

A REVIEW OF ACTIVE NOISE CONTROL ALGORITHMS TOWARDS A USER-IMPLEMENTABLE AFTERMARKET ANC SYSTEM

Marko Stamenovic

University of Rochester
Department of Electrical and Computer Engineering

mstameno@ur.rochester.edu

ABSTRACT

The past 30 years have shown a steady uptick in the application of Active Noise Control (ANC) technology to the practical problem of noise pollution in various fields including audio, transportation, medicine, appliances, industrial and more. Part of this success stems from the exponential increase in the computational power of modern microprocessors and their corresponding decrease in size, power consumption and price. These advances have allowed for the proliferation of relatively low-cost embedded applications of ANC systems. However, whereas in the past it may have been necessary to use custom embedded silicon for ANC signal processing, it may now be possible to implement using low-cost off-the-shelf non-embedded hardware such as a single-board computer or mobile phone. This paper is a survey of various active noise control algorithms with an eye towards eventual application in a user-implementable aftermarket ANC system on off-the-shelf hardware.

Index Terms— Active Noise Control, Active Noise Cancellation, ANC, LMS, FxLMS, NLMS, FuLMS

1. INTRODUCTION

1.1. Overview

Active noise attenuation is a technique which uses live measurement of an offending signal and digital signal processing to synthesize a new signal containing destructive interference in order to cancel the unwanted noise. The idea of attenuating noise actively first appeared in a 1936 patent [9] in which Lueg proposed such a system utilizing a microphone and electronic loudspeaker. ANC includes many benefits over passive noise control systems of acoustic isolation including lighter weight, smaller size, and ability to target only specific offending frequencies.

Figure 1 shows the most basic physical orientation of an ANC system. The system consists of a reference microphone at the noise source, a canceling loudspeaker, and an error microphone at the listener's ears or location where cancellation

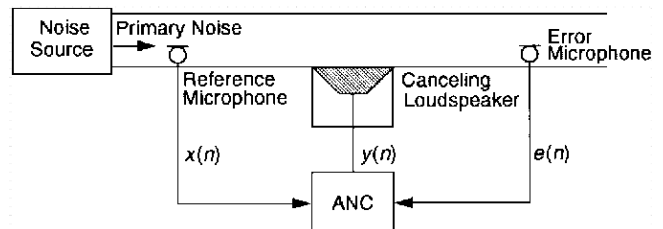


Fig. 1. Simple ANC system (reproduced from [5]).

is required. The primary microphone and reference microphone are used to update the ANC algorithm which then outputs an updated cancellation signal.

Since the characteristics of the unwanted noise signal may be time-varying in frequency, amplitude and phase, an ANC system must adapt to this changing signal in order to effectively cope. The Least-Mean-Squares (LMS) algorithm is an efficient and popular solution to the problem of updating the anti-noise signal. LMS for ANC was first proposed in the mid 20th century in Wiener's classic work[10] and remains one of the most popular algorithms for ANC today due to its simplicity and robustness. The LMS algorithm is a stochastic gradient-descent based algorithm. It utilizes the gradient vector of the active filter weights to converge on the optimal Wiener solution [2], thus minimizing the error signal. With each iteration of the LMS algorithm, the filter weights of the adaptive filter are updated according to Formula 1. In this formula, $\mathbf{w}(n)$ represents the weights of the adaptive filter, μ represents the learning rate or step size of the LMS algorithm, $\mathbf{x}(n)$ represents the reference microphone input buffer, and $e(n)$ represents the instantaneous error.

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

1.2. Practical Considerations

In practical noise control applications, the so-called secondary path signal - the path from the canceling speaker to the error microphone - introduces considerable phase and fre-

quency distortion which can cause LMS not to successfully converge. Thus a modified LMS algorithm, filtered extended LMS or FxLMS[1], is introduced to cope with the secondary path signal problems. A second issue which can arise in practical applications is output from the cancellation speaker bleeding into the reference sensor, which can cause feedback loops. A second modified LMS algorithm, filtered-u LMS or FuLMS[7], is introduced to counteract issues caused by the feedback.

