# Development of Frequency Domain Block Filtered-s LMS (FBFSLMS) Algorithm for Active Noise Control System

[1]Debi Prasad Das, [2]Ganapati Panda, and [3]Dilip Kumar Nayak

[1]Central Electronics Engineering Research Institute, Pilani, India 333031, debi_das_debi@yahoo.com
[2]National Institute of Technology, Jamshedpur, India 769008, ganapati.panda@gmail.com
[3]C. V. Raman College of Engineering, Bhubaneswar, India 752054, dilipnayak@rediffmail.com

## ABSTRACT

This paper presents a computationally efficient frequency domain block nonlinear active noise control algorithm. The computational complexity of nonlinear ANC algorithms is very high in comparison to the linear ANC algorithms like filtered-x LMS. Filtered-s LMS algorithm has been recently proposed as an effective ANC algorithm for nonlinear noise processes. In this paper, a frequency domain block implementation of the filtered-s LMS algorithm is proposed to achieve computational advantage. This algorithm is the exact implementation of the FSLMS algorithm but with reduced complexity.

## 1. INTRODUCTION

Active noise control (ANC) [1] has gained lot of research interest because of its potential use in low frequency acoustic noise control. Filtered-x LMS (FXLMS) algorithm is the most popular algorithm for ANC because of its simplicity. However, in case of nonlinear noise processes, the FXLMS algorithm shows poor performance [2]. It has been shown that the performance of the controller can be improved by using nonlinear control structures. The basic concept of nonlinear controller used here is making a nonlinear mapping between the input and the output of the controller. As, at the output of the controller the secondary path is present in ANC system, each of the nonlinear elements of the controllers is to be filtered through the estimate of the secondary path transfer function for parameter update. Hence, the nonlinear controllers involve heavy computational overload. Hardly any work has been reported for fast implementation of nonlinear ANC structure. It has been shown in [3-4] that the filtered-s LMS (FSLMS) algorithm is a better candidate compared to Volterra filtered-X LMS algorithm [5-6] both in terms of performance and computational complexity for control of nonlinear noise processes. Transform domain adaptive filters are conventionally used for reducing computational complexity keeping the performance in tact. Frequency domain block implementation of FXLMS algorithm is proposed in [7-9] to achieve computational advantage over its time domain counterpart. This paper attempts to reduce the computational complexity of the recently proposed FSLMS algorithm making its frequency domain implementation.

## 2. FILTERED-S LMS ALGORITHM AND ITS FILTER BANK IMPLEMENTATION

In filtered-x LMS algorithm, the reference signal is filtered through the secondary path estimate filter for updating the controller parameters. In filtered-s LMS algorithm the reference signal $x(n)$ is expanded through trigonometric functional expansion to make a signal vector $\mathbf{s}(n)$. The signal elements of the vector $\mathbf{s}(n)$ are filtered through the secondary path estimate filter in case of the filtered-s LMS algorithm. Defining the reference signal vector $\mathbf{x}(n) = \{x(n), x(n-1), ..., x(n-N-1)\}^T$ we can define the functionally expanded vector

$$\begin{aligned}\mathbf{s}(n) = \{&x(n), \sin[\pi x(n)], \cos[\pi x(n)], ...,\\ &\sin[(P\pi x(n)], \cos[P\pi x(n)],\\ &x(n-1), \sin[\pi x(n-1)], \cos[\pi x(n-1)], ...,\\ &\sin[(P\pi x(n-1)], \cos[P\pi x(n-1)], ...,\\ &x(n-N+1), \sin[\pi x(n-N+1)], \cos[\pi x(n-N+1)], ...\\ &\sin[P\pi x(n-N+1)], \cos[P\pi x(n-N+1)]\}^T\end{aligned}$$

The $\mathbf{s}(n)$ vector contains $M$ elements in case of $P$ th order expansion of $N$ elements of the reference signal vector $\mathbf{x}(n)$, where $M = N(2P+1)$. The derivation of FSLMS algorithm is presented in detail in [3]. Hence, only its filter bank implementation is presented here.

