Active Noise Reduction using LMS and FxLMS Algorithms

Krishna A/L Ravinchandra¹, Ka Fei, Thang² and Chee Yong, Lau³

¹School of Engineering, Faculty of Computing, Engineering and Technology,  
²³Asia Pacific University of Technology & Innovation

³k9999_5407@hotmail.com

Abstract—In recent times, the use of industrial Heating, Ventilation, and Air Conditioning (HVAC) systems has had a substantial increase due to various social-economic reasons. Various studies are conducted to address the concerns of noise generated by HVAC systems in particular from compressors; noise pollution affects the environment we live in and should therefore be mitigated as much as possible. For this purpose, an active noise reduction system is introduced to help alleviate noise levels at both low and high frequency ranges of the HVAC system. This paper demonstrates an implementation of Active Noise Reduction (ANR) system using Least Mean Square (LMS) and Filtered-x Least Mean Square (FxLMS) algorithms, which have been tested based on simulated noise scenarios. A comparison between both algorithms will be performed based on simulation of different frequency ranges which closely resembles the operating scenario of the industrial HVAC systems. Also, different values of step-size affecting the processing of the ANR system during the convergence rate and stability state will also be investigated and discussed.

Index Terms—Active Noise Reduction, Filtered-x Least Mean Square, HVAC, Least Mean Square, Noise Pollution

1. Introduction

In this modern era of technology, the demand for HVAC technologies worldwide is expected to reach 105.5 million units in 2017 as compared to the previous year which was at 102.31 million units; this presents, a percentage increase of 3.11% [1]. The increase in demand infers to a wide use of HVAC systems such as in offices, homes and even factories. This is because HVAC systems help in the cooling and heating of a room allowing occupants control over their environment. The system involves seven different processes namely heating, cooling, humidifying, dehumidifying, cleaning, ventilating and air movement; the relevance and requirements of each of these processes differs greatly [2]. The HVAC system can create a noise which affects our environment as well as the durability of the system. To address this problem, the Passive Noise Control (PNC) was introduced to handle the noise from an HVAC system. This was adopted as the conventional method, where barriers and enclosure are used to limit noise exposure.

Unfortunately, the PNC is only able to tackle the higher frequency range of noise. Hence, the Active Noise Control (ANC) was introduced to control the lower and higher frequency noise efficiently and
adaptively. The ideal way to describe ANC is, where a 180 degree phase signal (anti-noise) generated is used to destructively interfere with the unwanted noise [3]. Basically, it is a system that helps reduce the unwanted noise coming from the HVAC system. Thus, the ANC system is implemented to best handle the lower and higher frequency range of noise. Figure 1 below shows the active noise reduction and how noise is being reduced.

![Figure 1: Active Noise Reduction](image)

Essentially, the noise source undergoes a process of computation, which creates an anti-noise. The sensor is then detecting the noise source and commands a control process that creates an anti-noise wave which is 180 degrees out of phase with the noise source. It is impossible to create a pure silence in an environment using the ANC system but it has the potential to achieve a considerable amount of noise reduction. Hence, the ANC system is suitable in the construction of a more environmental friendly HVAC system.

Hence, in ANC system the adaptive filters is introduced and primarily used in the communication system. Its performance has encouraged the development of better solution to improve convergence rate and steady state error. In [5] [6] the authors solely discuss different noise reduction approaches using adaptive filters. As discussed above, the aim of this paper is to demonstrate an implementation of ANR system using LMS and FxLMS algorithm.

