Research Paper
Impact Factor: 3.567
Peer Reviewed Journal

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

PERFORMANCE EVALUATION OF ADAPTIVE SIGNAL PROCESSING ALGORITHMS

Sunny T.D
Assistant Professor, Department of Electronics and Communication Engineering, Government Engineering College, Kerala.

Abstract
Adaptive filter, by their very nature, are self-designing systems, which can adjust themselves to different environments. As a result, adaptive filters find applications in such diverse fields as control, communications, radar and sonar signal processing, interference cancellation, active noise control, biomedical engineering, etc. The availability of a rich literature that reports the development of adaptive algorithms motivated me to take up a detailed evaluation of some algorithms in order to get a proper insight into the possible choices. This work includes detailed evaluation of LMS (Least Mean Square), NLMS (Normalized LMS) and RLS (Recursive Least Square) algorithms. Here the convergence behavior and the influence of various parameters in the performance of these algorithms were evaluated. The results of the evaluation of different algorithms are compared, thereby bringing out the scope of each algorithm and the direction in choosing the right approach for a given problem.

Key Words: Adaptive Filter, Mean Square Error, LMS, NLMS, RLS, Performance Evaluation, Eigen Value Spread.

1. Introduction

Adaptive filters find applications in different fields such as control, communications, radar and sonar signal processing, interference cancellation, active noise control bio-medical engineering etc, The common feature of these applications which brings them under the same basic formulation adaptive signal processing is that they all involve a process of filtering some input signal to match a desired response. The filter parameters are updated by making a set of measurements of the underlying signals and applying that set to the adaptive filtering algorithm such that the difference between the filter output and the desired response ie the, error is minimized in either in a statistical or a deterministic sense. By evaluating the performance of varies algorithms it is possible to find out the parameters that determines the performance of the algorithm. So this work will help to analyze the merits and demerits of the present algorithms and be able to choose the right algorithm for the problem in hand.

2. Literature Survey

Many everyday problems encountered in communications and signal processing involves removing noise and distortion due to physical process that are time varying or unknown or possibly both. The common element in most of the applications of adaptive signal processing is that some element of the problem is unknown and must therefore be learned, or some component of the system is changing in an unknown manner and therefore must be tracked. Quite frequently both of these problems are resident in the application of adaptive signal processing. [7]

Adaptive filtering Algorithms are useful when the input signal statistics are unknown and/or changing with time. In a stationary environment the signal statistics are unknown but fixed. The adaptive algorithm gradually learns the required input statistics. After convergence in steady state, the filter weights jitter around the fixed desired values. The algorithm performance is then described simultaneously by the speed of convergence and by the weight fluctuations in the steady state. [2]

In a truly stationary environment, the algorithm learning phase can be increased, without bound, and the weight jitter made arbitrarily small. In reality, however, the environment is only stationary over a finite time interval, changing every so often to different state. The convergence properties of the algorithm describe the ability of the algorithm to adjust to the new desired steady state behavior. In the non-stationary case, the algorithm must continuously follow (track) the time varying statistics of the input.[6] Hence the adaptive algorithm must pass from a transient mode of operation (convergence) to a steady state. Hence we see that convergence speed and tracking capabilities are different properties of an algorithm: convergence is transient phenomenon and tracking is steady state phenomenon [5].

3. Performance Measurements

Depending up on the time required to meet the final target of the adaptation process and the complexity/resources that are available to carry out the adaptation, we can have a variety of adaptation algorithms and filter structures. This choice may become critical in the case of real-time processing, where performance, implementation complexity, and cost often trade off under rigid constraints.

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

The performance of an algorithm can be measured by a number of factors quantifying (a) how close the resulting solution (estimation) is to the theoretically expected setup and (b)in how many iterations, with respect to the number of measurement samples, this solution is obtained. In a stationary environment the algorithm is expected to converge to the fixed optimal values starting from arbitrary initial conditions in a general set. The number of measurement samples which the algorithm need to converge, that is, to learn the optimal parameters, is related to the 'convergence rate', which constitutes one of the basic performance measure of an algorithm. Another performance measure is 'tracking'. In a non-stationary environment the algorithm and the underlying model are expected to accommodate the time variation of the pertinent statistics. Tracking is related to the ability of the algorithm to track this time evolution of the physical system [4].

