Demonstration of adaptive algorithms Using Mat lab

Jagadeesh Thati
Department of ECE, Tirumala Engineering College, Guntur
E-mail: jagadeeshthati@gmail.com

Bindu Madhavi Gasi
Technical Associate in Tech Mahendra
E-mail: BG0079605@techmahindra.com

Abstract— This paper analyzes the performance characteristics of different adaptive filter algorithms in stationary and non-stationary environment. These algorithms are based on weight updated equation. After estimating the coefficients in the model during the training period, the desired signal can be estimated using the weight updated equation of the LMS, LLMS, NLMS and RLS. Finally we demonstrate through simulation that these results are useful in predicting the adaptive filter performance [1].

Keywords-—Adaptive filter, Weight updated equation, stationary environment, Non-stationary environment.

I. INTRODUCTION

Adaptive filter is based on set of rules or algorithm and explain how the correction ΔW_n is to be formed. The correction of sequence (is used to) should decrease the mean – square error. When we consider the design of Finite Impulse Response(FIR) non-recursive adaptive filters, FIR filters are used in adaptive filtering applications that ranges from adaptive equalizers in digital communication systems to adaptive noise control systems. In FIR Wiener filter we assume that x(n) and d(n) are jointly Wide sense stationary(WSS), the solution to these equations depends on n [2].

$$R_x(n)\, W_n = r_{dx}(n)$$

Mainly adaptive algorithms are depend on weight updations In this paper we are going to discuss adaptive and their MATLAB implementations.

II. ADAPTIVE ALGORITMS

The LMS algorithm is a linear adaptive filter algorithm which uses the concept of gradient algorithms, the weight updated equation

$$W_{n+1}=W_n + \mu E\{e(n)^*X(n)\}$$

Where e (n) = d (n) y(n)

W (n) is the weight vector which is adjusted adaptively in order to achieve the convergence to wiener solution. i.e.

$$W_n = R_x^{-1}\, r_{dx}$$

Step size ranges from $0< \mu <2 / \lambda$.

λ is the maximum Eigen value of autocorrelation matrix of X.

While observing the LMS algorithm for stationary process, the weight vector initialized to $w_0 = 0$, the sequence of weights moves toward the solution to the Wiener Hopf equations, $W = R_x^{-1} r_{dx}$. If the Weight vector is initialized With the solution to the Wiener-Hopf equation and gradient at this point is zero, W_n moves randomly within a neighborhood of this solution.

There are two main drawbacks of LMS algorithm one of it is slow rate of convergence and another is sensitivity to Eigen value spread of the correlation matrix of the input vector. The LMS algorithm has a disadvantage in selection of μ.

When the input process to an adaptive filter has an autocorrelation matrix with zero Eigen values, the LMS Algorithm has one or more modes that are undamped. In this case it is very much necessary to stabilize the LMS adaptive filter by forcing the modes to zero else the undamped modes become unstable. So, we introduce a leakage coefficient γ into the LMS algorithm.

In leaky LMS Algorithm, introduce a leakage coefficient γ into the LMS algorithm

$$w_{n+1}= (1-\mu\gamma)w_n+ \mu\, e(n)x^*(n)$$

Where $0 < \gamma << 1$

The effect of this leaky coefficient will force the filter coefficients to zero if either the error e(n) or the input x(n) becomes zero and also to force to undamped modes of the system to zero. The step size γ in the convergence in the mean becomes $0<\mu<2/ (\lambda max+\gamma)$.

When the input data vector X(n) in the algorithm tends to increase, like non stationary input signal. This force a stable algorithm to diverge if μ is not sufficiently small. Therefore we develop a new algorithm called Normalized LMS algorithm in which μ is adjusted based on the input signal power and as a result changes its value based on the present situation. Disadvantage of LMS is overcome by changing μ as

$$0< \mu < \frac{2}{||x(n)||^2}$$

This is further modified by using normalized step size β.

$$0 < \beta < 2$$

The weight updated equation of the normalized LMS algorithm is

$$W_{n+1}=W_n + \beta \frac{X^*(n)}{\varepsilon+||X(n)||^2} e(n)\}\}$$

Above mentioned algorithms do not provide a convergence or sufficiently small mean square error, and require autocorrelation of the input process and also the cross correlation between the input and the desired output. The functioning of the Recursive Least Square algorithm requires the present the adapted values, with least mean square error we minimize a squared error that depends explicitly on the specific values of x (n) and d (n).

$$\varepsilon(n)=\sum_{i=0}^{n} |e(n)|^2$$

The main difference between LMS and RLS is that, in LMS the correction is done by updating the old Estimate vector based on the instantaneous sample value of the input vector and e(n) whereas in RLS the correction utilizes past information, and other issue that is considered is that the approximations as a result the iterations approach infinity the least square estimate of the coefficient vector approaches the optimum Wiener value thus the error signal is minimized.

The weight updated equation for a complete recursion

$$W_n = W_{n-1} + e\,(n)\,k\,(n)$$

Where

$e(n) = d\,(n) - W_{n-1}^{T} X(n)$,is the true estimate

K (n) is the gain vector.

