g(n,0)\,g(n,1)...g(n,P_s)]^T \\ &= \left[\frac{\partial d_s(n)}{\partial y(n)}\,\frac{\partial d_s(n)}{\partial y(n-1)}\cdots\frac{\partial d_s(n)}{\partial y(n-P_s)}\right]^T \end{aligned} \tag{14}$$

where P_s is the memory length of the virtual secondary path filter. To calculate the gradient instantaneous estimate, we define the residual noise which is sensed by the microphone error as follows (see Fig. 1).

$$e(n) = d(n) - d_s(n) = d(n) - G(n) * y(n) \tag{15}$$

where *d(n)* is primary noise signal at the cancellation point and $d_s(n)$ is the signal generated by the adaptive control and propagated through the secondary path to the cancellation point, and * is convolution operation.

Similar to the study [33], the cost function of the MmLBFE-LMS algorithm can be defined as

$$J(n) = E(e^2(n)) + \gamma W^T(n)W(n) \tag{16}$$

where γ << 1 denotes a positive number, $E(.)$ denotes the expectation operator, and $\| W(n) \|^2 = W^T(n)W(n)$

The objective of the MmLBFE-LMS algorithm for the BFLANN filter is to minimize the cost function $J(n)$ by using the steepest descent method as follows

$$W(n+1) = W(n) - \frac{1}{2}\mu\nabla_{W(n)}J(n) \tag{17}$$

where μ is the learning rate; $\nabla_{W(n)}J(n)$ is the gradient of cost function $J(n)$ with respect to the weight vector $W(n)$, which can be calculated by

$$\begin{aligned} \nabla_{W(n)}J(n) &= \frac{\partial J(n)}{\partial W(n)} = \frac{\partial E(e^2(n))}{\partial W(n)} + \gamma W(n) \cong \frac{\partial e^2(n)}{\partial W(n)} + \gamma W(n) \\ &= -2e(n)\frac{\partial d_s(n)}{\partial W(n)} + \gamma W(n) \end{aligned} \tag{18}$$

We can define partial derivatives in (18) as follows

$$\frac{\partial d_s(n)}{\partial W(n)} = \sum_{l=0}^{Ps}\frac{\partial d_s(n)}{\partial y(n-l)}\frac{\partial y(n-l)}{\partial W(n)} \tag{19}$$

Assuming that the step size is small, the coefficients $W(n)$ are slowly adjusted, and thus we have

$$\begin{aligned} & \frac{\partial y(n-l)}{\partial W(n)} \approx \frac{\partial y(n-l)}{\partial W(n-l)} \\ & = \left[\frac{\partial y(n-l)}{\partial a_0(n-l)}\cdots\frac{\partial y(n-l)}{\partial a_{N-1}(n-l)}\frac{\partial y(n-l)}{\partial b1_0(n-l)}\cdots\frac{\partial y(n-l)}{\partial b2_{N-1}(n-l)}\right. \\ & \frac{\partial y(n-l)}{\partial c_1(n-l)}\cdots\frac{\partial y(n-l)}{\partial c_{N-1}(n-l)}\frac{\partial y(n-l)}{\partial d1_{1,0}(n-l)}\cdots\frac{\partial y(n-l)}{\partial d1_{N-1,N-1}(n-l)} \\ & \left.\frac{\partial y(n-l)}{\partial d2_{1,0}(n-l)}\cdots\frac{\partial y(n-l)}{\partial d2_{N-1,N-1}(n-l)}\frac{\partial y(n-l)}{\partial d3_{1,0}(n-l)}\cdots\frac{\partial y(n-l)}{\partial d3_{N-1,N-1}(n-l)}\right]^T \end{aligned} \tag{20}$$

Similar to the bilinear Fx-LMS algorithm [11], we also assume that the recursion of the past output gradients is negligible, and thus the (20) is obtained as follows

$$\begin{aligned} & \tfrac{\partial y(n-l)}{\partial W(n)} \approx [x(n-l)...x(n-l-N+1)\sin(\pi x(n-l))...\sin(\pi x(n-l-N+1)) \\ & \cos(\pi x(n-l))...\cos(\pi x(n-l-N+1))y(n-l-1)...y \\ & (n-l-N+1)x(n-l)y(n-l-1)... \\ & x(n-l-N+1)y(n-l-N+1)\sin(\pi x(n-l))y(n-l-1)... \\ & \sin(\pi x(n-l-N+1))y(n-l-N+1) \\ & \cos(\pi x(n-l))y(n-l-1)...\cos(\pi x(n-l-N+1))y(n-l-N+1)]^T = U(n-l) \end{aligned} \tag{21}$$

Substituting (21),(19) and (18) into (17), we have

$$W(n+1) = (1-\mu\gamma)W(n) + \mu e(n)\sum_{l=0}^{Ps} g(n,l)U(n-l) \tag{22}$$

where $g(n,l)$is the $(l+1)$th component of the virtual secondary path coefficient vector$G(n)$.

Let *k = n-l + Ps*, so that *n = k + l-Ps*. Thus, the last term in (22),$e(n)\sum_{l=0}^{Ps}g(n,l)U(n-l)$, is expressed as

$$e(n)\sum_{l=0}^{Ps}g(n,l)U(n-l) = \left[\sum_{l=0}^{Ps}e(k+l-Ps)g(k+l-Ps,l)\right]U(k-Ps) \tag{23}$$

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

$$G_{adj}(n) = [g_{adj}(n,Ps)g_{adj}(n-1,Ps-1)...g_{adj}(n-Ps,0)]^T \tag{24}$$

We can see the relationship between the virtual secondary path filter $G(n)$ and the adjoint virtual secondary path filter $Gadj(n)$ as follows: $Gadj(n)$ not only reverses the orders of the coefficients of $G(n)$, but it also delay the time-varying coefficients.

