A Review of Active Noise Control Algorithms

Yutao Chen[1]

***Abstract*— After more than 30 years of development, the Active Noise Control (ANC) technology has been used in many practical problems like noise pollution in various fields including audio, transportation, industry and more. In many applications of noise cancellation the change in signal is usually quite fast, hence adaptive algorithm with fast convergence rate is need. In this point of view, Least Mean Square (LMS) algorithm suits this situation best. Unfortunately, LMS algorithm has high computation complexity and stability problems. But, this technology and its application have still made great process and development in the past 30 years. Part of the reason for its success is due to the exponential increase in the computational power. At the same time, more and more efficient algorithms are proposed. Among various algorithms, LMS, Recursive Least Square (RLS) and Affine Projection Algorithm (APA) are the most commonly used ones. This paper is a survey of these active noise control algorithms. In this paper, different algorithms in LMS family, RLS and APA are explored and compared.**

Keywords — Active Noise Control, Active Noise Cancellation, ANC, LMS, RLS, APA

I. INTRODUCTION

A. Overview

Active Noise Control (ANC) was invented by Paul Lueg in Germany in 1932 and registered a patent in 1936[1]. The ideas in the patents worked in theory but they lacked the equipment back then in detection, processing, and generation of sound, thus it is hard to make products that are commercially viable. In the 1950s, Harry Olson started experimenting in making silent areas using active noise control systems to some success. Also in the 1950s, active noise control was introduced to headphones. Comparing to passive noise control systems of acoustic isolation, ANC has many benefits including lighter weight, smaller size, and ability to target only specific offending frequencies.
Figure 1 shows the most basic system identification scheme in which a digital filter $W(z)$ is used to estimate an unknown plant $P(z)$. The digital filter $W(z)$ uses the reference signal $x(n)$ to produce a predict signal $y(n)$. The predict signal $y(n)$ is compared with desired signal $d(n)$ in a acoustic duct producing an error signal $e(n)$. The error signal is used to update the parameters of the digital filter $W(z)$. The key of this process the algorithms used to update the weights of the digital filter $W(z)$ using the error signal $e(n)$. If the unknown plant is dynamic, the adaptive algorithm then has the task of continuously tracking time variations of the plant dynamics. The goal of such process is to find the adaptive filter $W(z)$ to minimize the residual error signal $e(n)$. Ideally, when the adaptive filter $W(z)$ converges, we will have $P(z) = W(z)$ for $x(n) \neq 0$ which implies that $y(n) = d(n)$. When $y(n)$ and $d(n)$ are acoustically combined, the error signal $e(n) = 0$, which results in perfect cancellation of the noise.

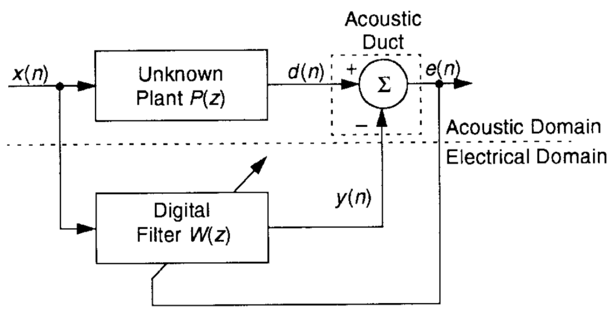

Fig. 1. Simple ANC system (reproduced from [2])

B. Algorithms

Since the the unkonwn plant may be dynamic which means it may be time-varying in frequency, amplitude and phase, an ANC system must adapt to such changes. There are several algorithms which can be used to solve this problem efficiently including Least-Mean-Square (LMS) algorithm and Recursive-Least-Squares (RLS) algorithm. There are also many modified LMS algorithm such as NLMS, Filtered extended LMS, Filtered-u LMS and Affine Projection LMS, which are designed to deal with different problems in traditional LMS algorithm.
The LMS algorithm is a efficient and popular solution to this kind of problem. LMS for ANC was first proposed in the mid 20th century in Wieners work[3] and remains one of the most popular algorithms for ANC today due to its simplicity and robustness. LMS algorithm is a stochastic gradient descent based algorithm. It uses the gradient vector of the weights to converge to the optimal Wiener solution. For each iteration, the weights are updated according to Equation 1.

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

where $w(n)$ represents the weights of the filter in time instance n, $x(n)$ represents the the input signal and $e(n)$ represents the instantaneous error. μ represents the learning rate or step size. The step size shows how big the weights will change between iterations. Big step size will result in fast convergence but may always wondering around the optimal solution instead of achieving the optimal solution. Small step

[1] University of Maryland College Park, Department of Electrical and Computer Engineering `cheny AT terpmail DOT umd DOT edu`

size have higher possibility to achieve the optimal solution but the convergence speed will be much slower. In this point of view, the step size can greatly influences the performance of the overall ANC system. The traditional LMS algorithm uses fixed learning rate through all the learning process. To achieve a better algorithm in terms of speed and performance, many algorithms with variable step size have been developed over the years. We focused on one such algorithm, the normalized LMS or NLMS algorithm[4]. NLMS algorithm updates the step size in proportion to the inverse of the total expected energy of the input signal. The expected energy of the input signal can be expressed as the dot product, or L2 norm of the input vector with itself. The formula for NLMS's updating rule is shown in Equation 2.

