Project Report

LMS, NLMS and RLS algorithms for adaptive filtering of corrupted signal

Table of Contents

Title Page Number

Introduction--02

Adaptive Filter for noise Cancellation--03

Adaptive filtering using Least Mean Square (LMS) algorithm--04

Adaptive filtering using Normalized LMS algorithm---05

Adaptive filtering using Recursive Least Square (RLS) algorithm---06

Experiment and Results---07

Conclusion--13

References---14

1. Introduction:

Signals are mostly corrupted by real-world noise. Noise free signal is required for effective communication and information understanding. The noise from surrounding environment corrupts the original signal and the resultant may not be recognizable or understandable to hear.

The signal under observation holds low speech quality and different filtering techniques can be utilized to compute the original signal with no or less noise. To do this, I have used Least mean squares (LMS) algorithm. LMS is the class of adaptive filter which can reduce the noise from the corrupted signal using a reference signal. This reference signal also has noise which is corelated to the noise in the signal under consideration. This corelation only available in the noise but not in between the signals. Using mean squared error as measure the algorithm try to learn the system which can effectively replace the source signal with the given input of reference signal. Because there is no correlation between the reference and source signal, the adaptive filter will only learn the noise mapping. The subtraction of noise from the given signal will leads to a enhanced quality signal with less noise. The condition for this algorithm is to have the correlation between the noise of the reference and corrupted signal.

The adaptive filter learning method is useful when the noise and input speech signal is random in nature. In the past, there has been plenty of research on signal processing mainly for noise reduction using adaptive filtering for example, Mugdha and kharadkar represented a brief survey on audio noise cancellation techniques [1]. Based on complexity and convergence time they provide a comparison for different filter design and performance. They have also discussed different adaptive filter algorithm, some of them are as follows:

(1) Least mean square (LMS) algorithm
(2) Normalized LMS algorithm
(3) Recursive Least Square algorithm
(4) Variable Step size algorithm
(5) Least mean Fourth (LMF) algorithm

LMS is the simplest algorithm for adapting the filter. It has low computational complexity, and it is easy to implement. This algorithm uses gradient search method to reach the minimum mean squared error. The Normalized LMS is the extension of LMS with the difference in weight upgrade. In normalized LMS, the learning rate is dependent on energy of the signal at an instance. Now to converge speed is controlled by step size. The problem with the Normalized LMS as reported in [2] is that the learning rate does not depend on input signal which also has noise in it. To solve this Huang and Lee proposed variable step size Normalized LMS algorithm [2]. They take mean squared error alongside estimated value of the noise power to alter the learning rate or step size. Least mean Fourth algorithm achieves a comparable trade-off between the temporal and steady state performance of the adaptive filter. Although it is well performing but it has stability problem. The stability problems are addressed by normalizing the weights update term.

This project describes noise cancellation from a corrupted signal using adaptive filter. There are two signals, first signal is the one we want to remove the noise from, the second signal is the reference signal. The first signal is the corrupted signal containing and has noise alongside the clean signal. The reference signal is some other signal which has noise, and that noise is correlated with the noise of source signal.

2. Adaptive Filter for noise Cancellation:

Adaptive filters serve many purposes like equalisation, system identification, and noise cancellation. Noise cancellation is similar to system identification problem where the task is to find out a system which tells the relation between the noise translation from reference to source. The below figure gives a overall understanding of adaptive filter.

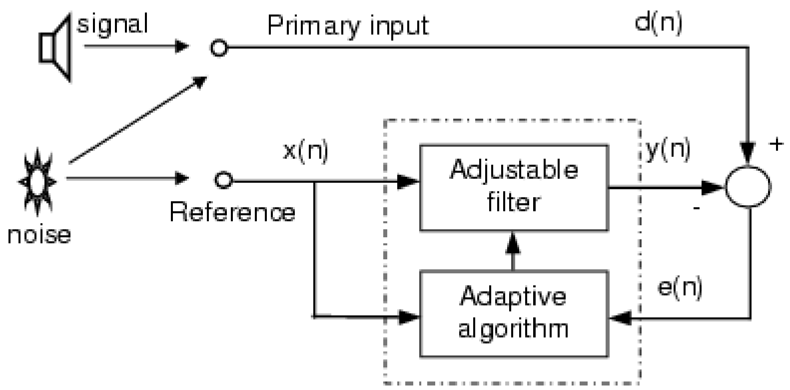

Figure 1: Adapting filter to cancel noise (taken from internet)

One thing to note in the figure is the configuration of the filter will remain same for any kind of signal input. The only thing we can change is how to effectively learn the adaptive filter. As discussed earlier there are different algorithms with there own pros and cons which are readily useable for adaptive filter learning. The main operation of adaptive filter is to learn such mapping that the end results is minimal error i.e., difference between the reference and source signal is minimum. Given the fact that the source signal and the reference signal are not corelated to each other how can a mapping be learned from reference signal to source signal. This is not possible unless there is some correlation. Again, the purpose of this configuration of adaptive learning is to reduce the noise in the source signal, so we need a system which can map the reference signal in such a way that it cancels out the noise in primary signal. To achieve that, there needs to be correlation in between the noise of the two sources i.e., primary source noise and reference source noise should have a relation achievable mathematically.