Obviously, the term on the right-hand side of the Eq. (23), $\sum_{l=0}^{Ps} e(k+l-Ps)g(k+l-Ps,l)$, can be determined by filtering the error signal $e(n)$ through adjoint virtual secondary path filter $G_{adj}(n)$. Thus, we further define the filtered error as

$$\begin{aligned} e_f(n) &= \sum_{l=0}^{Ps} e(n+l-Ps)g(n+l-Ps,l) \\ &= \sum_{l=0}^{Ps} e(n-(Ps-l)g_{adj}(n,Ps-l) \end{aligned} \tag{25}$$

where $g_{adj}(n,l)$ is the $(l+1)$th component of the vector $G_{adj}(n)$. Consequently, by combining (25), (23) and (22) we yields

$$W(n+1) = (1-\mu\gamma)W(n) + \mu e_f(n)U(n-Ps) \tag{26}$$

To increase flexibility during the update process, the update equation in (26) can be divided into following sub-equations

$$A(n+1) = (1-\mu_a\gamma)A(n) + \mu_a U_1(n-Ps)e_f(n) \tag{27a}$$

$$B(n+1) = (1-\mu_b\gamma)B(n) + \mu_b U_2(n-Ps)e_f(n) \tag{27b}$$

$$C(n+1) = (1-\mu_c\gamma)C(n) + \mu_c U_3(n-Ps)e_f(n) \tag{27c}$$

$$D(n+1) = (1-\mu_d\gamma)D(n) + \mu_d U_4(n-Ps)e_f(n) \tag{27d}$$

As discussed above, the coefficient vector of the BFLANN filter consists of the linear coefficients vector $A(n)$, the sinusoidal coefficients vector $B(n)$, the feedback coefficients vector $C(n)$ and the cross-term coefficients vector $D(n)$. Among these coefficient vectors, the cross-terms coefficient vector $D(n)$ is the largest. For example, when the memory length of external input is $N = 7$, the lengths of vectors $A(n)$, $B(n)$, $C(n)$, and $D(n)$ are respectively 7, 14, 6 and 126. Therefore, in order to match BFLANN's reality, we only apply the partial update technique for the update equation of the cross-terms part, as illustrated in Fig. 1. In this way, only a fraction of the total weights of the cross-terms is updated during every iterations. Therefore, the update equations of the MmLBFE-LMS algorithm is composed of the equations 27 (a-c) and the following equation

$$D(n+1) = (1-\mu_d\gamma)D(n) + \mu_d\Phi(n)U_4(n-Ps)e_f(n) \tag{28}$$

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

$$\Phi(n) = \begin{bmatrix} \lambda_1(n) & 0 & \cdots & 0 \\ 0 & \lambda_2(n) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_{L_{U4}}(n) \end{bmatrix} \tag{29}$$

with

$$\lambda_j(n) = \begin{cases} 1 & if\,|u4_j(n)| \in \max\limits_{1\leqslant k\leqslant L_{U4}}(|u4_k(n)|, M) \\ 0 & \qquad otherwise \end{cases} \tag{30}$$

where $L_{U4} = 3\,N(N-1)$ is the length of the cross-terms coefficients $D(n)$; $u4_j(n)$ is the jth element of the cross-terms signal vector $U_4(n)$ in (10); and $1 \leq M \leq L_{U4}$ is the pre-selected M-max parameter. The parameters M need to appropriately choose to achieve a good compromise between computational complexity and performance.

3. Stability of the MmLBFE-LMS algorithm

From (16), we find that the cost function of the MmLBFE-LMS algorithm uses the penalty term $\gamma\| W(n) \|^2$ which ensures that $W(n)$ will reach the minimum value of $e^2(n)$ and at the same time prevents the terms $\| W(n) \|^2$ from becoming too large. More specifically, in equations (27a-d), the weights $A(n)$, $B(n)$, $C(n)$, and $D(n)$ are not only updated by the gradients related $U_1(n\text{-}Ps)e_f(n)$, $U_2(n\text{-}Ps)e_f(n)$, $U_3(n\text{-}Ps)e_f(n)$, and $U_4(n\text{-}Ps)e_f(n)$, but also decays to a certain extent by multiplying the term $0 < (1\text{-}\mu\gamma) < 1$ at each iteration. Obviously, this can avoid the problem of instability when the weights become too large.

In addition, to ensure the stability of the algorithm, the condition for the step-size parameter μ should also be considered.