*A. Filter-Bank Implementation of FSLMS algorithm.*

The FSLMS algorithm presented above is called single channel implementation as the functionally expanded vector $\mathbf{s}(n)$ is multiplied with the weight vector $\hat{\mathbf{h}}(n)$ and the sum is the output of the network. Also each element of $\mathbf{s}(n)$ should be individually filtered through the secondary path

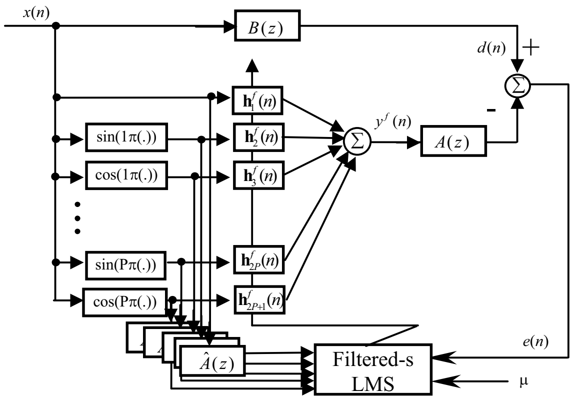

Fig.1 Filter bank implementation of FSLMS algorithm

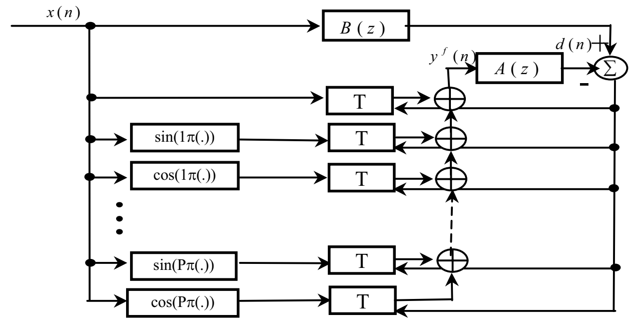

Fig.2 FBFSLMS algorithm for ANC

estimate $\hat{A}(z)$. To overcome this computational burden a simple filter-bank implementation of the same algorithm is established and is shown in Fig.1. The nonlinear adaptive filter can be shown to consist of a bank of linear filters with the inputs modulated by sinusoidal nonlinear functions. Now we present the derivation of the nonlinear filter for its filter bank structure. Let the output of the nonlinear filters may be represented as

$$\mathbf{y}^f(n) = y_1^f(n) + y_2^f(n) + ... + y_{2P+1}^f(n) \quad (1)$$

where $y_1^f(n) = \mathbf{x}_1^f(n)\{\mathbf{h}_1^f(n)\}^T$, $y_2^f(n) = \mathbf{x}_2^f(n)\{\mathbf{h}_2^f(n)\}^T, \dots,$

$y_{2P+1}^f(n) = \mathbf{x}_{2P+1}^f(n)\{\mathbf{h}_{2P+1}^f(n)\}^T$ and

$\mathbf{x}_1^f(n) = \mathbf{x}(n)$, $\mathbf{x}_2^f(n) = \sin[\pi\mathbf{x}(n)]$, $\mathbf{x}_3^f(n) = \cos[\pi\mathbf{x}(n)]$,

$\mathbf{x}_4^f(n) = \sin[2\pi\mathbf{x}(n)]$, $\mathbf{x}_5^f(n) = \cos[2\pi\mathbf{x}(n)], \dots,$

$\mathbf{x}_{2P}^f(n) = \sin[P\pi\mathbf{x}(n)]$, $\mathbf{x}_{2P+1}^f(n) = \cos[P\pi\mathbf{x}(n)]$.