Let us say the d(n) is the primary signal and y(n) is the required signal from adaptive filter then

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

Let us assume that d(n) and x(n) are given by:

(2) $$d(n) = s(n) + Noise(n)$$

(3) $$x(n) = c(n) + E\ .\ Noise(n)$$

where s(n) is clean signal and d(n) is the corrupted signal and I termed as source or primary signal. x(n) is reference signal it can alone be a "E" weighted noise or may be combination of signal c(n) with weighted noise.

If all conditions are satisfied, i.e., if there is no correlation between clean signals, s(n) and c(n) but noise than this error will get the best possible value as:

(4) $$e(n) = s(n)$$

If there is no correlation, then the above equation will represent the best possible output of the error. It is obvious that we will not instantly reach this optimal point but will do iterative analysis to get to this stage. To do this iterative process we have options to choose the process by which we want to use the information to update the parameters of the filters.

In this section we had a quick overview of some of the algorithms for adaptive filtering. The next section will touch upon the math and brief understanding of these algorithm.

3. Adaptive filtering using Least Mean Square (LMS) algorithm:

This technique of finding optimal filter parameters is straight forward and uses gradient direction to take the next step. If we represent the adaptive filter values with "W" we can write the output of filter as:

(5) $$y(n) = W(n)^T . x(n)$$

Once we have y(n), we can compute the error using equation (1). As discussed earlier for upgrading the weights LMS algorithm uses gradient to get to the optimal solution so we can write:

(6) $$W(n+1) = W(n) + 2\beta e(n)x(n)$$

Where β is the step size, by changing the value of β we can control the convergence speed. Higher value of β means less convergence time whereas smaller value will result in more convergence time. Also, with a large β the algorithm may diverge, and we may not get the optimal filter coefficients. This depicts that the suitable value is required to be chosen for step size. Below figure show how the error is propagated with respect to different step size.

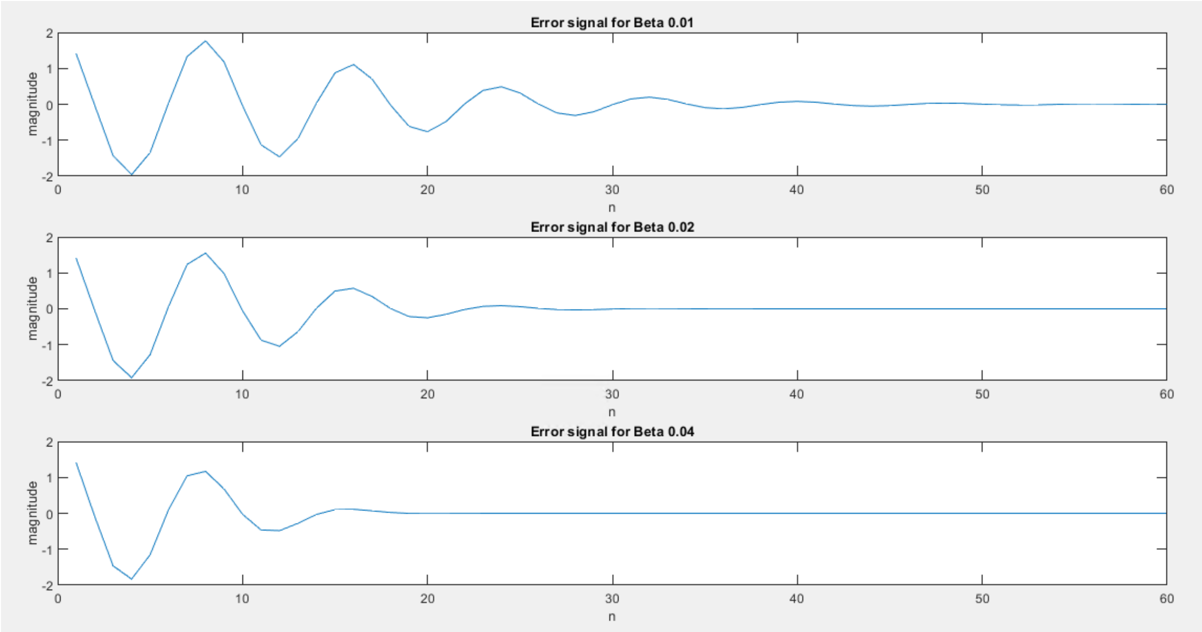

Figure 2: Error signal using LMS algorithm for different step size

It is evident from equation (6) that weight update is directly proportional to instantaneous value of x(n) which means, for high value of x(n) there will be amplified gradient which will result in abrupt change in the weight. Another problem with LMS algorithm is its fixed step size. Effectively choosing the step size needs statistical understanding of the signal which is highly unlikely to achieve. Considering all this LMS is not a very reliable algorithm.

4. Adaptive filtering using Normalized LMS algorithm:

This algorithm is same as LMS but now, instead of having a constant step size, a normalized step size is chosen. The normalization is done based on total energy of the instantaneous value of x(n). This Normalization helps to remove the gradient amplification problem and can be understand using the below equation:

$$W(n+1) = W(n) + \beta \frac{x(n)}{||x(n)||^2} e(n) \tag{7}$$

To get rid of division by zero we add a small constant η in the denominator the equation (7) then can be written as:

(8) $$W(n+1) = W(n) + \beta \frac{x(n)}{\eta + ||x(n)||^2} e(n)$$

Introducing normalization will make the convergence faster as compared to the simple LMS algorithm.