Another issue, which determines the choice of an algorithm for a particular application is 'computational complexity', that is the number of arithmetic operations required by the algorithm. For real-time applications, this is a highly critical issue since the computations required by the algorithm to update the parameters must be completed before the next set of measurement samples is received. Other performance parameters are round of error accumulation and numerical stability. Different algorithms exhibit different degrees of robustness with respect to round off error accumulation. Numerical stability manifests itself by the numerical explosion of certain algorithmic parameters, which become unbounded. The use of shorter word lengths accelerates the occurrence of this explosion, which is otherwise intrinsic to the algorithm's structure. So it may require selecting numerically stable algorithmic structure at the possible cost of higher complexity [1].

4. Implementation

All the algorithms covered in the present work take the output error of the filter, correlate that with the samples of filter input in some way, and use the result in a recursive equation to adjust the filter coefficients iteratively. The mathematical tool, MATLB is used for evaluating the performance of the algorithms.

4.1 Performance Evaluation of LMS Algorithm

Performance evaluation of LMS algorithm is carried out in the context of a system identification problem depicted in figure1. The input signal, x(k), is a random white noise . Here the unknown system that generates the desired signal d{k), is taken as an FIR filter of order 15. Frequency range of the filter is taken as 350 to 420 Hz. The system function $W(z)=w_0+w_1z^{-1}+$ ….. is used to identify the system. The input signal x(k) is applied to both the adaptive filter and to the unknown system. The filter output y(k) is compared with desired signal d(k). The error signal e(k) is the difference between d(k) and y(k) at time k. This error signal is used to update the adaptive filter weights. This process is repeated till the error is zero or the minimum. At this instant d(k) and y(k) are equal. So the adaptive filter is adapted to the unknown system.

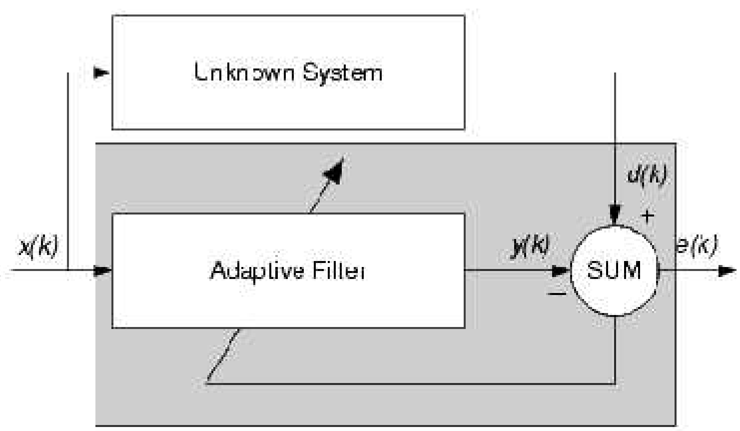

Figure 1: Structure of an Adaptive Filter to Identify an Unknown System [6]

For validating the adaptive filter, the adaptive filter just modeled is applied to the input sequence. MSE curve is plotted.

Parameters Taken

Filter order = 15
Number of Samples = 1000.
Step size parameter (μ)= 0.05
Weight vector is initialized to zeros.

Iterative Steps

The various steps of LMS algorithm is given below.

1. Filtering

$$y(n) = w^T(n)x(n) \qquad \text{---1}$$

2. Error Estimation

$$e(n)=d(n)-y(n) \qquad \text{---2}$$

3. Tap-Weight Vector Updating Equation

$$w(n+1) = w(n) + 2\mu e(n)x(n) \qquad \text{---3}$$

Input

Tap weight vector, w(n),

Input vector, x(n)
and desired output, d(n).

Output

Filter output, y(n),
Tap – weight vector update, w (n+1).

The above steps 1, 2 and 3 constitute one iteration of the LMS algorithm. These steps were repeated till all the samples of the input sequence were processed.

Table1. Variables used in LMS Algorithm

Variable	Description
n	The current algorithm iteration
x(*n*)	The buffered input samples at step *n*
w(n)	The vector of filter-tap estimates at step *n*
y(*n*)	The filter output at step *n*
e(*n*)	The estimation error at step *n*
d(*n*)	The desired response at step *n*
μ	The adaptation step size

[2], [3]

4.1.1 Effect of Step Size in Convergence

The effect of step size parameter in the performance of algorithm is evaluated by running the above system identification problem for different values of step size parameter. Step size selected is 0.05, 0.01, and 0.1. The algorithm is evaluated both in training as well as in validation phase. Simulated results of this experiment is given in figure