In Adaptive noise canceller, the primary input containing the corrupted signal and a reference input containing noise signal which is correlated with the primary noise. The reference input signal is sent to the adaptive filter and is subtracted from the primary input in order to obtain the desire estimated signal.

In Adaptive Line Enhancer the primary signal is a delayed replica. The autocorrelation coefficients of noise decay much faster than that of the primary and the delayed primary signals will be correlated but the noise components will not correlate.

Here in this adaptive line enhancer there is no need of the reference signal since the given input is delayed and is fed to the adaptive filter as noise signal.

III. SIMULATION

Using GUI, in which we designed four check boxes labeled LMS, LLMS, NLMS and RLS.

Two edit boxes are designed for LMS named 'mu' which represent the step size μ and the other box 'ORDER' in which the order of the filter can be given.

For LLMS, we have three edit boxes. The first, second boxes represents the step size and the order of the filter. Third box named 'gamma' in which leaky coefficients can be given.

For NLMS ,we designed two edit boxes named 'beta' and 'order' which represents normalized step size and the order of the filter respectively.

RLS, there are two edit boxes, lambda i.e., exponential weight factor and order respectively.

There are three popup menus. First popup menu is the selection of the input signals i.e., stationary and non stationary input signals. Second popup menu is the selection of the adaptive techniques i.e., adaptive noise canceller and adaptive line enhancer, third popup menu is the selection of the output i.e., error, learning curve, convergence of coefficients.

The run button is used to simulate the code. Clear button is used in order to clear the previous data and axis.

There are three graph represents original signal, optional output, estimate signal.

LMS is selected to work, the step size μ is set to 0.05, and order is set to 6 and the length of 1000 samples as shown in Fig.1. A non stationary signal is selected for adaptive noise canceller. Selection of the output i.e., error is taken as the optional output.

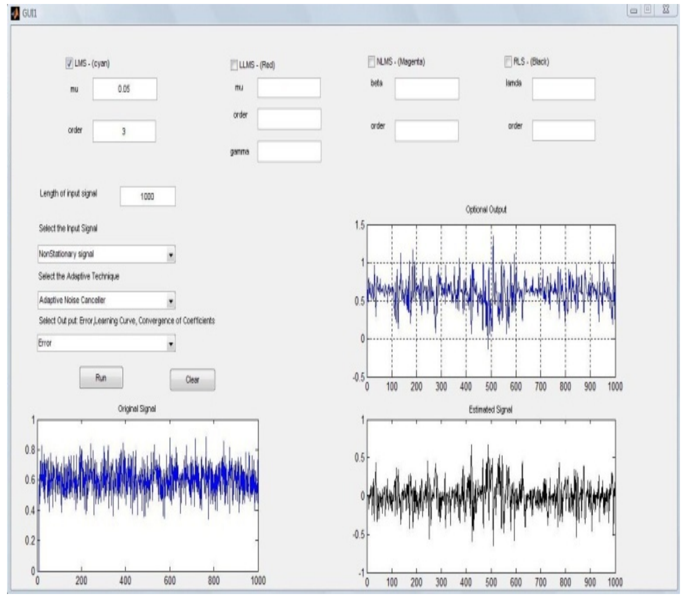

Fig.1.LMS Algorithm

LLMS is selected to work, the step size μ is set to 0.05, and order is set to 6 and the length of 1000 samples as shown in Fig.2. A non stationary signal is selected for adaptive noise canceller. Selection of the output i.e., error is taken as the optional output.

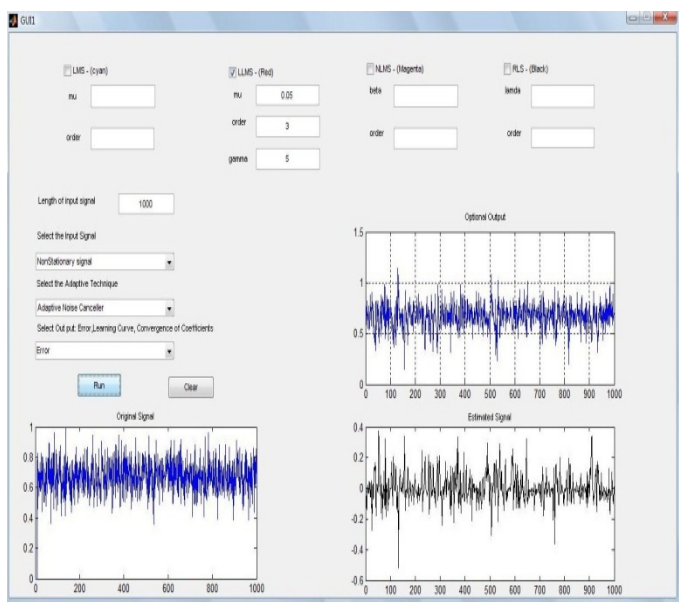

Fig.2.LLMS Algorithm

NLMS is selected to work, the step size β is set to 0.03, and order is set to 2 and the length of 1000 samples as shown in Fig.3. A stationary signal is selected for adaptive line enhancer. Selection of the output i.e., learning curve is taken as the optional output.