We first assume that the autocorrelation matrix $R = E\{U_f(n)U_f^T(n)\}$, where $U_f(n)$ is the input signal filtered through the estimate of the secondary path, which is derived from (19). To get a condition of the step-size parameter μ, we analyze the symmetric matrix R into$R = Q\Lambda Q^T$, where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_n)$ is the matrix of the eigenvalue, and Q is matrix of the eigenvector of R. We then define the parameter error vector as $V(n) = W(n) - W_0$, where W_0 is the optimal weight, and define the vectors rotated as

$$V'(n) = Q^T V(n), \quad U'_f(n) = Q^T U_f(n), \quad W'_0 = Q^T W_0 \tag{31}$$

For the convenience of analysis, we rewrite the weights update equation (26) as

$$W(n+1) = (1-\mu\gamma)W(n) + \mu e(n)U_f(n) \tag{32}$$

Using the assumptions defined above and the relation $d(n) = Uf^T(n)W0+e0(n)$ (where $e0(n)$ is the measurement noise), (32) can be expressed as follows

$$V'(n+1) = [I-\mu(\gamma I+\Lambda)]V'(n) - \mu\gamma W'_0 + \mu e_0(n)U'_f(n) \tag{33}$$

Taking the expected value of (33) with the independence assumption of $V'(n)$ and $U'_f(n)$, and using the relation $E\{U_f(n) e_0(n)\} = \gamma W_0$ obtained by minimizing the expected value of (16) with respect to $W_0 = E\{W(n)\}$, (33) can be rewritten as

$$E\{V'(n+1)\} = [I-\mu(\gamma I+\Lambda)]E\{V'(n)\} \tag{34}$$

Clearly, from (34), the boundedness of the step-size μ is guaranteed by the following condition

$$0 < \mu < \frac{2}{\gamma + \lambda_{\max}} \tag{35}$$

where $\lambda_{\max}$ denotes the largest eigenvalue of R. Following the adaptive filter theory [35], we get$\lambda_{\max} < tr(R) = \sum_{i=1}^N \lambda_i = N\delta_x^2$, where δ_x^2 is input signal power. Therefore, a practical bound on the step-size μ can be found from (35) as

$$0 < \mu < \frac{2}{N\delta_x^2 + \gamma} \tag{36}$$

4. Computational complexity analysis

For the convenience of analysis, we name the algorithms which were proposed for BFLANN in the study [27] and for the GFLANN in the study [25] are bilinear Fs-LMS (BFs-LMS) and generalized Fs-LMS (GFs-LMS), respectively.

In this section we would conduct the computational complexity analysis of the BFs-LMS, GFs-LMS, Fs-LMS, and proposed MmLBFE-LMS algorithms in terms of the number of multiplications and additions for both the linear secondary path (LSP) and the nonlinear secondary path (NSP). Assume that N and M is the memory length and the pre-selected M-max parameter of the MmLBFE-

LMS, respectively; N_g and N_d are the memory length and cross-term selection parameter of the GFs-LMS; N_f and B are the memory length and the order of expansion function of the FLANN; Ps is the memory size of the virtual secondary path; L_b is the expanded memory length of BFLANN.

4.1. Computational complexity for NANC/LSP

The MmLBFE-LMS algorithm requires major operations as follows.

Step 1: The operations for generating the cross-terms require 6 (N-1) multiplications.

Step 2: The operations for generating the adaptive BFLANN filter output, which require L multiplications and L-1 additions, where $L = 3N$+(N-1) + 3N(N-1) denote the expanded memory length.

Step 3: Calculate $e_f(n)$, the filtered error of the new virtual secondary path filter, which require Ps multiplications and P_s-1 additions

Step 4: The operations for updating filter coefficients requires $M + 4N$-1 multiplications and $M + 4N$-2 additions.

Similarly, we can also obtain the computational complexity of GFs-LMS, Fs-MS and BFs-LMS for NANC/LSP case. Table 1 summarizes the total computational load of the NANC/LSP with the proposed MmLBFE-LMS, BFs-LMS [27], GFs-LMS [25] and Fs-LMS [15] algorithm.

4.2. Computational complexity for NANC/NSP

When the secondary path is nonlinear, the virtual secondary filter is time-varying, and thus we cannot take advantage of the delay relationship existing in the nonlinear states. For the BFs-LMS, GFs-LMS and Fs-MS algorithms, we need to calculate the cost of filtering each element through the virtual secondary path. For the proposed MmLBFE-LMS algorithm, computational complexity does not depend on the secondary path whether nonlinear or linear. Hence, the total computational load of the MmLBFE-LMS, BFs-LMS [27], GFs-LMS [25] and Fs-LMS [15] algorithms for the NANC/NSP case is summarized as in Table 2.

5. Computer simulation

In order to prove the effectiveness of our proposed algorithm for the BFLANN-based ANC system, several simulation results are provided in this section. The performance of the MmLBFE-LMS algorithm is compared with the Fs-LMS, BFs-LMS and GFs-LMS algorithms for both NANC/LSP and NANC/NSP cases. In the experiments, the performance of the different filters will be measured in terms of the normalized mean-square error (NMSE) which is obtained by averaging over 100 independent runs.

$$NMSE = 10\log10\left(E(e^2(n))/\delta_d^2\right) \tag{37}$$

where δ_d^2 is the variance of the primary noise at the cancellation point, $e^2(n)$ is the square of the error at nth iteration and $E(.)$ is the expectation operator.

The memory length of the external input signal of Fs-LMS GFs-LMS, BFs-LMS and MmLBFE-LMS algorithms are chosen as $N_f = 10$, $N_g = 10$, $N_b = 6$, and $N = 6$, respectively. The function expansion of the input signal is third-order type ($B = 3$) for the Fs-LMS and first-order type ($B = 1$) for GFs-LMS, BFs-LMS and MmLBFE-LMS. The parameter for expanding nonlinear function of GFs-LMS is chosen as $N_d = 9$. The pre-selected M-max parameter is chosen as $M = 30$. The ensemble curves are smoothed with a rectangular window of length equal to 20 samples, in order to better discern the curves behavior.