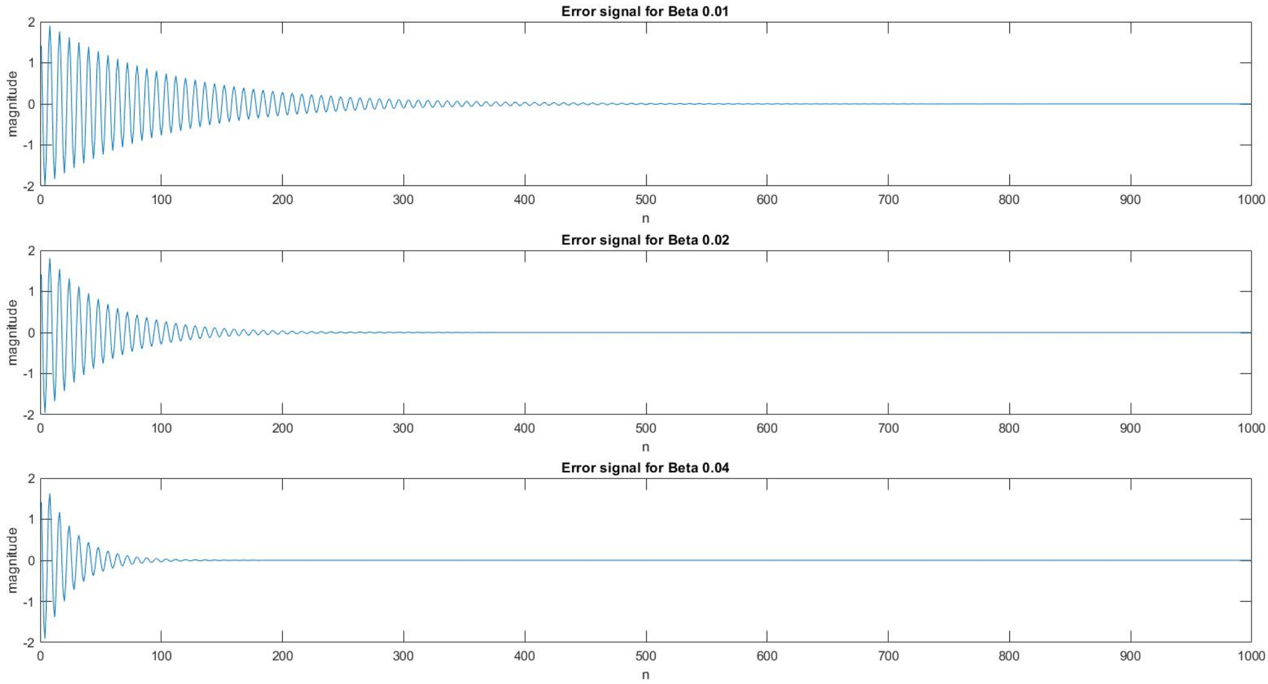

Figure 3: Error signal using Normalized LMS algorithm for different step size

It is clear from the figure 3 that the error signal is reducing steadily and does not change drastically. There are plenty of other algorithms which are complex but are more reliable. In the next section a different but more time efficient scheme is discussed.

5. Adaptive filtering using Recursive Least Square (RLS) algorithm:

As the name suggest it uses the same loss, least square, but in a recursive fashion. It recursively finds out the coefficients to minimize the weighted least squares instead of just least squares. The loss function can be understood as below equation:

(9) $$Cost = \sum_{i=0}^{n} \lambda^{n-i} e^2(i)$$

Where "λ" is called the forgetting factor. Its value can be chosen in the interval of $0 < \lambda \leq 1$. It gives less weight to previous error samples, that is why we call it has weighted least square loss. The smaller value of "λ" means less contribution from past value of error. With less value of λ we will encounter higher fluctuation in the weights because of the more sensitivity for the newer example data. Typical trade-off is to choose value of λ in range of 0.98 to 1. For Recursive Least Square algorithm we have:

$$W(n+1) = W(n) + K(n) \times e(n) \quad (10)$$

In the above equation “K(n)” defines how much importance is to assign the error signal, also called gain factor. It can be computed using below equation:

$$K(n) = \frac{P(n-1)\ X(n)}{\lambda + X^T(n) P X(n)} \quad (11)$$

As discussed earlier "λ" is the forgetting factor and "$P(n)$" represents the inverse of weighted covariance matrix of reference signal $x(n)$.

For the starting value of "$P(n)$" i.e., "$P(0)$" we use diagonal matrix with diagonal entries equal to δ. So, we can write:

$$P(0) = \delta I \quad (12)$$

After each iteration, the value of $P(n)$ will change and the new value can be calculated using the below equation:

$$P(n) = \frac{P(n-1) - K(n) X^T(n) P(n-1) X(n)}{\lambda} \quad (13)$$

In the above equation the $\mathrm{P(n-1)}$ represents the previous value of P.

