NOISE SUPPRESSION IN SPEECH SIGNALS USING ADAPTIVE ALGORITHMS

V.JaganNaveen [a], T.prabakar [b], J.Venkata Suman [c], P.Devi Pradeep [d]

[a]Associate Professor, Department of E.C.E, GMR Institute of Technology, Rajam-532127, India, Mobile: +91- 9849722092 E-mail: jagannaveen_v@yahoo.co.in
[b,c,d]Asst Professor, Department of E.C.E, GMR Institute of Technology, Rajam-532127, India, Mobile:+91- 9440399180 ,E-mail: [b] emailtprabhakar@gmail.com
[c] jami.venkatasuman@gmail.com, [d] pradeep.nit07@gmail.com.

Abstract

Adaptive Filtering is a widely researched topic in the present era of communications. When the received signal is continuously corrupted by noise where both the received signal and noise change continuously, then arises the need for adaptive filtering. The heart of the adaptive filter is the adaptive algorithm. This paper deals with cancellation of noise on speech signals using two algorithms-Least Mean Square (LMS) algorithm and Recursive Least Squares (RLS) algorithm with implementation in MATLAB. Comparisons of algorithms are based on SNR and tap weights of FIR filter. The algorithms chosen for implementation which provide efficient performances with less computational complexity.

Keywords: *LMS, RLS, Adaptive FIR filter, SNR, Gaussian noise.*

1. Introduction

In this modern world we are surrounded by all kinds of signals in various forms. Some of the signals are natural, but most of the signals are man-made. Some signals are necessary (speech); some are pleasant (music), while many are unwanted or unnecessary in a given situation.

In an engineering context, signals are carriers of information, both useful and unwanted. Therefore extracting or enhancing the useful information from a mix of conflicting information is a simplest form of signal processing [1]. More generally, signal processing is an operation designed for extracting, enhancing, storing, and transmitting useful information. The distinction between useful and unwanted information is often subjective as well as objective. Hence signal processing tends to be application dependent. In contrast to the conventional filter design techniques, adaptive filters do not have constant filter coefficients and no priori information is known. Such a filter with adjustable parameters is called an adaptive filter.

The basic idea of an adaptive noise cancellation algorithm is to pass the corrupted signal through a filter that tends to suppress the noise while leaving the signal unchanged. This is an adaptive process, which means it does not require a priori knowledge of signal or noise characteristics.

Although both FIR and IIR filters can be used for adaptive filtering, the FIR filter is by far the most practical and widely used. The reason being that FIR has adjustable zeros, and hence it is free of stability problems associated with adaptive IIR filters that have adjustable poles as well as zeros. However the adaptive FIR filters are not always stable and their stability depends critically on the algorithm [6].

2. Least mean square adaptation algorithm

Because the exact measurements of the gradient vector are not possible and since that would require prior knowledge of both the correlation matrix **R** of the tap inputs and the cross correlation vector **p** between the tap inputs and the desired response, the optimum wiener solution could not be reached. Consequently, the gradient vector must be estimated from the available data when we operate in an unknown environment.

After estimating the gradient vector we get a relation by which we can update the tap weight vector recursively as:

$W(n+1) = w(n) + mu\, u(n)\, [d^*(n) - u`(n)\, w(n)]$

Where mu = step size parameter

$u`(n)$ = hermit an of the matrix u

$d^*(n)$ = complex conjugate of $d(n)$

Eventually we may write the result in the form of three basic relations as follows:

1. *Filter output*:
   $y(n) = w(n)\, u(n)$ (2.1)
2. *Estimation error or error signal:*
   $e(n) = d(n) - y(n)$ (2.2)
3. *Tap weight adaptation:*
   $w(n+1) = w(n) + mu\, u(n)\, e^*(n)$ (2.3)

Equations 2.1&2.2 define the estimation error e(n), the computation of which is based on the current estimate of the tap weight vector, w(n).Note that the second term, mu u(n) e*(n),on the right hand side of equation 2.3 represents the adjustments that is applied to the current estimate of the tap weight vector, w (n).The iterative procedure is started with an initial guess w (0).The algorithm described by equations 2.1and 2.3 is the complex form of the adaptive least mean square algorithm. At each iteration or time update, this algorithm requires knowledge of the family of stochastic gradient algorithms. In particular, when the LMS algorithm operates on stochastic inputs, the allowed set of directions along which we "step" from one iteration to the next is quite random and therefore cannot be thought of as consisting of true gradient directions.[3]

