

Multiple Iteration of Weight Updates for Least Mean Square Adaptive Filter in Active Noise Control Application

Rahimie Mustafa¹, Anuar Mikdad Muad², Shahrizal Jelani¹ and Ahmad Nur Alifa Abdul Razap¹

¹UCSI University, Cheras Malaysia

²Universiti Kebangsaan Malaysia, Bandar Baru Bangi Malaysia

Abstract. The method of least mean square (LMS) is the commonly used algorithm in Adaptive filter due to its simplicity and robustness in implementation. In Active Noise Control application, a filtered reference signal is used prior to LMS algorithm to overcome the constraint on stability and convergence performance of the system due to the existence of the auxiliary path. This is known as Filtered-X LMS algorithm. In conventional Filtered-X LMS algorithm, each filter weight is updated once on every audio sample. This paper proposes the improved version of Filtered-X LMS algorithm with the use of multiple iteration of filter weight on every sample of audio signal. The proposed work uses field programmable gate arrays to realize real-time simulation on hardware for the noise signal of 500 Hz. Results from the real-time hardware simulations have shown much faster error convergence and better adaptation performance for different selections of learning constant μ , as compared with the conventional method.

1 Introduction

The least mean square (LMS) can be considered as the most applied algorithm for linear adaptive filter particularly in active noise control (ANC) application that required real-time processing for a successful and efficient hardware implementation [1]-[3]. The use of specialized digital signal processor (DSP) chip and with the capability of handling numerous floating point operations manage to address the real-time processing issue in narrowband attenuation of ANC headset [4]-[5], and in broadband attenuation for duct application [6] based on least mean square (LMS) algorithm. However, the performance of signal convergence in terms of the speed of filter adaptation and the excess of mean square error (MSE) is usually overlooked, which can be critical in a time-varying environment [2], [3]. Research being carried out to improve the error convergence time of LMS-based algorithm however increased the algorithm complexity [7]-[9] and consequently, implementations are mostly evaluated successfully in a purely simulated domain [10]-[12].

The conventional digital signal processor (DSP) evaluates a signal in a sequential behaviour, where each updates on the filter weight might require several instruction cycles to complete. As a result, heavy processing for complicated LMS-based algorithm will require more instruction cycles and will invoke additional time delay to ANC system. Therefore, in general the error signal will converge much slower. To overcome this, an analysis on the LMS convergence behaviour has been carried out using field programmable gate arrays (FPGA)

on ANC system for duct application [13]. It is known that FPGA has the capability of processing a signal in a parallel manner hence allowing a complete computation of weight updates within a few clock cycles. The experimental work by the author [13] also further verify the theoretical principle established from previous studies [14]-[17], that are initially based on simplifying assumptions.

In this paper, motivated by the idea presented in [13] we utilize the FPGA advantage to achieve real-time and faster convergence of error signal for LMS-based algorithm. A real-time hardware simulation is carried out to implement the modified Filtered-X LMS algorithm namely MFxLMS. Performance of the system is then compared with the conventional Filtered-X LMS algorithm also by the same hardware structure. The paper is organized as follows: Section 2 describes briefly on theory and underlying principle of LMS-based adaptive filter used, followed by our proposed modified version. In section 3, the implementation of the modified LMS-based algorithm on FPGA is described and in section 4, the real-time hardware simulation of ANC based on MFxLMS algorithm is briefly presented. The results obtained are then briefly discussed in section 5 and finally, section 6 presents the conclusion of the findings.

2 LMS-Based adaptive filter

LMS algorithm is an approximation method of steepest gradient descent that relies on the value of the instantaneous squared error signal [18], [19]. Due to this

approximation, the calculation of adaptive filter weight has resulted in the simplification that is expressed as:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{x}(n)e(n) \quad (1)$$

where $\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{L-1}(n)]^T$ are the weight updates or adaptive filter coefficients vectors for L-length filter, $\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T$ correspond to the reference signal vectors, $e(n)$ is the error signal at sampling time n and μ is the learning constant.