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

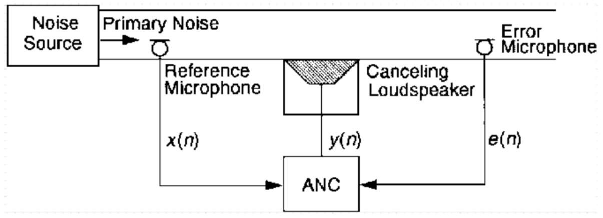

Fig. 2. Example of Practical ANC system (reproduced from [2])

A practical noise control system was shown in figure 2. In such system the path from the cancelling speaker to the error microphone will introduce considerable phase and frequency distortion which can cause the failure of LMS. To deal with this problems some modified version of LMS algorihms are proposed. Filtered extended LMS or FxLMS[5] is one of them to cope with this kind of problem. A second issue which can arise in practical applications is output from the cancellation speaker bleeding into the reference sensor. A second modified LMS algorithm, Filtered-u LMS or FuLMS[6], is introduced to counteract this issue.
An other notable algorithm in LMS algorithm family is Affine Projection Algorithm (APA). Affine projection algorithm is designed to improve the performance of other adaptive algorithms, mainly LMS-based ones, especially when input data is highly correlated. The computational cost of the affine projection algorithm depends on the projection order, which in turn conditions the speed of convergence, thus high speed of convergence implies usually high computational cost[7]. In general, APA have fast convergence speed and better tracking capabilities than LMS algorithms. But APA is less stable than LMS whereas more stable than RLS algorithm.
The RLS algorithm is another efficient algorithm where the coefficients are continually adjusted on a step-by-step basis during the filtering operation. RLS recursively finds the coefficients that minimize a weighted linear least squares cost function. This approach is in contrast to LMS algorithm that aim to reduce the mean square error. The structure of RLS filter is shown in figure 3. The RLS computes the temporal statistics directly at each time-step to determine the optimal filter coefficients. The RLS filter is adaptive and can adjust to time varying input statistics. Under most conditions the RLS filter will converge faster than a LMS filter while the algorithm is much more complex than all the other algorithms above.

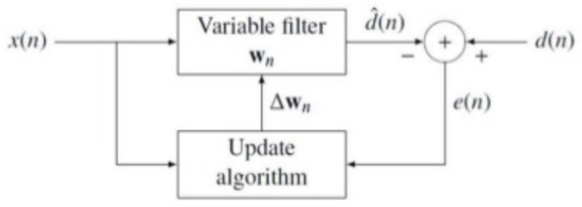

Fig. 3. The RLS filter structure

II. METHODS

For all the simulations in the next section, random noise was used as an input. We want to compare the affect of different values of learning rate μ, thus all the parameters with in the same algorithm are the same except the learning rate. For LMS and NLMS, we assume and initialize the primary path filter as a filter with linear coefficients since this path is known to us in practice. For FxLMS and FuLMS, the primary path filter has the same form as in LMS. In the plots, only the envelop of the error signal are shown instead of the real error signal. In affine projection algorithm, the projection order is set to be four for all different μs. In RLS, the "forgetting factor" λ is set to be 1 and the value δ which is to initialize P matrix is set to be 0.005.

III. ALGORITHMS

A. LMS

LMS, discussed at some length in the introduction section, is basis of most active noise control system today. The signal-flow graph of LMS algorithm in ANC system is shown in figure 4.

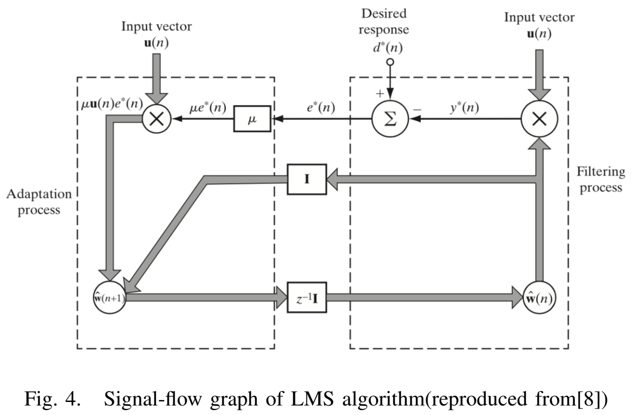

Fig. 4. Signal-flow graph of LMS algorithm(reproduced from[8])