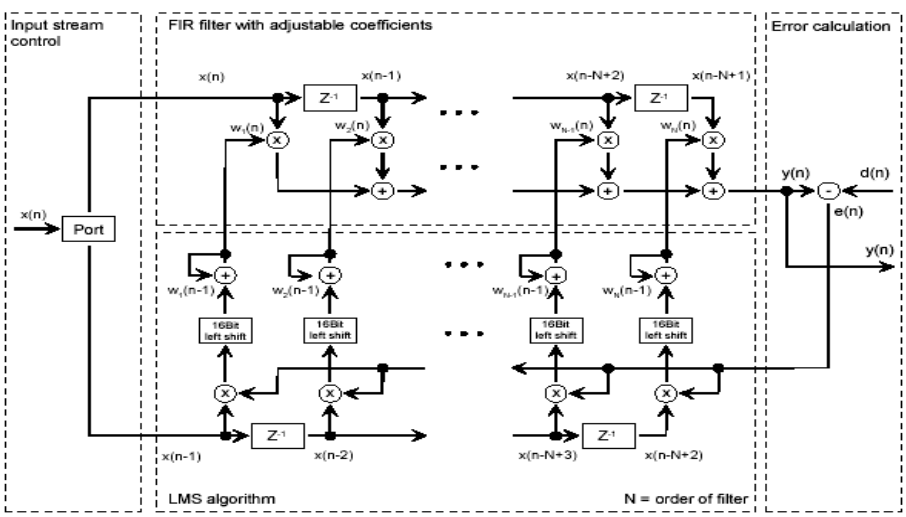


Fig 2.2: A detailed description of the LMS adaptive filter

3. Recursive Least Squares (RLS) Algorithm:

The RLS algorithm seeks to obtain the minimum mean squared error by obtaining the maximum estimate of the filter taps. It is based on the fact that if the error has

Zero mean
Statistically independent
Gaussian distribution

3.1 Algorithm Implementation:

The RLS algorithm is implemented as follows assuming c is a small constant, I is an identity matrix, the inverse of a correlation matrix P(n) can be initialized as

$P(0)= R^{-1}(0) = c^{-1} .I$ initially

Start with the initial conditions

$P(0)= c^{-1} .I$

$\lambda = 0.95$ (this can be modified but kept < 1)

$h(0) = 0$ (All initial tap weights set as 0)

Compute the gain vector

$$K(n) = \frac{P(n-1)\, u^*(n)}{\lambda + u^T(n)\, P(n-1)\, u^*(n)}$$

Compute the error

$e(n) = r(n) – h(n-1)\, u^T(n)$

Update the estimate of coefficient vector

$h(n) = h(n-1) + k(n)\, e(n)$

Update the inverse of the weighted auto correlation matrix

$P(n) = 1/\lambda\, [P(n-1) – k(n)\, u^T(n)\, P(n-1)]$

Increment n by 1; go back to step 2, and repeat the procedure.

4. Noise cancellation

The basic noise-canceling situation is illustrated in figure 4.1. A signal is transmitted over a channel to a sensor that receives the signal plus an uncorrelated noise n0. The combined signal and noise, s+n0, form the "primary input" to the canceller. A second sensor receives a noise n1, which is uncorrelated with the signal but correlated in some unknown way with the noise n0. This sensor provides the "reference input” to the canceller. The noise n1 is filtered to produce an out put y that is a close replica of n0. This output is subtracted from the primary input s+n0 to produce the system out put, s+n0-y.[9]

