Implementation of NLMS Algorithm for Noise Cancellation

Suyog Daga, *Graduate Student, University of Florida*

***Abstract*—Among various applications of adaptive filters, an important application is Noise or Interference Cancellation. The basic idea of this method is to obtain an estimation of noise from the reference signal and to remove it from the corrupted primary signal. In this paper performance of NLMS adaptive filter have been analyzed to remove the noise from the primary signal. NLMS algorithm has been simulated in Octave. The adaptive algorithm eliminates noise from the primary signal, by establishing a correlation between noise and estimated value. The SNR (Signal to Noise Ratio) is a parameter used to evaluate the performance with respect to various step sizes. All the calculations related to SNR have been done using Octave.**

***Index Terms*— Adaptive filters, NLMS algorithm, SNR, Noise Cancellation, Signal to Noise Ratio**

I. Introduction

Signal Processing is an enormous field of Electrical Science which studies different aspects related to signals like their power, patterns, and properties. Interference cancellation, an important technique in the Signal Processing domain has been widely used in telephone communications. Noise cancellation can be achieved by adaptive algorithms like LMS or NLMS, where filters basically modify their coefficients to give us a best possible estimate of the noise.

One such technique used here is Least Mean Square algorithm[1]. Least mean squares (LMS) algorithms are a class of adaptive filters used to mimic the desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal (the difference between the desired and the actual signal). It is a stochastic gradient method in that the filter is only adapted based on the error at the current time[2]. The Normalized Least Mean Square can be seen as a kind of LMS variant that takes into consideration the variation of the signal level at the filter output. The performance of both the algorithms LMS & NLMS depends on upon their rate of convergence, filter coefficients and step size µ. The main objective of this paper is to investigate the application of NLMS adaptive filtering in noise cancellation problem.
Adaptive noise cancellation produces the noise free signal by applying filtering operation on the reference input and then subtract the noise from the primary input which is the addition of input signal and noise. To accomplish this it is required that the noise estimate should be as close to original noise. That is the main reason for using an adaptive filter as it makes $\hat{n}$ a close approximation of n[3].

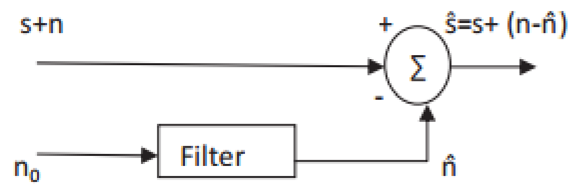

Fig 1.1 Input parameters for noise cancellation

The adaptive filter is able to adjust its impulse response so that an error signal will be minimized. The filter weights are updated by the adaptive algorithm.

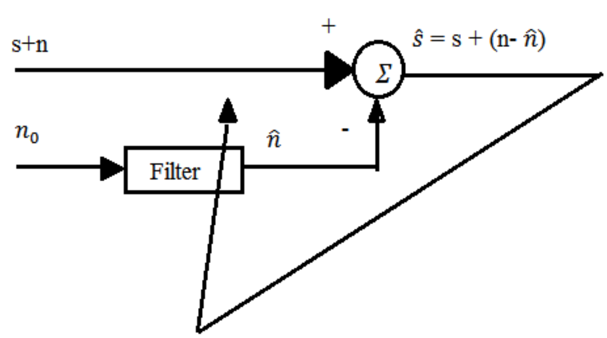

Fig 1.2 General Block diagram for adaptive filter

In the above diagram ,primary signal (signal corrupted with noise) is (s + n), where s is the original signal and n is noise. Reference signal is n_0 which is noise .

This paper organized as follows: Section II describes the general introduction and mathematical base of LMS and NLMS algorithm. Section III shows the results of implementation, different plots associated with it . Section IV concludes the work done.

II. ADAPTIVE NOISE CANCELLATION ALGORITHM

A. Least Mean Square (LMS) Algorithm

The best algorithm for adaptive filtering till now named as Least Mean Square (LMS) algorithm. For all linear estimation problems, the LMS algorithm plays a vital role for finding the excess mean square error and its efficiency has been studied extensively[1]. The aim was to reduce the error between the output of the filter and desired signal. This algorithm update's the coefficient of the filter $\hat{h}$(n) in each iteration to reduce error. This algorithm is a part of a group of stochastic gradient based algorithms.

