Normalized Least Mean Square Algorithm-Interference Cancelling

Harshit Vijayvargia
Department of Computer & Information Science, University of Florida
Gainesville, USA
hvijayvargia@ufl.edu

Abstract— **The purpose of this report is to explain adaptive noise cancellation using NLMS (Normalized Least Mean Square) algorithm. Adaptive noise cancellation is a method to recover a signal from a noise corrupted signal. In this report, it is accomplished by designing an adaptive FIR filter. The reference input given to the filter is mostly noise and the desired output is the noise corrupted signal which is also fed into the filter as a primary input. The report evaluates performance of NLMS algorithm considering both cases when the signal is assumed to have similar characteristics throughout and when it is not. The main objective of our algorithm is to determines filter coefficients which will try to match the filter output as close as to the desired output. In this case the error obtained will be close to a noise free signal.**

*Key words— **Adaptive filters, least mean square (LMS), Normalized Least mean square (NLMS).***

I. INTRODUCTION

Adaptive Interference cancelling has applications in many areas such as noise control, vibration control, channel equalization, beam forming, echo cancellation etc. Adaptive filtering comes into place when we don’t have information about data. In this situation, the parameters of adaptive filter are evaluated based on the estimated statistical nature of statistics. The basic principle of adaptive noise cancelling is shown in the block diagram (Fig (a)). In all applications, the objective is the same, to recover a noise corrupted signal by suppressing uncorrelated noise in it. In the block diagram, it can be seen there are two sensors: One is reference sensor and other is the primary sensor. The primary sensor receives the combination of signal and interference $s(n) + x1(n)$ where $s(n)$ is the signal generated from source and $x1(n)$ is the interference in it. The second sensor receives an interference $x(n)$ which is correlated with $x1(n)$ but uncorrelated with $s(n)$.

The reference input is filtered by the adaptive filter to generate an output $y(n)$ which is as close to the interference at primary input. In this way, the canceller will produce an output $e(n) = d(n) - y(n)$ which is not only our error but also the recovered signal.

The algorithm which we will be using in this report is a stochastic gradient descent algorithm where the filter is only adapted based on the error at the current time. The most common form of adaptive filter is the transversal filter using least mean square (LMS) algorithm and NLMS algorithm. Since we are given less statistics about signal and noise, we will prefer NLMS over LMS. The main drawback of LMS is that the learning rate of the algorithm is fixed which does not provide stability to the algorithm. In NLMS we can normalize this learning rate with the power of input so if there are variations in signal statistics such as changes in signal amplitude then the filter will adapt to them.

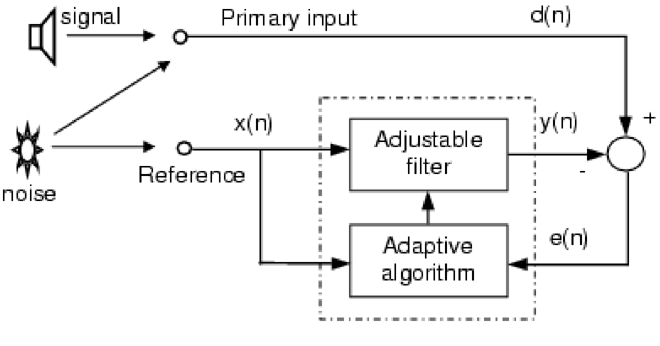

Fig (a) Block diagram of interference cancelling setup

II. Normalized Least Mean Square Algorithm

The main drawback of LMS algorithm is that it is sensitive to the scaling of its input $x(n)$. This makes it hard to choose a learning rate μ(n) that guarantees stability of the algorithm.

The Normalized least mean squares (NLMS) filter is a variant of the LMS algorithm that solves this problem by normalizing the learning rate with the power of the input. Here the learning rate is proportional to the inverse of the total expected energy of the instantaneous values of the coefficients of input vector $x(n)$. As the step size parameter is chosen based on the current input values, the NLMS algorithm shows far greater stability with unknown signals. Besides that, it is also observed that NLMS converges faster. These features have made it ideal for real time adaptive interference cancelling problems.

Each iteration of the NLMS algorithm requires these steps in the following order.