Our implementation uses the structure of finite impulse response (FIR), a stable digital filter that has a usual canonical form of tapped-delay input as a basic element to realize the designated adaptive filter. The LMS algorithm updates the FIR filter weight to produce the output signal, $y(n)$ that can be expressed as an arithmetic sum of products:

$$y(n) = \sum_{k=0}^{N-1} w_k(n) \cdot x(n-k) \quad (2)$$

where $y(n)$ is the adaptive filter output at time n , $x(n-k)$ is the tapped-delay input reference signal and $w_k(n)$ for $k = 0, 1, \dots, N-1$ are the N -tap FIR filter coefficients. In ANC system, the residual error signal is acquired from a sensor (usually microphone) that is also corresponding to the difference between the primary noise, $d(n)$ and the adaptive filter output, $y(n)$ which is expressed as:

$$e(n) = d(n) - y(n) \quad (3)$$

In general, numerical calculation for updating the filter weights utilizing conventional DSP chip is performed one time in every sample of audio signal that is expressed as [1]:

$$w_k(n+1) = w_k(n) + \mu x(n-k)e(n) \quad (4)$$

where the error signal $e(n)$ is multiplied with each tapped-delay input of reference signal $x(n-k)$ alongside with learning constant μ before being summed up with the previous weights.

However, if we could allow the weight updates to be calculated more than once within each sample of audio signal [20], then logically we can expect faster adaptation of weight updates, hence faster convergence of error signal. In this method, the calculation of error signal, $e(n)$ from (3) has become:

$$e_j(n) = d_j(n) - y_j(n) \quad (5)$$

where j is the number of repeated calculation (iteration) of error within every sample of audio signal. Thus, by repeating the calculation of the error signal, the computation of filter weight can now be modified to:

$$w_{j,k}(n+1) = w_{j,k}(n) + \mu x(n-k)e_j(n) \quad (6)$$

3 FPGA Implementation

The FIR-based LMS adaptive filter is designed using conventional multiplier and accumulator and is

implemented onto the Altera Cyclone II FPGA chip embedded on a development platform. The development platform is also featuring with a real-time audio coding and decoding (codec) that can facilitate analogue and digital conversion (DAC) of analogue audio signal at 48 kHz sampling rate along with a simple anti-aliasing and reconstruction filter. Therefore, additional design of analogue audio interface circuitry with FPGA chip is thus avoided. In digital domain, the audio signal is further down-sampled to 8 kHz to increase the number of maximum iteration possible for weight updates on every sample of audio signal. This will not affect the ANC system, since the noise frequency of interest for practical ANC application is well below 1 kHz range [2], [3]. The computation of weight updates is carried out iteratively within each sample of audio signal as depicted by the flow chart in Fig. 1.

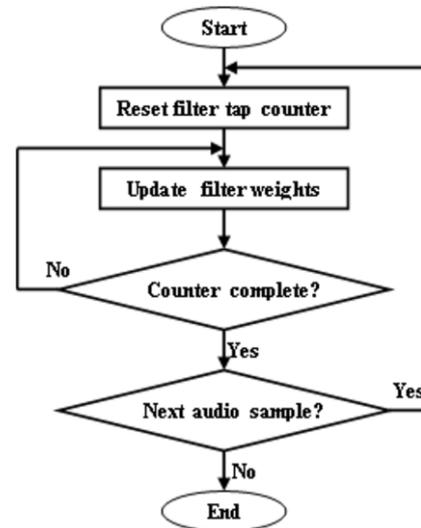


Figure 1. Flow chart for the process of filter weights updates for one sample of audio signal.

3.1. Learning constant

Due to the fixed-point data representation on FPGA, a truncation process during calculation is unavoidable especially to match the size of data at the end terminal. The truncation process also has the implication of dividing the data by the factor of 2 besides giving the effect of quantization and rounding errors. When this is implemented in the path of filter weight updates, due to this division effect, it will implicitly infer a multiplication with a learning constant, μ with the factor of reciprocal of 2 as the function of truncation size:

$$\mu = \frac{1}{2^B} \quad (7)$$

where B is a positive integer represent the size of the least significant bit that is truncated. Based on the requirement for stability and convergence of error signal established in the previous studies [1]-[3], the value of learning constant is bounded in a limited region:

$$0 < \mu < \frac{2}{NP_x} \quad (8)$$

where N is the filter length and P_x is the expectation of reference signal power. With the use of 64-taps filter length and the value of reference signal power set to be greater than one unit in average ($P_x > 1$), we obtained the practical bound of learning constant in our design, that is:

