Performance Evaluation of Adaptive Equalization Techniques for Digital Mobile Radio

Minal Gonsalves and Srija Unnikrishnan

***Abstract*--The paper discusses the convergence analysis of a simple but widely used LMS Algorithm, Normalized LMS algorithm, Block LMS Algorithm and Frequency Domain Adaptive Algorithm for Adaptive Equalization. Also the analysis of Godard's 'Constant Modulus Algorithm', the popular blind equalization algorithm is done.**

Algorithms are implemented using MATLAB Software. Analysis of Mean Square Error for all the algorithms is done. The effects of eigen value spread, step-size parameter and Filter Length on the Convergence of LMS Algorithm are studied and results are obtained. Also results for NLMS showing its advantages over LMS are obtained. The comparative analysis of LMS with BLMS and FDAF algorithms is presented. Convergence results for CMA with and without modulation are obtained.

***Index Terms*--Adaptive Equalization, Block LMS, Blind Equalization, Constant Modulus Algorithm, FDAF, Least Mean Square Algorithm, Mean Square Error, Normalized LMS**

I. INTRODUCTION

INTERSYMBOL interference (ISI) caused by multipath in band limited time dispersive channels distorts the transmitted signal, causing bit errors at the receiver. ISI has been recognized as the major obstacle to high speed data transmission. In wireless world high speed data means utilizing available bandwidth more efficiently, leading to more bits in every available Hertz of spectrum. As more bits per Hertz are sent, the bits overlap and interfere with each other, and reduce the received signal quality unless special techniques, called 'Channel Equalization' are used. [1]

The term equalization can be used to describe any signal processing operation that minimizes ISI. In radio channels, a variety of adaptive equalizers can be used to cancel interference while providing diversity. Since the mobile fading channel is random and time varying, equalizers must track the time varying characteristics of the mobile channel, and thus are called adaptive equalizers. These equalizers thus use some adaptive algorithms for continuously changing its filter characteristics over time. These adaptive algorithms are classified into two distinct approaches 'Stochastic Gradient Approach' and 'Least-squares Estimation'. [2]

The conventional Least Mean Square (LMS) algorithm and Normalized LMS Algorithms are Stochastic Gradient Algorithms, and Recursive Least Squares (RLS) is an example of Least-squares Algorith[1]m.

Minal Gonsalves is working as Lecturer in St. Francis Institute of Technology (e-mail: minal_ag@rediffmail.com)
Srija Unnikrishnan is HOD of Department of Electronics Engineering, Fr.Conceicao Rodrigues College of Engineering. (e-mail: srija@frcrce.ac.in)

The paper is organized as follows: In section 2 theory about LMS, NLMS, BLMS and FDAF Adaptive Algorithms is discussed. Section 3 includes Matlab simulation results for performance parameters and comparison of above Algorithms. Section 4 presents theory and simulation results for Constant Modulus blind equalization algorithm with and without modulation.

II. ADAPTIVE ALGORITHMS

A. *LMS Algorithm*

The LMS algorithm is an important member of 'Stochastic gradient algorithms'. A significant feature of the LMS algorithm is its simplicity that has made it the standard against which other linear adaptive filtering algorithms are benchmarked.

It is a linear adaptive filtering algorithm, which consist of two basic processes:

1. A filtering process, which involves (a) computing the output of a linear filter in response to an input signal and (b) generating an estimation error by comparing this output with a desired response.

2. An adaptive process, which involves the automatic adjustment of the parameters of the filter in accordance with the estimation error. [2]

The combination of these two processes working together constitute a feedback loop, as shown in block diagram below, From the block diagram it can be seen that the filter output $y(n)$ is given as,

$$y(n) = w^T(n)\,x(n) \tag{1}$$

The error,

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

and adapting filter weight,

$$w(n+1) = w(n) + \mu\, x(n)\, e(n) \tag{3}$$

Where,

μ - Step Size parameter.

$x(n)$ - Input Signal vector of tap inputs.

$e(n)$ - Error signal.

$w(n)$ - Coefficients of filter at time n.

Equation (1) to (3) represents LMS Algorithm, where the optimization criterion is the LMS error. Hence the Mean Square Error is minimized at every time instant.