1.3. Learning Rate

The learning rate of the LMS algorithm is one aspect that greatly influences the convergence rate of the adaptive filters and thus the performance of the overall ANC system. Although the regular LMS function uses a constant learning rate, many algorithms have been developed over the years in order to speed convergence with variable step size learning rates. We focused on one such algorithm, the normalized LMS or NLMS algorithm[8]. NLMS is a variant of LMS that updates the step size in proportion to the inverse of the total expected energy of the input buffer. This can also be expressed as the inverse of the dot product, or L2 norm of the input vector with itself. The formula for NLMS is shown in Equation 2 and the equation for just the learning rate update is shown in Equation 3.

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{\|\mathbf{x}(n)\|^2} \mathbf{x}(n)e(n) \quad (2)$$

$$\mu(n) = \frac{\mu}{\|\mathbf{x}(n)\|^2} \quad (3)$$

1.4. Goals

The goal of this paper is to review several variants of the LMS algorithm for application to a future user-implementable after-market ANC system. The algorithms were evaluated based on 1) convergence rate 2) robustness for various use-cases and 3) ease of use.

2. METHODS

The three above-mentioned ANC algorithms - LMS, FxLMS, and FuLMS - were implemented in the Matlab environment for both constant and variable step sizes. This provided six overall: LMS, NLMS, FxLMS, FxNLMS, FuLMS and FuNLMS. The algorithms were tested against synthetic linear transfer functions, white noise and synthetic feedback. In order to compare convergence rates, all algorithms were tested with initial step sizes of $\mu = [0.01, 0.05, 0.1, 0.5, 1, 2]$. The sample rate was set at 8kHz with a system input white noise 10k samples long. In order to get a more general idea of the convergence rates, each test was run 100 times and averaged. The results are plotted as the logarithmic peak envelope of the average convergence rate using a 250 point sliding window.

In addition, the LMS and FxLMS algorithms were implemented on the Texas Instruments OMAP-L138 embedded DSP board to better understand the challenges of programming for a real-time ANC environment. Algorithms on the OMAP-L138 were tested using a synthetic transfer function consisting of a third order low-pass filter, pink noise, and various recorded signals such as an idling truck and an airplane cabin noise.

3. ALGORITHMS

3.1. LMS

LMS, discussed at some length in the introduction, is the basis of most active noise control systems today. A block diagram of the LMS algorithm in an ANC application is shown in Figure 2. The signal is picked up at a reference microphone $x(n)$ and sent through an adaptive filter $W(z)$ which is constantly being updated by LMS. LMS uses stochastic gradient descent to update the weights of $W(z)$ such that noise at the error microphone $e(n)$ is minimized. Thus, as $e(n) \rightarrow 0, W(z) \rightarrow P(z)$.

Results of our convergence rate testing are shown in Figure 3. Although the best convergence times are comparable between best case LMS and NLMS, the results show that NLMS is more tolerant to initial convergence rate estimates. Many of the systems in Figure 3a have become unstable and thus are not shown on the graph, while all systems in 3b are at least stable, if not convergent.

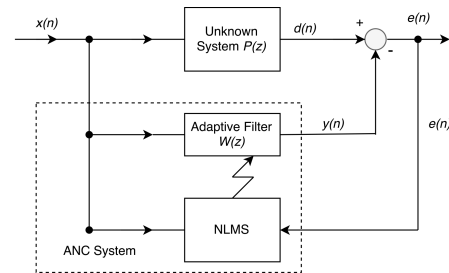


Fig. 2. Block diagram of an ANC system using the LMS algorithm.

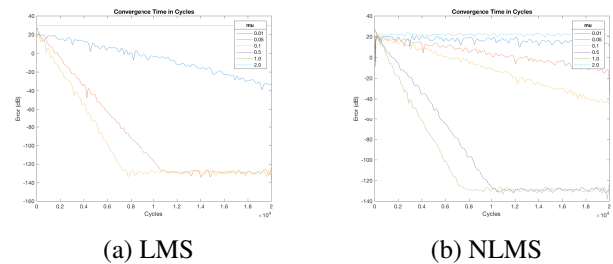


Fig. 3. Convergence times for LMS and NLMS algorithms.

