RESEARCH ARTICLE OPEN ACCESS

Noise Cancellation using Least Mean Square Algorithm

Vedansh Thakkar
Medicaps Institute of Technology and Management

ABSTRACT
A revolution in science of electronics and communication has emerged in the last few decades, with the potential to create a paradigm shift in thinking about adaptive filtering. This Project involves the study of the principles of Adaptive Noise Cancellation (ANC) and its Applications. Adaptive noise Cancellation is an alternative technique of estimating signals corrupted by additive noise or interference. Its advantage lies in that, with no apriori estimates of signal or noise, levels of noise rejection are attainable that would be difficult or impossible to achieve by other signal processing methods of removing noise. Adaptive noise cancellation is an approach used for noise reduction in speech signal. As received signal is continuously corrupted by noise where both received signal and noise signal both changes continuously, then this arise the need of adaptive filtering. This paper deals with cancellation of noise on speech signal using an adaptive algorithm called least mean square (LMS) algorithm

Keywords: Adaptive noise cancellation (ANC), LMS Algorithm, NLMS algorithm, Adaptive filtering.

Date of Submission: 26 -09-2017 Date of acceptance: 10-10-2017

I. INTRODUCTION

Acoustic noise issues becomes pronounce as increase in range of commercial instrumentation like engines, transformers, compressors and blowers in use. The normal approach to acoustic noise cancellation uses passive techniques like enclosures, barriers and silencers to get rid of the unwanted noise signal. Silencers are necessary for noise cancellation over broad frequency range however ineffective and expensive at low frequencies. Mechanical vibration may be a style of noise that creates issues in all areas of communication and electronic appliances. Signals are carriers of knowledge, each helpful and unwanted. Extracting or enhancing the helpful info from a combination of conflicting information may be a simplest style of signal processing. Signal processing is designed for extracting, enhancing, storing, and transmittal helpful information. Therefore, signal processing tends to be application dependent. Converse to the traditional filter design techniques, adaptive filters don't have constant filter coefficients and no priori information is known. Such a filter with adjustable parameters is named adaptive filter. Adaptive filter change their coefficients to nullify an error signal and could be realized as finite impulse response (FIR), infinite impulse response (IIR), lattice and transform domain filter. The foremost common type of adaptive filter is that the transversal filter using least mean square (LMS) algorithm. In this paper, adaptive algorithms are applied to totally different types noise. The essential plan of adaptive noise cancellation algorithm is to pass the corrupted signal through a filter that tends to suppress the noise whereas exploit the signal unchanged. This is an adaptive method, which implies it doesn't need a priori data of signal or noise characteristics. Adaptive noise cancellation (ANC) attenuates low frequency noise that passive filters cannot .

Figure (1): usual method of estimating noisy signal.

Noise
It is any kind of unwanted signal which gets transmitted with the message signal .
Presence of noise in communication channel is very undesirable as the original message signal do not get interpreted correctly at the receiver side.

Types of noise considered in the project:
1) Random Noise
2) White Gaussian Noise

Random noise:
Random noise usually refers to electric or acoustic signal that consists of equal amounts of all frequencies.
Power spectral Density of random noise:

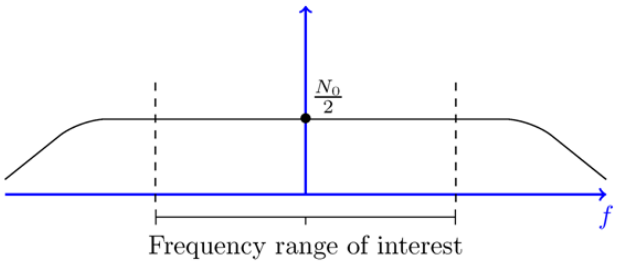


Fig (2) : PSD of Thermal noise (one type of random noise)

White noise:

White noise is a random signal having equal intensity at different frequencies, giving it a constant power spectral density.

The power spectral density (PSD) of additive white Gaussian noise :

$S_X(f) = N_0/2$

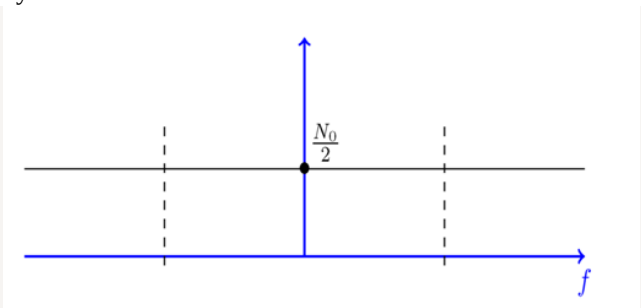


Fig (3) : PSD of white noise

Adaptive Noise Cancellation

The purpose of adaptive noise cancellation is to improve the signal-to-noise ratio (SNR) of a signal by removing noise from the signal that we receive.