If one knew the characteristics of the channels over which the noise was transmitted to the primary and reference sensors, one could, in general, design a fixed filter capable of changing n1 into y=n0. The filter out put could then be subtracted from the primary input, and the system output would be the signal alone. Since, however, the characteristics of the transmission paths are assumed to be unknown or known only approximately and not of a fixed nature, the use of a fixed filter is not feasible. Moreover, even if a fixed filter were feasible, its characteristics would have to be adjusted with a precision difficult to attain, and the slightest error could result in increased out put noise power. In the system shown in figure 4.1, the reference input is processed by an adaptive filter that automatically adjusts its own impulse response through a least-squares algorithm such as LMS that responds to an error signal dependent, among another things, on the filter's output. Thus with the proper algorithm, the filter can operate under changing conditions and can readjust itself continuously to minimize the error signal. We have seen that the error signal used in an adaptive process depends on the nature of the application. In noise -canceling systems the practical objective is to produce a system output, s+n0-y, that are a best fit in the least -squares sense to the signal s. This objective is accomplished by feeding the system output back to the adaptive filter and adjusting the filter through an adaptive algorithm to minimize the total system output power. In an adaptive noise-canceling system, in other words, the system out put serves as the error signal for the adaptive process.

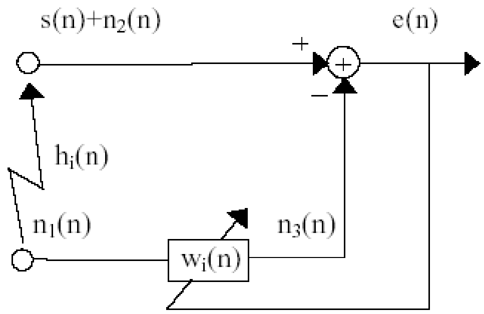


Fig 4.1: Adaptive Noise Canceller

Assume that s, n0, n1, and y are statistically stationary and have zero means. Assume that s is uncorrelated with n0 and n1, and suppose that n1 is correlated with n0. The output is

$$\varepsilon = s + n0 - y \qquad (4.1)$$

Squaring, one obtains

$$\varepsilon^2 = s^2 + (n0 - y)^2 + 2s(n0 - y) \qquad (4.2)$$

Taking expectations of both sides of (4.2), and realizing that s is uncorrelated with n0 and with y, yields

$$E[\varepsilon^2] = E[s^2] + E[(n0 - y)^2] + 2E[s(n0 - y)]$$
$$= E[s^2] + E[(n0 - y)^2] \qquad (4.3)$$

The signal power E $[s^2]$ will be unaffected as the filter is adjusted to minimize E $[e^2]$

Accordingly, the minimum out put power is:

$$E_{min}[\varepsilon^2] = E[s^2] + E_{min}[(n0 - y)^2] \qquad (4.4)$$

When the filter is adjusted so that E [ε2] is minimized, E [(N0-Y) 2] is therefore also minimized. The filter out put y is then a best least -squares estimate of the primary noise n0. Moreover, when E $(n0-y)^2]$ is minimized , $E[(\varepsilon-s)^2]$ is also reference input .

$$(\varepsilon - s) = (n0 - y) \qquad (4.5)$$

Adjusting or adapting the filter to minimize the total out put power is thus tantamount to causing the out put e to be a best least -squares estimate of the signal s for the given structure and adjustability of the adaptive filter and for the given reference input. [3] The out put e will generally contain the signal s plus some noise. From (4.1), the out put noise is given by (n0-y). Since minimizing the out put noise power and, since the signal in the out put remains constant, minimizing the total output power maximizes the out put signal-to-noise ratio.

We see from (4.3) that the smallest possible out put power is Emin $[\varepsilon^2] = E[s^2]$.When this is achievable ,$E[(n0-y)^2] = 0$. Therefore, y = n0 and ε = s. In this case, minimizing output power causes the out put signal to be perfectly free of noise.

5. Comparison of Adaptive algorithms

The performance of these adaptive algorithms is highly dependent on their filter order and signal condition. Furthermore, the performance of the LMS algorithm also depends on the selected convergence parameter. As for the RLS algorithm, it is also dependent on as" forgetting factor" or" exponential weighting factor.[7] Compared to the LMS algorithm, the RLS algorithm has the advantage of fast convergence but this comes at the cost of increasing the complexity. Hence, the RLS algorithm requires longer computation time as well as a higher sensitivity to numerical instability.