Experiment 1: In this case, we simulate NANC system with the nonlinear secondary path and the primary path exhibits high nonlinear behavior. Here, the primary path $P(z)$ is modeled by a Volterra series whose input $x(n)$ and output $d(n)$ relationship is described as

$$\begin{aligned} d(n) = {} & x(n) + 0.8x(n-1) + 0.3x(n-2) + 0.4x(n-3) \\ & -0.8x(n)x(n-1) + 0.9x(n)x(n-2) \\ & +0.7x(n-3)x(n-3) - 3.9x^2(n-1)x(n-2) \\ & -2.6x^2(n-1)x(n-3) + 2.1x^2(n-2)x(n-3) \end{aligned} \tag{38}$$

and the secondary path has the input $y(n)$ to output $d_s(n)$ relationship

$$\begin{aligned} d_s(n) = {} & y(n) + 0.35y(n-1) + 0.09y(n-2) - 0.5y(n)y(n-1) \\ & +0.4y(n)y(n-2) \end{aligned} \tag{39}$$

The reference signal is white Gaussian noise. The learning rate of Fs-LMS is set to $\eta_f1 = 0.01$, $\eta_f2 = 0.0005$ for the linear and nonlinear parts, respectively. Learning rate of GFs-LMS are $\eta_g1 = 0.6$, $\eta_g2 = 0.0035$ and $\eta_g3 = 0.07$ for the linear, the sin(.) cos (.) functions and the cross-terms parts, respectively. The learning rate of the BFs-LMS algorithm are set $\eta_b1 = 0.15$, $\eta_b2 = 0.005$ for

Table 1
Total computational requirement for NANC/LSP case.

Algorithms for ANC	Multiplications	additions
Fs-LMS	$14L_f + 7P_s + 1$	$14L_f + 7(P_s\text{-}1)\text{-}1$
GFs-LMS	$(3 + 2N_d)Ps$ $+ 2N_d^2 + 6N_g + 4N_d + 1$	$2N_d^2 + 6N_g + 2N_d + (2N_d + 3)$ $(Ps\text{-}1)\text{-}1$
BFs-LMS	$2L_b + [6(N_b\text{-}1) + 4]Ps + 6$ $(N_b\text{-}1) + 1$	$2L_b + [6(N_b\text{-}1) + 4](Ps\text{-}1)\text{-}2$
MmLBFE-LMS	L + 6(N-1) + Ps + M + 4N	L + Ps + M + 4 N-2

Table 2
Total computational requirement for NANC/NSP case.

Algorithms for ANC	Multiplications	additions
Fs-LMS	$14L_f + 7L_fP_s + 1$	$14L_f + 7L_f(P_s\text{-}1)\text{-}1$
GFs-LMS	$[N_d(N_d + 1) + 3N_g](P_s + 2)$ $+ 2N_d + 1$	$[N_d(N_d + 1) + 3N_g]$ $(P_s + 1)\text{-}1$
BFs-LMS	$2L_b + L_bPs + 6(N_b\text{-}1) + 1$	$2L_b + L_b(Ps\text{-}1)\text{-}1$
MmLBFE-LMS	L + 6(N-1) + Ps + M + 4N	L + Ps + M + 4N-2

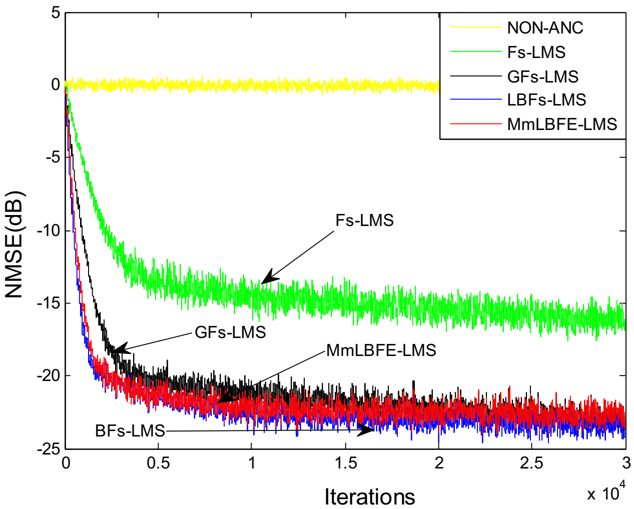


Fig. 2. Learning curves of different algorithms for NANC/NSP, the secondary path is the Volterra model.

the feedforward coefficients respectively, $\eta_b3 = 0.01$ and $\eta_b4 = 0.08$ for the feedback coefficient and the cross-term coefficients, respectively. The learning rate of the MmBLFE-LMS algorithms are set $\mu_a = 0.1$, $\mu_b = 0.003$, $\mu_c = 0.025$ and $\mu_d = 0.1$.

Fig. 2 illustrates the averaged NMSE performance curves of the algorithms for the reference signal is white Gaussian noise.

To compare computational complexity, in this section we assume that the length of the secondary path increases from 3 to 10. Hence, the number of additions and multiplications required in each of the algorithm is plotted in Fig. 3 as a function of Ps. From Figs. 2 and 3, it is clear that the proposed MmLBFE-LMS algorithm has the lowest computational complexity with a better performance than that of GFs-LMS and the equivalent of BFs-LMS algorithm.