In the filtering process, the output $y^*(n)$ is produced by filtering the input vector $u(n)$ with the estimated weight vector $\widehat{w}(n)$. The the output $y^*(n)$ is compared with the desired response $d^*(n)$ resulting in the error signal $e^*(n)$. In the adaption process, the estimated vector $\widehat{w}(n)$ is updated by increase an amount equals to the product term $\mu u(n)e^*(n)$. The LMS algorithm is shown in algorithm 1.

```
Parameter: N = filter order; $\mu$ = step size;
Initialization: $\widehat{w}(n)$ = zeros(N);
for $i$ = 1, 2, 3... do
    $\mathbf{x}(n) = [x(n), x(n-1), ..., x(n-N+1)]^T$;
    $e(n) = y(n) - \widehat{w}^H(n-1)\mathbf{x}(n)$;
    $\widehat{w}(n) = \widehat{w}(n-1) + \mu\mathbf{x}(n)e^*(n)$
end
```

Algorithm 1: LMS algorithm

In the simulation test, the number of taps were set to 128 and the known filter has the length of 128. This algorithm was ran 100 epochs. Result of the convergence rate test is shown in figure 5.

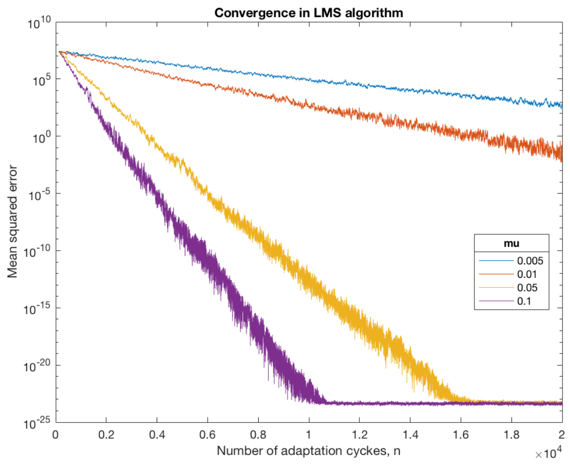

Fig. 5. LMS

We can conclude from figure 5 that the smaller learning rate requires more time to converge which means the larger learning rate will result in fast convergence. But when the learning rate is too large, LMS algorithm will become unstable. In this simulation, LMS algorithm will become unstable when the learning rate is larger than 0.5 which is not shown in figure 5.

B. FxLMS

The usage of LMS algorithm in a real ANC application becomes complicated because of the fact that the anti-noise created by the algorithm must travel from canceling loudspeaker to the error microphone and thus transition from the digital domain to the analog domain and back again. This process will introduce the frequency and phase distortion to the signal. The influence of such process is represented by a filter $S(z)$. In order to counteract distortion introduced by such process, an additional filter $\hat{S}(z)$ is used and placed between the reference microphone and the LMS algorithm. This algorithm is called Filtered extended LMS. The basic diagram for a ANC system using FxLMS algorithm is shown in figure 6.

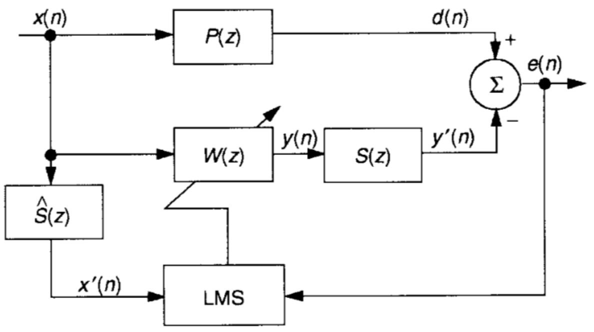

Fig. 6. The basic diagram of a ANC system suing FxLMS (reproduced from[2])

In order to know the impluse response of $S(z)$, the system should be tested before ANC is actually implemented. Among many different ways to estimate the impluse response of $S(z)$, an easy is to use the LMS algorithm with white noise as the input. Then the inverse response can be obtained by first taking it into frequency domain, taking the inverse in the frequency domain and then taking it back to time domain. The the update step in FxLMS algorithm will be in the form shown in equation 3.

$$w(n+1) = w(n) - \mu x^{'}(n)e(n) \tag{3}$$

$$where\ x^{'}(n) = \hat{s}(n) * \mathbf{x}(n)$$

The $*$ sign here represents the convolution operation and $\hat{s}(n)$ is the impulse response of $\hat{S}(z)$. The FxLIMS algorithm is shown in algorithm 2.

```
Offline Test: $\hat{s}(n)$ = inverse impulse response of $S(z)$;
Parameter: N = filter order; $\mu$ = step size;
Initialization: $\widehat{w}(n)$ = zeros(N);
for $i$ = 1, 2, 3... do
    $\mathbf{x}(n) = [x(n), x(n-1), ..., x(n-N+1)]^T$;
    $e(n) = y(n) - \widehat{w}^H(n-1)\mathbf{x}(n)$;
    $x^{'}(n) = \hat{s}(n) * \mathbf{x}(n)$;
    $\hat{w}(n) = \hat{w}(n-1) + \mu x^{'}(n)e^*(n)$
end
```

Algorithm 2: FxLMS algorithm

Result of the convergence rate testing is shown in figure 7.