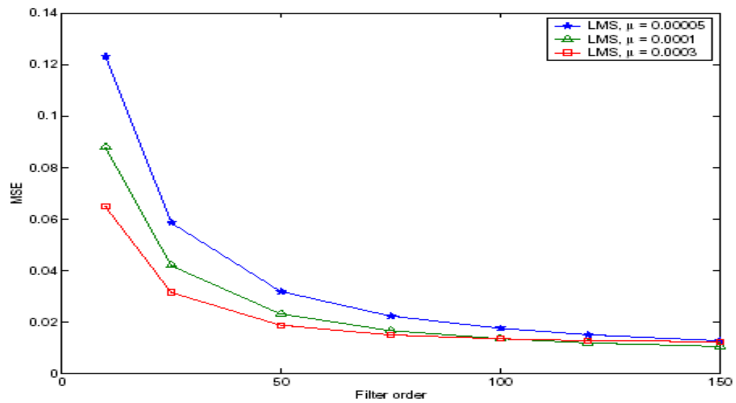


Fig 5.1: Mean Squared Error performance of the LMS algorithm with different filter orders

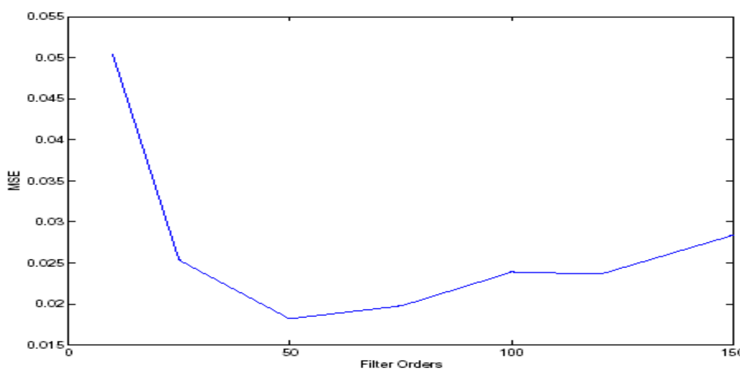


Fig 5.2: Mean Squared Error performance of the RLS algorithm with different filter orders

Fig 5.1 and Fig 5.2 show the MSE performance of the LMS algorithm and the RLS algorithm with different filter orders. Again, LMS algorithm provides similar performance results.

- The LMS algorithm performs much better than the RLS algorithm in the high filter order region. As the RLS is highly sensitive to numerical instability, the filter order will severely affect the performance of the algorithm.
- Fig 5.2 shows that the RLS performance does not improve when the filter order increases. Hence, a careful selection of the filter order is needed for optimal performance.
- LMS usually provides better tracking behavior than exponentially weighted RLS for linearly chirped sinusoids in additive white Gaussian noise (AWGN). This behavior is because LMS is model independent, whereas RLS must employ a model of the data correlation matrix (such as an exponential weighting), which may not match the characteristics of the input signals.[10]
- Correction term in LMS is based on the instantaneous sample values in the filter and the error signal, but in the RLS algorithm all PAST information is used.
- The LMS requires approximately 20M iterations (where M is the number of taps of the filter) to converge in mean square and the RLS converges in less than 2M iterations. This figure of merit is subjective to the step size of the LMS algorithm.[5]

- The RLS does not contain any approximations in its derivation, unlike the LMS.

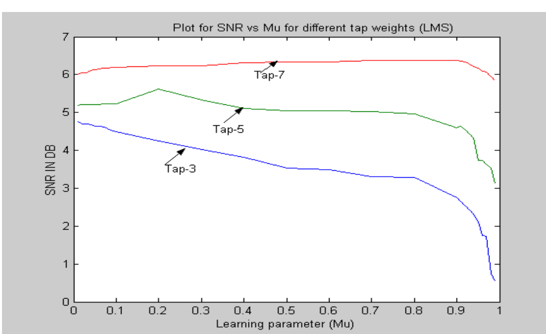
The RLS is more complicate to implement (especially for a large number of taps)