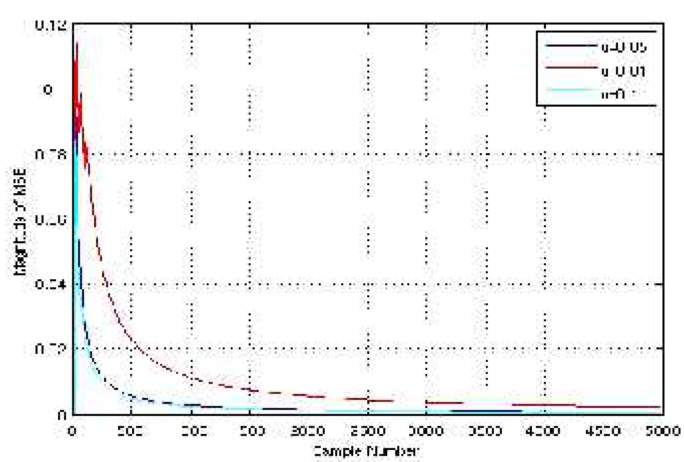

Figure 2: MSE Curve with Deferent Values of μ

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

4.1.2 Effect of Eigen Value Spread of Input Sequence

The effect of Eigen value spread of input sequence in the performance of the algorithm is evaluated by considering the following system identification problem depicted in figure 3The input signal x(n) is generated by passing a white noise signal through a coloring filter with the system function ,

$H(z) = \mathrm{sqrt}(1-\alpha^2)/(1-\alpha z^{-1})$, Where α controls the Eigen value spread of the correlation matrix R of the input sequence x(n).

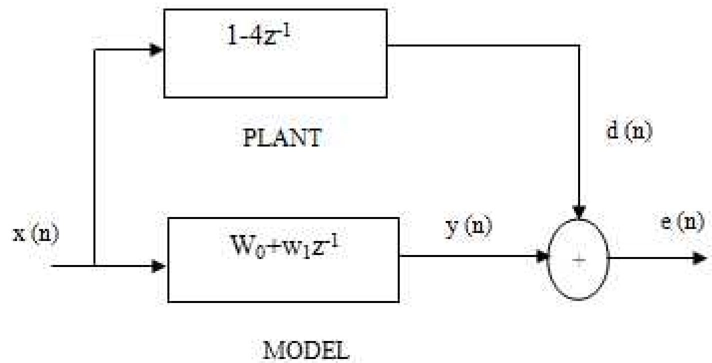

Figure 3: Model for System Identification Problem

The plant is a two-tap FIR system with the system function $p(z) = 1-4z^{-1}$. This plant will generate the desired signal. An adaptive filter with the system function $w(z) = w_0 + w_1 z^{-1}$ is used to identify the plant. Here, the LMS algorithm is used to find the optimum values of the tap weights w_0 and w_1. We want to see how much the tap weights w_0 and w_1 converge to the plant coefficients, 1 and –4 respectively for different values of α.

The experiment is conducted for the following Eigen value spread parameters. α= 0, 0.5 and 0.9. The result is given in figure 4.

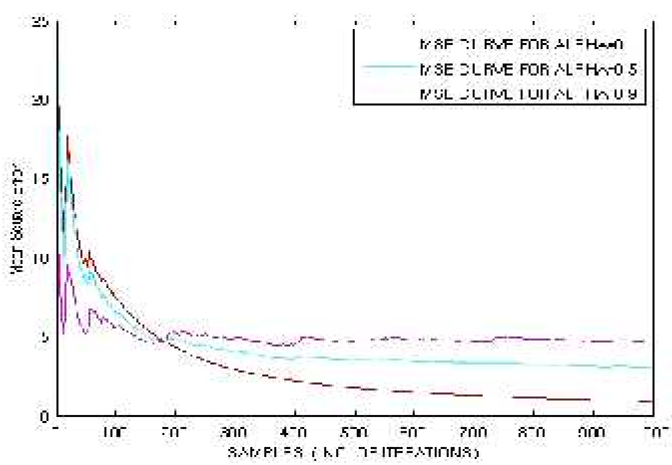

Figure 4: MSE Curves for 3 different Eigen Values, Demonstrates Effect of Eigen Value Spread

4.2 Performance Evaluation of NLMS Algorithm

Performance evaluation of NLMS algorithm is carried out by considering the system identification problem depicted in figure 1.