1. The output of the adaptive filter is calculated from input *x(n)*

$$y(n) = \sum_{i=0}^{N-1} w(n)x(n-i) = \mathbf{w}^T(n)\mathbf{x}(n)$$

2. Error signal is calculated from the difference between the desired *d(n)* and output of the filter.

$$e(n) = d(n) - y(n)$$

3. Step size is calculated which is the inverse of variance of input.

$$\mu(n) = \frac{1}{\mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n)}$$

4. Parameters for next iteration are updated and this process goes on till we get closer to optimal parameters.

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

III. Methodology

We will consider that the noise corrupted signal is stationary or in other words its statistics are not changing with time. Based on this assumption we will calculate the optimal parameters for whole set of data. If our signal was non-stationary or showing different trends in data, then we might have evaluated optimal parameters for different windows in given data.

In the end, we will be comparing our results based on Learning curves, Performance surfaces, Mean Square error, SNR and ERLE values. SNR value defines the quality of signal recovered and ERLE (Echo Return Loss Enhancement) gives the amount of loss introduced by the adaptive filter alone.

ERLE value depends on the size of the filter as well as the design of our algorithm. The higher the ERLE value, better the noise cancelling system. We will compute ERLE as:

ERLE(dB) = 10 log(E[d2]/E[e2]).

Where E[d2] is the send in power and E[e2] is the power of residual error immediately after the cancellation.

Choosing step size: For NLMS algorithm

$$\text{step size } (\mu) = \frac{a}{b+x(n)x'(n)}$$

where a is the adaptation constant for NLMS algorithm and it is between 0 and 2 to optimize the convergence rate. b is the regularization parameter added to prevent step size from becoming infinite when variance of signal is too low.

Considering the noise corrupted signal as stationary, I started my analysis with filter order= 2 and a = 0.05, b =0.001. The results which I obtained can be analyzed from the following figures:

a) Performance Surface*:* 3D plot between weights and Mean Square Error. We can observe from it that MSE is high and weights are not converging towards an optimal weight which can be also seen from weight tracks.

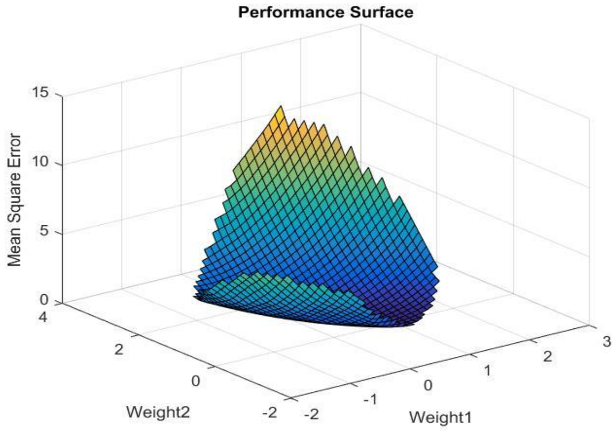

Fig:1

b) Weight Tracks for both weights:

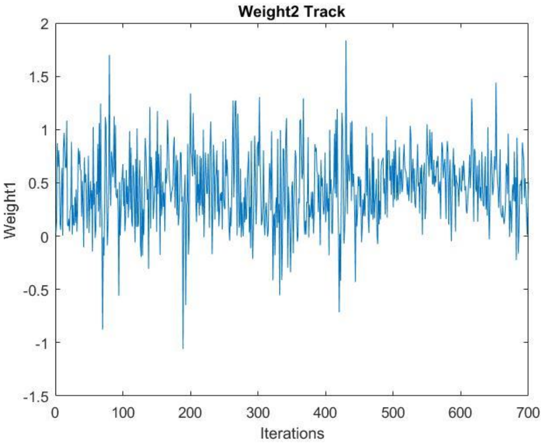

Fig:2a

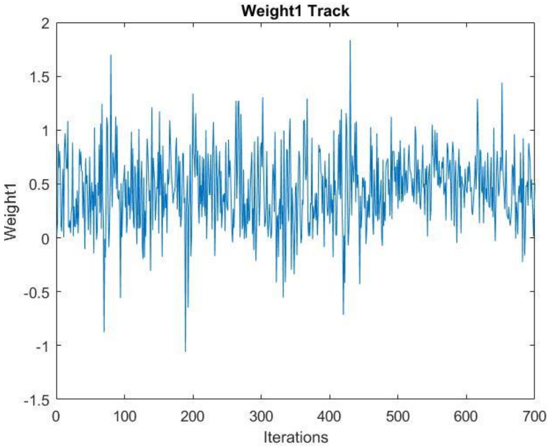

Fig:2b

From the weight tracks with weights taken after every 100th sample it can be observed that weights are not converging towards optimal weights. Their values are varying in a wider domain. Same argument could be made empirically by calculating ERLE value which comes out to be:

ERLE (order 2) = 5.9846 dB

Low ERLE value tells that the performance of our noise cancellation algorithm is not good. I verified it by assessing the recovered signal(Fig:4a) and the person's voice was still not clear. Either it may be due to the size of filter or our algorithm design.

c) Contour plot: It can be also seen from the Contour plot (Fig:3) that the performance of the filter is not optimal. The weights are not converging towards the center point which is the optimal point for minimum error.

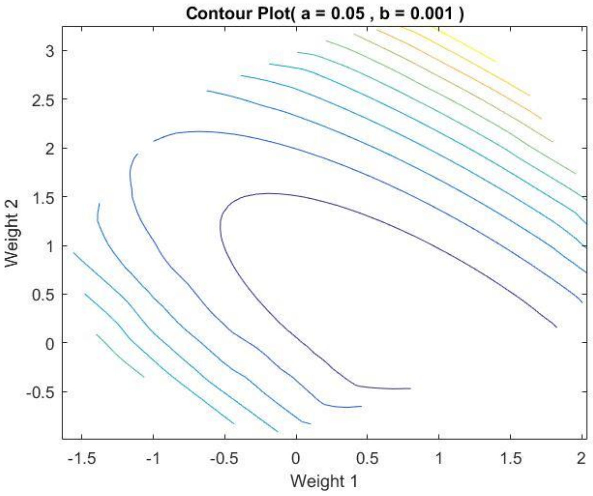

Fig:3

d) Recovered signal & Learning curve for order 2 at adaptation constant = 0.005:

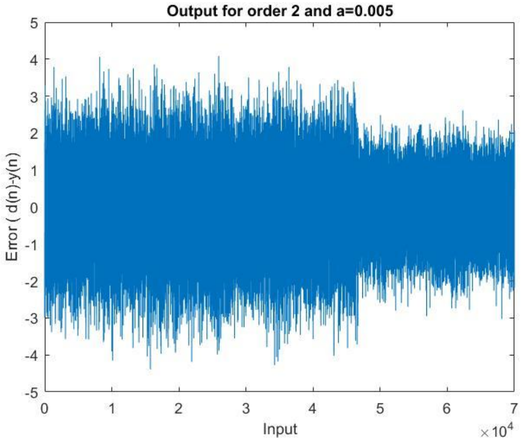

Fig:4a

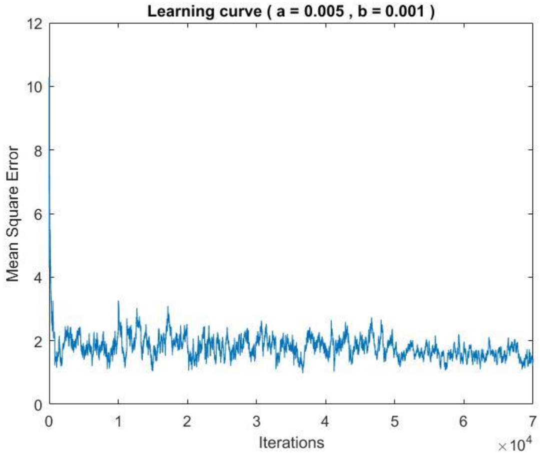

Fig:4b

Since the parameters are not converging I varied the step size and plotted learning curves (Fig:5a, Fig:5b) for a = 0.0005, 0.00005.