6. Results

Table 6(a) and 6(d) shows the performance of LMS and RLS algorithm in terms of SNR for different tap weights. Fig 6(b) 6(c) 6(e)& 6(f) shows the plots for SNR vs Mu for different taps (LMS) and SNR Verses Lamda for different tap weights (RLS Algorithm).Fig 6(g) &6(h) shows reconstructed speech signal using LMS algorithm and RLS algorithm for tap 3.Fig 6(i) &6(j) shows

S.NO	Mu value in LMS algorithm	SNR in dB for TAP-3	SNR in dB for TAP-5	SNR in dB for TAP-7
1	0.01	**4.7591**	5.1931	6.0011
2	0.02	4.7006	5.2017	6.0400
3	0.03	4.6892	5.2067	6.0413
4	0.04	4.6853	5.2024	6.1132
5	0.05	4.6390	5.2125	6.1266
6	0.06	4.6334	5.2186	6.1434
7	0.07	4.6188	5.2211	6.1744
8	0.08	4.5768	5.2252	6.1759
9	0.09	4.5147	5.2332	6.1851
10	0.1	4.4986	5.2380	6.2013
11	0.2	4.2499	**5.6145**	6.2400
12	0.3	4.0124	5.3250	6.2413
13	0.4	3.8230	5.1005	6.3134
14	0.5	3.5305	5.0470	6.3266
15	0.6	3.4865	5.0375	6.3434
16	0.7	3.3011	5.0285	6.3744
17	0.8	3.2931	4.9608	6.3759
18	0.9	2.7446	4.6003	**6.3851**
19	0.91	2.6445	4.6445	6.3512
20	0.92	2.5344	4.5344	6.3386
21	0.93	2.4248	4.4297	6.2728
22	0.94	2.3244	4.3289	6.2196
23	0.95	2.1232	3.7446	6.1751
24	0.96	1.7435	3.7311	6.0972
25	0.97	1.7311	3.6423	6.0595
26	0.98	0.7447	3.5356	5.9517
27	0.99	0.5348	3.1267	5.8500

Table 6(a): SNR Verses Mu for different tap weights (LMS Algorithm)



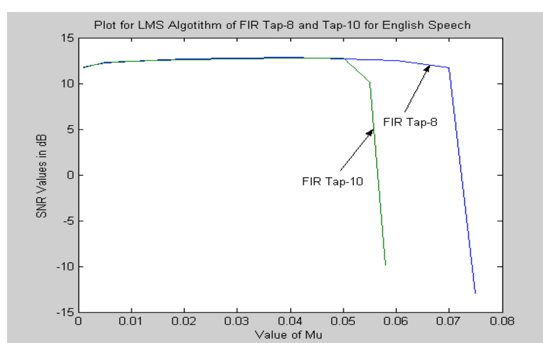


Fig 6(b): Plot for SNR vs Mu for different taps (LMS)   Fig 6(c): Plot for LMS algorithm for tap 8 &10

S.NO	Lamda value in RLS algorithm	SNR in dB for TAP-3	SNR in dB for TAP-5	SNR in dB for TAP-7
1	0.1	41.0756	52.7477	49.1887
2	0.2	44.7020	52.8656	50.7799
3	0.3	45.7861	54.1902	51.7287
4	0.4	47.9930	54.3162	53.0838
5	0.5	48.9430	55.0163	53.1789
6	0.6	50.8519	57.7852	53.3113
7	0.7	53.9461	57.9585	53.6879
8	0.8	54.1481	58.1815	54.1667
9	0.9	56.5446	58.3939	55.5531
10	0.91	56.4716	58.4172	56.5418
11	0.92	**62.0054**	63.1453	59.2906
12	0.93	59.1803	65.2394	65.4242
13	0.94	58.3524	**67.4584**	**75.0479**
14	0.95	57.5812	61.5060	69.6198
15	0.96	52.9756	57.4053	57.2496
16	0.97	51.1639	54.0939	55.3010
17	0.98	46.1694	51.9899	54.1178
18	0.99	45.1539	50.8761	50.7958

Table 6(d): SNR Verses Lamda for different tap weights (RLS Algorithm)