Experiment 2: In practical NANC system, the secondary path can be seen as the nonlinear effect of the power amplifier and loudspeaker at the output of the filter, and thus it can be modeled by a Hammerstein filter with a memoryless nonlinearity and the linear filter as

$$\begin{aligned} v(n) &= \tanh(y(n)) \\ d_s(n) &= v(n) + 0.2v(n-1) + 0.05v(n-2) \end{aligned} \tag{40}$$

The reference signal is the noise generated by a fan, and its mathematical model can be approximately described with a logistic chaotic signal as follows

$$x(n+1) = \lambda x(n-1)[1 - x(n-1)] \tag{41}$$

where $\lambda = 4$ and $x(0) = x(1) = 0.9$ are selected. For simplicity, we assume that the amplitude of the reference signal is limited to the range (-0.4 to 0.4) and the length of the signal is equal to 60,000 samples. In this example, the two cases of the primary path have been selected.

Case 1: The primary path is assumed to be exhibiting high nonlinear behavior and modeled similar to the example 1. The learning rate for all the filters are chosen as: the Fs-LMS ($\eta_f1 = 0.002$, $\eta_f2 = 0.0001$); the GFs-LMS ($\eta_g1 = 0.006$, $\eta_g2 = 0.0005$; $\eta_g3 = 0.006$); the BFs-LMS ($\eta_b1 = 0.005$, $\eta_b2 = 0.0004$; $\eta_b3 = 0.005$ and $\eta_b4 = 0.007$); the MmLBFE-LMS ($\mu_a = 0.003$, $\mu_b = 0.001$, $\mu_c = 0.002$ and $\mu_d = 0.004$).

Case 2: The primary path is linear and taken as following model $P(z) = z^{-3}\text{-}0.3z^{-4} + 0.2z^{-5}$. The learning rate for all the filters are chosen as: the Fs-LMS ($\eta_f1 = 0.4$, $\eta_f2 = 0.003$); the GFs-LMS ($\eta_g1 = 0.035$, $\eta_g2 = 0.003$; $\eta_g3 = 0.1$); the BFs-LMS ($\eta_b1 = 0.45$, $\eta_b2 = 0.01$; $\eta_b3 = 0.02$ and $\eta_b4 = 0.01$); the MmLBFE-LMS ($\mu_a = 0.45$, $\mu_b = 0.01$, $\mu_c = 0.2$ and $\mu_d = 0.1$).

Table 3 shows the results of the noise attenuation at the cancellation point of the controllers for both cases. Here, the noise attenuation is the ratio between the mean noise power values without and with ANC. Note that the mean noise power of the controllers are measured in the last 10,000 samples and selected with the same convergence rate.

It is clear that the proposed MmLBFE-LMS algorithm achieves the noise attenuations equivalent to BFs-LMS and better the GFs-LMS and Fs-LMS algorithms in the case of the primary path exhibits a high nonlinear behavior. In the case of the linear primary path, the noise attenuation at the cancellation point of the proposed MmLBFE-LMS algorithm does not improve compared to BFs-LMS and GFs-LMS, however it can offer reduced computational complexity.

Experiment 3: In this experiment, we assume that the behavior of the primary noise at the cancellation point is modeled as a third-order polynomial as follows [7]

$$d(n) = u(n-2) + 0.8u^2(n-2) - 0.4u^3(n-2) \tag{42}$$

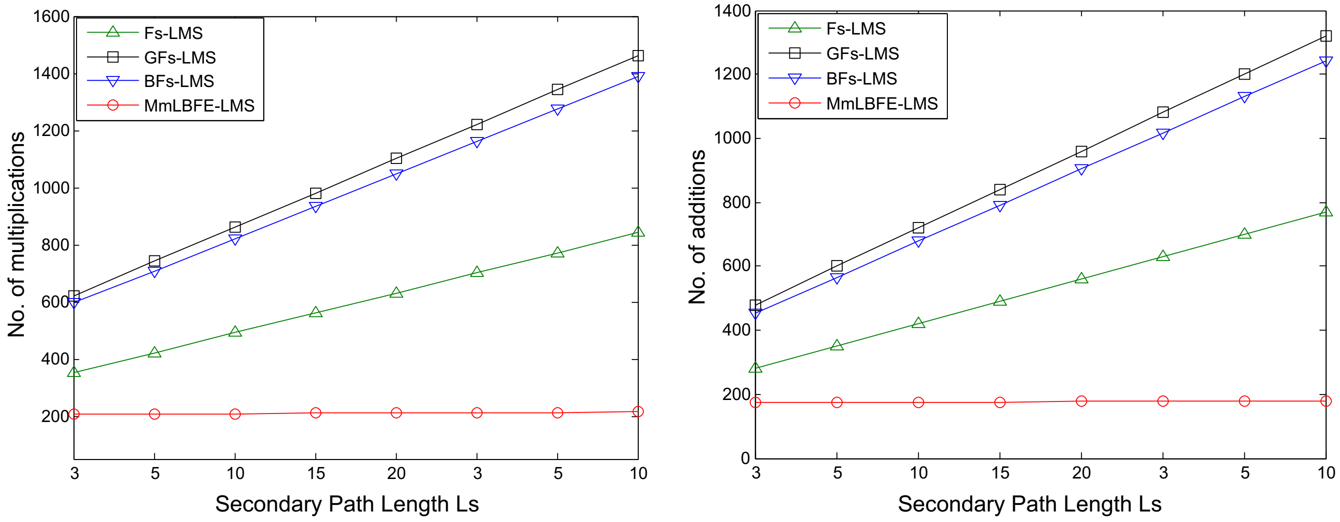
where $u(n)$ is obtained by $u(n) = x(n)*f(n)$ and * denotes the convolution operation, $f(n)$ is the impulsive response of the transfer function as

$$f(z) = z^{-3} - 0.3z^{-4} + 0.2z^{-5} \tag{43}$$

The reference noise $x(n)$ is a sinusoidal wave of 500 Hz sampled at the rate of 8000 samples/s, which is obtained by

Table 3
Noise attenuation (dB) for experiment 2.