As is shown in figure 7, the FxLMS algorithm converges to an absolute minimum at a slower rate than the LMS algorithm. Figure 7 shows the result when the learning rate between the secondary pass identification step and the actual running of the system are the same. In testing, it was observed that convergence could be speed up by varying

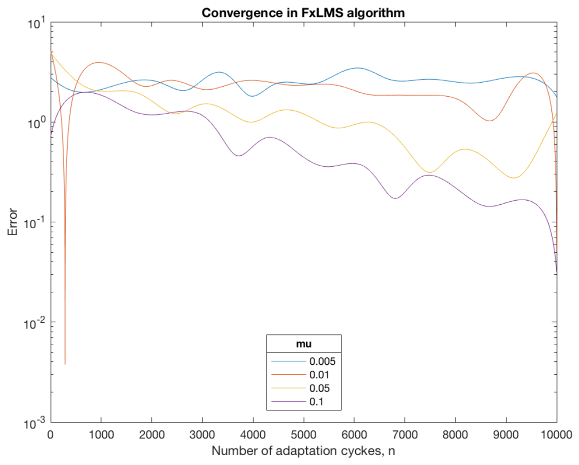

Fig. 7. FxLMS

learning rate between these two steps. For example, in this simulation, using a smaller learning rate for secondary path identification and then a larger step size while running the system showed comparable results to the "pure" LMS algorithm.

C. FuLMS

Another thing that makes the application of LMS algorithm in the real world complicated is the issue of feedback to the reference microphone from the cancellation speaker. This means that the sound from the cancellation speaker will affect the reference sound. To deal with this issue, FuLMS algorithm is proposed. In FuLMS, an adaptive recursive IIR filter $B(z)$ is added to the signal chain. The filter $B(z)$ is to minimize error based on a one sample delayed version of $\hat{y}^{'}(n)$. A block diagram of an adaptive IIR ANC system is illustrated in figure 8.

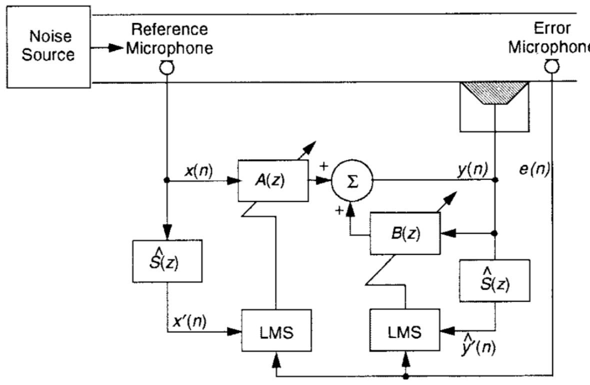

Fig. 8. The basic diagram of a ANC system suing FuLMS (reproduced from[2])

The output $y(n)$ is computed as

$$y(n) = \mathbf{a}^T(n)\mathbf{x}(n) + \mathbf{b}^T(n)\mathbf{y}(n-1) \tag{4}$$

The $\mathbf{a}^T(n)$ is the weight vector of $A(z)$, $\mathbf{b}^T(n)$ is the weight vector if $B(z)$ and $y(n-1)$ is the output signal with one sample delay. The update rule for the weight vector of $A(z)$ and $B(z)$ are illustrated in equation 5 and 6.