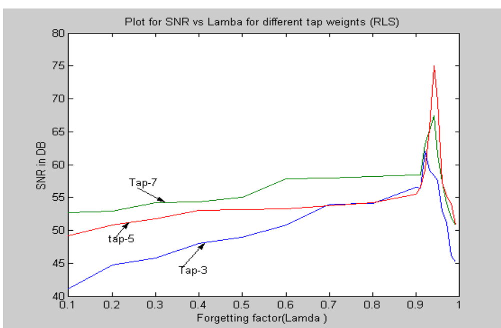


Fig 6(e): Plot for SNR vs. Lamda for different taps (RLS)

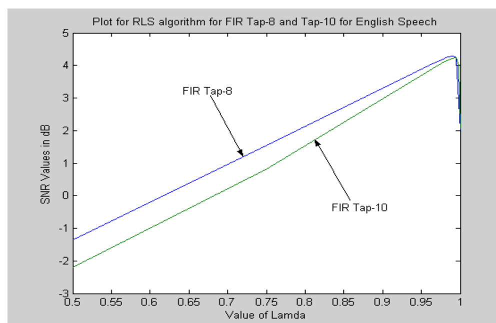


Fig 6(f): Plot for RLS algorithm for FIR tap 8&10 for speech signal

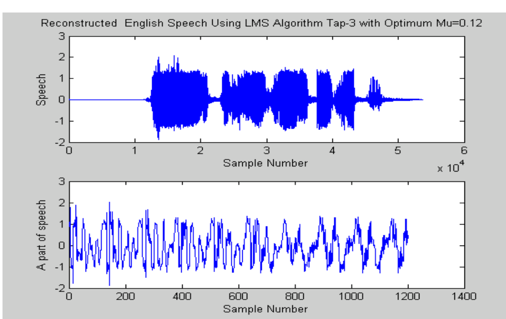


Fig 6(g): Reconstructed speech signal using LMS algorithm for tap-3

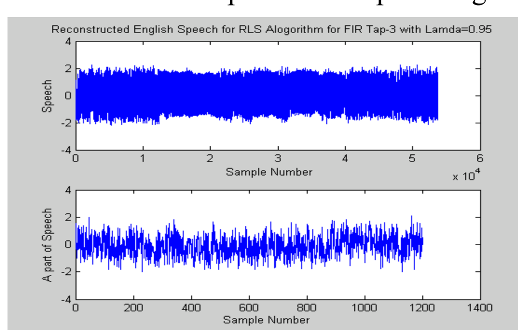


Fig 6(h): Reconstructed speech signal using RLS algorithm for tap 3

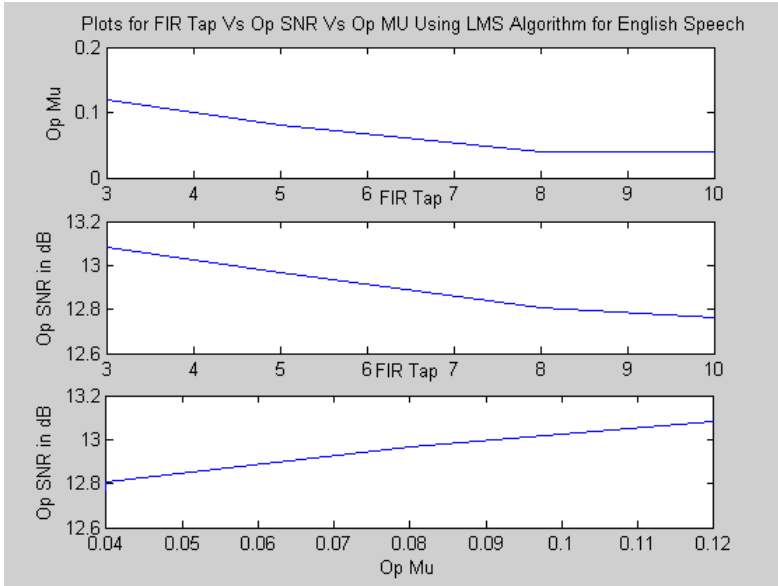


Fig 6(i): Plot for FIR tap Vs Optimum SNR Vs Optimum mu for speech signal using LMS