In this configuration the input x (n), a noise source N1 (n), is compared with a desired signal d (n), which consists of a signal s (n) corrupted by another noise N0 (n). The adaptive filter coefficients adapt to cause the error signal to be a noiseless version of the signal s (n).

Do to the nature of the error signal; the error signal will never become zero. The error signal should converge to the signal s (n), but not converge to the exact signal. In other words, the difference between the

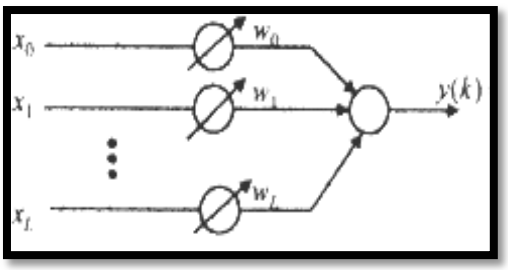


Fig (4): general structure of adaptive filter

signal s (n) and the error signal e (n) will always be greater than zero. The only option is to minimize the difference between those two signals

Adaptive filter:

An adaptive filter adapts to changes in its input signals automatically according to a given algorithm. The algorithm will vary the coefficients according to a given criteria, typically an error signal to improve its performance. Basically an adaptive filter is a digital filter combined with an adaptive algorithm, which is used to modify the coefficients of the filter.

Adaptive filters are used in many diverse applications in today's world for example telephone echo canceling, radar signal processing, equalization of communication channels and biomedical signal enhancement.

Input Signal:

For the case of multiple input

$\mathbf{X_k} = [\, x_{0k} \quad x_{1k} \ldots\ldots\ldots\ldots X_{Lk}]^T \quad \ldots\ldots\ldots (1)$

Weight Vectors:
Corresponding to weight vector input signal are $\mathbf{W_k} = [\, w_{0k} \quad w_{1k\ldots\ldots} W_{Lk}]^T \quad \ldots\ldots.(2)$
So, on multiplying eqution (1) and (2) , we get output signal as
$\mathbf{y_k} = X_k^T W_k = W_k X_k^T \ldots\ldots\ldots\ldots(3)$
and since error signal is:
$\mathbf{e_k} = d_k - y_k \ldots\ldots\ldots\ldots(4)$
Calculating error signal from equation (3) :
$\mathbf{e_k} = d_k - X_k^T W = d_k - W\ X_k^T \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$
On squaring equation (5), the instantaneous squared error is :
$\mathbf{e^2_k} = d^2_k + W^T X_k X_k^T W - 2d_k X_k^T W \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(6)$

from equation (6) calculating the expected mean square value as mentioned in equation (7)

$\mathbf{E\,[e^2_k]} = E[d^2_k] + W^T E\,[X_k X_k^T]W - 2E[d_k X_k^T]W \quad \ldots\ldots\ldots (7)$

$X_k X_k^T$ is the autocorrelation matrix of input signal and it is represented by the term R as given in the eqution (8)

$$\mathbf{R} = E\,[X_k X_k^T] = E\begin{bmatrix} x^2_{0k} & x_{0k}\ x_{1k} \cdots & x_{0k}\ x_{Lk} \\ \vdots & \ddots & \vdots \\ x_{Lk}\ x_{0k} & x_{Lk}\ x_{1k} \cdots & x^2_{Lk} \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots(8)$$

Diagonal terms are mean square of input components and cross terms are cross correlation among the input components.

Next term $E\,[d_k X_k]$ is the cross correlation between the input signal and desired signal and it is expressed by P and represented in equation (9)

$\mathbf{P} = E\,[d_k X_k] = E[d_k x_{0k} \quad d_k x_{1k} \ldots\ldots \quad d_k x_{Lk}]^T \ldots\ldots(9)$

Now calculating mean square error from equation (7)
$MSE \triangleq \zeta = E[e^2_k] = E[d^2_k] + W^T R W - 2P^T W \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$

Gradient and Mean square error:
$\nabla \triangleq \partial\zeta/\partial W = [\partial\zeta/\partial w_0 \quad \partial\zeta/\partial w_1 \ \ldots\ \partial\zeta/\partial w_L]^T \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(11)$
$= 2RW - 2P \ldots\ldots\ldots\ldots\ldots\ldots(12)$

To obtain the minimum mean-square error the weight vector W is set at its optimal value W*, where the gradient is zero:
$\nabla = 0 = 2RW^* 2P \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (13)$

W* is wiener weight vector
$W^* = R^{-1} P \ldots\ldots\ldots\ldots\ldots\ldots\ldots(14)$

Mean square error is obtained by substituting W* from equation (14) for W in equation (10)

$\zeta_{min} = E[d^2_k] + \mathbf{W^{*T} R W^* - 2P^T W^*} \ldots\ldots\ldots\ldots\ldots\ldots\ldots..(15)$
$= E[d^2_k] + \mathbf{[R^{-1}P]^T R R^{-1} P - 2P^T R^{-1} P} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(16)$