1.1 LMS Algorithm

![Figure 2: Block diagram of LMS algorithm](image)

Widrow and Hoff developed the LMS algorithm for the approach of noise reduction [8]. Figure 2 shows the block diagram of LMS algorithm on how the anti-noise processing works and the definition of each symbol is shown in Table 1 [7].

| Symbols | Definitions |
|---------|-------------|
| x(n)    | Noise signal |
| H(z)    | Primary path transfer function |
| d(n)    | Primary noise signal at the error microphone |
| e(n)    | Modified error signal |
| W(z)    | LMS Adaptive Filter |
| y(n)    | Output of adaptive filter |
A gradient descent is used in this algorithm which estimates a time varying signal. Once this method identifies the minimum, the order to minimize the error is then computed by adjusting the filter coefficient. In order to find the divergence of a function the gradient of the del-operator is applied, which in this case is the error with respect to the n-th coefficient. The error signal is that the distinction between the desired \( d(n) \) and output \( y(n) \). The following equation shows the elements of LMS,

Weights evaluation

\[
w_t(n+1) = w_t(n) + \mu \ast e(n) \ast x(n-i)
\]  

(1)

Filtering output

\[
y(n) = \sum_{i=0}^{M-1} w_t(n) \ast x(n-i)
\]  

(2)

Error estimation

\[
e(n) = d(n) - y(n)
\]  

(3)

where the weight estimation of LMS adaptive algorithm, output signal \( y(n) \) and the error signal \( e(n) \) are given by (1), (2) and (3) respectively, the equation are explained in meticulous manner by Dixit and Nagaria [9].

1.2 FXLMS Algorithm

Figure 3: Block diagram of FXLMS algorithm [10].

Widrow, Shur & Shaffer proposed the integration of a secondary path model in the reference signal path (from speaker to error microphone) [11]. Figure 3 shows the block diagram of FXLMS algorithm presented by Kuo and Morgan on how the noise reduction algorithm works and the definition of each symbol is shown in Table 2 [10].

| Symbols | Definitions |
|---------|-------------|
| x(k)    | Noise signal |
| xs(k)   | Noise signal combined with assumed Sh(z) based on S(z) |
| P(z)    | Primary path transfer function |
| yp(k)   | Primary noise signal at the error microphone |
| e(k)    | Modified error signal |
| S(z)    | Secondary path transfer function |
| C(z)    | Controller for the FxLMS algorithm |
The FXLMS algorithm is a straightforward variant of the LMS algorithm, thus the reason of it being selected based on the minimization of mean square error condition. Haykin has provided a thorough analysis on this adaptive algorithm in a textbook [12]. For this adaptive filter, there are two different control system, which is feedback and feedforward which is distinguished by the reference signal x(n). Figure 4 shows the generally how the feedforward approach uses a different microphone to measure the signal at output. The feedback uses reference signal based on the feedback predictor which can attenuate the components of the unwanted noise by predicting, thus although the feedback is less robust but it is more compact and cost effective compared to feedforward [13].

In order to derive the FxLMS algorithm, the similar method of LMS algorithm is used but with steepest descent, the following update equation can lead to this minimization:

\[ W_{\text{New}} = W_{\text{Old}} + \mu \nabla J(n) \]  
(1)

Where \( W \) is the controller weight error, \( \mu \) is an adaption step size (scalar), \( \nabla \) denotes the gradient operator (with respect to \( W \) parameters), \( J(n) \) is the power of error signal.

The derivation of \( \nabla J(n) \),

\[ J(n) = \mathbb{E}\{e^2(n)\} \]

Where \( \mathbb{E}\{.\} \) denotes statistical expectation operator and \( \mathbb{E}\{.\} \) is a theoretical function. To avoid this operator, \( J(n) \) is approximated by

\[ J(n) \approx e^2(n) \]
Then, estimate $\nabla J(n)$ as follows,

$$\nabla J(n) = \nabla e^2(n)$$

(2)

Now to estimate $\nabla e(n)$, the derivation is as follows based on the block diagram,

$$e(n) = d(n) + s(n) \ast y(n)$$

(3)

Now to estimate $\nabla y(n)$, the derivation is as follows based on the block diagram,

$$y(n) = W^T x(n)$$

(4)

Now $\nabla y(n)$ can be expressed by,

$$\nabla y(n) = x(n)$$

(5)

Substitute (4) into (3),

$$\nabla e(n) = s(n) \ast x(n)$$

(6)

Substitute (5) into (2),

$$\nabla J(n) = 2e(n).s(n) \ast x(n)$$

(7)