In the next section, this report will present results and brief analysis of LMS, NLMS, and RLS algorithm.

6. Experiment and Results:

For experiment analysis I have used an audio signal which has 70000 samples with a sampling frequency of 21K Hz. There are two signals given, reference and primary. Both signals are corrupted with a noise from a same source i.e., there is correlation between the noise but there is no similarity between signals. I perform noise removal configuration of adaptive filter as described in Figure 1. For experiment evaluation I used three different algorithms to adapt the filter to noise. For analysis mean squared plot is used to understand the effectiveness of each of the algorithm. Signal to noise ratio (SNR) is reported together with different design evaluation of algorithm. To calculate mean squared error and signal to noise ratio one can use below equations:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (d(n) - y(n))^2 \quad (14)$$

$$SNR = \frac{\frac{1}{n} \sum_{i=1}^{n} (d(n))^2}{\frac{1}{n} \sum_{i=1}^{n} (e(n))^2} \quad (15)$$

Equation (14) calculates the mean squared error whereas equation (15) is signal to noise ratio. Given the fact that signals may have wide and dynamic range we convert SNR to logarithmic scale using below equation:

$$SNR_{\,dB} = 10\, log_{10}(SNR) \tag{16}$$

We have seen how the error propagate for different learning rate in the below table the results for simple LMS algorithm are reported.

Step size	MSE final value	Signal to Noise ratio
0.000001	2.9319	-4.3756
0.00001	0.5222	3.1178
0.0001	0.0666	12.0593
0.001	0.0516	13.1673
0.1	NaN	NaN

Table 1: LMS algorithm for different values step size, the order of filter is set to 21 coefficients.

As evident from table, having a large value of step size does not leads to optimal convergence point. With very large value of step size the algorithm may never converges like in case of step size of 0.1. The below figure gives an overview of how the filter is adapting over the time with different step size.

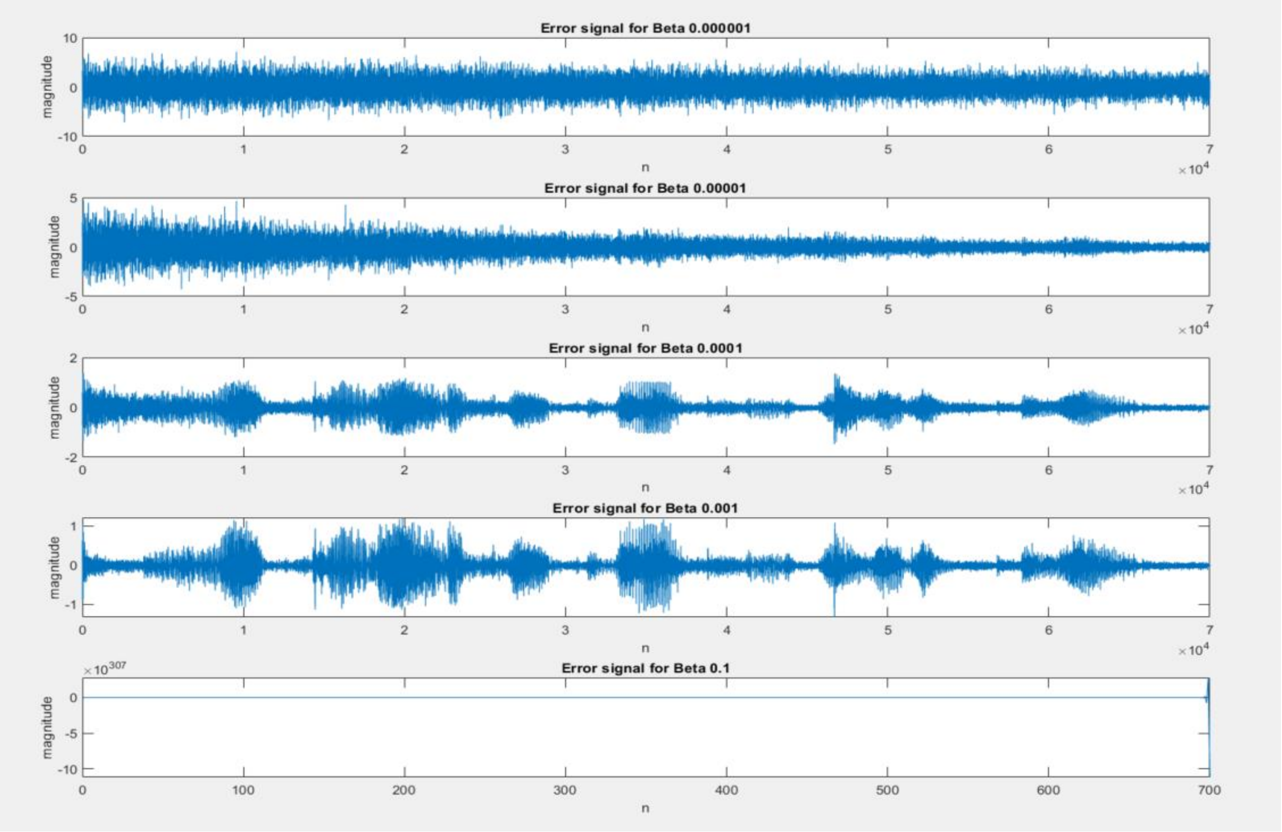

Figure 4: Error signal using **LMS algorithm** for different step size (Noise removal)

Figure 4 illustrates that using a very small step size will take very large amount of time for the filter weights to converge. Similarly, choosing very large value may not help us in getting to final solution and the algorithm may diverge. Figure 5 help us understand how the loss decreasing with respect to different step size.