On simplifying equation (16) we get :
$\zeta_{min} = E[d^2_k] - \mathbf{P^T R^{-1} P} = E[d^2_k] - \mathbf{P^T W^*} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(17)$

THE LMS ALGORITHM
The Least Mean Square (LMS) is an adaptive algorithm, LMS algorithm uses the estimates of the gradient vector from the available data. The LMS incorporates an iterative procedure that makes corrections to the weight vector in the direction of the negative of the gradient vector which eventually leads to the minimum mean square error.
Compared to other algorithms, the LMS algorithm is considered simpler because it does not require

correlation function calculations nor does it require matrix inversions. There are many algorithms used to adjust the coefficients of the digital filter in order to match the desired response as well as possible. The LMS Algorithm is the more successful of the algorithms because it is the most efficient in terms of storage requirement and indeed computational complexity, the basic LMS algorithm updates the filter coefficients after every sample.

The Least-Mean-Square algorithm in words**:**

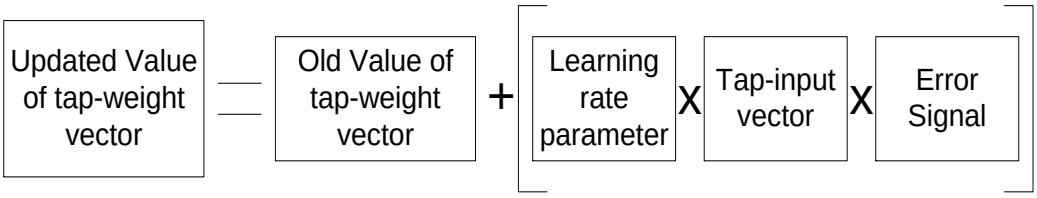


Figure (5): the LMS Algorithm in words

The simplicity of the LMS algorithm and ease of implementation means that it is the best choice for many real-time systems.

The LMS Algorithm consists of two basic processes :

- Filtering process-
  1. Calculate the output of FIR filter by convolving input and taps.
  2. Calculate estimation error by comparing the output to desired signal
- Adaptation process-
  1. Adjust tap weights based on the estimation error

The implementation steps for the LMS algorithm –

- Filter output: $y[n]=\sum_{k=0}^{M-1} u[n-k]w_k^*[n]$ ……(18)
- Estimation error : $e[n]=d[n]-y[n]$ …….(19)
- Tap-weight adaptation : $w_k[n+1]=w_k[n]+\mu u[n-k]e^*[n]$ …….(20)

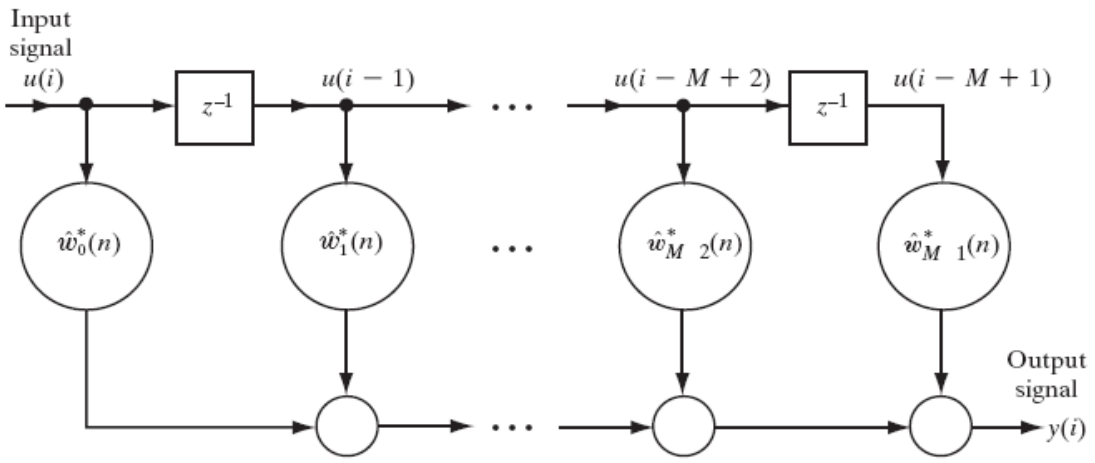


Figure (6): Tap weight adaptation

LMS WEIGHT UPDATION :

In LMS algorithm $\nabla w(n)$ is assumed as $\frac{\partial e^2(n)}{\partial \underline{W}(n)}$

STABILITY OF LMS :

- The LMS algorithm is convergent in the mean square if and only if the step-size parameter satisfy.

$$0 < \mu < \frac{2}{\lambda_{max}}$$

- Here λ_{max} is the largest eigenvalue of the correlation ssmatrix of the input data.

- More practical test for stability is

$$0 < \mu < \frac{2}{\text{input signal power}}$$

- Larger values for step size
  - Increases adaptation rate (faster adaptation) as convergent rate is fast.
  - Increases mean-squared error (MSE) model the given system or the initial state of the adaptive filter is an inadequate starting point to cause the adaptive filter to converge.