$$0 < \mu \leq 2^{-6} \quad (9)$$

The size of learning constant is 16-bit. After multiplying with the reference signal, $x(n)$ and the error signal, $e(n)$ of 24-bit, the accumulated size due to the fixed-point calculation is 64-bit in total. Since the size of the filter weight is set to 33-bit in size, then a total of 31-bit of least significant value has to be truncated. Due to this effect, we obtained the maximum inherent practical bound of the learning constant as:

$$0 < \mu \leq 2^{-31} \quad (10)$$

The maximum inherent practical bound of the learning constant is found to be within the range of the requirement as in equation (9). Thus, we can guarantee stability of the adaptation process since we can never exceed the maximum value. The choosing and selection of learning constant, μ can be made manually in between experiments by using digital switches. To simplify and to make a uniform selection (in logarithmic sense) within the range of the bounded value given in equation (10), a total number of 16 differences selection of learning constant value is used.

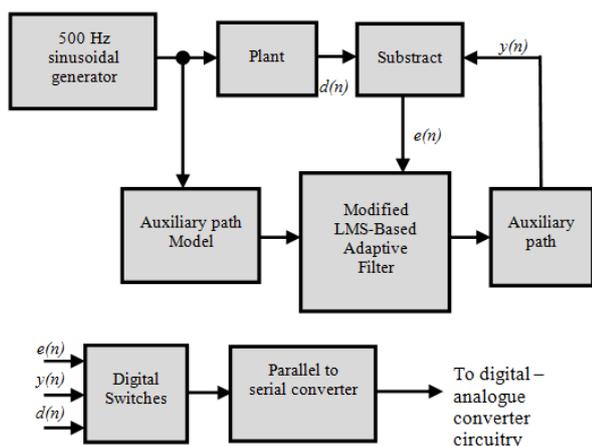


Figure 2. Block diagram of MFxLMS implemented on FPGA.

4 Real-time hardware simulation

Fig. 2 shows the block diagram of MFxLMS implemented on FPGA environment. The conventional Filtered-X LMS system also uses the same structure except the sub-block on *Modified LMS-Based Adaptive Filter* is replaced by *Conventional LMS-Based Adaptive Filter*. In Actual ANC system, the transfer function of sub-block on *Auxiliary path Model* need to adapt with the transfer function of sub-block on *Auxiliary path* by means of adaptive algorithm and need to be continuously modeled when the system is operated.

In order to conduct the experiment systematically, first the transfer function of sub-block *Auxiliary path* is

fixed to two different types (pure samples delay and impulse response of all-pass filter) and the transfer function of sub-block *Auxiliary path Model* is made to follow exactly as the *Auxiliary path*. Secondly, the transfer function of sub-block on *plant* is also fixed to two different types (pure samples delay and impulse response of band-pass filter). By setting up to this configuration, there will be four differences alternative combination of *Plant* and *Auxiliary path*. In this sense, the four different combinations will gives more intuitive insight on the comparison of performance between both methods. Finally, a single 500 Hz frequency signal is used as the reference noise signal to consider the simple and basic noise for the experiment.

5 Results

The output of the results on the real-time hardware simulation is basically the analogue audio signal converted by the on-board audio codec embedded on the development platform. This audio signal is reconstructed based on the DAC circuitry that is sampled at 48 kHz sampling rate. This audio signal can be either the error signal $e(n)$, adaptive filter output $y(n)$ or the primary noise signal $d(n)$. This analogue audio signal is then captured and represented in computer terminal (PC) using Matlab for further analysis and graph presentation.