The weight update equation can be rewritten in this way as

$$\hat{\mathbf{h}}_i^f(n) = \hat{\mathbf{h}}_i^f(n) + \mu\mathbf{c}_i^f(n) \quad (2)$$

where

$$\mathbf{c}_i^f(n) = e(n)x_i^f{}'(n) \quad (3)$$

and $x_i^f{}'(n) = x_i^f(n) * \hat{a}(n)$ (4)

$$e(n) = d(n) - y^f(n) * a(n) \quad (5)$$

Here all the sub adaptive filters represented by impulse responses $\mathbf{h}_i^f(n)$, $0 < i \le 2P+1$ are of memory size $N$. From the diagram it is obvious that lot of computational savings can be achieved by this implementation as for a $P$ th order functional expansion only $2P+1$ elements are to be filtered through $\hat{A}(z)$, where as in original FSLMS algorithm $N(2P+1)$ elements are to be filtered through $\hat{A}(z)$. In Fig. 1, $\hat{\mathbf{h}}_1^f(n)$ through $\hat{\mathbf{h}}_{2P+1}^f(n)$ are $N-1$ th order FIR filters. In FXLMS algorithm only one $N-1$ th order FIR filter exists. From the filter-bank implementation of FSLMS algorithm, it is seen that the FSLMS algorithm is nearly equivalent to $2P+1$ numbers of FXLMS algorithms having $2P$ numbers of nonlinear functional units at the inputs. This reasoning, gives us a way to make its frequency domain block implementation, which is advantageous over its time domain sequential implementation in terms of computational involvement.

## 3. FREQUENCY DOMAIN BLOCK FSLMS (FBFSLMS) ALGORITHMS

In this section, we propose the frequency-domain implementation of the FSLMS algorithm exploiting the properties of fast Fourier transform (FFT). These transforms are used to efficiently compute the associated linear convolutions and correlations in the FSLMS algorithms. This algorithm is termed as frequency domain block FSLMS (FBFSLMS) algorithm

From the previous section it is learnt that the FSLMS algorithm involves $2P+1$ numbers of convolutions and $2P+1$ numbers of cross correlations. These convolutions and correlations are implemented in transform domain using overlap and save method. Defining $F_{2N}$ and $F_{2N}^{-1}$ as the 2$N$-point FFT and IFFT, the $i$ th linear convolution may be implemented as follows.

$$X_i^f(k+N) = F_{2N}\begin{bmatrix} \mathbf{x}_i^f(k) \\ \mathbf{x}_i^f(k+N) \end{bmatrix} \quad (6)$$

$$\hat{H}_i^f(k) = F_{2N}\begin{bmatrix} I_N \\ O_N \end{bmatrix}\hat{\mathbf{h}}_i^f(k) \quad (7)$$

$$\mathbf{x}_i^f(k+N) = [O_N \quad I_N]F_{2N}^{-1}[X_i^f(k+N) \otimes \hat{H}_i^f(k)]. \quad (8)$$

where $\otimes$ denotes frequency-by-frequency bin multiply or in more general way point-by-point multiplications of two vectors. Two vector of same length can be multiplied element-by-element wise in this operation to get a vector of same length. The matrix $I_N$ is an $N \times N$ identity matrix; the matrix $O_N$ is an $N \times N$ matrix with all zero elements.

Similarly, $i$ th cross correlation may be implemented as follows.

$$\hat{A}(k) = F_{2N} \begin{bmatrix} I_N \\ O_N \end{bmatrix} \hat{a}(k) \quad (9)$$

$$\mathbf{x}_i^f{}'(k+N) = [O_N \quad I_N] F_{2N}^{-1} [X_i^f(k+N) \otimes \hat{A}_i^f(k)]. \quad (10)$$

The FFT-based implementation of the cross-correlation can be expressed as

$$X_i^f{}'(k+N) = F_{2N} \begin{bmatrix} \mathbf{x}_i^f{}'(k) \\ \mathbf{x}_i^f{}'(k+N) \end{bmatrix} \quad (11)$$

$$E(k+N) = F_{2N} \begin{bmatrix} I_N \\ O_N \end{bmatrix} \mathbf{e}(k+N) \quad (12)$$

$$\mathbf{c}_i^f(k+N) = [O_N \quad I_N] F_{2N}^{-1} [X_i^{f\,\prime *}(k+N) \otimes E(k+N)]. \quad (13)$$

where $\{.\}^*$ is for complex conjugate operator. The time domain weights are updated as

$$\hat{\mathbf{h}}_i^f(k+N) = \hat{\mathbf{h}}_i^f(k) + \mu \mathbf{c}_i^f(k+N) \quad (14)$$

The structure of the FBFSLMS is shown in Fig. 2 and Fig. 3. This structure is different from the frequency-domain implementation of the FXLMS (FBFXLMS) algorithm proposed by Shen and Spanias [7, 8], the way the filtering of the reference signal, through the secondary-path estimate, is implemented. In the proposed method, the secondary path filtering is done in block frequency domain.