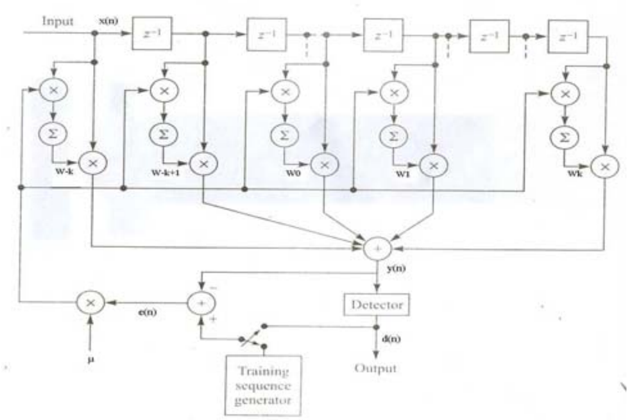

Fig. 1 Block Diagram of Adaptive LMS Equalizer [1]

B. Normalized LMS Algorithm (NLMS)

In standard LMS Algorithm, the adjustment applied to the tap-weight vector is directly proportional to the tap-input vector x(n).Therefore, when x(n) is large, the LMS filter suffers from a gradient noise amplification problem. This difficulty is overcome by '*Normalized LMS Algorithm'*. Here the adjustment applied to the tap-weight vector at iteration n+1 is *'Normalized'* with respect to the squared Euclidean norm of the tap-input vector x(n) at iteration n. [2]

Thus structurally, Normalized LMS filter is same as that of standard LMS filter. Only difference is the way weight vector is updated as shown below,

$$w(n+1) = w(n) + \frac{\mu}{\left\| x(n)^2 \right\|} x(n)\, e(n) \tag{4}$$

Comparing the recursions of Eq(3) and (4), setting $\mu(n) = \mu / \left\| x(n)^2 \right\|$ makes normalized LMS filter as an LMS filter with a time-varying step-size parameter. Thus normalized LMS algorithm exhibits a rate of convergence that is potentially faster than that of the standard LMS algorithm. [8]

The problem associated with NLMS is that, in overcoming the gradient noise amplification problem, it introduces a problem of its own, namely that when the tap-inputs are small, the step size value is divided by a small value for the squared norm $\left\| x(n)^2 \right\|$. To overcome this problem the recursion equation (4) is modified as,

$$w(n+1) = w(n) + \frac{\mu}{\left\| \delta + x(n)^2 \right\|} x(n) e(n) \tag{5}$$

Where, δ – Small initial value (δ >0)

C. Block LMS Algorithm (BLMS)

The recursion in eq(3) is done each time a new sample is received. Alternately if the weights are kept fixed until N (No. of filter taps) data samples are received and then incorporate this information to update the weights only once during this period, the weight update will be,

$$w(n+L) = w(n+L-1) + 2\,\mu\, x(n+L-1)\, e(n+L-1) \tag{6}$$

Where, $1 \le L \le N$ is an integer. [9]

By substitution eq (6) becomes,

$$w(n+L) = w(n) + 2\,\mu \sum_{m=0}^{L-1} x(n+m)\, e(n+m) \tag{7}$$

Eq(7) represents 'Block Recursion'. Note that the error terms in the summation all depend on the same weight vector $w(n)$ i.e.

$$e(n+m) = d(n+m) - y(n+m) \qquad m = 0,1,----,(L-1).$$

Where, $$y(n+m) = x^T(n+m)\, w(n) \tag{8}$$

Since eq(7) is a block update that operates at a lower sampling rate than that of the incoming data, a new time index k is defined where one increment corresponds to L increments of the original index n. We can substitute $n = KL$ where 'n' is an integer multiple of 'k'. By factoring the argument $(Kl + L)$ on the left-hand side of eq(7) as $(k+1)L$ and dropping the explicit dependence of the weight vector on L, we have the following equivalent block update:

$$w(kL+L) = w(kL+L-1) + 2\,\mu \sum_{m=0}^{L-1} x(kL+m)\, e(kL+m)$$

$$w(k+1) = w(k) + 2\,\mu \sum_{m=0}^{L-1} x(kL+m)\, e(kL+m) \tag{9}$$

Thus, k refers to block time and n denotes the original time index of the incoming data. The block LMS algorithm in eq(9) essentially minimizes the same MSE performance function as the non-block LMS algorithm in eq(3).