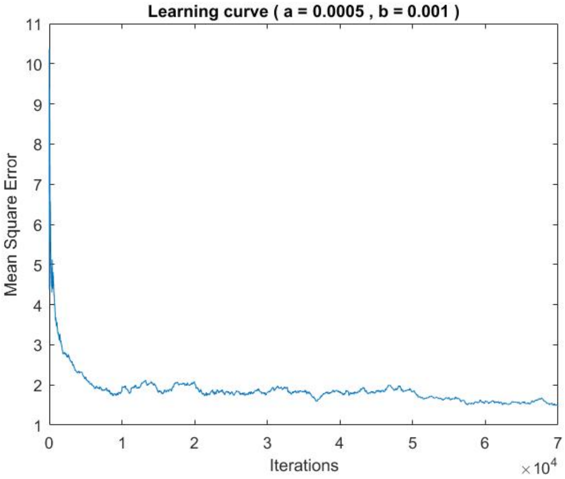

Fig:5a

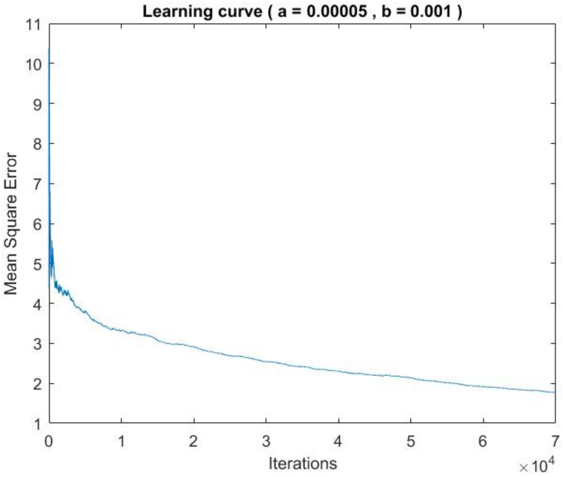

Fig:5b

Changing the step size improved the learning curve as it can be seen from the figures above but not significantly. There was no improvement in SNR value.

One more observation that can be made is for filter order 2 at a = 0.00005, it is taking more number of iterations to reach a steady state. In fact, weights are not converging well even after 70000 iterations.

III. Analysis

Since changing step size was not beneficial, I must look for other approaches which can improve the performance of filter. These approaches are:

a) *Cross-validation:* It is a way to check how well our model generalizes to the new data. If we don't validate the performance of our model on new data then a problem of overfitting may occur. There are three types of methods used for cross validation: Holdout method, K-Fold method and Leave one out cross validation method. These all methods are distinguished based on what data sets they take in their training set. For online mode or when we are working on non-stationary data it is better to use either K-fold or Leave-one-out method. In K-fold method we perform validation keeping each of k size data set in training. This randomness improves the performance of our algorithm.
   In this problem we can use K-Fold method but we must make sure that our data is showing similar characteristics throughout. For example: when data is in segments then we can perform validation for only those segments. At any point of time we can divide the samples which we have in k subsets and perform K-fold validation. We can evaluate our performance using ERLE values which is one of the measures to determine filter performance.

b) *Increasing filter order:* This is another approach to minimize error. Higher order implies more phase shift and higher noise reduction. I increased the order of the filter to 40 and the plotted the recovered output signal (Fig:5a) and learning curve (Fig:5b) at this filter order:

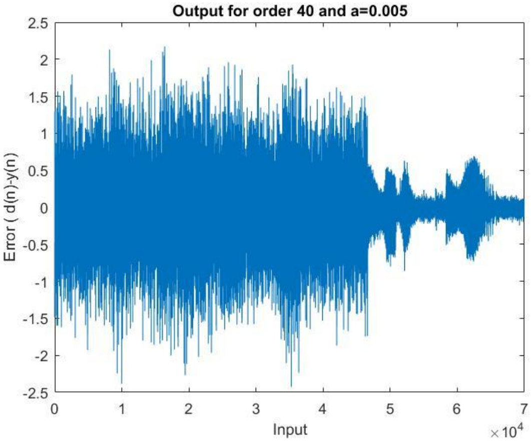

Fig:5a

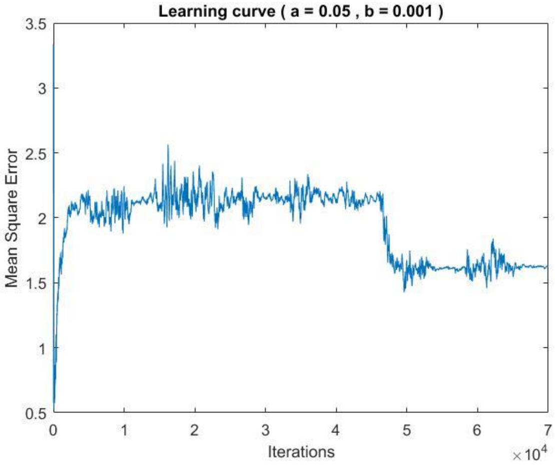

Fig:5b

From the figures above it can be clearly seen that increasing the filter order has significantly improved the filter performance but a different thing has come to notice. A part of data is only getting cleaned.