## 4. COMPUTATIONAL COMPLEXITY

Assuming the length of secondary path estimate filter $L = N$, the number of multiplication requirements for $N$-samples is $3N^2(2P+1)$ and $(14N\log_2 N + 38N)(2P+1)$ in FSLMS and FBFSLMS algorithms respectively. Therefore as an illustration for $N = 1024$, FSLMS and FBFSLMS algorithms involves $1048576\,(2P+1)$ and $182272\,(2P+1)$ numbers of multiplications, respectively.

## 5. SIMULATION RESULTS

To validate the effectiveness of the present reduced complexity FBFSLMS algorithm over FBFXLMS algorithm, computer simulation is carried out. Four experiments are carried out for nonlinear ANC. In all experiments the linear part of the primary path is chosen as $B(z) = 0.071z^{-5} - 0.59z^{-6} + 0.9z^{-7}$ ($b(n)$ is its impulse response). The experiments are conducted for two different types of primary path nonlinearity (second and third order) and two different types of secondary path transfer function (minimum and maximum phase). The primary signal is generated from the following equations. $s(n) = \sqrt{2}\sin(\pi n/8 + 2\alpha\pi) + 0.01\eta$; where $\eta$ is a unit power Gaussian random noise, and $\alpha$ is used to randomly vary the phase of the periodic signal to compute the ensemble average. The ensemble average of square error (EASE) for 100 independent trials is chosen for performance evaluation. Also, the power spectral magnitude

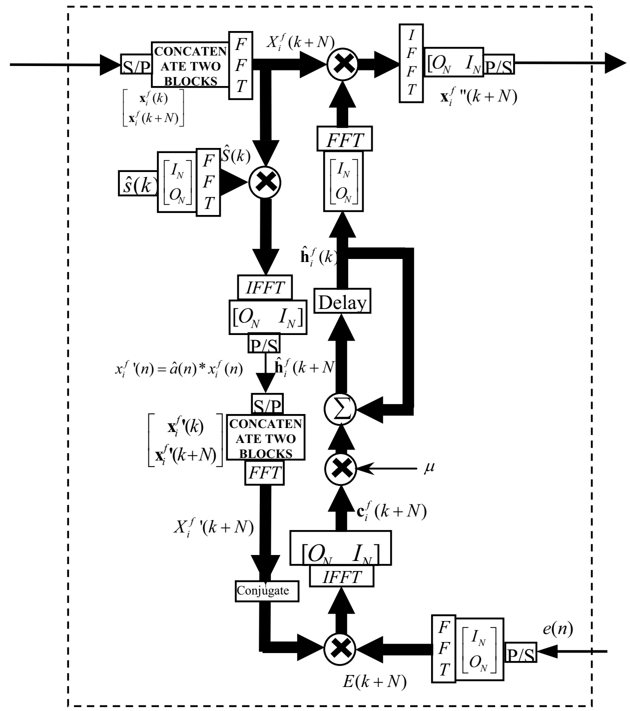

Fig. 3 Internal of the T block of FBFSLMS

of the square error signal in the steady state is used for further performance comparison.

Experiment 1: In this experiment, the primary noise at the canceling point is generated based on the following third-order polynomial model, $d(n) = t(n-2) + 0.5t^2(n-2) - 0.04t^3(n-1)$ where $t(n) = x(n) * b(n)$ and * denotes convolution operation. The secondary path is chosen as a nonminimum phase system $A(z) = z^{-2} + 1.5z^{-3} - z^{-4}$. The results are plotted in Fig. 4 and Fig. 5.