The reference signal is filtered by $\hat{s}(n)$ before passing through the standard LMS algorithm. Therefore resulting the compensation for secondary path. $s(n)$ should be estimated through off-line or online secondary path techniques. If $\hat{s}(n)$ denotes an estimate of $s(n)$, then

$$W_{\text{New}} = W_{\text{Old}} + 2\mu e(n).\hat{s}(n) \ast x(n)$$

OR

$$W_{\text{New}} = W_{\text{Old}} + 2\mu e(n).x_f(n)$$

The stability of the FXLMS algorithm is highly dependent on the $x_f(n)$ power where the convergence rate is directly proportional to the step-size and this parameter is indirectly proportional to the steady state performance.

2. Methodology

In this section, the ANR process using LMS and FXLMS algorithm will be discussed. The main components involved to build an ANR are HVAC compressor as noise control application, microphone as a sensor for recording noise and error signal and speakers as an actuator which is the anti-noise source.

Figure 5: Proposed ANR Block Diagram

Figure 5 shows the proposed ANR block diagram for the LMS with FXLMS algorithm. $P(z)$ is the primary path, $x(n)$ is the noise signal in the form of low frequency noise generated by the compressor, $y(n)$ is the generated anti-noise signal based on the algorithm used for the ANR process and $e(n)$ is the residual noise signal which is then undergoes another filtration and speakers. The error microphone keeps
receiving signal until the noise is reduced to a suitable dB level. This block diagram determines how the flow of the noise signals and anti-noise signal flow.

A program is developed for the ANR system for LMS and FXLMS algorithm, the high level tool used is MATLAB. The program is tested with different step size value to allow the understanding of how the ANR perform differently according to the values. The programming process is demonstrated in the form of flowchart for both LMS algorithm and FXLMS algorithm, where the flowchart is drawn based on the MATLAB programming steps.

Figure 6: Flowchart of (a) LMS and (b) FXLMS

Figure 6 shows the flowchart of LMS and FXLMS, for the LMS, firstly the system receives an input signal which is the simulated noise or singleton noise. Then, the signal undergoes LMS algorithm and is later output through the speakers for anti-noise signal, this system runs as a real time until user stops the ANR process except for simulation which depends on how many data sample (in seconds) would like to be used. A graph is drawn with the noise signal, desired signal and error where evaluations on the performance can be done based on different stepsizes. For the FXLMS, the flow of the system is the same as LMS algorithm but with an additional procedure which is the FXLMS algorithm.

3. Simulation and Singleton Testing

Figure 7: Setup for Singleton Testing (a) Front View and (b) Top View

In this section, the simulation and singleton testing is discussed using three different step sizes for both LMS and FXLMS algorithm. For the simulation, the noise signal is generated using the MATLAB software which is then feed into the respectively algorithms; frequency used for simulation is 500Hz, 1500 Hz and 2500 Hz. Figure 7 shows the setup for the singleton testing which was conducted in an
acrylic container where there is four components the noise signal speaker, microphone, anti-noise signal speaker, dB meter and also a laptop. The noise signal speaker is generated using a frequency generator application on a mobile phone, then the signal is recorded by the microphone which is connected to the laptop that runs the ANR process system using LMS and FXLMS algorithm; the output signal after processing with algorithm is then pushed to the anti-noise signal speaker. And the dB reduction is measured using a dB meter.

3.1 LMS based on simulation

Figure 8 shows the graphs of three different step sizes used to simulate ANR process using LMS Algorithm. The simulation period used is 300 seconds. The noise signal is represented in blue, the desired output is represented in dotted line red and the error is represented in green. Figure 8(a) shows that the LMS algorithm takes 25 seconds to reach low error amplitude while Figure 8(b) takes 4 seconds and Figure 8 (c) takes less than a second. This is because Figure 8(a), Figure 8(b) and Figure 8(c) uses 0.00001, 0.0001, 0.001 as their step size respectively. It can be observed that the error rate reduces upon increasing the step size of the LMS algorithm system, this shows that the convergence rate is good but the steady-state is affected.
3.2 FXLMS based on simulation

Figure 9: Graphs of FXLMS Algorithm Simulation using different stepsizes (a) Stepsize= 0.00001, (b) Stepsize= 0.0001 and (c) Stepsize= 0.001 for 300 seconds.