This can be also seen from the error plot I obtained by applying NLMS algorithm on another part of data.

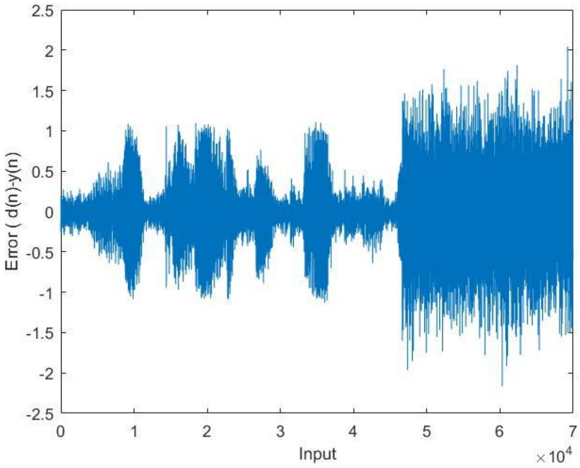

Fig:6

The only reason for such a behavior could be that our data set has segments with different statistics and patterns. This observation can also be justified empirically if I compare ERLE values of non-segmented and segmented data (part 1). The ERLE value of segmented data is more. So, our assumption that data has same pattern throughout does not hold true.

ERLE (Segmented): 19.89 dB
ERLE (Non-Segmented): 12.56 dB

Because of this analysis, we should change our approach. We should deal with segments of data which are showing similar statistics. So, I segmented my data into two parts and this partition was done at a point from where the plots are representing different models.

To determine the best filter order, I calculated ERLE values for each filter order (10,20,40,50). These values are shown in Table.1.

Table 1 (SNR V/s Filter Order)

Filter Order	2	10	20	40	50
SNR(dB)	5.98	17.61	17.66	18.06	17.6

As we are increasing the order of filter SNR values are also increasing. Since we know higher the SNR value better is the signal quality. I take filter with order 40 as my choice.

Now the next step is to determine the best step size for filter order 40. This can be done by plotting learning curves for different step sizes. I plotted Learning curves for each part of segmented data. The learning curves shown in Fig:7 & Fig:8 show a trend. They tend to converge better as I increase the step size. This trend can be empirically justified by checking SNR values with different step sizes (Table 2). It can be observed if I am increasing step size SNR value is also increasing.

Table 2 (SNR V/s Step Size, Order:40)

Data Segment 1				
Adaptation constant	0.5	0.05	0.01	0.001
SNR value	18.1	16.63	11.68	6.36
Data Segment 2				
Adaptation constant	0.5	0.05	0.01	0.001
SNR value	20.6	16.2	10.32	9.36