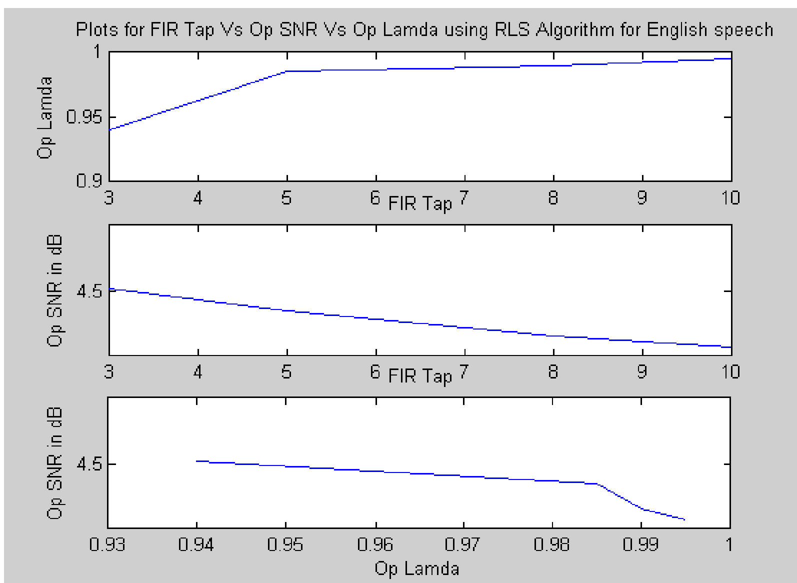


Fig 6(j): Plot for FIR tap Vs Optimum SNR Vs lamda using RLS for speech signal

7. Conclusion

Adaptive filtering is a hot topic of research in this era of information explosion. From the results it is observed that reconstructed speech signal SNR-LMS= 6.3851 dB (Mu =0.9) and SNR-RLS= 75.0479 dB (Lamda = 0.94) for TAP-7.From the tables 6(a) &6(d) it is seen that SNR increases as the order of the filter increases and SNR is high for RLS than LMS algorithm. For the lower order FIR adaptive filter, RLS algorithm produce highest SNR and it is superior to LMS in its performance. But LMS is converging faster that RLS for the Finite Impulse response (FIR) filter Taps. [8] Optimum Mu (LMS) and Lamda (RLS) values have been obtained by fixing the FIR Tap weights .These optimum Mu values for LMS are 0.9 (FIR Tap-7) to 0.01 (FIR Tap-3) and Lamda values for RLS are 0.92(FIR Tap-3) to 0.94(FIR Tap-7) on speech signal.[11] These results are obtained with primitive adaptive algorithms like LMS and RLS.

References

[1] Adaptive Filter Theory by Simon Haykin; 3rd edition, Pearson Education Asia, LPE.

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

[3] Lecture Notes by Professor Md. Zaki Ahmed, Plymouth University, U.K

[4] "Adaptive Filtering with Averaging in Noise Cancellation for Voice and Speech Recognition" by Georgi Iliev and Nikola Kasabov, University of Otago, 1998

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

[6] 'Adaptive noise Cancellation' paper by Rajvignesh, C. Thogulua, University of Florida, 2000

[7] White and Color Noise Cancellation of Speech Signal by Adaptive Filtering and Soft Computing Algorithms Ersoy Kelebekler[1] and Melih İnal[1] Kocaeli University, Technical Education Faculty, Electronics and Computer Department, Umuttepe Campus, 41380 İzmit, Kocaeli, Turkey. Advances in Artificial intelligence publication Springer

[8] Comparison of convergence characteristics of adaptive IIR and FIR filters for active noise control in a duct by R.W. Jones , B.L. Olsen science direct journal applied acoustics volume 68 issue 7 July 2007 Pages 729-738.

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

[10] W. Harrison, J. Lim, E. Singer, "A new application of adaptive noise cancellation," IEEE Trans. Acoust., Speech, Signal Processing, vol. 34, pp. 21-27, Jan. 1986.

[11] S. Ikeda, A. Sugiyama, "An adaptive noise canceller with low signal distortion for speech codecs," IEEE Trans. Signal Processing, vol. 47, pp. 665-674, Mar. 1999.