$$\mathbf{a}(n+1) = \mathbf{a}(n) + \mu x^{'}(n)e(n) \tag{5}$$

$$\mathbf{b}(n+1) = \mathbf{b}(n) + \mu \hat{y}^{'}(n-1)e(n) \tag{6}$$

where $x^{'}(n)$ is the same as in FxLMS. $\hat{y}^{'}(n-1) = \hat{s}(n) * \mathbf{y}(n-1)$ is the filtered version of the canceling signal vector at time $n-1$. The FuLMS Algorithm is shown in algorithm 3.

```
Offline Test: $\hat{s}(n)$ = inverse impulse response of $S(z)$;
Parameter: N = filter order; $\mu$ = step size;
Initialization: $A(z)$ = zeros(N);
$B(z)$ = zeros(N);
$x^{'}(n)$ = convolution($\hat{S}(z)$,$x(n)$);
$y^{'}(n)$ = convolution($\hat{S}(z)$,$y(n)$);
for $n$ = 1, 2, 3... do
    $\mathbf{y}(n) = [y(n), y(n-1), ..., y(n-N+1)]^T$;
    $\mathbf{y}^{'}(n) = [y^{'}(n), y^{'}(n-1), ..., y^{'}(n-N+1)]^T$;
    $\mathbf{x}(n) = [x(n), x(n-1), ..., x(n-N+1)]^T$;
    $\mathbf{x}(n) = \mathbf{x}(n) + \alpha_1 * \mathbf{y}^{'}(n)$;
    $\mathbf{x}^{'}(n) = [x^{'}(n), x^{'}(n-1), ..., x^{'}(n-N+1)]^T$;
    $\mathbf{x}^{'}(n) = \mathbf{x}^{'}(n) + \alpha_2 * \mathbf{y}^{'}(n)$;
    $y(n) = \mathbf{a}^T(n)\mathbf{x}(n) + \mathbf{b}^T(n)\mathbf{y}(n-1)$;
    $e(n) = d(n) - y(n)$;
    $\mathbf{a}(n+1) = \mathbf{a}(n) + \mu\mathbf{x}^{'}(n)e(n)$;
    $\mathbf{b}(n+1) = \mathbf{b}(n) + \mu\mathbf{y}^{'}(n-1)e(n)$;
end
```

Algorithm 3: FuLMS algorithm

The FuLMS algorithm does come with some drawbacks, most notably that it has never been mathematically proven to guarantee convergence. In this simulation, 80% of the sound from the canceling speaker was add back into the reference microphone to simulate a high level of feedback. The result of the test on FuLMS algorithm is shown in Figure 9.

As shown in Figure 9, the convergence is slower than FxLMS and LMS algorithm. This may be due to the fact that there are two adaptive filters need to learn in FuLMS which are both sharing the same learning rate μ. Perhaps the initial value of μs should be chosen independently. Another thing worth noticing is that the convergence may also vary with the amount of feedback added back to the reference microphone.

D. NLMS

The main drawback of the "pure" LMS algorithm is that it is sensitive to the scaling of its input. This makes it very hard to choose a learning rate μ that guarantees stability of the algorithm[9]. NLMS algorithm is derived base on the intuition that, in the light of new input data, the parameters of an adaptive system should only be disturbed in a minimal fashion. It can be expressed mathematically as

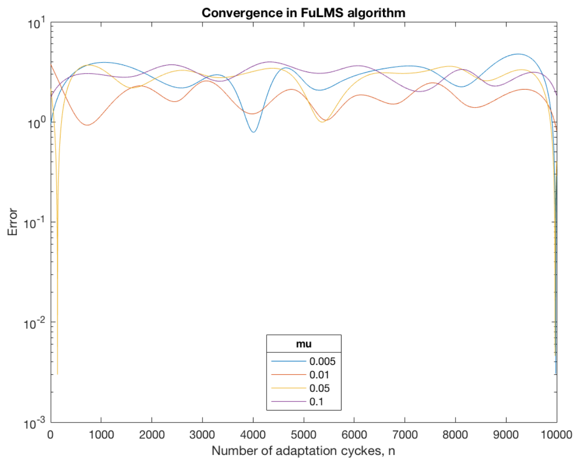

Fig. 9. FuLMS

$$min(\|\delta \mathbf{w}(n+1))\|^2)$$
$$subject\ to\ \mathbf{w}^T(n+1)\mathbf{x}(n) = d(n) \tag{7}$$

where $\mathbf{x}(n)$ is the input signal, $d(n)$ is desired signal and $\mathbf{w}(n)$ is the weight vector of the filter at time n. We define

$$\delta \mathbf{w}(n+1) = \mathbf{w}(n+1) - \mathbf{w}(n) \tag{8}$$

This is the change of weight vector between time $n+1$ and time n. This optimization problem can be solved using Lagrange multipliers method:

$$J(\mathbf{w}(n+1), \lambda) = \|\delta \mathbf{w}(n+1))\|^2 + \lambda(d(n) - \mathbf{w}^T(n+1)\mathbf{x}(n)) \tag{9}$$

After breaking down the vector and letting M representing the length of the filter, the equation 9 becomes:

$$J = \sum_{k=0}^{M-1}(w_k(n+1) - w_k(n))^2 + \lambda(d(n) - \sum_{i=0}^{M-1} w_i(n+1)x(n-i)) \tag{10}$$

This problem can be solved by following the standard procedures. After this, we can achieve that the minimum of the criterion $J(\mathbf{w}(n+1), \lambda)$ will be obtained using

$$w_j(n+1) = w_j(n) + \frac{\tilde{\mu}}{\|\mathbf{x}(n)\|^2} e(n)x(n-j) \tag{11}$$

where j represents the each element in the weight vector $\mathbf{w}(n)$, $\tilde{\mu}$ is a constant in order to add an extra freedom degree to the adaptation strategy. To overcome the possible numerically difficulty when $\mathbf{u}(n)$ is very small, a constant $a > 0$ is used. The final updating rule for NLMS algorithm is

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\tilde{\mu}}{a + \|\mathbf{x}(n)\|^2} e(n)\mathbf{x}(n) \tag{12}$$

The rest of NLMS algorithm is the same as "pure" LMS algorithm shown in Algorithm 1. The result of the test on NLMS algorithm is shown in figure 10.