The LMS algorithm[2] for a p^{th} adaptive filter order & step size μ can be described by following equations :

Parameters: p = filter order
μ = step size
Initialization : $\hat{h}$(0) = zeros (p)
Computation : For n = 0,1,2….
x(n) = [x(n),x(n-1),….,x(n-p+1)]^T
e(n) = d(n) - $\hat{h}^H$(n)x(n) …… (2.1)
$\hat{h}$(n+1) = $\hat{h}(n)$ + μe^*(n)x(n) ……. (2.2)

d(n) denotes the filter output, e(n) is estimated error signal which is obtained by substituting the value d(n) in equation (2.1). The computation of estimated error is based on currently estimated tap weight vector $\hat{h}$(n). Right-hand side of equation (2.2) adjusts the weight vector $\hat{h}$(n).

Above mentioned are general steps in an LMS algorithm for updating the weight vector. We initially start with $\hat{h}$=0.The LMS is a stochastic gradient algorithm that iterates each tap weight in the filter in the direction of the gradient of the squared amplitude of an error signal with respect to that tap weight. It is an approximation of the steepest descent algorithm, that uses an instantaneous estimate of the gradient vector[3]. The estimate of the gradient is done based on the basis of sample values of the tap input vector and an error signal. LMS algorithm iterates over each tap weight in the filter, rotating it in the direction of the approximated gradient[4].A proper step size is a must for the algorithm to converge as soon as possible. Small step sizes converge but require many iterations to converge. On the other hand, a very large step size may result in non-convergence, and we may not get the optimal weights for the expression. We have the following relationship for step size ,

$$0 < \mu < \frac{2}{\lambda_{max}} \qquad \ldots(2.3)$$

where λ_{max} = is the greatest eigenvalue of the autocorrelation matrix R , where R = x^Tx.

If the above-specified condition is not met , then algorithm becomes unstable and $\hat{h}$(n) diverges[2].In results section we see, keeping filter order constant how a change in step size , results in a different number of steps for convergence.

B. Normalized Step Size LMS (NLMS)

The main drawback of the LMS algorithm is that it is sensitive to the scaling of its input x(n). This makes it very hard to choose a step size μ that guarantees the stability of the algorithm[2]. The Normalized least mean square (NLMS) is an extension of the LMS algorithm that solves this problem by normalizing with the power of the input. The NLMS algorithm can be summarized as follows :

Parameters: p = filter order
μ = step size
Initialization : $\hat{h}$(0) = zeros (p)
Computation : For n = 0,1,2….
x(n) = [x(n),x(n-1),….,x(n-p+1)]^T
e(n) = d(n) - $\hat{h}^H$(n)x(n)
$\hat{h}$(n+1)=$\hat{h}(n)$ +(μe^*(n)x(n))/x^H(n)x(n) … (2.4)

The NLMS algorithm works similar to the LMS algorithm, the only difference is, coefficient updating equation in (2.3) is divided by the conjugate transpose of the input vector x(n). The output is obtained on the basis of the signal and the weight coefficient. The corresponding error is calculated and then the weights are subsequently updated for NLMS algorithm. A proper step size μ can help in faster convergence of NLMS algorithm.

III. SIMULATION AND RESULTS

- We are provided with two data sets , the primary which consists of pure vacuum cleaner noise and reference which consists vacuum cleaner and speech signal .
- Our objective is to learn on primary data , which is vacuum noise and to find optimal weights with respect to a given filter parameter and step size, so that we can generate the best value for noise , which would help us to eliminate noise signal from the reference signal, giving us a clear voice.
- Both the datasets, primary and reference data set has 70000 points.
- We train on the primary data set, get parameters or weights for that using proper filter order and step size, and use those parameters to predict on values which when subtracted from reference gives us the desired speech noise.
- The desired speech noise is " **I will not condone the course of action which will lead us to war**", a dialogue from Queen Amidala in Star Wars.
- Choosing proper filter order and the proper step size is very important as it will give us a clear and crisp speech.
- We start with filter order 2 and any random step-size

say 0.1 for initial assumptions. We train the model on this parameters and obtain corresponding values of error, weights , different weight values which were used by the model to set itself before reaching the optimal values, as well as the calculation of corresponding MSE with respect to each parameter.

- Weight parameters w1 and w2 are then used, with MSE to give us a performance surface plot and related contour plot which helps to visualize them.
- Below are both the plots related to two weights filter