Experiment 2: In the second experiment, the primary noise at the canceling point is generated based on the following second-order polynomial model, $d(n) = t(n-2) + 0.5t^2(n-2)$. The secondary path is same as that of expt. 1. The results are plotted in Fig. 6 and Fig. 7.

Experiment 3: Here, the nonlinear part of the primary path is chosen same as expt.1. The secondary path is chosen as a minimum phase system $A(z) = z^{-2} + 0.5z^{-3}$. The results are plotted in Fig. 8 and Fig. 9.

Experiment 4: Here, the nonlinear part of the primary path is chosen same as expt. 2. The secondary path is chosen as a minimum phase system. The results are plotted in Fig. 10 and Fig. 11.

From the results of all the four experiments, it is concluded that the FBFSLMS algorithm outperforms the FBFXLMS algorithm in terms of mean square error. To evaluate the effect of nonlinearity on the residual error, the power spectral magnitudes of the square error signals for every experiment are plotted. From the plot it is seen that the FBFSLMS algorithm being a nonlinear controller performs

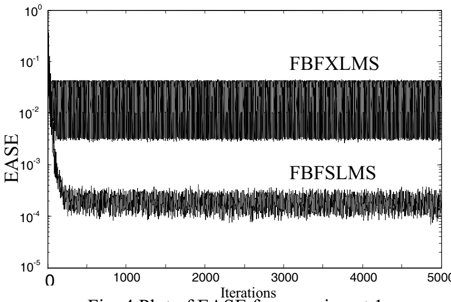

Fig. 4 Plot of EASE for experiment 1

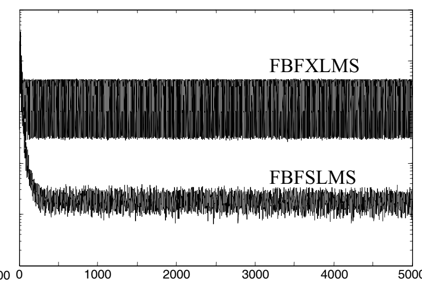

Fig. 6 Plot of EASE for experiment 2

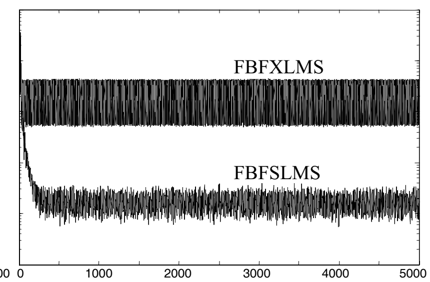

Fig. 8 Plot of EASE for experiment 3

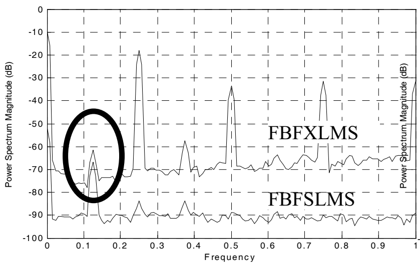

Fig. 5 Spectrum of residual error in experiment 1

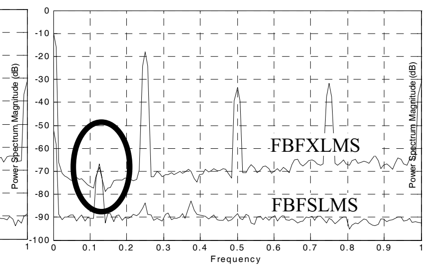

Fig. 7 Spectrum of residual error in experiment 2

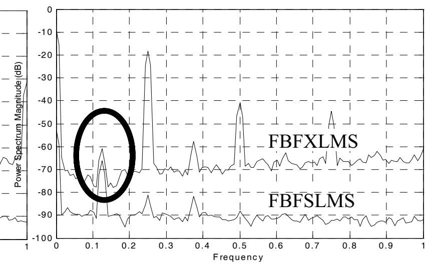

Fig. 9 Spectrum of residual error in experiment 3

better for third order nonlinearity case, in comparison to the second order one.