II. RESULT ANALYSIS

Simulation Tool used: MATLAB is used as the simulation tool for the project. The function adaptfilt.lms() is used for creating the adaptive filter working on LMS algorithm. This function gives the characteristics of the adaptive filter according to the input like step size, filter length etc. provided.

The performance of the LMS algorithm has been assessed for noise cancellation. The MATLAB tool r2015a has been used for analysis. First a recorded voice signal is taken and then different noises i.e. AWGN, traffic noise, airplane noise are used to corrupt the voice signal and the corrupted signal were filtered adaptively and results have been analyzed.

In fig(3) results for AWGN is shown with filter length l=256 and μ=0.017. In fig(4) results for traffic noise is shown with l=256 and μ=0.017. In fig(5) results for airplane noise is shown with l=256 and μ=0.017.

In fig(6)Fading phenomenon was implemented by the interference of voice signal with its delayed and advanced form for l=256 and μ=0.013.

In fig (7) sine signal was used to analyze the variation of MSE with iteration for different values of μ. It was observed that on increasing μ the convergence was faster ,the phenomenon was observed till a certain value of μ=0.017(AWGN) showing that there exists a tradeoff for values of μ.

In fig (8) sine signal was used to analyze the variation of error with iteration for different values of μ. It was observed that on increasing the value of μ the number of iterations required for convergence was decreased.

GRAPHS:

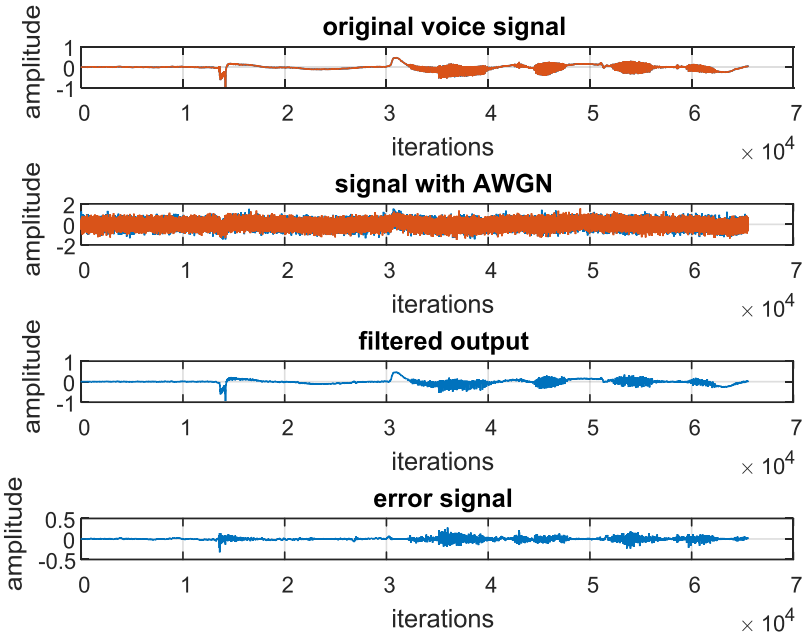
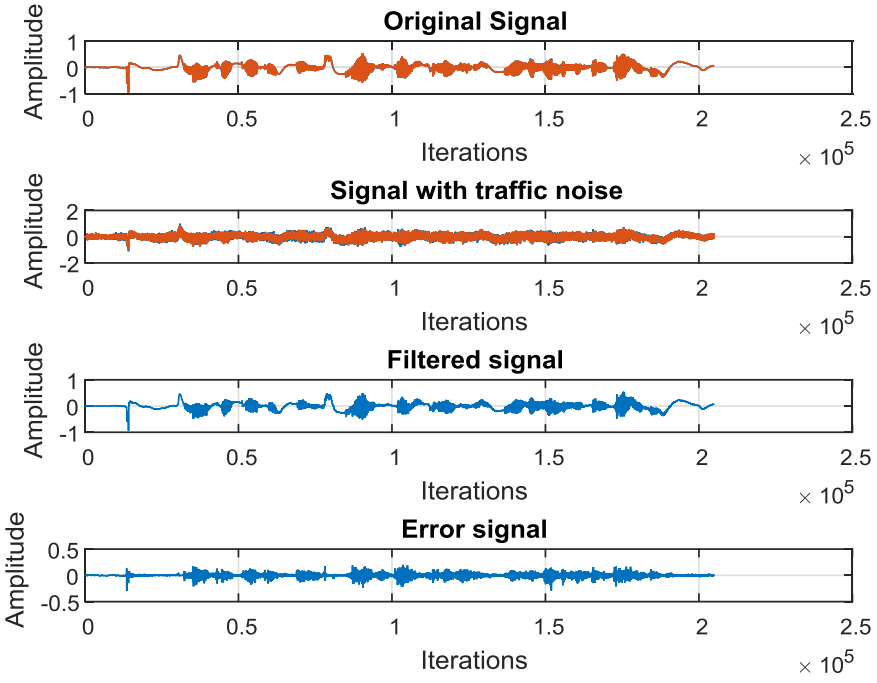