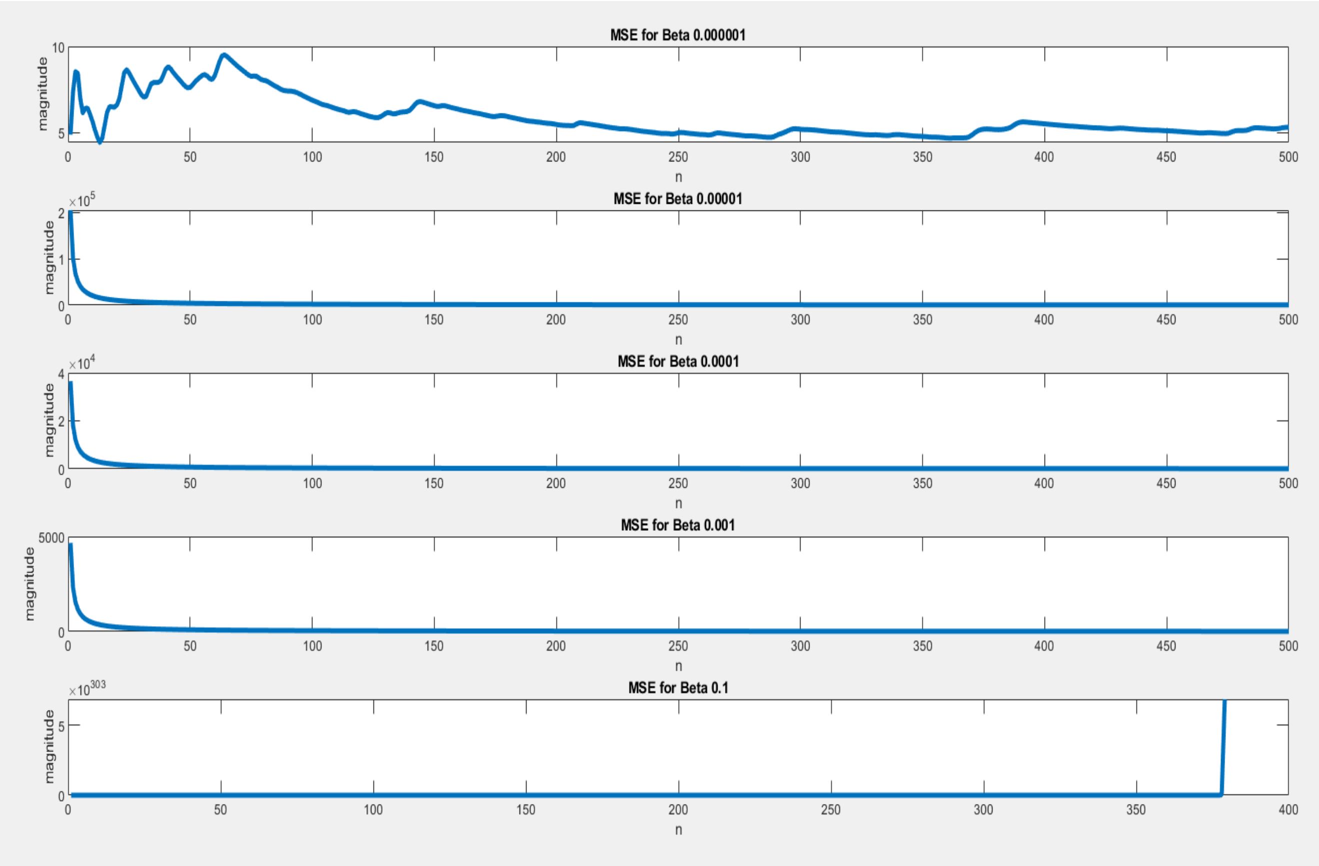

Figure 5: Mean Square error plot using LMS algorithm for different step size (Noise removal)

It is clear that with respect to each iteration MSE is decreasing for all the step size but for large step size it takes a big jump near 350th iteration and it start increasing. It is because of obvious reason of amplified gradient for instantaneous value of x(n).

Next, I have reported results for Normalized Least mean squared algorithm. The trend of NLMS is pretty much same as expected. It does not show any sudden spike like we had in LMS for a learning rate of 0.1. In Table 2 we can see for NLMS that for lower value of learning rate it is slow, there is requirement of having a large learning rate as compared to LMS. This configuration is also evident in Figure 6 where MSE is reported for each of the step size. Unlike in LMS MSE error drops nicely but not suddenly because of its less dependence on sudden changes in reference signal.

Epsilon	Step size	MSE final value	Signal to Noise ratio
0.001	0.000001	4.1271	-5.8606
	0.00001	3.6727	-5.3541
	0.0001	1.6701	-1.9316
	0.001	0.1803	7.7348
	0.1	0.0424	14.0198

Table 2: NLMS algorithm for multiple step size, the order of filter is set to 21 coefficients.