D. Frequency Domain Adaptive Filter Algorithm (FDAF)

The basic operation in a frequency-domain adaptive filter is the transformation of the input signal into a more desirable form before the adaptive processing. This is accomplished by discrete Fourier transforms (DFTs). The Overlap-save method is explained below. [9]

Consider first the process of computing the filter output in eq(8). Let $w(k)$ and $X(k)$ be derived from the corresponding time-domain quantities as,

$$W(k) = F\left[w^T(k), 0, ..., 0 \right]^T \tag{10}$$

and $$X(k) = Diag\left\{ F\left[x(kN-N), ..., x(kN-1), x(kN), ..., x(kN+N-1) \right]^T \right\} \tag{11}$$

According to the overlap-save method, N output samples $y(k) = \left[y(kN), ..., y(kN+N-1) \right]^T$ from a linear convolution can be computed as

$$y(k) = last\ N\ components\ of\ F^{-1}\left[Y(k) \right] \tag{12}$$

Where, $Y(k)$ is the frequency-domain output vector given as $Y(k) = X(k)W(k)$ .The input sequence in eq(11) contains N samples from the current block of data and another N samples from the previous block, i.e. the data are being overlapped by N points so that only N new samples are introduced before the DFT is computed for each block update. Only the last N points of the IDFT of $Y(k) = X(k)W(k)$ are retained because the first N terms correspond to a circular convolution.

A similar technique can be employed to implement the block adaptive algorithm because the gradient in eq(12) is a linear correlation and the weights are fixed for the entire block of N samples. The error terms are computed in the time domain according to

$$e(kN+m)=d(kN+m)-y(kN+m) \qquad m=0,............,(N-1)$$

and this block is grouped as

$$e(k)=\left[e(kN),........,e(kN+N-1)\right]^T=d(k)-y(k)$$

where $d(k)=\left[d(kN),.........,d(kN+N-1)\right]^T$, is transformed to the frequency domain as follows:

$$E(k)=F\left[0,............,0,e^T(k)\right]^T \qquad (13)$$

The error vector is augmented with N zeros because N terms of the output are discarded to implement the linear convolution in eq(12). Alternatively, one may view $e(k)$ as having the same role in the correlation as $w(k)$ does in the convolution, except that the zeros precede $e(k)$ because a correlation is basically a "reversed" convolution. Applying the same reasoning as was used to derive the block output, it is shown that the block gradient estimate is

$$\hat{v}(k)=\textit{first N components of } F^{-1}\left[X^H(k)\,E(k)\right] \qquad (14)$$

Where, the first N elements are retained.

The final step of the algorithm transforms this time-domain gradient into its frequency-domain counter-part, which is then added to $W(k)$ in order to generate the updated weights $W(k+1)$. Because $w(k)$ is followed by N zeros in eq(10), the gradient in eq(14) must be similarly augmented. The algorithm is thus given by,

$$W(k+1)=W(k)+2\mu F\left[\hat{v}^T(k),0,..................,0\right]^T \qquad (15)$$

this is equivalent to the update in eq (9) except that DFTs have been used to implement the output convolution and the gradient correlation.

III. MATLAB SIMULATION RESULTS

A. LMS Algorithm

1) Convergence Rate

The convergence rate determines the rate at which the filter converges to its resultant state. Usually a faster convergence rate is a desired characteristic of an adaptive system. The simulation results for Convergence characteristics of standard LMS Algorithm are shown in figure 2 and 3. [4]

Fig 2 shows convergence characteristics for good telephone channel. Figure 3 shows another example of convergence characteristics where the channel impulse response is described by the Raised Cosine filter as shown below,

$$h_n=\begin{cases}0.5\left\{1+\cos\left[\dfrac{2\Pi(n-2)}{W}\right]\right\} & n=1,2,3\\ 0 & ,otherwise\end{cases} \qquad (16)$$

Where, parameter W controls the amount of amplitude distortion increasing with W.

The value of μ is selected to satisfy the condition,

$0 < \mu < \dfrac{2}{\lambda_{max}}$ Where, λ_{max} - maximum Eigen value of Autocorrelation matrix of input data.