4.2. 1. One Iteration in NLMS Algorithm

1. Filtering

$$y(n) = w^T(n)x(n)$$

2. Error Estimation

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

3. Tap-Weight Vector Updating Equation

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 1/(2\mathbf{x}^T n)\, \mathbf{x}(n))\, e(n)\mathbf{x}(n)$$

Input

Tap weight vector, w(n),
Input vector, x(n)
and desired output, d(n).

Output

Filter output, y(n),
Tap – weight vector update, w(n+1).

Table 2: Variables used in NLMS Algorithm

Variable	Description
N	The current algorithm iteration
x(*n*)	The buffered input samples at step *n*
w(n)	The vector of filter-tap estimates at step *n*
y(*n*)	The filtered output at step *n*
e(*n*)	The estimation error at step *n*
d(*n*)	The desired response at step *n*
μ(n)	The adaptation step size

[2], [3]

The result of performance evaluation of the algorithm is presented figure 5.

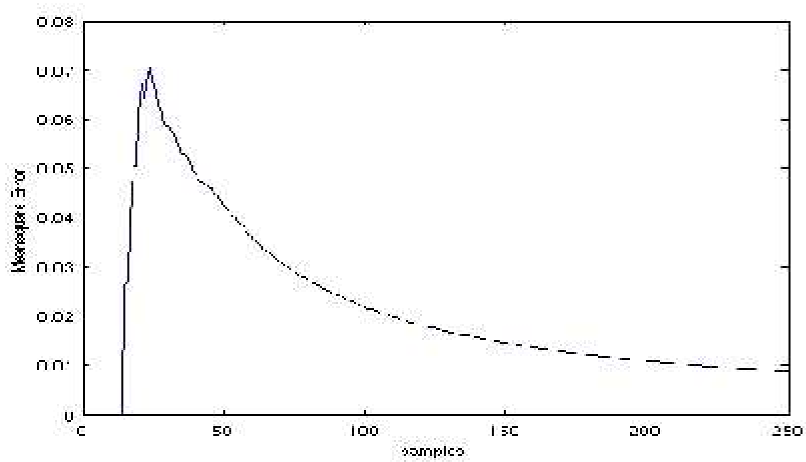

Figure 5: Shows MSE Curve for the System Identification Problem

4.3 Performance Evaluation of RLS Algorithm

The RLS Adaptive Filter recursively computes the least squares estimate (RLS) of the FIR filter coefficients. Performance of RLS algorithm is evaluated by considering the system identification problem depicted in figure 1
The parameters used in algorithm are,

Input

Tap weight vector estimate, $\hat{\mathbf{w}}(n-1)$,
Input vector, $\mathbf{x}(n)$, desired output, $\mathbf{d}(n)$
And the matrix $\boldsymbol{\psi}^{-1}(n-1)$

Output

Filter output, $\hat{y}_{n-1}(n)$,
Tap weight vector update $\hat{\mathbf{w}}(n)$,
And the updated matrix $\boldsymbol{\psi}^{-1}(n)$

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

4.3.1 One Iteration of RLS Algorithm

1. Computation of the Gain Vector

$\mathbf{u}(n) = \mathbf{p}(n\text{-}1)\mathbf{x}(n)$

$\mathbf{k}(n) = \mathbf{u}(n)/(\quad + \mathbf{x}^T(n)\mathbf{u}(n))$

2. Filtering

$\hat{y}_{n-1}(n) = \hat{\mathbf{w}}^T(n\text{-}1)\ \mathbf{x}(n)$

3. Error Estimation

$\hat{e}_{n-1}(n)=d(n)- \hat{y}_{n-1}(n),$

4. Tap-Weight Vector Updating Step

$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n\text{-}1)+\mathbf{k}(n)\ \hat{e}_{n-1}(n)$

5. p(n) update

$\mathbf{p}(n)= \ ^{-1}(\mathbf{p}(n\text{-}1)\text{-}\mathbf{k}(n)[\mathbf{x}^T(n)\ \mathbf{p}(n\text{-}1)])$

The variables used in RLS algorithm are listed in table 3.

Table 3: List of Variables used in RLS Algorithm

Variable	Description
N	The current algorithm iteration
x(*n*)	The buffered input samples at step *n*
p(*n*)	The inverse correlation matrix at step *n*
ψ (n)	Correlation matrix of input sequence, x(n)
Θ (n)	Cross correlation matrix of x(n) and d(n)
k(*n*)	The gain vector at step *n*
$\hat{\mathbf{w}}$(n)	The vector of filter-tap estimates at step *n*
y(*n*)	The filtered output at step *n*
$\hat{e}_{n-1}$	The estimation error at step *n*
d(*n*)	The desired response at step *n*
λ	The exponential memory weighting factor

The adaptation process is same as the LMS algorithm. The difference is in the way in which the filter weights are updated and in how the signals are taken. The input signal is a random signal. The desired signal is generated by using an FIR filter with the following specifications. Frequency range 0.2 to 0.4 (Normalized with sampling frequency), System order =15. Adaptive filter weights are initialized to **0** (vector). Inverse correlation matrix is initialized to 0.001xI. Forgetting factor is selected as 0.9.