3.2. FxLMS

The use of the LMS algorithm in practical ANC applications is complicated by the fact that the antinoise created by the algorithm must travel from the cancellation speaker to the error microphone and thus transition from the digital domain to the analog domain and back again. This introduces frequency and phase distortions into the signal which are known cumulatively as the secondary path $S(z)$. The secondary path includes any signal components between the output of the adaptive filter and the input of the error signal to the LMS algorithm which includes the D/A converter, reconstruction filter, amplifier, loudspeaker, acoustic path from loudspeaker to error microphone, pre-amplifier, anti-aliasing filter, and A/D converter [5].

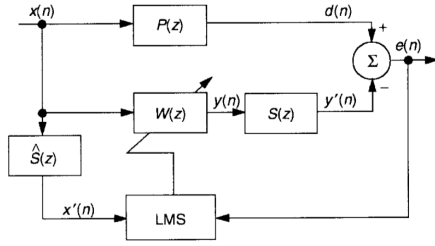


Fig. 4. Block diagram of an ANC system using the FxLMS algorithm (reproduced from [5]).

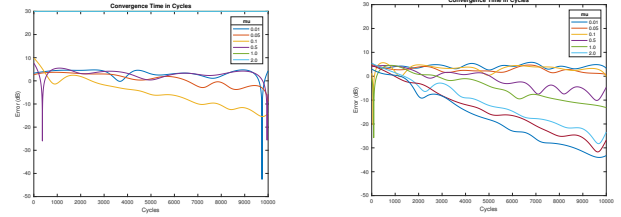
A block diagram of the FxLMS algorithm is shown in Figure 4. To counteract distortion introduced by $S(z)$, an inverse filter $\hat{S}(z)$ is placed between the reference microphone and the LMS algorithm. This allows for the algorithm to converge. In order to obtain the impulse response of $S(z)$, the system must be tested offline before ANC is implemented. Although there are many ways to determine $S(z)$, an easy way is to simply use the LMS algorithm with white noise as an input to the system. The inverse impulse response is obtained by bringing the impulse response of $S(z)$ into the frequency domain, taking its inverse and then bringing it back into the time domain. The equations to update the filter weights using the FxLMS algorithm is shown below in Equation 4 where $\mathbf{x}'(n) = \hat{s}(n) * \mathbf{x}(n)$ - the convolution of the inverse secondary path and the input vector - and $\hat{s}(n)$ is the impulse response of the inverse estimated secondary path filter $\hat{S}(z)$.

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \mathbf{x}'(n) e(n) \quad (4)$$

Results of convergence rate testing for FxLMS are shown in Figure 7. The convergence times are much better for FxLMS with the variable step size algorithm. Also, as with LMS, the normalized variant is more tolerant to different convergence rate estimates. The results also show that the FxLMS algorithm converges to an absolute minimum at a slower rate than the LMS algorithm. However, for practical

use, both algorithms come very close to zero error (0 dB) at around 1500 cycles.

In testing, it was observed that convergence could be speeded by varying learning rate between the secondary path identification step and the actual running of the system. For example, in the constant learning rate tests, using a smaller learning rate for secondary path identification and then a larger step size while running the system showed comparable results to the LMS algorithm alone. For variable step size, the learning rate converged too slowly if updated by the input vectors $\mathbf{x}(n)$ alone, as shown in Figure 5b. However, the system became unstable for most initial values of μ if $\mathbf{x}'(n)$ was used to update $\mu(n)$ instead of $\mathbf{x}(n)$. A compromise that showed convergence rates comparable to LMS was observed by the taking the square root of the mean of $\mu(n)$ and $\mu'(n)$ as calculated by $\mathbf{x}(n)$ and $\mathbf{x}'(n)$, respectively. This was the result of heuristic tests and does somewhat defeat the purpose of an automatically variable step size, but may be useful in later implementations.