Fig (7) Results for voice signal with AWGN

Fig (8) Results for voice signal with traffic noise



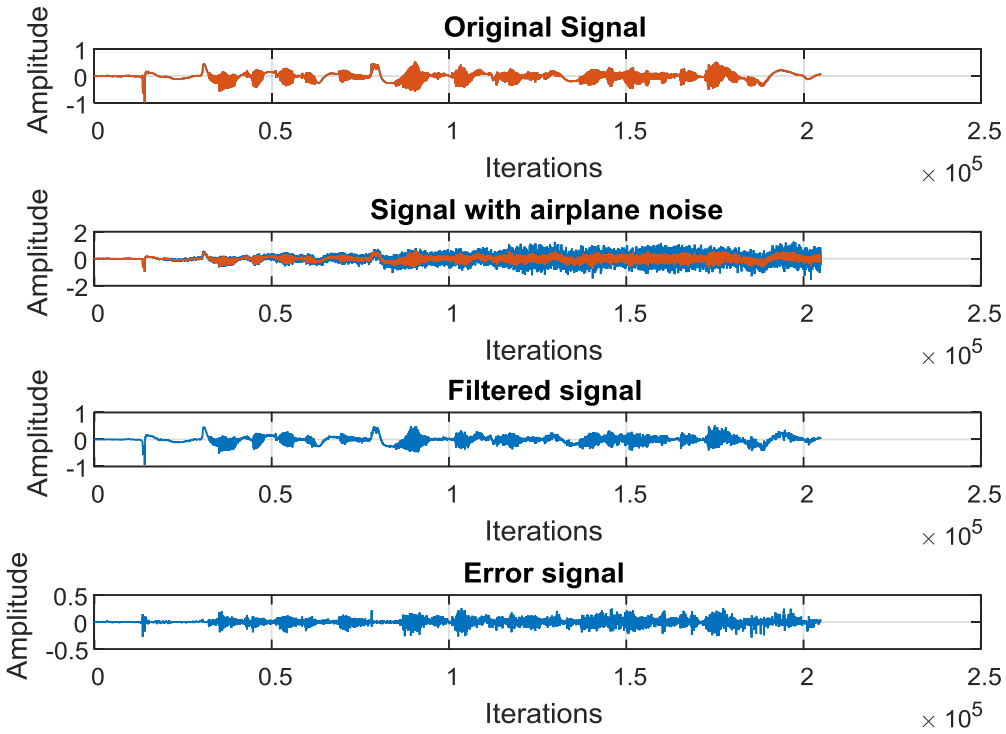


Fig (9) Results for voice signal with airplane noise

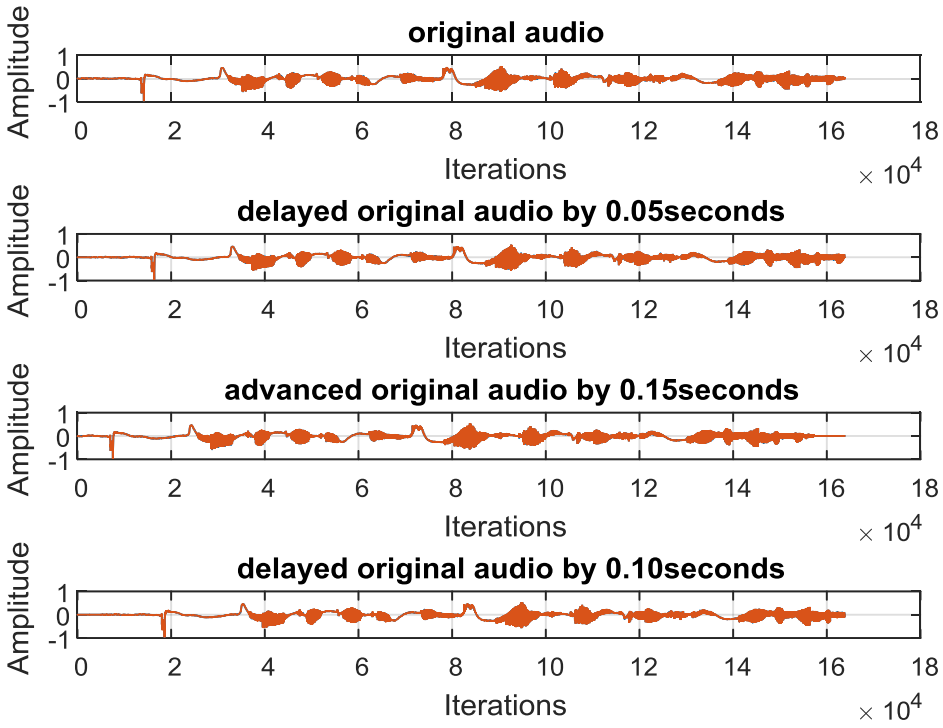


Fig (10a) Delayed and advanced form of voice signal

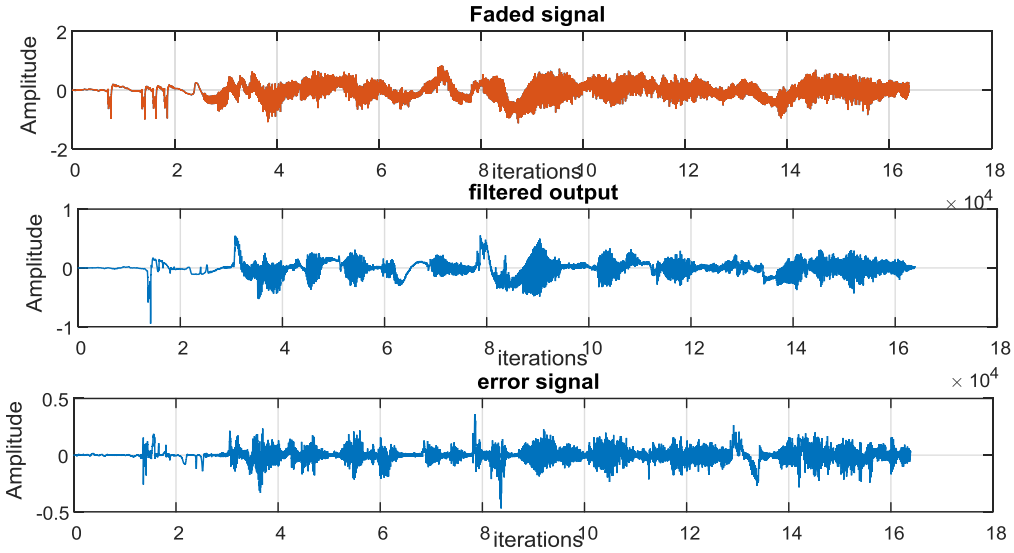


Fig (10b) Filtered signal of fading

Fig (11) Variation of Mean Square Error with iterations

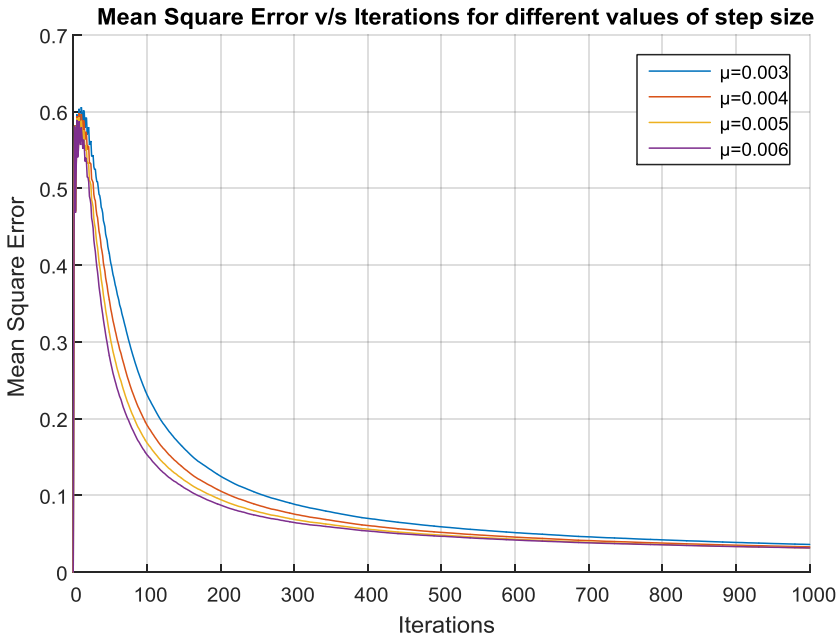


Fig (11) Variation of Mean Square Error with iterations

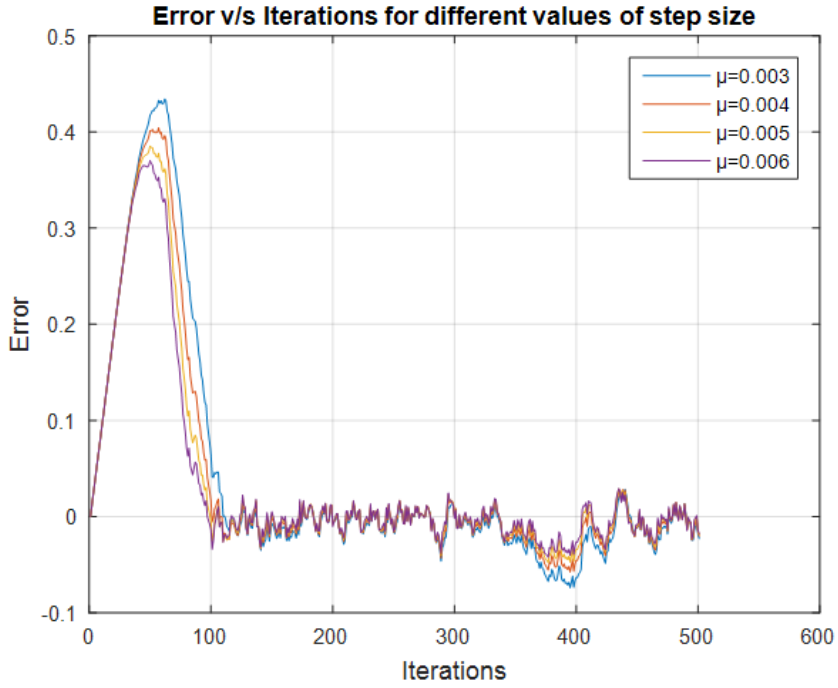


Fig (12) Error signal for different values of step size (μ)

TABLES:

Table I. MEAN ERROR FOR DIFFERENT NOISES

S. No.	Type of Noise	Mean Error
1	AWGN	2.1821e-04
2	Traffic noise	2.7879e-05
3	Airplane noise	2.1e-03

Table II. CONVERGENCE OF FILTER FOR DIFFERENT VALUES OF step size (μ)

S. No.	Step size(μ)	Number of iterations	MSE
1	0.003	622	0.05021
2	0.004	537	0.04966
3	0.005	489	0.04944
4	0.006	468	0.04892

III. CONCLUSION

Adaptive filters are a very useful tool in signal processing as they have the capability to remove the noise from the signal even if the statistical data of the signal. Adaptive filter using LMS (Least Mean Square) algorithm is simulated in the project using MATLAB. The function adaptfilt.lms() was used for creating the desired filter. In the simulation it was observed that in the starting of the signal the