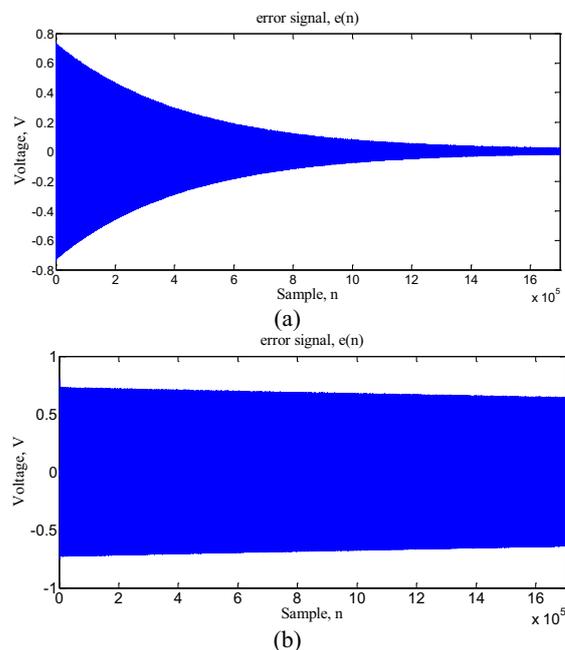
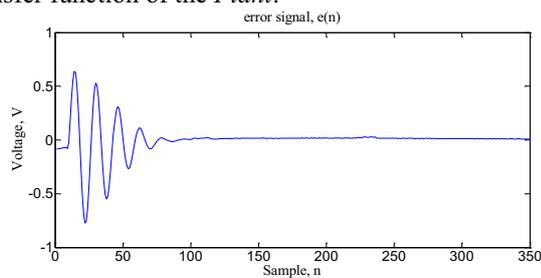


Figure 3. Error signal, $e(n)$ at plant is pure samples delay, auxiliary path is pure samples delay and the learning constant at 0.0001×10^{-10} (a) MFxLMS, (b) Conventional Filtered-X LMS

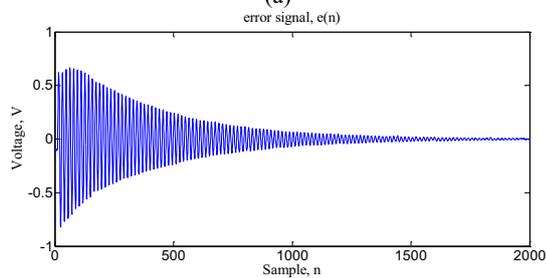
Fig. 3 and Fig. 4 shown two distinctive results on the experiments for MFxLMS and Conventional Filtered-X LMS for ANC system with the transfer function of *Plant* and *Auxiliary path* that is just pure samples delay. It is shown in Fig. 3 that when the learning constant is set to the lowest value which is $\mu = 0.0001 \times 10^{-10}$ and when MFxLMS algorithm is applied to the ANC system, the error signal converge to its minimum value after 1600K of audio samples. This result indicates the filter weight on

the adaptive filter has adapted successfully with the transfer function of the *Plant*. However, when the conventional Filtered-X LMS algorithm is applied, the error signal seems to require much more time (hence more audio sample) before it is expected to converge to its minimum value.

In Fig. 4, it is shown that when the learning constant is set to the highest value which is $\mu = 4.66 \times 10^{-10}$ and the ANC system applying the method of MFxLMS, the error signal converges to its minimum value only after 100 audio samples. Again this result indicates the filter weight on the *Plant* has been adapted successfully whereas for conventional Filtered-X LMS method, it will require more than 1500 samples of audio signal before the adaptive filter could adapt successfully with the transfer function of the *Plant*.

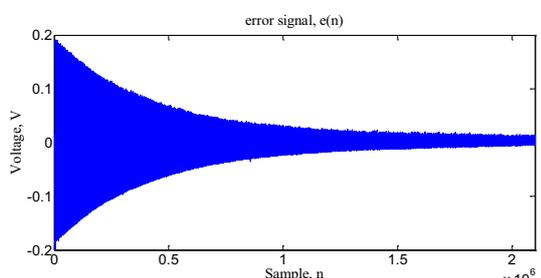


(a)

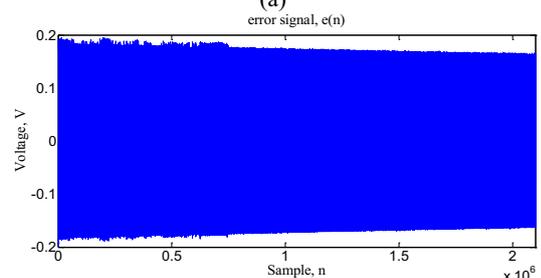


(b)

Figure 4. Error signal, $e(n)$ at plant is pure samples delay, auxiliary path is pure samples delay and the learning constant at 4.66×10^{-10} (a) MFxLMS, (b) Conventional Filtered-X LMS.