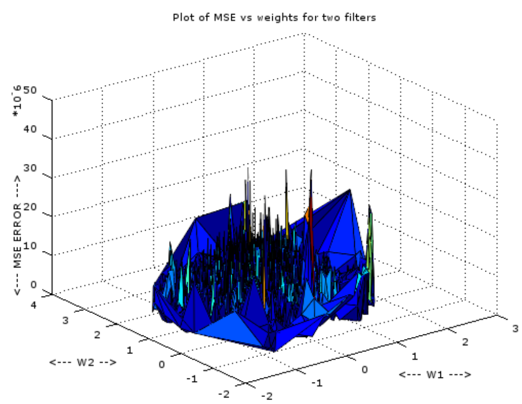

Fig 3.1 Plot of MSE vs weights for 2 weight filter

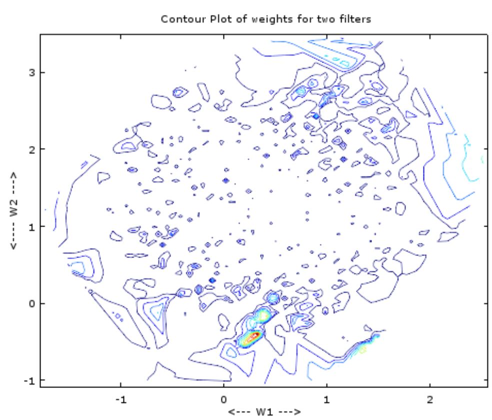

Fig 3.2 Contour plot for two weights

- From diagram (3.1) and (3.2), we see that the contour for two weights plot is little skewed and therefore would take some time to converge to get optimal weights, also seeing the MSE vs weights plot it gives us little idea about how the error is converging .
- For two weights filter, converging takes approximately 62000 iterations for converging. So this shows it learns very slowly We get an idea about how learning rate decreases over iterations , by plotting MSE against iterations. Different step-size can be used for plotting the same filter and we can get an intuition graphically as an how in general a learning curve coefficient stabilizes with respect to iterations. Figure (3.3)shows us a similar learning curve for filter order 2 and step size as 0.1

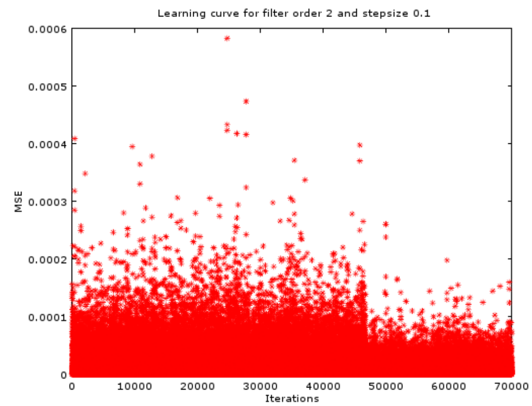

Fig 3.3 Learning curve for filter order 2 and step-size 0.1

- Similarly using different step sizes and same filter order we get learning curve for the corresponding step-sizes, we use filter order 18 and step-sizes 0.001,0.01 and 0,1 respectively. They converge in 42000 , 28300 and 24192 steps respectively.

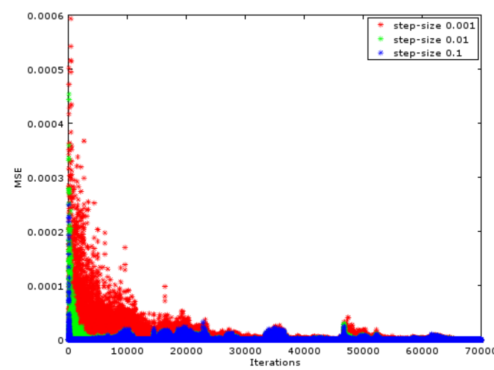

Fig 3.4 Learning curve for filter order 18 and different step-sizes

- But all the above results were for random filter orders and step-sizes. How do we calculate the optimum filter order and optimum step size ? Here is the simple solution for it.
    1. Take a range of filter order say 2 to 40.
    2. From (2.3) we know upper bound for step-size , we take 0.1 as the highest value and 10 different values from 0.001 to 0.1 range
    3. We plot this using performance surface and check where do we have minimum MSE on the surface.
    4. Minimum MSE corresponds to optimal filter order and step-size respectively.
    5. Minimum MSE means we have the best fit in the data and results in overall less value of MSE across all the points.
    6. Note filter order and step size range can have more values , like filter order in 2:50 and step-sizes 20 or so, after optimum weights MSE increases, so we restrict filter

order to 30.

- Here is the interesting plot of performance surface for the range of filter order in 2:40

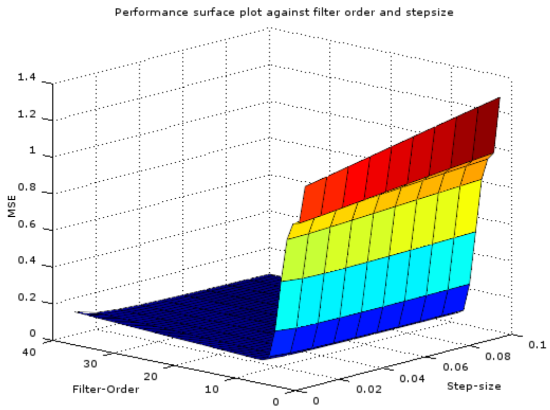

Fig 3.5 Plot of performance surface against filter order and step-size

- After checking the plot we get an idea that minimum filter order lies between 10-15 and best step-size is 0.1. Further checking matrix for minimum MSE we get filter order 14 and step-size 0.1 as the optimal parameters.
- ERLE calculation:
  ERLE measures the SNR improvement in signal. We calculate for filter order 10 , 14 and 20.