Figure 9 shows the graphs of three different stepsizes used to simulate ANR process using LMS Algorithm. The simulation period used is 300 seconds. The color code representation is the same as per LMS algorithm figure. Figure 9(a) shows that the LMS algorithm takes 160 seconds to reach a low error amplitude while Figure 9(b) takes 30 seconds and Figure 9(c) takes less 3 second. This is because Figure 8(a), Figure 8(b) and Figure 8(c) uses 0.00001, 0.0001, 0.001 as their stepsize respectively. It can be observed that the error rate reduces upon increasing the stepsize of the FXLMS algorithm system, this shows that the convergence rate is good but the steady-state performance is affected.

3.3 LMS based on singleton testing
Table 3: Result of different stepsize of singleton testing with LMS algorithm.

| Frequency (Hz) | Stepsize = 0.00001 | Stepsize = 0.0001 | Stepsize = 0.001 |
|---------------|---------------------|-------------------|------------------|
|               | ANR Off (dB) | ANR On (dB) | ANR Off (dB) | ANR On (dB) | ANR Off (dB) | ANR On (dB) |
| 200           | 75.0      | 74.9     | 75.0       | 75.5       | 74.9       | 74.8       |
| 400           | 88.3      | 88.2     | 88.3       | 88.6       | 88.3       | 88.5       |
| 600           | 81.9      | 81.8     | 81.5       | 81.9       | 81.7       | 81.5       |
| 800           | 89.0      | 88.9     | 89.0       | 88.9       | 89.0       | 88.9       |
| 1000          | 86.0      | 85.8     | 86.0       | 85.9       | 85.9       | 85.7       |
| 1200          | 104.9     | 104.8    | 104.8      | 105.8      | 104.8      | 104.8      |
| 1500          | 90.5      | 90.6     | 90.5       | 90.3       | 90.4       | 90.6       |

Table 3 shows the result for singleton testing for the LMS algorithm using three different stepsize values which are 0.00001, 0.0001 and 0.001. The best results achieved among the three stepsizes using 0.00001, where the steady-state performance to attain the dB reduction is high but the convergence rate of the system is slow.

3.4 FXLMS based on singleton testing

Table 4: Result of different stepsize of singleton testing with FXLMS algorithm.

| Frequency (Hz) | Stepsize = 0.00001 | Stepsize = 0.0001 | Stepsize = 0.001 |
|---------------|---------------------|-------------------|------------------|
|               | ANR Off (dB) | ANR On (dB) | ANR Off (dB) | ANR On (dB) | ANR Off (dB) | ANR On (dB) |
| 200           | 75.0      | 74.2     | 75.3       | 75.1       | 75.2       | 74.8       |
| 400           | 88.3      | 88.1     | 88.4       | 88.2       | 87.6       | 87.4       |
| 600           | 81.5      | 81.5     | 81.5       | 81.0       | 81.6       | 81.0       |
| 800           | 88.9      | 88.9     | 88.0       | 87.5       | 88.2       | 87.6       |
| 1000          | 85.9      | 85.6     | 86.2       | 86.0       | 86.0       | 85.6       |
| 1200          | 104.9     | 104.5    | 105.2      | 105.0      | 104.9      | 104.5      |
| 1500          | 89.9      | 89.6     | 89.6       | 89.5       | 88.4       | 88.0       |

Table 4 shows the result for singleton testing for the FXLMS algorithm using three different stepsize values which are 0.00001, 0.0001 and 0.001. The best results achieved among the three stepsizes using 0.001, even though at 200 Hz for the stepsize 0.00001 a dB reduction of 0.8 was achieved which was the best among all frequencies in each stepsize but dB reduction is to be measured in overall at all frequencies. The performance of the steady-state using 0.001 as stepsize shows a consistent dB reduction for all frequency but slow convergence rate still.