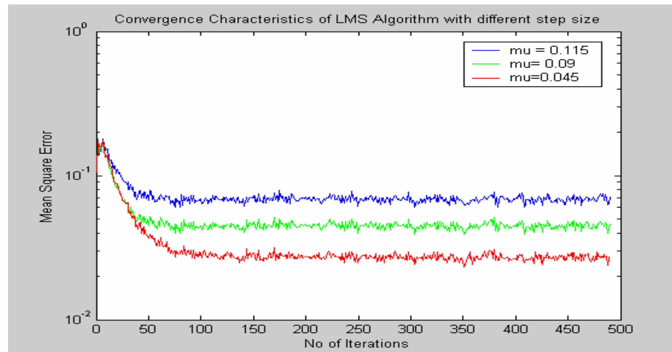

Fig. 2 Convergence of LMS for different step sizes

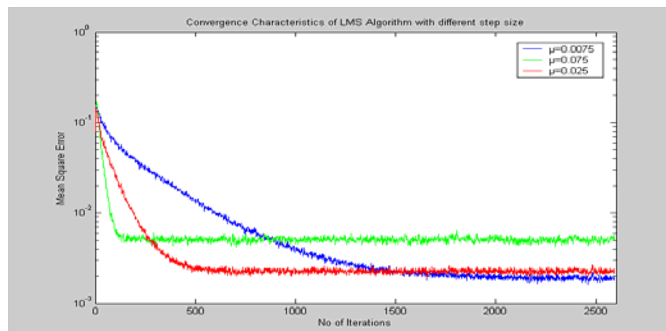

Fig. 3 Convergence of LMS for Raised Cosine Channel with W =3.1

2) Eigenvalue Spread

Figure 4 shows the effect of eigen value spread on the convergence of LMS Algorithm. The value of μ is chosen to be 0.075.

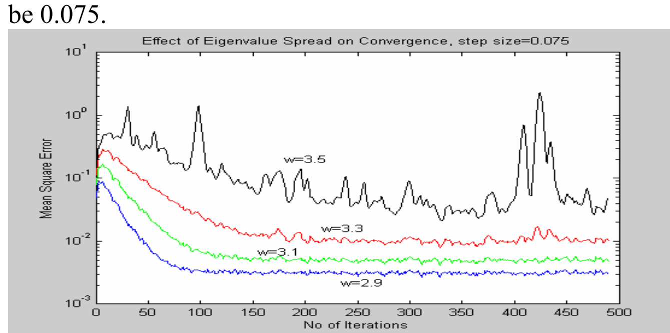

Fig. 4 Effect of Eigenvalue spread, step size = 0.075

From the result we see that increasing the eigen value spread has the effect of slowing down the rate of convergence of the adaptive filter [2] for example, when W =2.9, approximately 80 iterations are required to converge in the mean square, and the average squared error (after 500 iterations) equals approximately 0.003.But for W =3.5, the equalizer requires approximately 200 iterations to converge in the mean square, and resulting average squared error equals approximately 0.04.

In figure 5 the ensemble-average impulse responses of the adaptive filter after 500 iterations for each of the four eigen value spreads of interest are shown. From figure we see that in each case the ensemble-average impulse response of the adaptive filter is very close to being symmetric with respect to the center tap and with minimum ISI values, as expected.

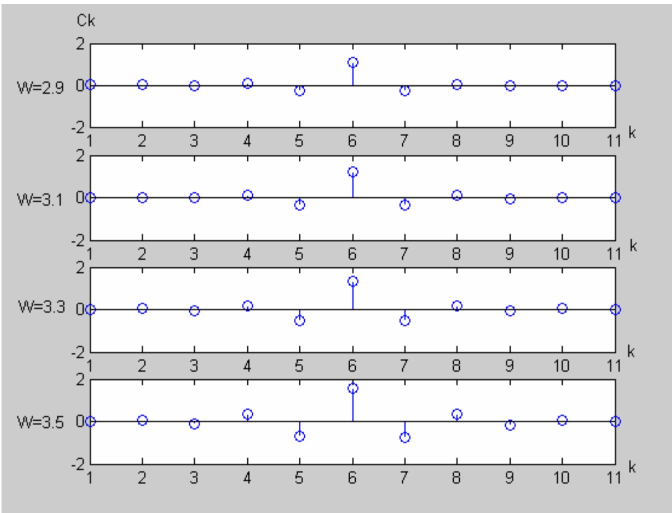

Fig. 5 Impulse Response of Equalizer for different Eigenvalue spread.