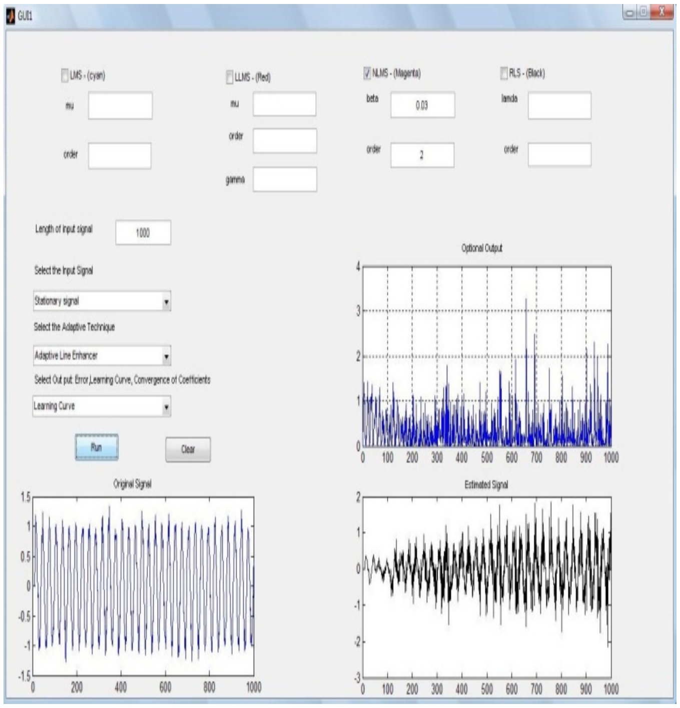

Fig.3.NLMS Algorithm

RLS is selected to work, the step size Lambda is set to 0.03, and order is set to 2 and the length of 1000 samples as shown in Fig.4. A stationary signal is selected for adaptive line enhancer. Selection of the output i.e., learning curve is taken as the optional output.

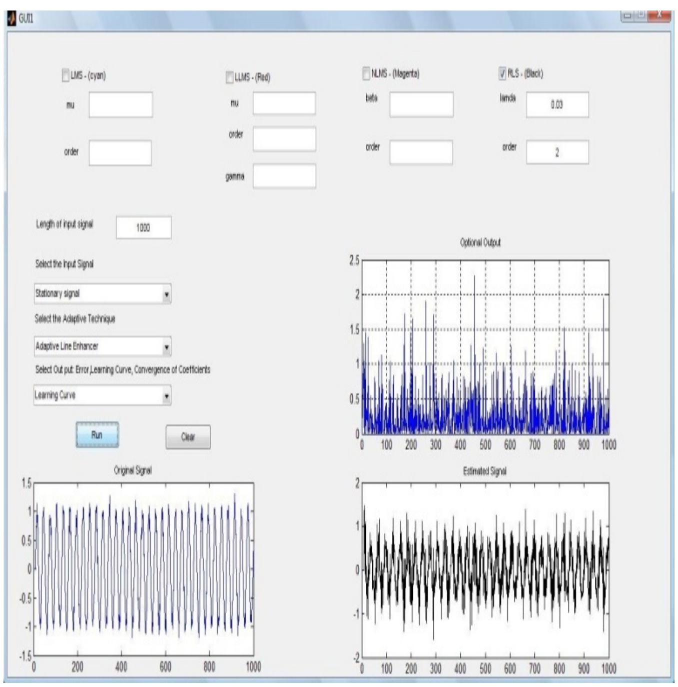

Fig.4.RLS Algorithm

CONCLUSION

In this paper we have presented a performance analysis of adaptive filter algorithms based on adaptive weight update equation presented in stationary and non-stationary environments.

REFERENCE

[1] Mohammad shams Esfand Abadi and John Hakon Husoy, "Mean – Square Performance of Adaptive Filter Algorithms in Non stationary Environments", International Journal of Signal Processing.

[2] Monson H. Hayes, "Statistical digital signal processing and modelling".

AUTHORS PROFILE

Jagadeesh Thati received the M.Tech. in DSP from JNTU Hyderabad and M.Sc. from the Department of Electrical Engineering, BTH, Sweden, in 2009 and 2010, respectively. From 2008 to 2010, he was a Researcher at Dasa Control systems, Vaxjo University and BTH. He is currently working as a Assistant professor, Department of ECE in Tirumala Engineering College Guntur. His current research interests include 2-D/3-D digital image processing, robotics, and contents security.

Bindu Madhavi Gasi received the B.Tech. in Electronics and communications engineering from JNTU Kakinada. She is currently working as a Assistant professor, Department of ECE in Tirumala Engineering College Guntur. His current research interests include 2- D/3-D digital image processing, Signal Processing, Software Engineering and contents security.