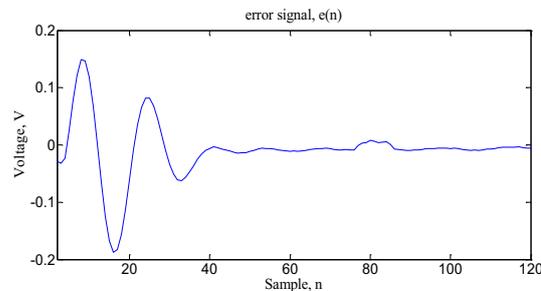


(a)

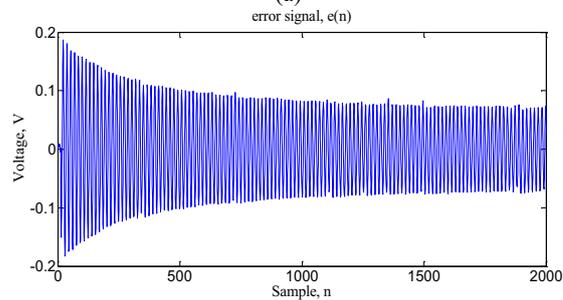


(b)

Figure 5. Error signal, $e(n)$ at plant is impulse response of band-pass filter, auxiliary path is pure samples delay and the learning constant at 0.0001×10^{-10} (a) MFxLMS, (b) Conventional Filtered-X LMS.

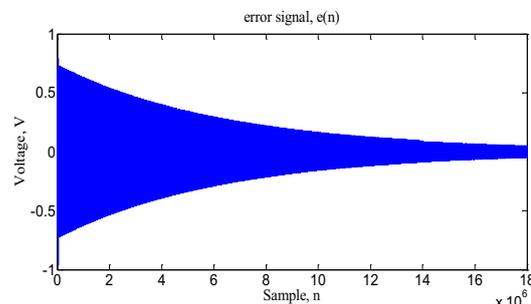


(a)

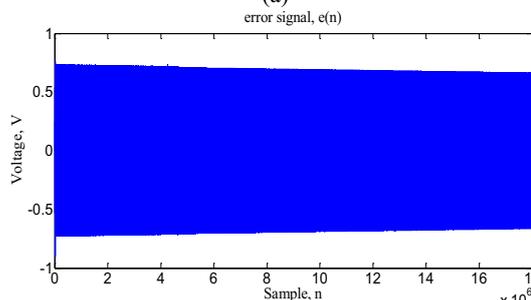


(b)

Figure 6. Error signal, $e(n)$ at plant is impulse response of band-pass filter, auxiliary path is pure samples delay and the learning constant at 4.66×10^{-10} (a) MFxLMS, (b) Conventional Filtered-X LMS.

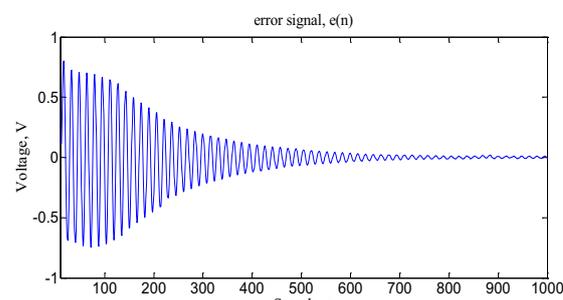


(a)



(b)

Figure 7. Error signal, $e(n)$ at plant is pure samples delay, auxiliary path is impulse response of all-pass filter and the learning constant at 0.0001×10^{-10} (a) MFxLMS, (b) Conventional Filtered-X LMS.



(a)

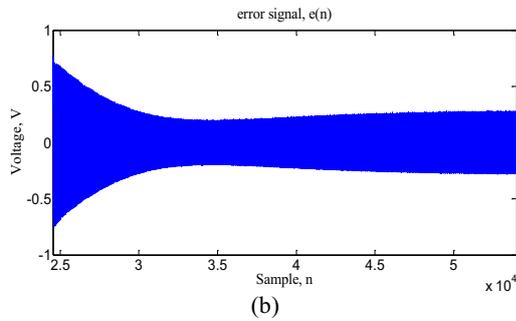


Figure 8. Error signal, $e(n)$ at plant is pure samples delay, auxiliary path is impulse response of all-pass filter and the learning constant at 4.66×10^{-10} (a) MFxLMS, (b) Conventional Filtered-X LMS.