Controllers for NANC system	Noise attenuation (dB) for case 1	Noise attenuation (dB) for case 2
Fs-LMS	14.8907	35.4191
GFs-LMS	20.6545	40.3898
BFs-LMS	21.7961	40.4678
MmLBFE-L MS	21.7709	40.3114



(a) (b)

Fig. 3. Comparison of computational load per sample for algorithms (a) multiplications and (b) additions.

$$x(n) = \sin\left(\frac{2\pi 500n}{8000}\right) + v(n) \tag{44}$$

where $v(n)$ denotes a Gaussian noise of 40 dB SNR. The secondary path transfer function is taken as the non-minimum-phase model

$$S(z) = z^{-2} + 1.5z^{-3} - z^{-4} \tag{45}$$

The learning rate for all the filters are chosen as: the Fs-LMS ($\eta_f1 = 0.0045$, $\eta_f2 = 0.0022$); the GFs-LMS ($\eta_g1 = 0.004$, $\eta_g2 = 0.002$; $\eta_g3 = 0.0007$); the BFs-LMS ($\eta_b1 = 0.006$, $\eta_b2 = 0.0025$; $\eta_b3 = 0.008$ and $\eta_b4 = 0.005$); the MmLBFE-LMS ($\mu_a = 0.006$, $\mu_b = 0.0025$, $\mu_c = 0.008$ and $\mu_d = 0.006$). Fig. 4 depicts the performance comparison of NMSE for algorithms for NANC/LSP case. It is clear that the proposed MmLBFE-LMS algorithm exhibits a NMSE equivalent to that of the BFs-LMS algorithm and much better than the GFs-LMS and Fs-LMS algorithms. In addition, results obtained from Table 4 show that the MmLBFE-LMS algorithm requires less computational complexity in comparison with the BFs-LMS and GFs-LMS algorithms.

Experiment 4: To evaluate the tracking capability of the proposed algorithm, in this experiment we use the primary path, secondary path and reference signal similar to experiments 3. Fig. 5 compares algorithms in a tracking situation when after 4000 sample, the primary path response is shifted to the right by 12 samples and the reference signal is abruptly increased by a Gaussian noise of 35 dB SNR. According to this simulation, the proposed MmLBFE-LMS algorithm tracks as fast as the competitive algorithms

Experiment 5: With the aim to further test the effectiveness of the proposed algorithm to real-world situations, in this experiment, we adopt the real measured secondary and primary path which was set by Kuo's experiment as in [1]. The amplitude response and phase response of such primary path and secondary path are shown as in Fig. 6.

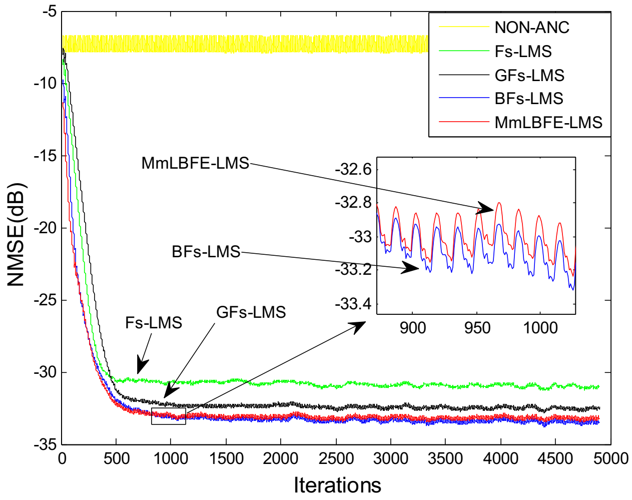


Fig. 4. Learning curves of different algorithms for NANC/LSP, the reference signal is a sinusoidal wav, the secondary path is chosen as the non-minimum-phase model, the primary path is nonlinear.

Table 4
Computational requirements for NANC/LSP case.

Algorithms	Multiplications	Additions
Fs-LMS	176	167
GFs-LMS	364	323
BFs-LMS	424	360
MmLBFE-LMS	199	177

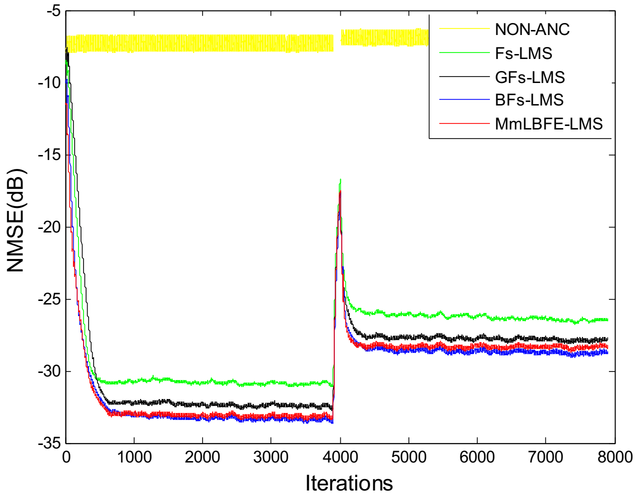


Fig. 5. Tracking performance of algorithms.