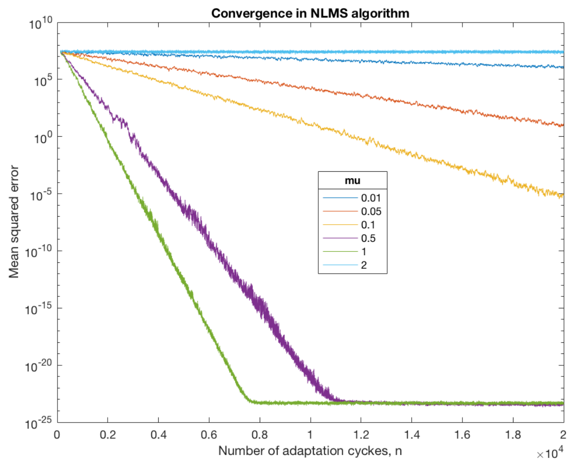

Fig. 10. NLMS

The μ shown in the figure 10 is the actually the $\tilde{\mu}$ in the above NLMS algorithm. Comparing with the result obtained in "pure" LMS algorithm section, the best convergence times are comparable between these two. But NLMS algorithm is more tolerant to initial convergence rate estimates. As shown in figure 10, when the learning rate is as large as 2, the system is still stable even though it is not actually learning.

E. AP Algorithm

Affine Projection (AP) algorithms emerged to improve speed of convergence of gradient based algorithms. A common feature that shared between all AP algorithms is the updating equation, which uses N (called projection order) vectors of the input signal instead of a single vector as NLMS algorithm. Let the change of weight vector between two successive algorithm iterations be the form in equation 8. Unlike the NLMS algorithm in which only one constrain is used to minimize the norm as shown in equation 7, AP algorithm tries to minimize the norm under N constrains

$$\mathbf{w}^T(n+1)\mathbf{x}(n-k) = d(n-k)\ where\ k = 0, 1, 2, ..., N-1 \tag{13}$$

The solution of the presented problem leads to the AP update equation given by

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{A}^T(n)(\mathbf{A}(n)\mathbf{A}^T(n))^{-1}\mathbf{e}_N(n) \tag{14}$$

where

$$\mathbf{A}[n] = (\mathbf{x}(n), \mathbf{x}(n-1), ..., \mathbf{x}(n-N+1))^T \tag{15}$$

$\mathbf{e}_N(n)$ is given by

$$\mathbf{e}_N(n) = \mathbf{d}_N(n) - \mathbf{A}(n)\mathbf{w}(n-1) \tag{16}$$

and $\mathbf{d}_N(n)$ represents the desired signal

$$\mathbf{d}_N(n) = (d(n), d(n-1), ..., d(n-N+1))^T \tag{17}$$

It is worth noticing that the N data vectors that are used to update the coefficients could not necessarily be the most recent ones. Thus different versions of the algorithms can be developed choosing the input data vectors and using them in different ways. Also , the different choice of N will result in different performance. When $N = 1$, AP algorithm become identical to NLMS algorithm.
The result of the test on AP algorithm is shown in figure 11.

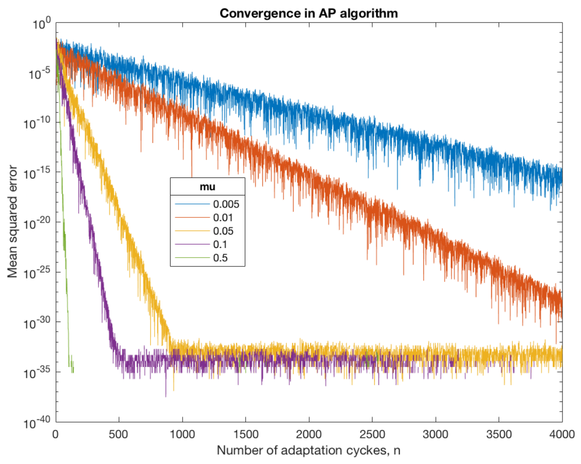

Fig. 11. APA

Afiine Projection agolrithm is actually a generalized version of NLMS algorithm, As shown in figure 11, the AP algorithm converges faster than NLMS. But the price paid is higher computational complexity.

