Applied Acoustics 106 (2016) 122–128

Contents lists available at ScienceDirect

Applied Acoustics

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

Technical Note

On implementation of adaptive bilinear filters for nonlinear active noise control

Li Tan [a,*], Chen Dong [b], Sidan Du [b]

[a] Department of Engineering, Purdue University North Central, Westville, IN 46391,USA
[b] School of Electronic Science and Engineering, Nanjing University, Nanjing, 210046, China

ARTICLE INFO

Article history:
Received 19 August 2015
Received in revised form 8 November 2015
Accepted 4 January 2016
Available online 19 January 2016

Keywords:
Active noise control (ANC)
Adaptive filter
Adaptive bilinear filter
Diagonal structure
Least mean square algorithm

ABSTRACT

In this technical note, the simplified diagonal-structure bilinear filtered-X least mean square (SDBFXLMS) and channel-reduced diagonal-structure bilinear filtered-X least mean square (CRDBFXLMS) algorithms are proposed. Computational complexity for each proposed algorithm is analyzed to show the significant computational reduction in comparison with the diagonal-structure bilinear FXLMS (DBFXLMS) algorithm. For $L = 15$ (memory length of the bilinear filter), $P = 2$ (the corresponding number of the diagonal channels for the SDBFXLMS algorithm is $L + 2P = 19$ and the corresponding number of the diagonal channels for the CRDBFXLMS algorithm is $2P = 4$), and $M = 64$ (memory length of the secondary path estimate), the SDBFXLMS algorithm achieves 45% and 40% reduction of multiplications and additions, respectively, while the CRDBFXLMS algorithm acquires 78% reduction of multiplications and 76% reduction of additions. Computer simulations validate the satisfied control performances of the proposed algorithms.

1. Introduction

Active noise control (ANC) applies the principle of superposition of the primary noise source and the secondary source [1–6]. The secondary source consists of acoustic output which has the same amplitude and opposite phase of the primary noise source so that a quiet zone can be produced at the canceling point. It is well known that linear controllers can successfully be adopted to reduce the low-frequency noise. However, a linear ANC system will suffer from performance degradation if the following conditions exist: a noise process to be controlled is a nonlinear and deterministic process such as chaotic noise; the primary as well as secondary paths have the nonlinear behavior; the reference signal may have the saturation effect [7–13]. In order to tackle various nonlinear effects, many nonlinear controllers such as the Volterra filtered-X least mean square (VFXLMS), filtered-s LMS (FSLMS), and bilinear filtered-X LMS (BFXLMS) algorithms have been developed [7–15]. The other new nonlinear controllers for nonlinear active noise control may include the generalized functional link artificial neural network (FLANN) filters, recursive FLANN filters, combination of the FLANN and Legendre neural network (LeNN) filters, and convex combination of FLANN and VFXLMS filters [16–20]. Based on the coefficient reduction strategy used in [21], a simplified BFXLMS (SBFXLMS) algorithm is recently proposed in [22] in order to reduce the number of filter coefficients in the multi-channel NANC system [23].

The implementations of the bilinear filtered algorithms by Kuo and Wu [14], and Zhao et al. [22] have shown the improved control performance over the VFXLMS and FSLMS algorithms [10,12]. However, the diagonal structure feature for the cross terms of delayed input signal and delayed output signal in the bilinear model was not considered in [14,22] when performing computational reduction for the filtered signal elements. Hence, both BFXLMS and SBFXLMS algorithms could suffer the performance degradation. To improve the noise control performance, the diagonal-structure BFXLMS (DBFXLMS) algorithm is proposed in [24,25]. Further reduction methods on the developed DBFXLMS algorithm could be developed.

Section 2 proposes new algorithms for reduction of computational load. Section 3 shows computer simulations to validate the control performance. Finally, Section 4 presents the conclusions.

2. Simplified and channel reduced diagonal-structure bilinear filtered-X LMS algorithms

2.1. Simplified diagonal-structure bilinear filtered-X LMS algorithm

A standard bilinear filter model with an input $x(n)$ and an output $y(n)$ can be expresses as

* Corresponding author at: Department of Engineering, Purdue University North Central, 1401 South U.S. Highway 421, Westville, IN 46391, USA.
E-mail address: lizhetan@pnc.edu (L. Tan).

http://dx.doi.org/10.1016/j.apacoust.2016.01.001
0003-682X/

$$y(n) = \sum_{j=0}^{L} a_j(n)x(n-j) + \sum_{j=1}^{L} b_j(n)y(n-j) + \sum_{i=0}^{L}\sum_{j=1}^{L} c_{i,j}(n)x(n-i)y(n-j) \quad (1)$$

where L is the memory length. $a_j(n)$, $b_j(n)$, $c_{i,j}(n)$ are the filter coefficients; and have $L+1$, L, $L(L+1)$ coefficient elements, respectively. Zhao et al. [22] presented a simplified BFXLMS algorithm based on the following simplified model [21]:

$$y(n) = \sum_{j=0}^{L} a_j(n)x(n-j) + \sum_{j=1}^{L} b_j(n)y(n-j) + \sum_{i=R_1}^{R_1+2P}\sum_{j=1}^{L} c_{i,j}(n)x(n-i)y(n-j) \quad (2)$$

where $P \leqslant \lfloor L/2 \rfloor$ is a small positive integer, $R_1 = \lfloor L/2 \rfloor - P$, and $R_2 = \lfloor L/2 \rfloor + P$. Note that the inputs from $x(n-R_1)$, $x(n-R_1-1)$, $\ldots$, $x(n-R_2)$ are fed into the bilinear section and $R_2 = R_1 + 2P$.