(a) FxLMS

(b) FxNLMS

Fig. 5. Convergence times for FxLMS and FxNLMS algorithms.

3.3. FuLMS

The FuLMS algorithm addresses another practical problem in real world implementation of ANC; the issue of feedback to the reference microphone from the cancellation speaker. The FuLMS algorithm adds an adaptive recursive IIR filter $B(z)$ to the signal chain, whose function is to minimize error based on a one sample delayed version of $\hat{\mathbf{y}}'(n)$ as shown in Figure 6.

The recursive update equations for the FuLMS system are shown in equations 5-7 where $A(z)$ is the primary FIR adaptive ANC filter, $B(z)$ is an IIR filter to remove feedback and $\hat{\mathbf{y}}'(n-1) = \hat{s}(n) * \mathbf{y}(n-1)$ is a one sample delayed version of the cancellation signal filtered through the inverse of the secondary path signal filter.

$$\mathbf{y}(n) = \mathbf{a}^T(n) \mathbf{x}(n) + \mathbf{b}^T(n) \mathbf{y}(n-1) \quad (5)$$

$$\mathbf{a}(n+1) = \mathbf{a}(n) + \mu \mathbf{x}'(n) e(n) \quad (6)$$

$$\mathbf{b}(n+1) = \mathbf{b}(n) + \mu \hat{\mathbf{y}}'(n-1) e(n) \quad (7)$$

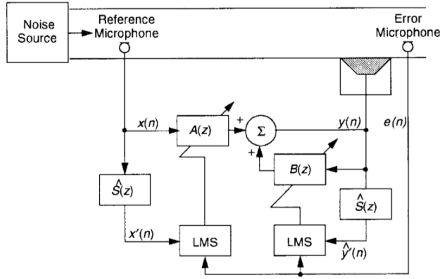


Fig. 6. Block diagram of an ANC system using the FuLMS algorithm (reproduced from [5]).

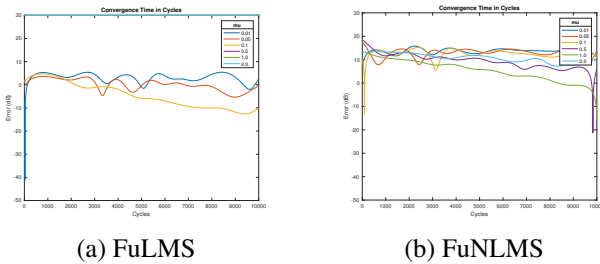


Fig. 7. Convergence times for FuLMS and FuNLMS algorithms.

The FuLMS algorithm does come with some drawbacks, most notably that it has never been mathematically proven to guarantee convergence. This stems from the fact that the error function may converge to local minima due to the non-quadratic nature of IIR filters [5]. Additionally, IIR filters are not unconditionally stable, which may affect the overall system as well. However, a variant of the FuLMS algorithm known as SHARF [7] has been shown to be exceptionally stable. In this version, a low-pass filter is used to smooth the error signal fed to the LMS algorithm updating $B(z)$.

For our reported results, we added 80% of the sound from the canceling speaker back into the reference microphone to simulate a high level of feedback. However, the algorithm was observed to converge over various levels of feedback.

As shown in Figure 7, the convergence trends between constant and variable step sizes are different for FuLMS than the other algorithms tested in that the variable step size version converges more slowly than the constant step size variant. This may be due to the fact that there are two adaptive filters in FuLMS which are both sharing the same updated μ value. Perhaps the initial μ should be chosen independently for the IIR filter. Convergence may also vary with the amount of feedback added to the reference microphone.