4.3. 2 Convergence Behavior of RLS Algorithm

Convergence behavior of RLS algorithm is found out by considering the system-modeling problem depicted in figure 1. The common input, x(n), to the plant, $w_0(z)$, and adaptive filter, w(z), is obtained by passing a unit variance white Gaussian sequence, v(n), through a filter with the system function H(z).

$$W_0(z) = \Sigma z^{-1} – \Sigma z^{-1}$$

The length of the adaptive filter, N, is chosen equal to the length of W_0 (z) ie; N=15 Here we taken two different inputs with different Eigen value spread. They are characterized by

Input 1

$$H(z) = H_1(z) = 0.35 + z^{-1}\text{-}0.35z^{-2}$$

And

Input 2

$$H(z) = H_2(z) = 0.35 + z^{-1}+0.35z^{-2}$$

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

Here $H_1(z)$ results in an input, x(n), whose corresponding correlation matrix has an Eigen value spread of 1.45. This is close to white input. On the contrary, $H_2(z)$ results in highly colored input with an associated Eigen spread of 28.7.

Parameters Taken

Filter order = 15
Number of Samples, N = 10000.
Forgetting factor λ=0.9
δ = 0.001 and 0.1 (initial value of inverse correlation matrix).

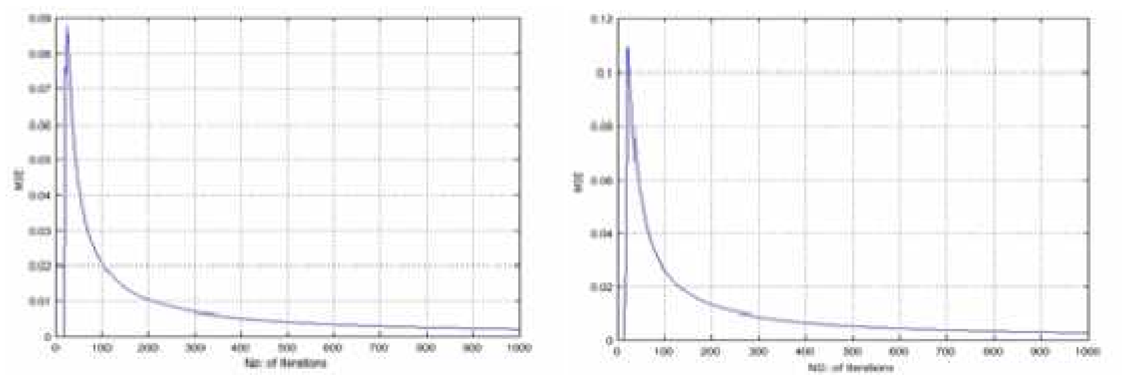

Figure6. MSE curve of system identification problem depicted in figure 1for two different values of , 0.001 and 0.1

Figure6 shows the MSE curve for two different values of . First result with 0.001 and second figure with =0. It is clear that for smaller value of initial error is less. However the steady state performance is same for both cases. Thus it is concluded that steady state error is independent of initial value of the inverse correlation matrix.

5. Results and Conclusion

This section gives the results of comparison of the performance of LMS NLMS and RLS algorithms in the context of system identification problem depicted figure 1.

Figure 7 shows a set of filter outputs that compares the convergence behavior of LMS, NLMS and RLS algorithm. It is clear from the results that adaptation is quick in the case of RLS algorithm. Adaptation of LMS algorithm is very slow as compared to NLMS and RLS algorithm.