3) Filter Length

Filter length is another important performance parameter of adaptive systems. The length of the filter specifies how accurately a given system can be modeled by the adaptive filter. As shown in figure 6, as filter length is increased the MSE achieved is smaller, but the filter converges slowly.

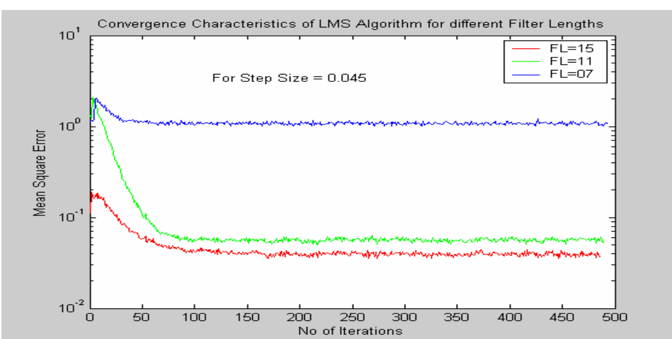

Fig 6 Convergence characteristics with different filter lengths

B. NLMS Algorithm

Figure 7 shows convergence characteristics of NLMS for good telephone channel

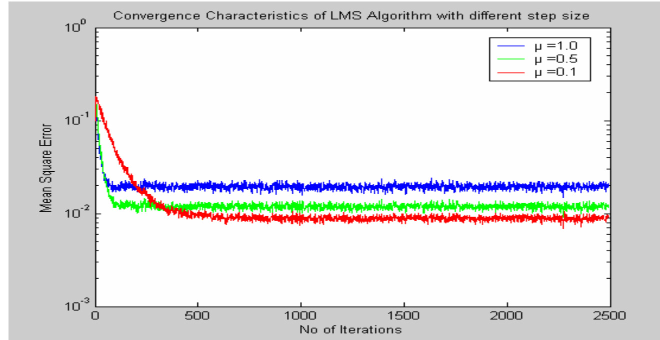

Fig. 7 Convergence of NLMS for Telephone Channel

As stated earlier, NLMS converges faster than the standard LMS algorithm at very little extra cost of increased complexity. This is shown in figure 8 and 9.

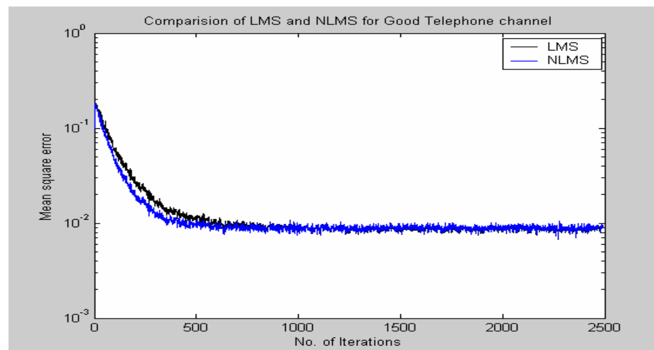

Fig. 8 Comparison of LMS & NLMS for good Telephone channel

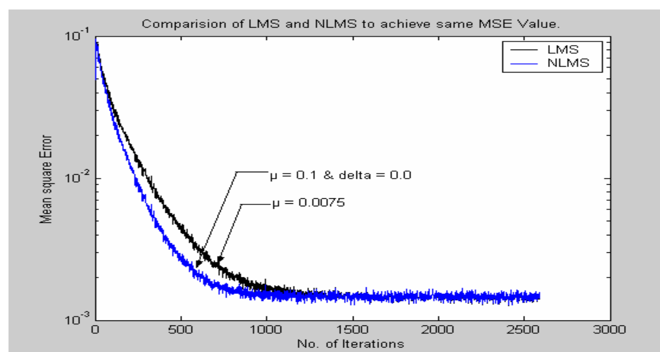

Fig. 9 Comparison of LMS & NLMS for Raised cosine channel

The simulation is carried for telephone channel and raised cosine channel with dispersion W =2.9. The experiment is carried to achieve equal value of MSE with both LMS and NLMS Algorithms. Simulation Results above show very obvious reason why NLMS is very commonly used.