F. RLS

The RLS algorithm can recursively finds the filter weights that minimize a weighted linear least squares cost function relating to the input signal. RLS algorithm generally converge faster than all the algorithms presented above but it is also the most complex algorithm among the algorithms above. The RLS algorithm is shown in algorithm 4.

```
Initialization: $p(0) = \delta^{-1}I$; $\hat{\mathbf{w}}(0) = \mathbf{0}$;
for $n$ = 1, 2, 3... do
    $\mathbf{k}(n) = \frac{\lambda^{-1}p(n-1)\mathbf{u}(n)}{1+\lambda^{-1}p(n-1)\mathbf{u}(n)}$;
    $\alpha(n) = \mathbf{d}(n) - \hat{\mathbf{w}}(n-1)\mathbf{u}(n)$;
    $\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\alpha^*(n)$;
    $p(n) = \lambda^{-1}p(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^H(n)p(n-1)$
end
```

Algorithm 4: RLS algorithm

In the simulation, the "forgetting factor" λ was set to be 1 and the value δ which is used to initialize the P matrix is set to be 0.005. The filter length of this set to be 5. The result of the test on AP algorithm is shown in figure 12.

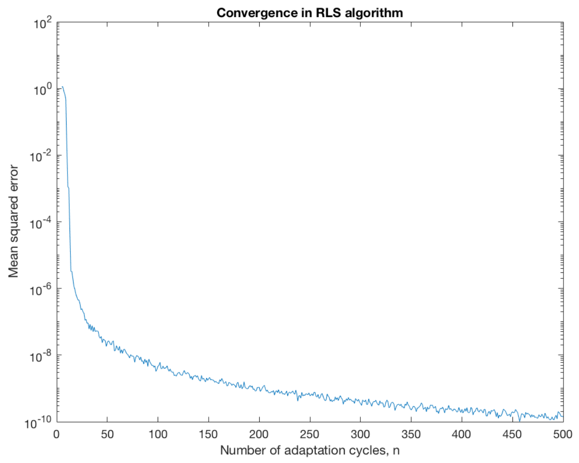

Fig. 12. RLS

As shown in figure 11, RLS algorithm can converge very fast. We can see that RLS algorithm converges after around 20 iterations. Also, its noise cancellation capacity is the most. But, comparing to algorithms in LMS family, the computational complexity of RLS is much higher. Generally, the complexity for LMS algorithm is linear while RLS algorithm has the complexity $O(N)$ where N is the length of of the filter.

IV. APPLICATION

In this section, we will apply these different algorithms to some typical noises. Among various noises, we chose pink noise, white noise, airplane noise and car noise. All this four noises are shown in figure 13.

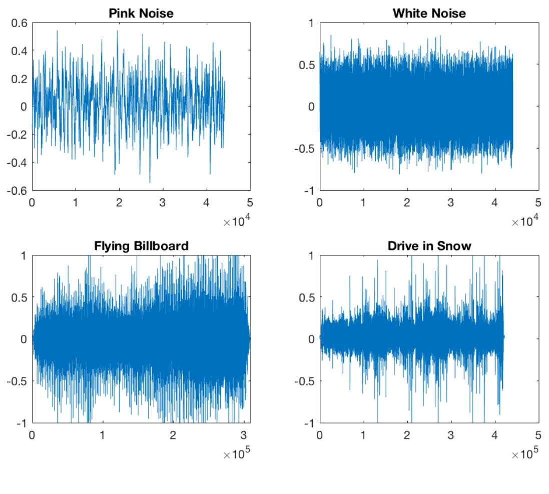

Fig. 13. Four typical noises

A. NLMS

In this subsection, Normalized LMS is applied on four typical noises. We assume the primary path filter has a length of 128 and its parameters are known to us. In NLMS algorithm, the base learning rate $\tilde{\mu}$ is set equal to 0.05 and the constant a which will be used to overcome the possible numerically difficulty is set equal to 1. The resulting noise is shown in figure 14.

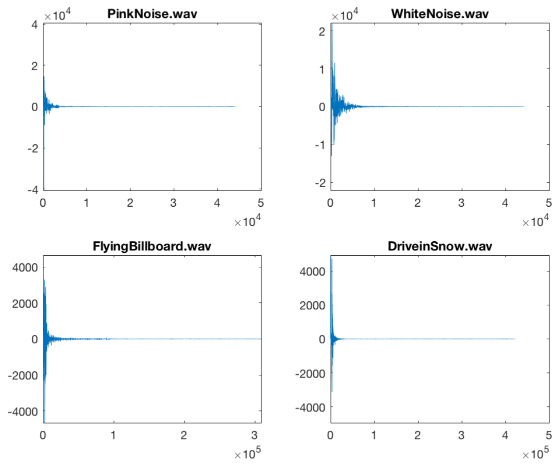

Fig. 14. NLMS on four noises

As shown in figure 14, we can conclude that the convergence rate is not fast comparing to the result of other algorithm. But the resulting noise after convergence is very close to zero.

B. RLS

When applying RLS algorithm on the real noise, we assume the primary path filter is known to us. We assume the primary path filter has a order of 5 and the parameters of this filter are set to be $[1\ 0.5\ 0.25\ 0.125\ 0.0625]$. For the parameters in RLS algorithm, we set the "forgetting factor" $\lambda = 1$ and the regularization factor $\delta = 0.005$. The resulting noise is shown in figure 15.