Learning curves for filter order 40 at different step sizes

Segment 1) 1 to 45000 samples

1) a =0.01, b=0.001

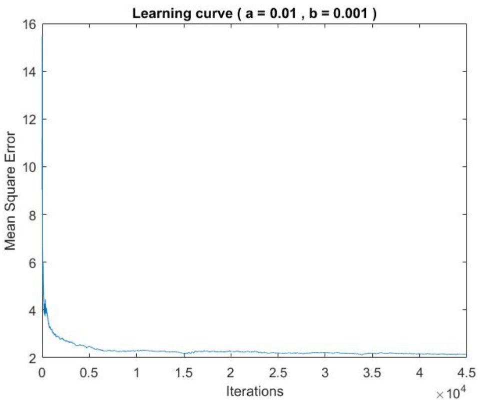

Fig:7a

2) a = 0.05, b = 0.001

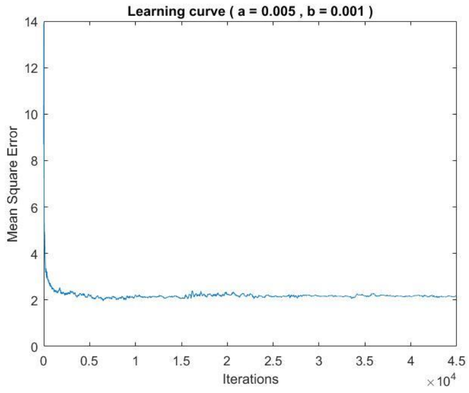

Fig:7b

3) a = 0.5, b = 0.001

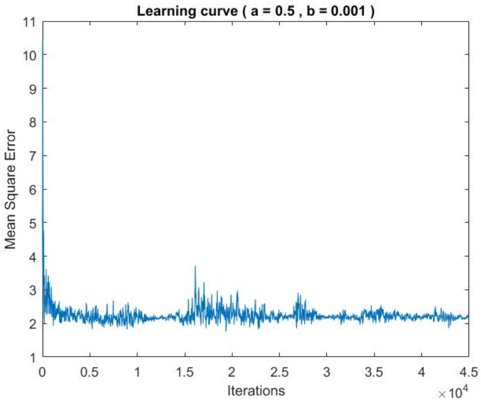

Fig:7c

Segment 2) 45001 to 70000 samples

1) a =0.01, b=0.001

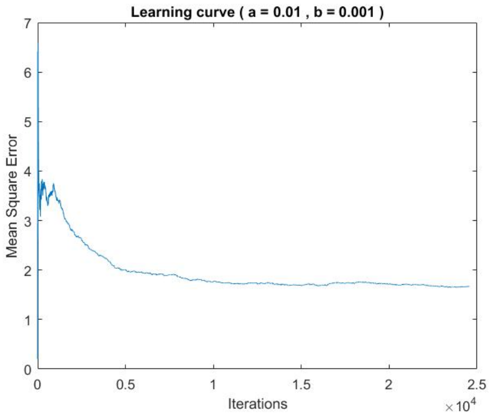

Fig:8a

2) a =0.05, b=0.001

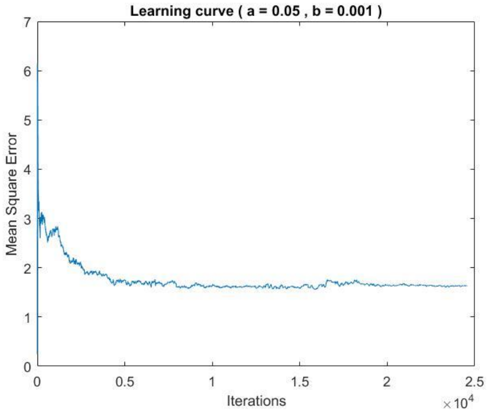

Fig:8b

3) a =0.5, b=0.001

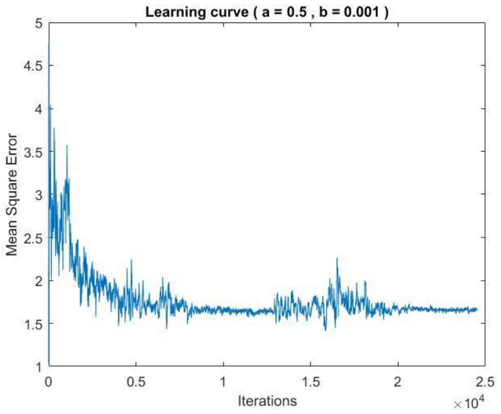

Fig:8c

IV. RESULT OF NLMS ALGORITHM

After comparing SNR values for different order of filters I found out that filter with order 40 is giving maximum SNR value. Considering this order as my choice I determined the step size which will give high convergence rate and less steady state error. This analysis was done by plotting learning curves and calculating SNR values for different step sizes.

Step Size at which the filter is showing optimal performance:

$$\text{step size } (\mu) = \frac{0.5}{0.001 + x(n)x'(n)}$$

These results can be verified by comparing plots of primary input(Fig:9a) which was noise corrupted signal and output(Fig:9b) which is the error or recovered signal. Besides that, these results agree when I performed the assessment of signal by hearing it.