C. Performance comparison of LMS and BLMS

For wide-sense stationary signals, the steady-state weight vector and time constants of the BLMS algorithm are identical to those of the standard LMS algorithm (refer fig.10)

The main difference is that the maximum value of the step size such that the algorithm is stable is now scaled down by a factor of L .

$$w(k+1) = w(k) + 2\,\mu\, L\left[\frac{1}{L}\sum_{m=0}^{L-1} x(kL+m)\, e(kL+m)\right] \qquad (17)$$

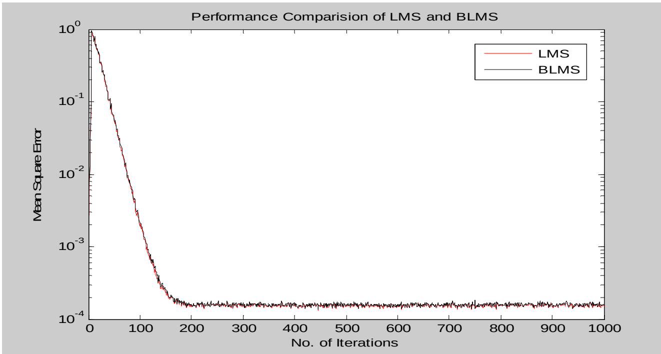

Fig. 10 Performance comparison of LMS and BLMS

Where, $\mu_L = \mu L$ is the effective step size. In fig.11 the channel used is a telephone channel with step size μ =0.045. If the input signal correlation matrix has a large eigen value spread, then the BLMS algorithm may converge more slowly than the LMS algorithm because of the tighter upper bound on μ . This is shown in fig.11 where the channel used is raised cosine channel with dispersion, W =3.5 and eigen value spread of 46.8216.

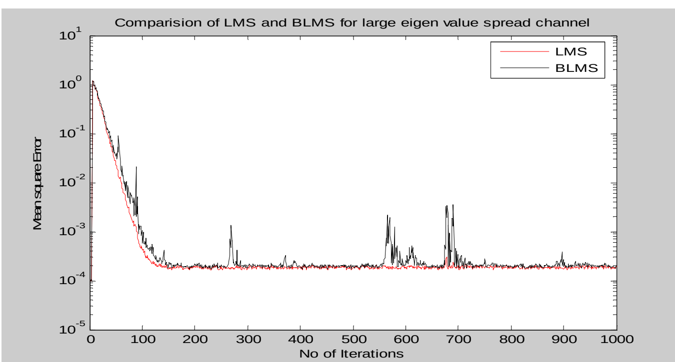

Fig.11 Comparison of LMS and BLMS

D. Comparison of LMS and FDAF

As the overlap-save FDAF is an efficient implementation of the BLMS algorithm, it has the same convergence properties.

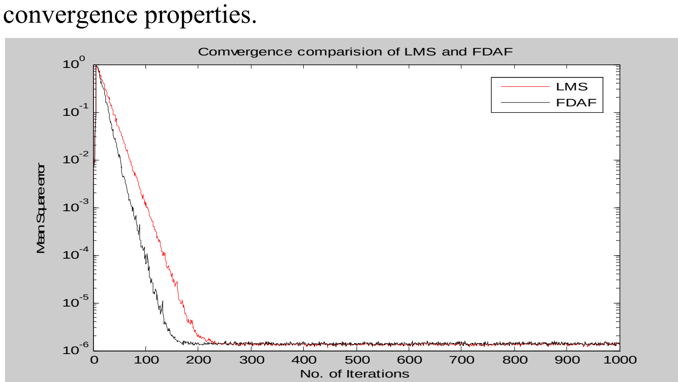

Fig. 12 Comparison of LMS and FDAF

Also, the adaptive weights converge to the same Wiener weight vector, yielding the same steady-state (minimum) MSE. If a different step size is used for each adaptive weight, the convergence rate of the algorithm can be improved without increasing this minimum MSE. (Refer Fig.12) Fig.12 shows that FDAF has faster convergence rate than time domain LMS to achieve same MSE. The channel used is a good Telephone channel. Fig.13 shows convergence of FDAF for raised cosine channel with dispersion W =3.5.