Using (2) and $R_2 = R_1 + 2P$, the relationship between the input and output can be expressed as

$$\begin{aligned} y(n) = &\sum_{j=0}^{L} a_j(n)x(n-j) + \sum_{j=1}^{L} b_j(n)y(n-j) \\ &+ \sum_{j=0}^{2P} g_{1,j}(n)x(n-R_1-j)y(n-1-j) \\ &+ \sum_{j=0}^{2P} g_{2,j}(n)x(n-R_1-j)y(n-2-j) + \cdots \\ &+ \sum_{j=0}^{2P} g_{L-2P,j}(n)x(n-R_1-j)y[n-(L-2P)-j] \\ &+ \sum_{j=0}^{2P-1} g_{L-2P+1,j}(n)x(n-R_1-j)y[n-(L-2P+1)-j] \\ &+ \sum_{j=0}^{2P-2} g_{L-2P+2,j}(n)x(n-R_1-j)y[n-(L-2P+2)-j] + \cdots \\ &+ g_{L,j}(n)x(n-R_1)y(n-L) \\ &+ \sum_{j=0}^{2P-1} h_{1,j}(n)x(n-R_1-1-j)y(n-1-j) \\ &+ \sum_{j=0}^{2P-2} h_{2,j}(n)x(n-R_1-2-j)y(n-1-j) + \cdots \\ &+ h_{2P,0}(n)x(n-R_1-2P)y(n-1) \end{aligned} \quad (3)$$

Eq. (3) can easily be illustrated in Fig. 1 for the cross elements for $L = 5$ and $P = 1$.

Reformulating (3) leads to the following:

$$y(n) = A^T(n)X(n) + B^T(n)Y(n-1) + \sum_{i=1}^{L} G_i^T(n)U_i(n) + \sum_{i=1}^{2P} H_i^T(n)V_i(n) \quad (4)$$

where

For $i = 1, 2, \ldots, L-2P$

$$U_i^T(n) = [x(n-R_1)y(n-i)\; x(n-R_1-1)y(n-i-1)\; \cdots\; x(n-R_1-2P)y(n-i-2P)]^T \quad (5)$$

$$G_i^T(n) = \left[g_{i,0}(n)\; g_{i,1}(n)\; \cdots\; g_{i,2P}(n)\right]^T \quad (6)$$

For $i = L-2P+1, \ldots, L$

$$U_i^T(n) = [x(n-R_1)y(n-i)\; x(n-R_1-1)y(n-i-1)\; \cdots\; x(n-R_1-L+i)y(n-L)]^T \quad (7)$$

$$G_i^T(n) = [g_{i,0}(n)\; g_{i,1}(n)\; \cdots\; g_{i,2P-(i-L+2P)}(n)]^T \quad (8)$$

$$V_i^T(n) = [x(n-R_1-i)y(n-1)\; x(n-R_1-i-1)y(n-i-1)\; \cdots\; x(n-R_1-2P)y(n-1-2P+i)]^T \quad (9)$$

For $i = 1, 2, \ldots, 2P$

$$H_i^T(n) = \left[h_{i,0}(n)\; h_{i,1}(n)\; \cdots\; h_{i,2P-i}(n)\right]^T \quad (10)$$

The SDBFXLMS algorithm is then summarized below:

$$A(n+1) = A(n) + \mu_a e(n)X'(n) \quad (11)$$

$$B(n+1) = B(n) + \mu_b e(n)Y'(n-1) \quad (12)$$

$$G_i(n+1) = G_i(n) + \mu_c e(n)U_i'(n) \quad i = 1, 2, \ldots, L \quad (13)$$

$$H_i(n+1) = H_i(n) + \mu_c e(n)V_i'(n) \quad i = 1, 2, \ldots, 2P \quad (14)$$

where

$$x'(n) = \bar{s}(n) * x(n) \quad (15)$$

$$y'(n-1) = \bar{s}(n) * y(n-1) \quad (16)$$

$$u_i'(n) = \bar{s}(n) * u_i(n) \quad i = 1, 2, \ldots, L \quad (17)$$

$$v_i'(n) = \bar{s}(n) * v_i(n) \quad i = 1, 2, \ldots, 2P \quad (18)$$

where $*$ denotes linear convolution and $\bar{s}(n)$ is the sequence of the secondary path estimate $\bar{S}(z)$. μ_a, μ_b, and μ_c are the convergence factors controlling the convergence speed and the stability of the algorithm for updating the feed forward coefficients $A(n)$, feedback coefficients $B(n)$, and cross term coefficients $G_i(n)$ and $H_i(n)$. Define the signal vector $L'(n)$ as

$$L'^T(n) = \left[X'^T(n)\; Y'^T(n-1)U_1'^T(n)\; \cdots\; V_1'^T(n)\; \cdots\right] \quad (19)$$