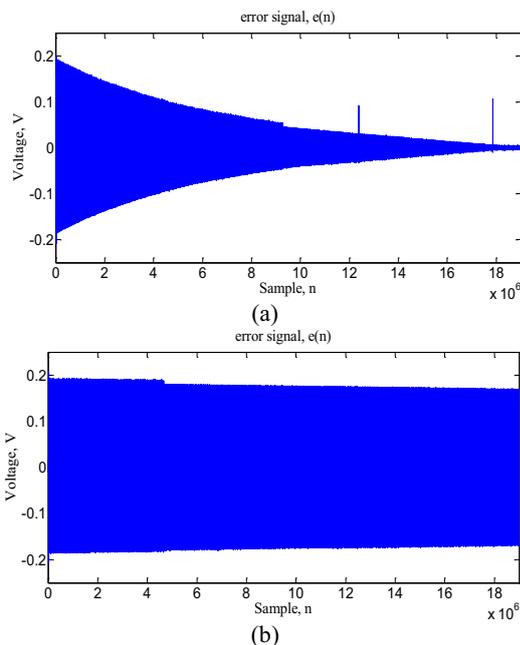


Figure 9. Error signal, $e(n)$ at plant is impulse response of band-pass filter, auxiliary path is impulse response of all-pass filter and the learning constant at 0.0001×10^{-10} (a) MFxLMS, (b) Conventional Filtered-X LMS.

Fig. 5 and Fig. 6 on the other hand show the results on the experiments for MFxLMS and Conventional Filtered-X LMS for ANC system when the transfer function of *Plant* is set to the impulse response of band-pass filter while the *Auxiliary path* is set to just pure samples delay. It is shown from the result that in general the MFxLMS algorithm performs much better in term of faster adaptation of filter weights. Similarly, Fig. 7 and Fig. 8 as well as Fig. 9 and Fig. 10 show the distinctive results on the comparison of error convergence performance between the method of MFxLMS and Conventional Filtered-X LMS algorithm.

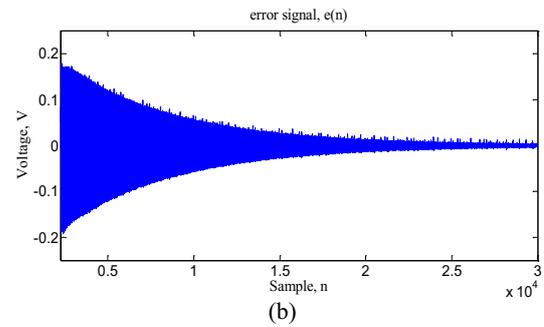
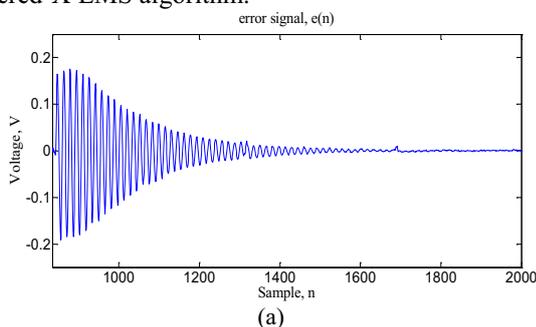


Figure 10. Error signal, $e(n)$ at plant is impulse response of band-pass filter, auxiliary path is impulse response of all-pass filter and the learning constant at 4.66×10^{-10} (a) MFxLMS, (b) Conventional Filtered-X LMS.

6 Conclusion

Based on the results in section 5, it is concluded that the new modified method on LMS-based adaptive algorithm that is implemented on ANC system namely MFxLMS, has shown superior performance in terms of faster adaptation of filter weights as compared to the conventional method, Filtered-X LMS.