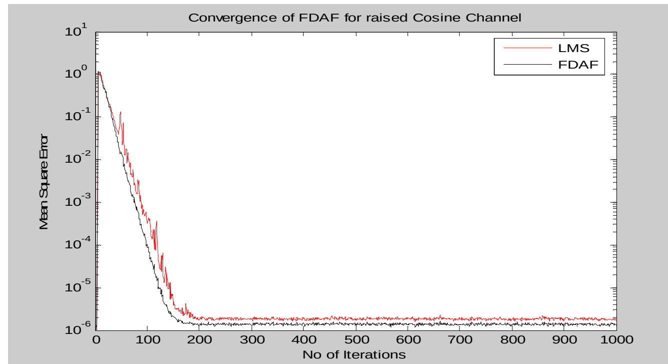

Fig. 13 Convergence of FDAF for raised cosine channel

IV. SELF-RECOVERING (BLIND) EQUALIZATION

In all the algorithms discussed earlier, it is assumed that a known training sequence is transmitted to the receiver for initial adjustment of equalizer coefficients. However in some applications it is desirable that the receiver synchronizes to the received signal and adjusts the equalizer without having a known training sequence. The typical examples are ‘Multipoint Data Networks’ and ‘Wireless communication’.

In wireless communication systems, particularly in a mobile communications channel, it is impractical to employ a training sequence of long duration because of two reasons:

1. The system cost involved in the repeated transmission of a known sequence to train the equalizer at the receiving end of the system is too high.
2. The unavoidable presence of multipath fading makes it difficult to establish data transmission over the channel when there is outage in the system. Fading arises because the transmitted signal tends to propagate along several paths, each of different electrical length. [11]

Thus need of fast converging equalization algorithms and capability of receiver to achieve adaptation without cooperation of the transmitter and cost of retransmission is of paramount importance for most applications.

Godard’s Algorithm uses a cost function that characterizes the amount of intersymbol interference at the equalizer output independently of the data symbol constellation and of carrier phase. Godard proposed a cost function that is independent of carrier phase and has the property that its minimum leads to a small MSE i.e.

$$G^{(P)} = E(\left|z_n\right|^p - \left|a_n\right|^p)^2 \qquad (18)$$

The value of p is chosen as 2 for simplicity, which leads to an algorithm simple to implement in a microprocessor based receivers.

$$C_{n+1} = C_n - \lambda_2 y_n^* z_n \left(|z_n|^2 - R_2 \right) \qquad (19)$$

With
$$R_2 = \frac{E\,|a_n|^4}{E\,|a_n|^2} \qquad (20)$$

Above case of Godard's Algorithm with $p = 2$ is called as 'Constant Modulus Algorithm'.

A. *Performance Evaluation of CMA Algorithm*

Figure 14 shows convergence of CMA without introducing modulation. The equalizer used is an 11 tap equalizer and the channel is a telephone channel.

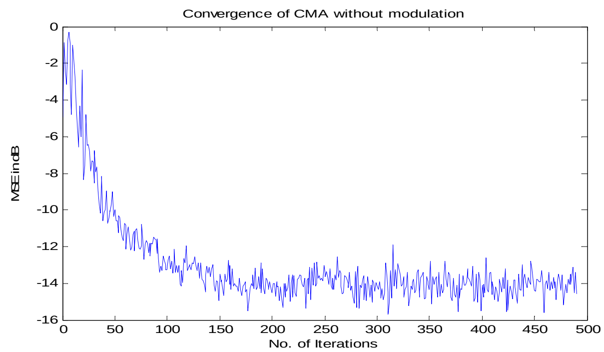

Fig.14 Convergence of CMA without Modulation

Figure 15, 16 and 17 shows an eye diagram representation of Information signal, Signal with added ISI and noise, and of Data equalized using CMA algorithm. As can be seen from fig 18, the eye diagram of an equalized data is widely opened indicating fairly good equalization.