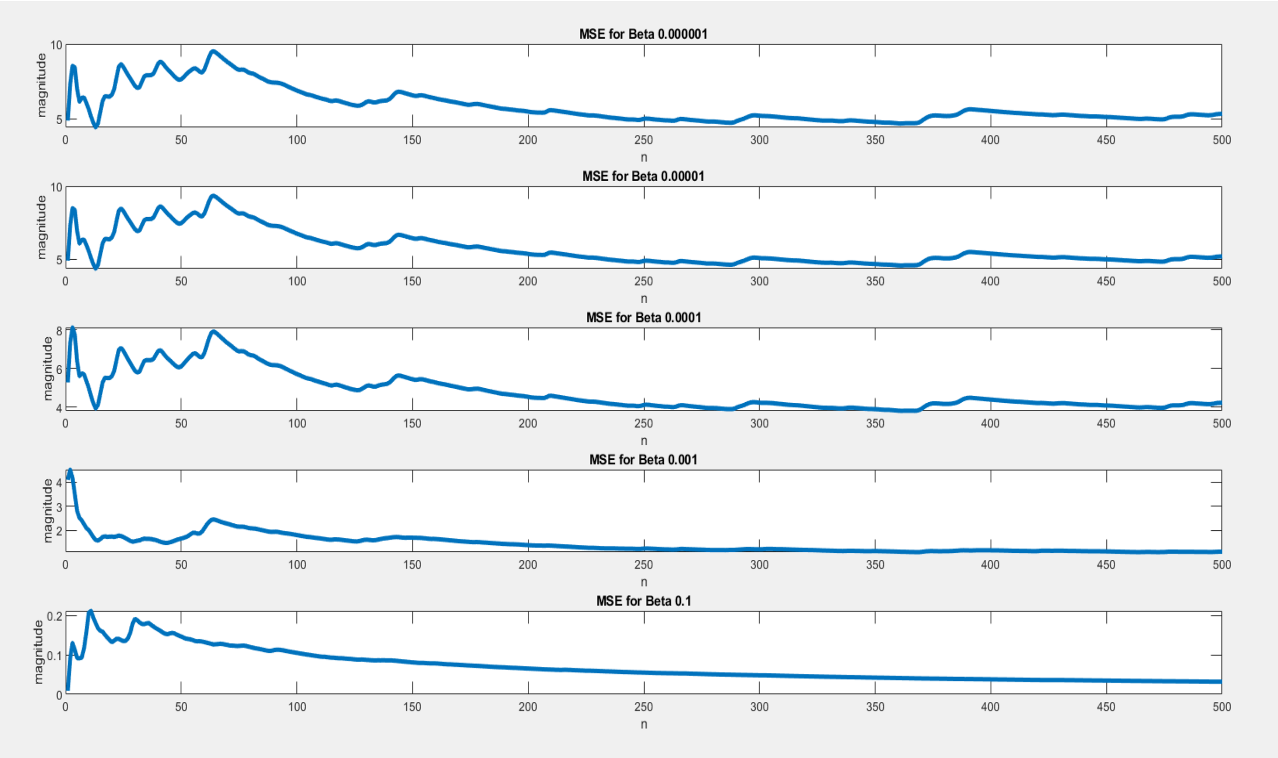

Figure 6: Mean Square error plot using **NLMS algorithm** for different step size

One of the major noticeable difference in MSE plots is the starting value of LMS and NLMS. LMS has very high MSE at start as compared NLMS almost of 10^5 order. The error plot for each iteration and update can be seen in Figure 7. This plot also aligns with the results we achieved in the from of MSE and SNR.