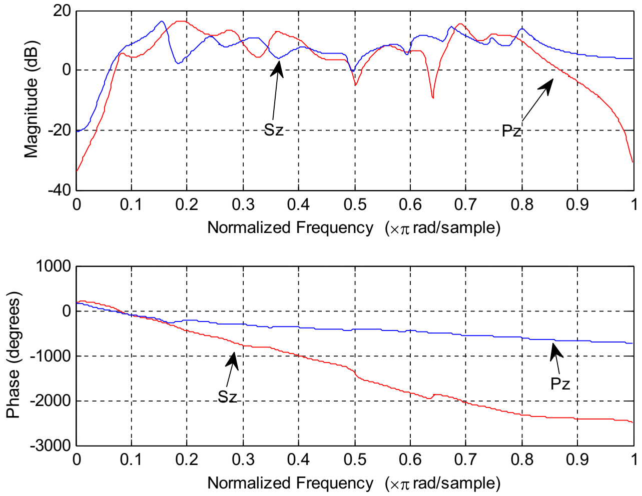


Fig. 6. The amplitude response and phase response for the measured primary path and secondary path.

The reference input signal is a mixture of three sine signals with normalized frequencies of 0.02, 0.04, 0.08, and normalized to have a unit power. The SNR at the noise cancelling point is set to 40 dB. The reference signal is then assumed to be strongly distorted by the nonlinear function described as follows

$$f[x(n)] = 0.2x(n)x(n-1) + 0.8x(n) \tag{46}$$

The learning rate for all the filters are chosen as: the Fs-LMS ($\eta_f1 = 0.0000035$, $\eta_f2 = 0.0000022$); the GFs-LMS ($\eta_g1 = 0.0000075$, $\eta_g2 = 0.000001$; $\eta_g3 = 0.000009$); the BFs-LMS ($\eta_b1 = 0.000035$, $\eta_b2 = 0.0000014$; $\eta_b3 = 0.00007$ and $\eta_b4 = 0.000012$); the MmLBFE-LMS ($\mu_a = 0.000025$, $\mu_b = 0.0000015$, $\mu_c = 0.000055$ and $\mu_d = 0.000018$). Fig. 7 shows the performance comparison of NMSE for algorithms under the real measured secondary and primary paths. It is apparent that there is a significant improvement of the performance in both BFs-LMS and proposed MmLBFE-LMS algorithms compared to Fs-LMS and GFS-LMS algorithms. However, the proposed MmLBFE-LMS algorithm requires lower computational costs than the BFs-LMS algorithm.

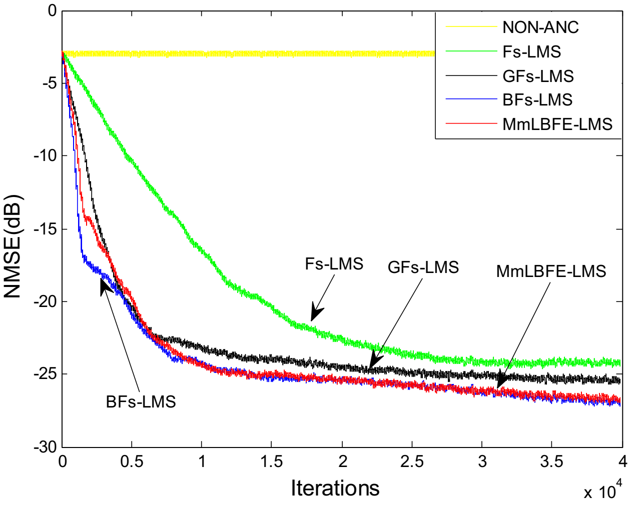


Fig. 7. The performance comparison of NMSE for algorithms under the real secondary and primary paths.

6. Conclusion

This paper presents a novel MmLBFE-LMS algorithm to reduce the computational complexity of the BFLANN-based ANC system. The proposed algorithm exploits the superiority of the filtered-error method in the active noise control and uses a data-dependence partial update strategy to optimally exploit the cross-terms, which can reduce the computational cost of BFLANN. Moreover, based on leaky technique, this algorithm can limit the gain of filtering weights, and thus can mitigate the instability problem of BFLANN-based ANC system. Stability and computational complexity of the MmLBFE-LMS algorithm is analyzed. Simulation results demonstrated that the proposed solution can significantly reduce the computational cost of the BFs-LMS algorithm without degrading noise-canceling performance. In conclusion, the MmLBFE-LMS algorithm can be considered a suitable candidate for the BFLANN-based ANC system.