3.5 Summary for Simulation and Testing
Figure 10: Graph to determine convergence rate and steady-state performance based on stepsize [14].

As observed in the simulation results for LMS and FXLMS algorithm, the steady-state performance is affected upon increasing the stepsize value. Figure 10 shows the convergence rate and steady-state performance when using different step size value. Where the slow convergence rate occurs when small stepsize is used but steady-state performance is high but when using big stepsize value, the steady-state performance is low and the convergence rate is fast. Thus, to handle with this trade off, a flexible step-size can be used. In the transient conditions, the step-size is set to a moderately big number and it is reduced bit by bit while the system converges to its steady state level. The singleton testing resultshas the same outcome as the simulation results where the stepsize affects the performance of steady-state and convergence rate of the system.

4. Conclusions

To conclude, this paper demonstrated the ANR system using LMS and FXLMS algorithm with simulation and singleton testing using three different stepsizes. Simulations results shows both the LMS and FXLMS algorithm performance was at the best when using 0.001 as its stepsize. Singleton testing for both LMS and FXLMS algorithm shows that each frequency for all the stepsize has different dB level, which makes it easy to catch on which frequency is the difficult to reduce its dB level can be used as a point of concentration. LMS algorithm steady-state performance was at the best when using 0.00001 as it stepsize and FXLMS algorithm steady-state performance was at the best when using 0.001 as its stepsize but LMS has a better convergence but FXLMS has poor convergence rate. Besides, the future recommendations that can be taken into consideration is by using multiple frequency tone testing, different position of microphone and speakers and also implementation towards the industrial HVAC system.

References

[1] JRAIA. 2015. Statistics. [ONLINE] Available at: http://www.jraia.or.jp/english/statistics/index.html. [Accessed 26 July 2017].

[2] McDowall, R., 2007. Fundamentals of HVAC systems: SI edition. Academic Press.

[3] Rajesh, M., Jeevamalar, J. and Jancirani, J., “Active noise reduction of automotive HVAC system using filtered LMS [Part-1 sound measurement],” In IEEE-International Conference on Advances in Engineering, Science and Management (ICAESM-2012).

[4] Swain, A. "Active Noise Control: Basic Understanding". Research Gate (2013): 1-19.

[5] Simon Haykin, “Adaptive filter theory,” Pearson Education Publication, 4th Edition, May 2012, pp. 231-278.

[6] Paulo S.R. Diniz, “Adaptive Filtering Algorithms and Practical Implementation,” Kluwer Academic Publications, 3rd Edition, pp. 77-189.

[7] J. Jintu K and P. U, "Implementation of LMS-ALE Filter Using Vedic Algorithm," Journal of Electrical & Electronic Systems, vol. 5, 2016.

[8] Widrow B, Hoff ME. Adaptive switching circuits. InIRE WESCON convention record 1960 Aug 4 (Vol. 4, No.1, pp. 96-104).

[9] S. Dixit and D. Nagaria, "LMS adaptive filters for Noise Cancellation: A Review," International Journal of Electrical and Computer Engineering (IJECE), vol. 7, pp. 2520-2529, 2017.

[10] S. M. Kuo and D. R. Morgan, "Active noise control: a tutorial review," Proceedings of the IEEE, vol. 87, pp. 943-973, 1999.

[11] B. Widrow, "D. Shur, and S," in Shaffer, "On adaptive inverse controls," in Proc. 15th Asilomar Conf Circuits, Syst., Comput, 1981, pp. 185-189.

[12] S. Haykin, "Adaptive Filter Theory, 3rd-ed," Printice Hall, 1996.
[13] I. T. Ardekani and W. Abdulla, "FxLMS-based active noise control: A quick review," in Proceedings of, 2011, pp. 1-11.

[14] I. T. Ardekani and W. Abdulla, “Active Noise Control: Fundamentals and Recent Advances,” Asia Pacific Signal and Information Processing Annual