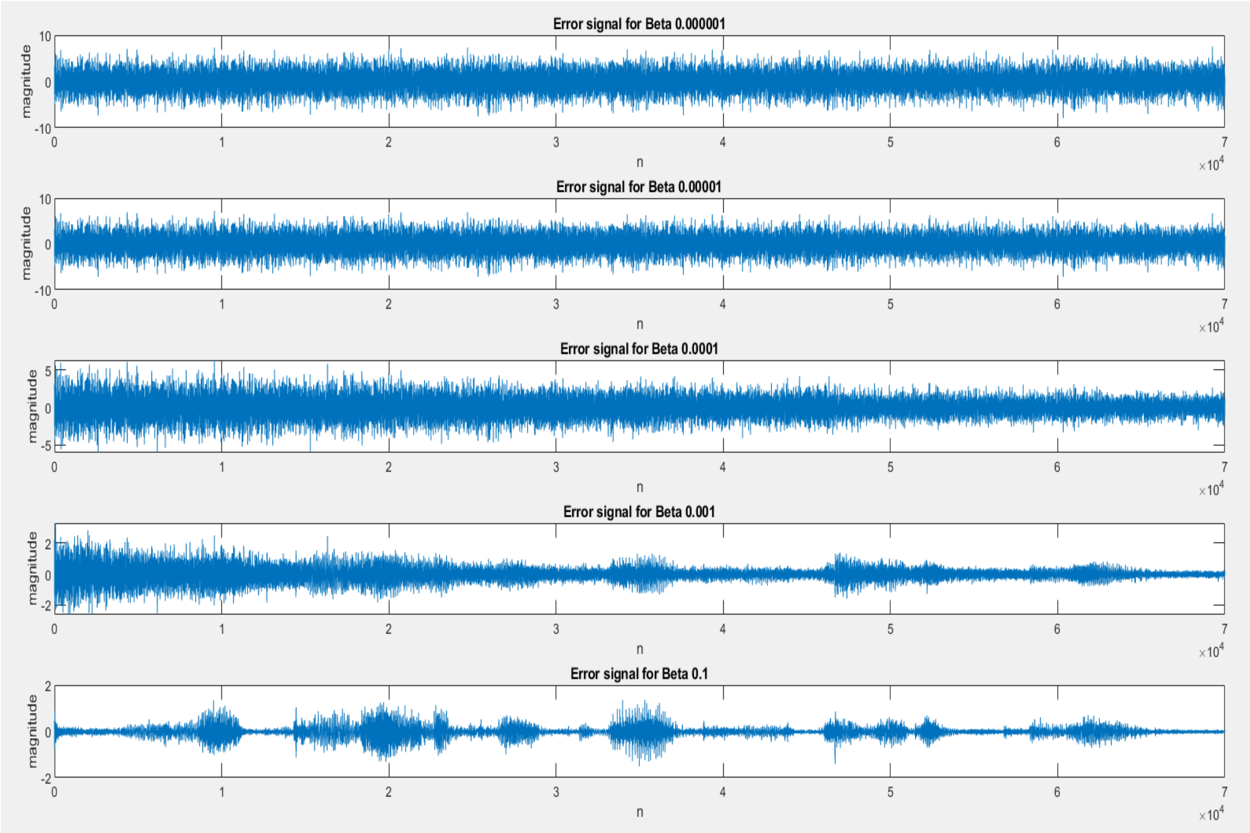

Figure 7: Error signal change with each iteration using **NLMS algorithm** for different step size (Noise removal)

The below table shows how quickly RLS adapt the coefficients evident from signal to noise ratio. Also, from MSE it is obvious that compared to previous algorithms RLS more quickly converge. In Figure 8, we can also see that the message which was originally corrupted is recovered in early stage. Figure 8 has error signal propagation for Recursive least square algorithm for multiple value of delta.

Lambda	delta	MSE final value	Signal to Noise ratio
0.99	0.01	0.0426	12.4781
	0.02	0.0583	12.6386
	0.04	0.0583	12.6409
	0.08	0.0583	12.6401
	0.1	0.0583	12.6410

Table 3: RLS algorithm for multiple step size, the order of filter is set to 21 coefficients

In Figure 9 MSE curve is attached for different values of delta another evidence of fast convergence.