Role of the funding source

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

References

[1] Kuo SM, Morgan DR. Active Noise Control Systems: Algorithms and DSP Implementations. New York: Wiley; 1996.
[2] Elliott SJ, Nelson PA. Active noise control. Signal Process Mag IEEE 1993;10(4):12–35.
[3] Das DP, Panda G, Kuo SM. New block filtered-X LMS algorithms for active noise control systems. IET Signal Process 2007;1(2):73–81.
[4] Bouchard M. Multichannel affine and fast affine projection algorithms for active noise control and acoustic equalization systems. IEEE Trans Speech Audio Process 2003;11(1):54–60.
[5] Kim HW, Park HS, Lee SK, Shin K. Modified-filtered-u LMS algorithm for active noise control and its application to a short acoustic duct. Mech Syst Sig Process 2011;25(1):475–84.
[6] Kuo SM, Wu HT, Chen FK, Gunnala MR. Saturation effects in active noise control systems. IEEE Trans Circuits Syst I Regul Pap 2004;51(6):1163–71.
[7] Tan L, Jiang J. Adaptive volterra filter for active control of nonlinear noise processes. IEEE Trans Signal Process 2001;49(8):1667–76.
[8] Tokhi MO, Wood R. Active noise control using radial basis function networks. Control Eng Pract 1997;5(9):1311–22.
[9] Zhang QZ, Gan WS, Zhou YL. Adaptive recurrent fuzzy neural networks for active noise control. J Sound Vib 2006;296(4–5):935–48.
[10] Chang CY, Luo FB. Enhancement of active noise control using neural-based filtered-X algorithm. J Sound Vib 2007;305(1–2):348–56.
[11] Kuo SM, Wu HT. Nonlinear adaptive bilinear filters for active noise control systems. IEEE Trans Circuits Syst I Regul Pap 2005;52(3):617–24.
[12] Zhang JS, Zhao HQ. A novel adaptive bilinear filter based on pipelined architecture. Digit Signal Process 2010;20(1):23–38.
[13] Tan L, Dong C, Du S. On implementation of adaptive bilinearfilters for nonlinear active noise control. Appl Acoust 2016;106:122–8.
[14] Zhao H, Zeng X, He Z, Li T, Jin W. Nonlinear adaptivefilter-based simplified bilinear model for multichannel active control of nonlinear noise processes. Appl Acoust 2013;74(12):1414–21.
[15] Das PD, Panda G. Active mitigation of nonlinear noise processes using a novel filtered-s LMS algorithm. IEEE Trans Speech Audio Process 2004;12(3):313–22.
[16] Das DP, Mohapatra SR, Routray A, Basu TK. Filtered-s LMS algorithm for multichannel active control of nonlinear noise processes. IEEE Trans Audio Speech Lang Process 2006;14(5):1875–80.
[17] Luo L, Bai Z, Zhu W, Sun J. Improved functional link artificial neural networkfilters for nonlinear active noise control. Appl Acoust 2018;135:111–23.
[18] Reddy EP, Das DP, Prabhu KMM. Fast exact multichannel FSLMS algorithm for active noise control. Signal Process 2009;89(5):952–6.
[19] Vikram Kumar P, Prabhu KMM, Das DP. Block filtered-s least mean square algorithm for active control of non-linear noise systems. IET Signal Proc 2010;4(2):168–80.
[20] Sicuranza GL, Carini A. On the BIBO stability condition of adaptive recursive FLANN filters with application to nonlinear active noise control. IEEE Trans Audio Speech Language Process 2012;20(1):234–45.
[21] Zhao H, Zeng X, Zhang J. Adaptive reduced feedback FLNN filter for active control of nonlinear noise processes. Signal Process 2010;90(3):834–47.
[22] George NV. Gonzalez A, Convex combination of nonlinear adaptive filters for active noise control. Appl Acoust 2014;76:157–61.
[23] George NV, Panda G. Active control of nonlinear noise processes using cascaded adaptive nonlinearfilter. Appl Acoust 2014;74:217–22.
[24] Zhou D, DeBrunner V. Efficient adaptive nonlinear filters for nonlinear active noise control. IEEE Trans Circuits Syst–I: Regular Pap 2007;54(3):669–81.
[25] Sicuranza GL, Carini A. A generalized FLANN filter for nonlinear active noise control. IEEE Trans Audio Speech Lang Process 2011;19(8):2412–7.
[26] Le DC, Zhang JS, Li DF, Zhang S. A generalized exponential functional link artificial neural networks filter with channel-reduced diagonal structure for nonlinear active noise control. Appl Acoust 2018;139:174–81.
[27] Le DC, Zhang JS, Pang YJ. A bilinear functional link artificial neural network filter for nonlinear active noise control and its stability condition. Appl Acoust 2018;132:19–25.
[28] Miyagi S, Sakai H. Mean-square performance of the filtered-reference/filtered-error LMS algorithm. IEEE Trans Circuits Syst–I: Regular pap 2005;52(11):2454–63.
[29] K. Dogancay, Partial-update adaptive signal processing: Design Analysis and Implementation, 2008
[30] Y. Chien, W. Tseng, a new variable step-size method for the M-max LMS algorithms, IEEE international conference on consumer electronics-Taiwan, Taipei, 2014, 21–22
[31] Zhang S, Zhang J. Set-membership NLMS algorithm with robust error bound. IEEE Trans Circuits Syst II 2014;61(7):1–5.
[32] Zhang S, Zhang JS, Pang YJ. Pipelined set-membership approach to adaptive Volterra filtering. Signal Process 2016;129:195–203.
[33] Tobias O, Seara R. Leaky-FXLMS algorithm: stochastic analysis for Gaussian data and secondary path modeling error [J]. IEEE Trans Speech Audio Process 2005;13(6):1217–30.
[34] Wu LF, Qiu XJ. Guo Y, A generalized leaky FxLMS algorithm for tuning the waterbed effect of feedback active noise control systems. Mech Syst Signal Process 2018;106:13–23.
[35] Haykin S. Adaptive Filter Theory. 4th ed. Englewood Cliffs, NJ: Prentice-Hall; 2002.