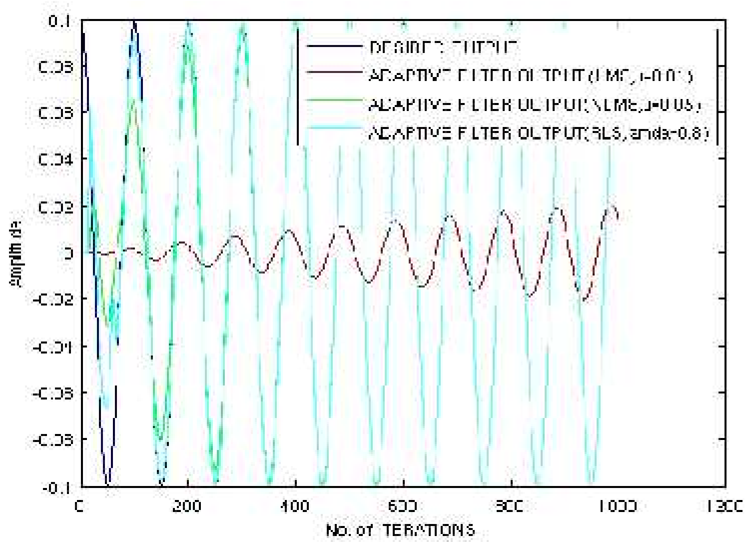

Figure 7: Shows the Filter Outputs of LMS, NLMS and RLS Algorithms for the System Identification Problem Described in Figure1

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

Figure 8: Shows the MSE Curves of LMS, NLMS and RLS Algorithm for the System Identification Problem Described in Figure 1.

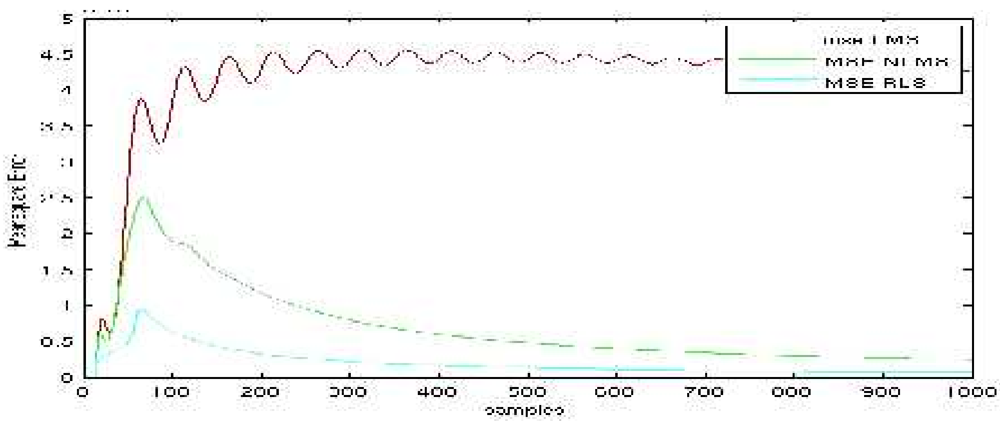

Figure 8: MSE Curves Comparing the Convergence Behavior of LMS, NLMS and RLS Algorithms

From the results obtained in this section we are able to conclude that the RLS algorithm converges quickly than LMS and NLMS algorithms. RLS algorithm is free from Eigen value spread of input sequence. The convergence performance of NLMS algorithm is better than conventional LMS algorithm. Its step size parameter varies with signal statistics. Hence it can adapt the input more quickly than LMS algorithm. The major limitation of LMS algorithm is its slow convergence especially when the input process is highly colored. Signal whitening by using some power normalization mechanism, prior to applying the adaptive algorithm, is applied to enhance the convergence performance of the LMS algorithm. One variant of the LMS algorithm, the NLMS algorithm, shows better convergence behavior. This work bringing out the scope of each algorithm and the direction in choosing the right approach for a given problem.

References

1. Nicholas Kalouptsidis and Sergios Theodoridis “Adaptive System Identification and Signal Processing Algorithms” By, PHI International series.
2. B. Farhang – Boroujeny, “Adaptive Filters Theory and Applications” John Wiely & Sons, Chichester, New York,
3. Bernard Widrow ad Samuel D. Sterans “Adaptive Signal Processing” PHI Inc. New York.
4. Rulph Chassaing “Digital Signal Processing and Applications” John Wiely & Sons, Inc. , 2005
5. Amit Kumar Kohli and D.K. Mehra, “Trcking of Time-Varying Channels Using Two-step LMS-type adaptive algorithm” IEEE transactions on Signal processing. Vol. 54.No.7, July 2006.
6. B.Farhang – Boroujeny and Saeed Gazor , “Performance of LMS-Based Adaptive Filters in Tracking a Time varying Plant”
7. S. Thomas Alexander “Adaptive Signal Processing theory and Applications” Springer varlag Publishes.