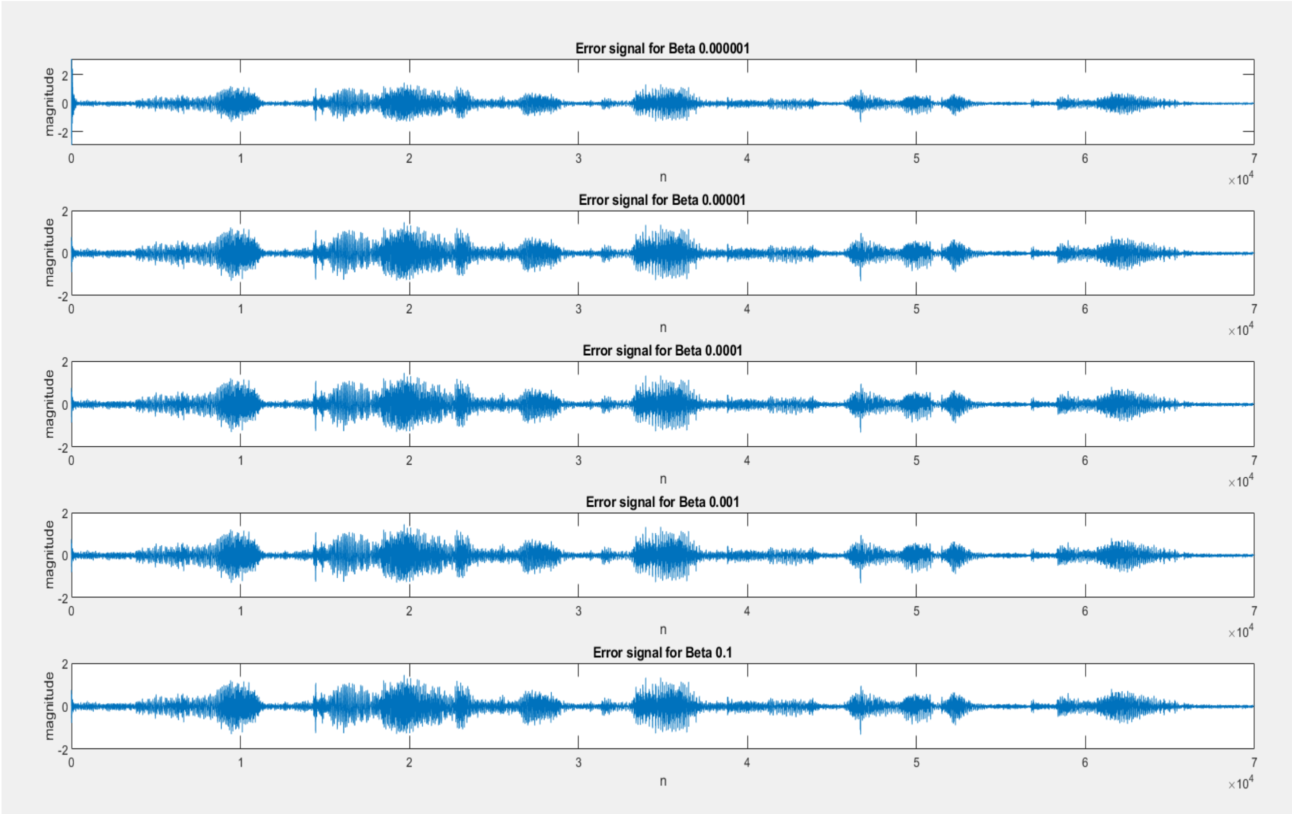

Figure 8: Error signal change with each iteration using **RLS algorithm** for different step size (Noise removal)

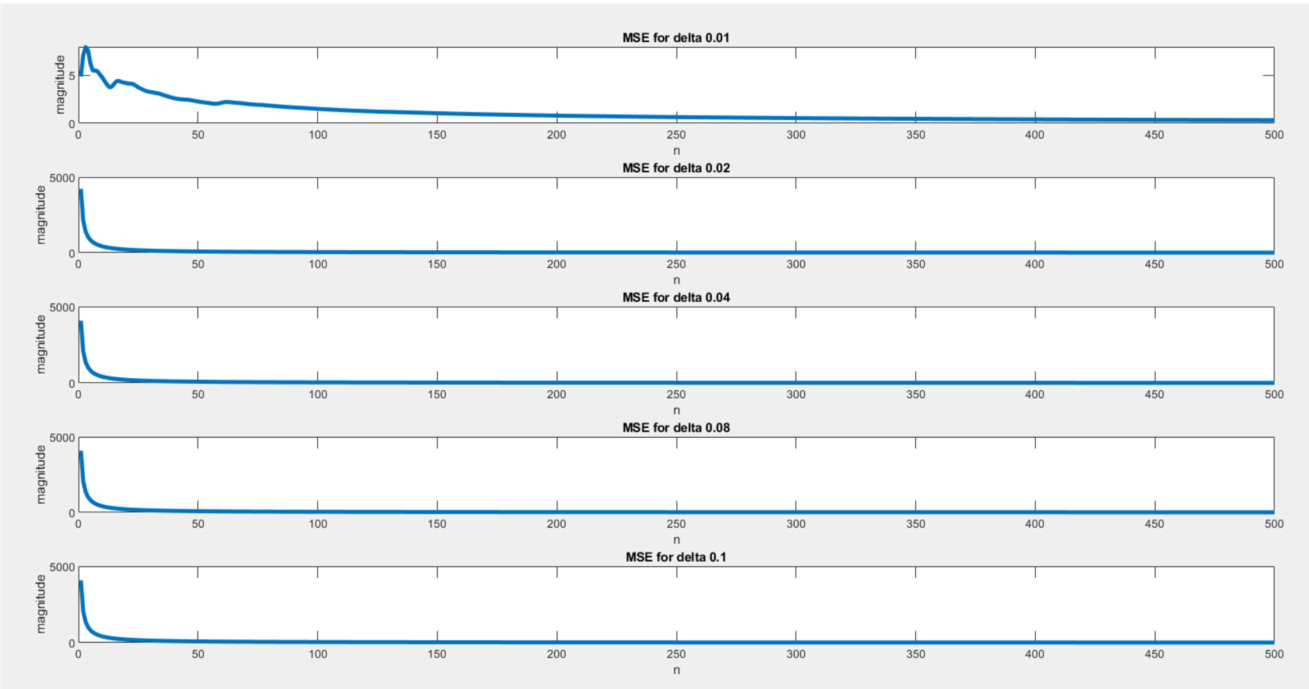

Figure 9: Mean Square error plot using **RLS algorithm** for different values of delta

Conclusion:

In this project I implemented three different algorithms for adapting the filter coefficients. I used 21 number of coefficients for all the algorithms. The Simple and less complex, LMS, little complex but rewarding, NLMS, and more complex and highly competitive, RLS, are implemented. The key findings are:

1. LMS results in amplified gradient and may not diverge.
2. RLS is computationally expensive but rewarding.
3. Faster convergence in RLS as compared to LMS
4. RLS accounts for past values of signal whereas LMS do not.
5. LMS only minimize the current mean square error only.
6. RLS takes weighted squared error into account.

References:

[1] Dewasthale, Mugdha M., and R. D. Kharadkar. "Acoustic noise cancellation using adaptive filters: A survey." In *2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies*, pp. 12-16. IEEE, 2014.

[2] Huang, Hsu-Chang, and Junghsi Lee. "A new variable step-size NLMS algorithm and its performance analysis." *IEEE Transactions on Signal Processing* 60, no. 4 (2011): 2055-2060.