## 6. CONCLUSION

In this paper, a computationally efficient frequency domain block FSLMS algorithm, a nonlinear active noise control algorithm, is newly proposed to achieve computational advantage. The filter bank implementation of FSLMS algorithm is established, which gives a way to apply the conventional frequency domain block implementation for FSLMS algorithm. The detailed derivation of the algorithm is presented. The computational complexity of nonlinear ANC algorithms is also analyzed. This algorithm is the exact implementation of the FSLMS algorithm but with reduced complexity. Computer simulation study of the proposed algorithm along with FBFXLMS algorithm validates the effectiveness of the proposed algorithm over the FBFXLMS algorithm.

## 7. REFERENCES

[1] S. M. Kuo and D.R. Morgan, *Active Noise Control Systems—Algorithms and DSP Implementations.* New York: Wiley, 1996.

[2] P. Strauch, B.Mulgrew, "Active Control of Non linear Noise Processes in a linear Duct,"*IEEE Trans. Sig. Proc*., vol. 46. No. 9, pp. 2404-2412, Sept. 1998.

[3] D. P. Das, G.Panda, "Active mitigation of nonlinear noise processes using a novel filtered-s LMS algorithm," *IEEE Trans. on Speech and Audio Proc.*, Vol: 12 , Issue: 3, pp. 313 – 322, May 2004.

[4] D.P. Das, S. R. Mohapatra, A. Routray and T. K. Basu, "Filtered-s LMS algorithm for multichannel active noise control of nonlinear noise processes" accepted *IEEE Trans. on Speech and Audio Pro.* Jan 2005 paper no. T-SA-00712-2005.

[5] L. Tan and J. Jiang " Adaptive Volterra filters for active control of nonlinear noise processes*," IEEE Trans. Sig. Proc.*, vol. 49, no. 8, pp. 1667-1676, Aug. 2001.

[6] G. L. Sicuranza and A.C.Carini, "Filtered-X affine projection algorithm for multichannel Active Noise Control Using second-order Volterra filters," *IEEE Sig. Proc. Let.* vol. 11 no. 11, pp. 853-857 Nov. 2004.

[7] Q. Shen and A. S. Spanias, "Time and frequency domain X block LMS algorithms for single channel active noise control." *Proc. 2nd Congress on Recent Developments in Air-& Structure-Borne Sound and Vibration,* Auburn, AL, pp. 353-360, March-1992.

[8] —, "Time and frequency domain X block LMS algorithms for single channel active noise control." *Noise Control Eng. J.* vol. 44(6), pp. 281-293, Nov-Dec. 1996.

[9] K. Reichard and D. Swanson, " Frequency-domain implementation of the filtered-x algorithm with online system identification," *Proc. 2nd Conf. on Recent Development in Active control of Sound and Vibration,* Blacksburg, VA, pp. 562-573, April 1993.

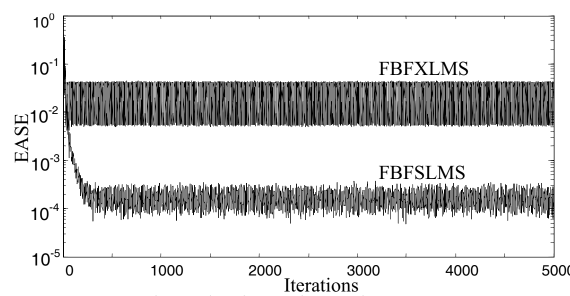

Fig. 10 Plot of EASE for experiment 4

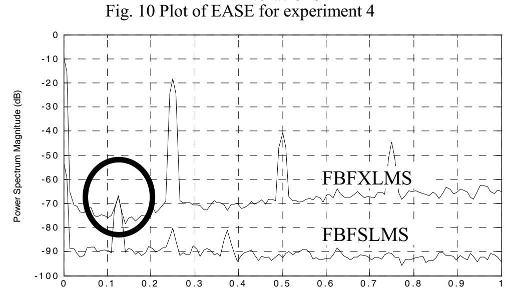

Fig. 11 Spectrum of residual error in experiment 4