Filter Order	ERLE (DB)
10	40.247
14	43.345
20	41.734

- We can confirm this from our performance surface plot as well, as we get minimum error at filter order 14, and step size 0.1 , ERLE will be maxed at that point
- Following are the plots of the reference signal when unfiltered, the primary signal and speech after filtering.

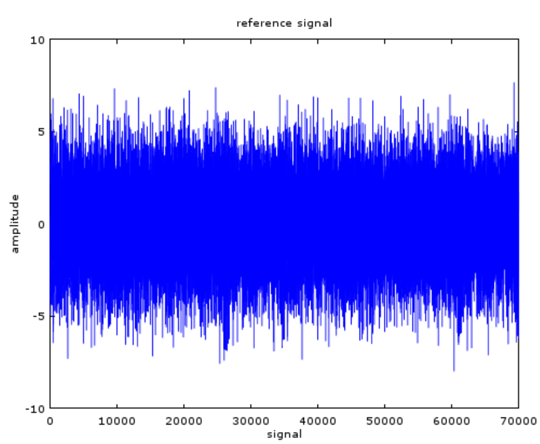

Fig 3.6 Reference Signal

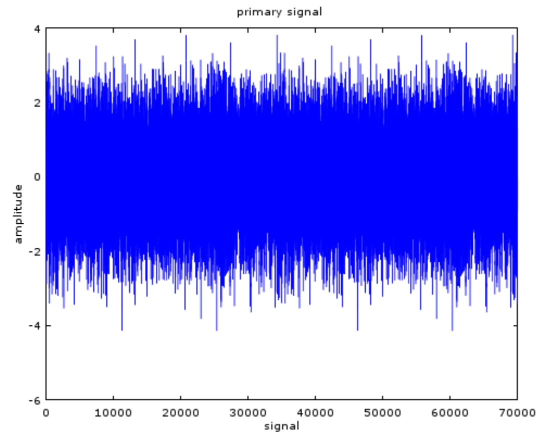

Fig 3.7 Primary Signal

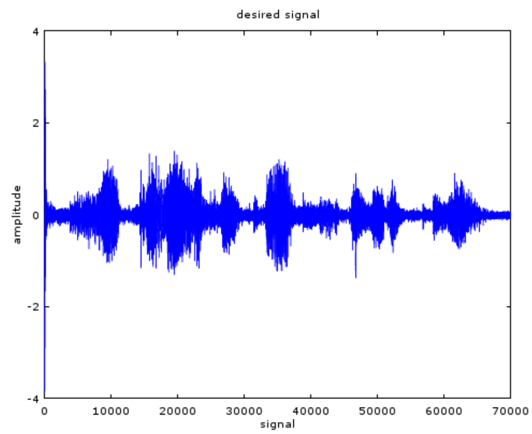

Fig 3.8 Desired Speech Signal

IV CONCLUSION

We have implemented NLMS algorithm for filtering out desired speech out of the reference algorithm. Various different filter size and step-sizes were used to compare the effect of the parameters on NLMS algorithm. We get optimal parameters for filter order 14 and step-size 0.1.While using this parameter we take approximately 24000 iterations to reach the minimum MSE. The desired speech is "**I will not condone the course of action which will lead us to war** ", we have first concluded our observations on basis of performance surface plot and then concluded on the basis of ERLE calculation. We see small filter order's give better results than bigger filter orders , keeping step size constant. A smaller step size converges slowly and a larger step-size converges fast. At the end, we get clear and crisp speech signal.

REFERENCES:

[1] Widrow, Bernard, and Marcian E. Hoff. "Adaptive switching circuits." In IRE WESCON convention record, vol. 4, no. 1, pp. 96-104. 1960.

[2]https://en.wikipedia.org/wiki/Least_mean_squares_filter

[3] Divya, Preeti Singh, Rajesh Mehra “Performance Analysis of LMS & NLMS Algorithms for Noise Cancellation” International Journal of Scientific Research Engineering & Technology (IJSRET) Volume 2 Issue 6 September 2013 ISSN 2278 – 0882

[4] Principe, Jose C., Neil R. Euliano, and W. Curt Lefebvre. Neural and adaptive systems: fundamentals through simulations with CD-ROM. John Wiley & Sons, Inc., 1999.