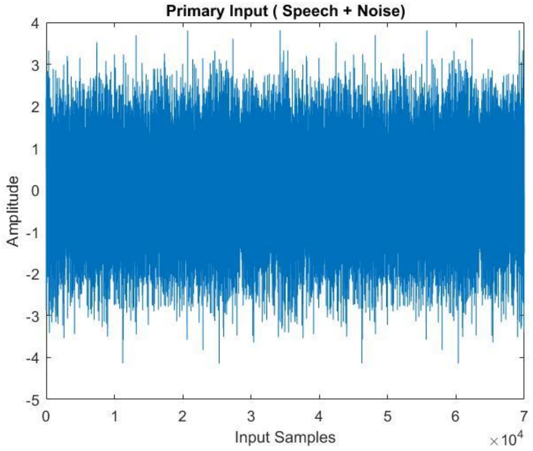

Fig:9a

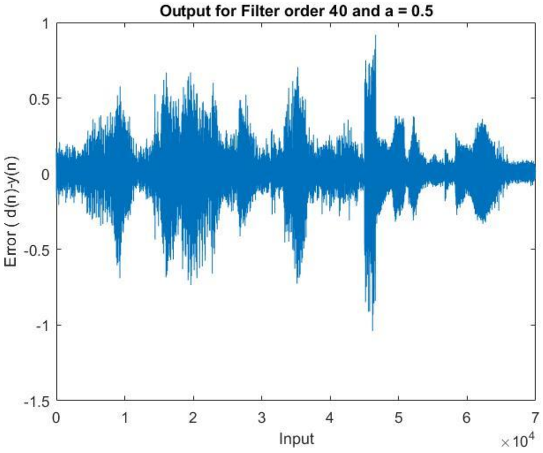

Fig:9b

V. CONCLUSION

In this report, we analyzed the performance of an adaptive algorithm known as NLMS (Normalized Least Mean Square Algorithm) for interference cancelling problem. We considered both cases when the signal is assumed to have similar characteristics throughout and when it does not. Since the signal was non-stationary our results in the first case were not good. But when we considered it non-stationary and performed data segmentation, we could determine filter coefficients which resulted in high convergence rate and low steady state error for each segmented part. Our empirical results based on ERLE values and learning curves also agreed with our speech assessment. We may, therefore, conclude that when we have limited statistics about data and our signal is showing different patterns then NLMS can act as a feasible robust solution.

REFERENCES

[1] V. Ramakrishna, T. Anil Kumar, SMIEEE “Low Power VLSI Implementation of Adaptive Noise Canceller Based on Least Mean Square Algorithm” International Conference on Intelligent Systems, Modeling and Simulation, pp.276-279, 2013 IEEE.

[2] J. Yadav, M. Kumar, R. Saxena, A. K. Jaiswal “Performance analysis of LMS adaptive FIR filter and RLS adaptive FIR filter For noise cancellation” Signal & image processing : an international journal (SIPIJ) vol.4, no.3, June 2013.4304

[3] G. Madhuri, B. Vijay Kumar, V. Sudheer Raja, M. Shasidhar, "Performance Analysis of Adaptive Algorithms for Noise Cancellation", CICN, 2011, Computational Intelligence and Communication Networks, International Conference on, Computational Intelligence and Communication Networks, International Conference on 2011, pp. 586-590, doi:10.1109/CICN.2011IEEE.127

[4] H. Yaghoobian, Ali Akbar Khazaei Mojtaba Salmani Zarchi S. Javad Sayed Hosayni “Modeling of a Robust and Fast Noise Cancellation System” Computer Modeling and Simulation, UKSIM European Symposium on 2011IEEE, pp. 210-213.

[5] Raj Kumar Thenua, S.K. Agarwal “Simulation and performance Analysis of adaptive filter in Noise cancellation” International Journal of Engineering Science and Technology Vol. 2(9), 2010, 4373-4378

[6] S. Salivahanan, A. Vallavaraj, C. Gnanapriya, Digital Signal Processing, 2nd ed., Published by Tata McGraw Hill Education Private Limited, 2010.

[7]Raj Kumar Thenua and S.K. AGARWAL” Simulation and Performance Analyasis of Adaptive Filter in Noise Cancellation” International Journal of Engineering Science and Technology Vol. 2(9), 2010, 4373-4378..

[8] S. Haykin, Adaptive Filter Theory, 3rd ed., Upper Saddle River, New Jersy : Prentice Hall ,1996.

[9] J. Benesty, et. al., “Adaptive Filtering Algorithms for Stereophonic Acoustic Echo Cancellation”, International Conference on Acoustics, Speech, and Signal Processing, 1995, vol.5, Page(s): 3099 – 3102