4. EMBEDDED TESTS

In addition to our offline algorithm, we conducted tests of the LMS and FxLMS algorithms on the OMAP L-138 embedded DSP board using an 8 kHz sample rate and 256-point circular input buffer and filter tap size. For our initial test we used the LMS algorithm with 3 different noise signals - pink noise, recorded airplane cabin noise, and recorded diesel engine idling noise - to test the convergence time under different kinds of semi-periodic tones. The transfer function of the system $P(z)$ was modeled by a third order Chebyshev FIR low pass filter. We found that convergence was relatively quick for our pink noise signal, approximately 0.6 seconds to find a minimum. The convergence on the airplane cabin signal took significantly longer, approximately 4 seconds until the error stopped decreasing significantly. The diesel engine took the longest of the three at over 12 seconds. The discrepancy in time is likely due to the more complex tones and higher variance of the airplane cabin and diesel engine, respectively.

We created an experimental setup similar to the one reproduced in Figure 8 in order to test our FxLMS implementation using actual acoustic transfer functions and recordings rather than synthetic ones. However, we were not successful in measuring a useful secondary path signal. This was due in part to a) feedback of the canceling speaker into the reference microphone b) poor acoustics in the shared laboratory - constant outside noise contaminating both microphones c) the inflexibility of the DSP board to be used in different locations - no onboard disk space, so the board is tethered by location to a windows PC.

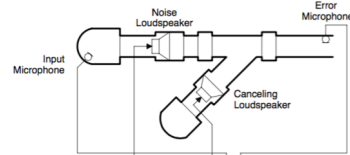


Fig. 8. One dimensional ANC experimental setup (reproduced from [6]).

5. RESULTS AND CONCLUSION

Our testing showed that all six algorithms tested will converge in the digital domain with synthetic noise as input when given the correct value of μ . We showed that the simple LMS algorithm will converge quickest of all three tested, with the FxLMS algorithm being the second fastest and the FuLMS algorithm the slowest. We also showed that in most cases NLMS variable step size algorithms show greater likelihood of convergence and higher stability than corresponding constant step size LMS algorithms. However, this does not hold true for the FuLMS algorithm which exhibits faster convergence with constant learning rates. We also showed that the

LMS algorithm is robust enough to converge when fed various types of complex sounds such as diesel engine and airplane cabin sounds.

6. FUTURE WORK

First, we would like to extend our experiments in the acoustic domain. We plan to accomplish this by implementing an ANC system on a PC using Matlab's new audio streaming toolkit. We would also like to keep examining new and updated ANC algorithms as better more robust options, such as algorithms that do online secondary path source modeling as shown by Eriksson [3] or use that use faster methods of gradient descent as shown by Fernandez [4].

7. REFERENCES

- [1] . C. Burgess, Active adaptive sound control in a duct: A computer simulation, *J. Acoustical Soc. Amer.*, vol. 70, pp. 715726, Sept. 1981.
- [2] R. Chinaboina, D. S. Ramkiran, H. Khan, and M. Usha, Adaptive algorithms for acoustic echo cancellation in speech processing, vol. 103, no. 6. *International Journal of Electronics*, 2011, pp. 975984.
- [3] L. J. Eriksson and M. C. Allie, Use of random noise for online transducer modeling in an adaptive active attenuation system, *J. Acoust. Soc. Am.*, 1989.
- [4] A. Fernandez and P. Cobo, Artificial neural network algorithms for active noise control applications. *Sociedad Espaola de Acstica*, 2002.
- [5] S. M. Kuo and D. R. Morgan, Active noise control: a tutorial review, *Proceedings of the IEEE*, vol. 87, no. 6, pp. 943973, Jun. 1999.
- [6] S.M. Kuo et al , "Design of active noise control system with the TMS320 family", Texas Instruments, 1996.
- [7] M. Larimore, J. Treichler, and C. Johnson, SHARF: An algorithm for adapting IIR digital filters, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 428440, Aug. 1980.
- [8] J. Lee, J. W. Chen, and H. C. Huang, Performance comparison of variable step-size NLMS algorithms, *Proceedings of the World Congress on Engineering and Computer Science*, San Francisco, 2009.
- [9] P. Lueg, Process of silencing sound oscillations, U.S. Patent 2 043 416, June 9, 1936.
- [10] N. Wiener, "Extrapolation, interpolation, and smoothing of stationary time series." The MIT Press, Cambridge, MA, 1949.