Acknowledgment

The Author acknowledged the Faculty of Engineering and Built Environment of Universiti Kebangsaan Malaysia (UKM) for providing the laboratory facility to conduct the research.

References

1. Sen M. Kuo and Dennis R. Morgan, *Active noise control systems: Algorithms and DSP implementations*, John Wiley & Sons, Inc., New York, (1996).
2. P. A. Nelson and S. J. Elliott, *Active Control of Sound*. Academic Press, London, (1992).
3. S. J. Elliott, *Signal Processing for Active Control*. Academic Press, New York, (2001).
4. W. S. Gan, S. Mitra and S. M. Kuo "Adaptive Feedback Active Noise Control Headset: Implementation, Evaluation and Its Extensions" *IEEE Transactions on Consumer Electronics*, **51**(3), pp. 975-982, (August 2005).
5. Sen M. Kuo, S. Mitra and W. S. Gan, "Active Noise Control System for Headphone Applications" *IEEE Transactions on Control Systems Technology*, **14**(2), pp. 331-335, (March 2006).
6. C. Y. Chang "Efficient Active Noise Controller using a Fixed-point DSP" *Elsevier Signal Processing*, **89**, pp 843-850, (2008).
7. L. J. Eriksson and M. A. Allie, "Use of random noise for online transducer estimate in an adaptive active attenuation system," *Journal of Acoustical. Society of America*, **85**, pp. 797-802, (1989).
8. M. Zhang, H. Lan, and W. Ser "Cross-Updated Active Noise Control System with Online Secondary

- Path Modeling” *IEEE Transactions on Speech and Audio Processing*, **9**(5), pp. 598-602, (July 2001).
9. M. T. Akhtar, M. Abe and M. Kawamata “A New Variable Step Size LMS Algorithm-Based Method for Improved Online Secondary Path Modeling in Active Noise Control Systems” *IEEE Transactions On Audio, Speech, and Language Processing*, **14**(2), pp. 720-726, (March 2006).
 10. D. Zhou, V. DeBrunner, “Efficient adaptive nonlinear filters for nonlinear active noise control, *IEEE Transaction on Circuits and Systems I*, **54**(3), pp. 669–681, (March 2007).
 11. P. Davari and H. Hassanpour “Designing a New Robust On-Line Secondary Path Modeling Technique for Feedforward Active Noise Control Systems” *Elsevier Signal Processing*, **89**, pp 1195-1204, (2009).
 12. H. Hassanpour and P. Davari “An efficient online secondary path estimation for feedback active noise control systems” *Elsevier Digital Signal Processing*, **19**, pp. 241–249. (2009).
 13. Ardekani, I. T. and Abdulla, W. H. “On the convergence of real-time active noise control systems” *Elsevier Journal on Signal Proceesing*, in Press, pp. 1-13. (2010).
 14. E. Bjarnason “Analysis of the filtered-x LMS algorithm” *IEEE Transactions on Speech and Audio Processing*, **3**(6), pp. 504–514, (Nov 1995).
 15. L. Vicente and E. Masgrau “Novel FxLMS convergence condition with deterministic reference” *IEEE Transactions on Signal Processing*, **54**(10), pp. 3768–3774, (Oct. 2006).
 16. Y. Xiao, A. Ikuta, L. Ma and K. Khorasani “Stochastic analysis of the FXLMS-based narrowband active noise control system” *IEEE Transactions on Audio, Speech and Language Processing*, **16**(5), pp. 1000–1014, (July 2008).
 17. I. Tabatabaei Ardekani and W. Abdulla “Theoretical convergence analysis of FxLMS algorithm” *Signal Processing*, **90**(12), pp. 3046–3055, (2010).
 18. B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hern, J. R. Zeidler, E. Dong and R. C. Goodlin, “Adaptive Noise Cancelling: Principles and Applications” *Proceeding IEEE*, **63**, pp. 1692-1716, (Dec 1975).
 19. S. Haykin, *Adaptive Filter Theory*, 4th ed., Prentice-Hall, New Jersey. (2002).
 20. R. Mustafa and M. A. Mohd Ali, “Fast and Efficient Least Mean Square Algorithm for Active Noise Control System Identification” *Acoustical Letter, Acoust. Sci. & Tech.* **33**(2), pp. 111-112, (2012).