filters output was not tracing the desired signal and the error signal was also high. But after some time duration the filter adapts to the signal and the output of the filter is almost tracing the desired signal and hence reducing the error signal to very low value.

Noise cancellation using LMS algorithm for different types of noises is analyzed using MATLAB tool r2015a. It is observed that LMS algorithm is an effective technique for removal of noise as it has less complexities .Observations of MSE for different values of step size and filter length is done and it is concluded that optimized results is achieved for filter length 256 and step size 0.017. The same algorithm is applied for fading and optimized results are obtained at step size 0.013. There exist a tradeoff between step size and filter length, directly affecting the convergence rate of adaptive filter. This work can be extended to various medical and military applications.

REFERENCES

[1] Priyanka Gupta1, Mukesh Patidar2, Pragya Nema3,"Performance Analysis of Speech Enhancement Using LMS, NLMS and UNANR algorithms "IEEE International Conference on Computer Communication and Control,September 2015

[2] A. Hadei, and M. lotfizad, "A Family of Adaptive Filter Algorithms in Noise Cancellation for Speech Enhancement," International Journal of Computer and Electrical Engineering, Vol. 2, No. 2, April 2010.

[3] Bernard Widrow, Samuel D. Stearns, "Adaptive Signal Processing", Pearson Publication, 2007

[4] International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 7, July 2013

[5] J. M. Górriz, et. al, "A Novel LMS Algorithm Applied To Adaptive Noise Cancellation", IEEE Signal Processing Letters, Vol. 16, No. 1,January 2009.

[6] Simon Haykin, Adaptive Filter Theory, Prentice Hall, II. Edition

[7] John R. Glover, Jr., "Adaptive Noise Canceling Applied to Sinusoidal Interferences", IEEE Trans. ASSP, Vol. ASSP-25, No. 6, pp. 484-491, Dec. 1977.

[8] J.R. Zeidler et al., "Adaptive Enhancement of Multiple Sinusoids in Uncorrelated Noise", IEEE Trans. ASSP, Vol. ASSP-26, No. 3, pp. 240-254, June 1978.

[9] D. W. Tufts, "Adaptive Line Enhancement and Spectrum Analysis", Proc. IEEE (Letts.), vol. 65, pp.169-170, Jan. 1977

[10] L. Griffiths, Proc. IEEE (Letts.), vol. 65, pp.170-171, Jan. 1977

[11] B. Widrow et al., Proc. IEEE (Letts.), vol. 65, pp.171-173, Jan. 1977

[12] Adaptive Signal Processing by John G Proakis, 3rd edition, Perntice Hall of India.

[13] B. Widow, "Adaptive noise canceling: principles and applications", Proceedings of the IEEE, vol. 63, pp. 1692-1716, 1975.

[14] Adaptive Signal Processing by Bernard Widrow and Samuel D.Stearns; Pearson Education Asia, LPE

[15] Emmanuel Ifeachor and Barrie W. Jervis, " Digital Signal Processing ," Pearson Education Publication, 2nd Edition, Jan. 2013, pp.342-440.