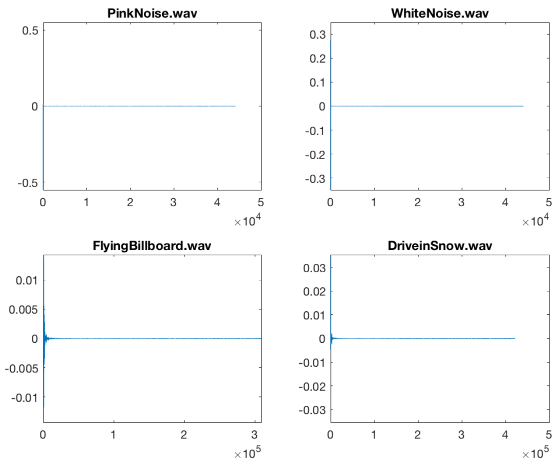

Fig. 15. RLS on four noises

As shown in figure 15, we can see that RLS algorithm converge very fast and the resulting noise is very close to zero.

C. APA

In this subsection, the assumption made about the primary path is the same as that in RLS part. For the parameters in AP algorithm, the learning rate μ is set to 0.05, regularization factor δ is set to 0.001 and the projection order was set to be 4. The result is shown in figure 16.

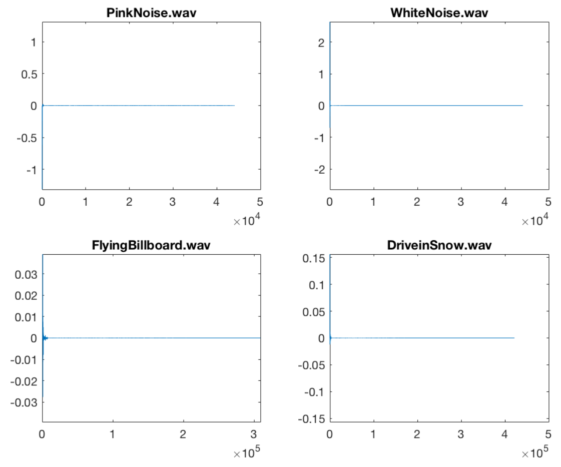

Fig. 16. AP algorithm on four noises

As shown in figure 16, we can see that the AP algorithm also have a fast convergence rate and the resulting noise is very close to zero.

D. Summary

From the above three applications, we can conclude that different kinds of noises will affect the performance of different algorithms to some extent. In general, that the NLMS algorithm has the smallest convergence rate. But, all of them have a very good tracking performance after convergence.

V. CONCLUSION

All the testings here shows that all the six algorithms will convergence after different numbers of iterations when given the right value of learning rate μ. Among four algorithms in LMS family, "pure" LMS and NLMS have a similiar convergence rate. But, in most cases, NLMS algorithms with various learning rate show greater likelihood of convergence and higher stability than corresponding constant step size LMS algorithms. FxLMS and FuLMS have relatively slower learning rate. For RLS, it has the fastest convergence rate than all the other algorithms tested here which comes at the price of the computational complexity. For APA, it converges faster than LMS in general but the computational complexity is higher when the projection order increases.
We also showed that NLMS, RLS and APA are all robust enough to converge in a relatively short time when fed various types of noises in real world.

VI. FUTURE WORK

First, I would like to keep examining new ANC methods such as use neural network to learning the time-varying signals as shown by Fernandez [10] as well as different versions of old ANC algorithms as better and more robust options. I would also like to try to combine the advantages of multiple algorithms and then come up with a new more robust ANC algorithm.

APPENDIX

You can find all the codes which produce all the results above on this website: https://github.com/YutaoC/Adaptive-Filter-Design

REFERENCES

[1] P. Lueg, Process of silencing sound oscillations, U.S. Patent 2 043 416, June 9, 1936.

[2] 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.

[3] N. Wiener, Extrapolation, interpolation, and smoothing of stationary time series. The MIT Press, Cambridge, MA, 1949.

[4] 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.

[5] C. Burgess, Active adaptive sound control in a duct: A computer simulation, J. Acoustical Soc. Amer., vol. 70, pp. 715726, Sept. 1981.

[6] 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.

[7] A. Gonzalez, M. Ferrer, F. Albu and M. de Diego, ”Affine projection algorithms: Evolution to smart and fast algorithms and applications,” 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, 2012, pp. 1965-1969.

[8] Haykin, Simon S. Adaptive Filter Theory. Pearson Education Inc., 2014.

[9] Dhiman, Jyoti et al. Comparison between Adaptive filter Algorithms (LMS, NLMS and RLS). (2013).

[10] A. Fernndez and P. Cobo, Artificial neural network algorithms for active noise control applications. Sociedad Espaola de Acstica, 2002.