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

APPENDIX

APPENDIX A

MATLAB CODE FOR VOICE SIGNAL WITH AWGN

```
load handel.mat;
d= 'Recording.m4a';
samples = [1,20*Fs];
```

```
clear d Fs
[d,Fs] =
audioread('Recording.m4a',samples)
;
sound(d,Fs)
pause(3)
x=awgn(d,20)

sound(x,Fs)
pause(3)
mu=0.017
ha=adaptfilt.lms(256,mu);
[y,e]=filter(ha,x(:,1),d(:,1));
sound(y,Fs)

subplot(4,1,1)
plot(d)
grid on
xlabel('iterations')
ylabel('amplitude')
title('original voice signal')
subplot(4,1,2)
plot(x)
grid on
xlabel('iterations')
ylabel('amplitude')
title('signal with AWGN')
subplot(4,1,3)
plot(y)
grid on
title('filtered output')
xlabel('iterations')
ylabel('amplitude')
subplot(4,1,4)
plot(e)
grid on
title('error signal')
xlabel('iterations')
ylabel('amplitude')
```

APPENDIX B

MATLAB CODE FOR VOICE SIGNAL WITH TRAFFIC NOISE

```
load handel.mat;
d = 'Recording.m4a';
samples = [1,25*Fs];
clear d Fs
[d,Fs] =
audioread('Recording.m4a',samples)
;
sound(d,Fs)
pause(3)
[n,Fs]=
audioread('traffic.mp3',samples);
x=d+n
sound(x,Fs)
pause(3)
mu=0.013
ha=adaptfilt.lms(256,mu);
[y,e]=filter(ha,x(:,1),d(:,1));
sound(y,Fs)
subplot(4,1,1)
plot(d)
grid on
xlabel('Iterations')
ylabel('Amplitude')
title('Original Signal')
subplot(4,1,2)
plot(x)
grid on
xlabel('Iterations')
ylabel('Amplitude')
title('Signal with traffic noise')
subplot(4,1,3)
plot(y)
grid on
xlabel('Iterations')
ylabel('Amplitude')
title('Filtered signal')
subplot(4,1,4)
plot(e)
grid on
xlabel('Iterations')
ylabel('Amplitude')
title('Error signal')
```

APPENDIX C

MATLAB CODE FOR VOICE SIGNAL WITH AIRPLANE NOISE

```
load handel.mat;
d = 'Recording.m4a';
samples = [1,25*Fs];
clear d Fs
[d,Fs] =
audioread('Recording.m4a',samples)
;
sound(d,Fs)
pause(3)
[n,Fs]=
audioread('airplane.mp3',samples);
x=d+n
sound(x,Fs)
pause(3)
mu=0.013
ha=adaptfilt.lms(256,mu);
[y,e]=filter(ha,x(:,1),d(:,1));
sound(y,Fs)
subplot(4,1,1)
plot(d)
ylabel('Amplitude')
xlabel('Iterations')
grid on
title('Original Signal')
subplot(4,1,2)
plot(x)
grid on
ylabel('Amplitude')
xlabel('Iterations')
```

```
title('Signal with airplane
noise')
subplot(4,1,3)
plot(y)
grid on
ylabel('Amplitude')
xlabel('Iterations')
title('Filtered signal')
subplot(4,1,4)
plot(e)
grid on
ylabel('Amplitude')
xlabel('Iterations')
title('Error signal')
```

APPENDIX D

MATLAB CODE FOR FADING

```
load handel.mat;
d= 'Recording.m4a';
samples = [1,20*Fs];
clear d Fs
[d,Fs] =
audioread('Recording.m4a',samples)
;
sound(d,Fs)
subplot(4,1,1)
plot(d)
grid on
ylabel('Amplitude')
xlabel('Iterations')
title('original audio')
pause(3)
n1 = delayseq(d,0.05,Fs)
n2 = delayseq(d,-0.15,Fs)
n3 = delayseq(d,0.10,Fs)
subplot(4,1,2)
plot(n1)
grid on
ylabel('Amplitude')
xlabel('Iterations')
title('delayed original audio by
0.05seconds')
subplot(4,1,3)
plot(n2)
grid on
ylabel('Amplitude')
xlabel('Iterations')
title('advanced original audio by
0.15seconds')
subplot(4,1,4)
plot(n3)
grid on
ylabel('Amplitude')
xlabel('Iterations')
title('delayed original audio by
0.10seconds')
x=d+n1+n2+n3;
sound(x,Fs)
pause(6)
mu=0.017
ha=adaptfilt.lms(128,mu);
[y,e]=filter(ha,x(:,1),d(:,1));
sound(y,Fs)
subplot(3,1,1)
plot(x)
grid on
xlabel('iterations')
ylabel('Amplitude')
title('Faded signal')
subplot(3,1,2)
plot(y)
grid on
ylabel('Amplitude')
xlabel('iterations')
title('filtered output')
subplot(3,1,3)
plot(e)
grid on
ylabel('Amplitude')
xlabel('iterations')
title('error signal')
```