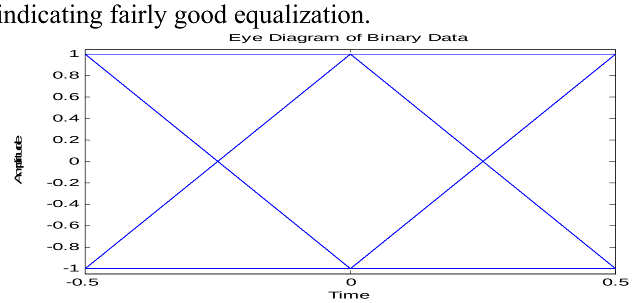

Fig.15

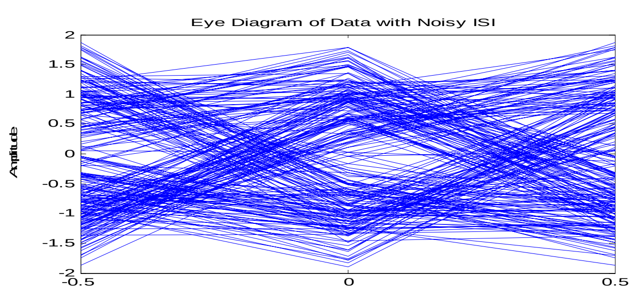

Fig.16

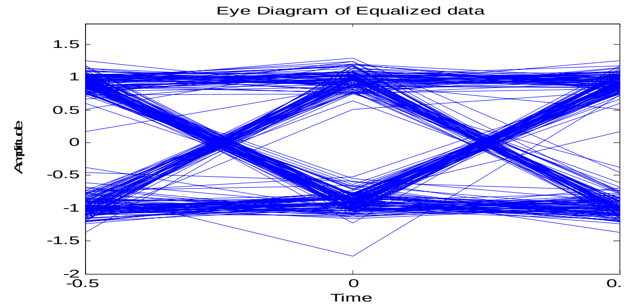

Fig. 17

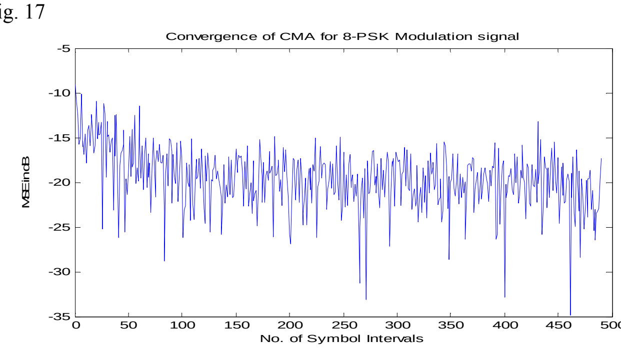

Fig. 18

Fig. 18 shows convergence of CMA with modulation. The modulation scheme used is 8-PSK with a constellation as shown in fig 19.

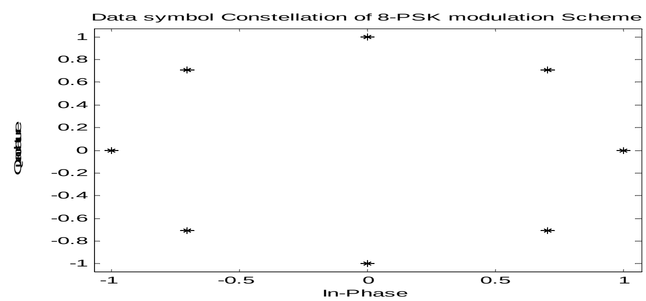

Fig.19

V. CONCLUSION

Thus using Matlab simulations, various performance parameters of the standard LMS and Normalized LMS as well as comparison of performance of the standard LMS, Block LMS and Frequency Domain adaptive filter are verified using an 11 tap equalizer built around a transversal filter.

It can be seen that the convergence characteristics of these algorithms highly depend on the step-size parameter μ. Thus selection of proper value of μ is important as it affects the convergence rate and the cost function i.e. Mean Square Error. Also it is verified that NLMS gives improved convergence rate over the standard LMS, but with slight increase in complexity. Also it is verified that block LMS and FDAF has no much improvement over standard LMS, but they definitely reduce the complexity problem in case of filters with long impulse responses. These techniques reduce the complexity because the filter output and the adaptive weights are computed only after a large block of data has been accumulated. Also due to advanced techniques like FFT in signal processing, the algorithms can efficiently perform the filter convolution and gradient correlation.

Godard's algorithm is a simplest of all blind equalization algorithms. It requires no much computing power than the conventional gradient algorithms, which makes its implementation easy and thus attractive in microprocessor based data receivers. The cost function used in CMA is a non-convex function which represents ISI independent of carrier phase. Thus equalizer convergence does not require carrier recovery and it can be carried out at the equalizer output in decision directed mode.

As equalization plays important role in wired as well as wireless world, it is needed to find optimum performance algorithms for adaptive filters. The work above can be used as a ground to find the optimum performance algorithm among many available.

VI. REFERENCES

Books:

[1] John G. Proakis, "Digital Communications" McGraw-Hill International Edition, Fourth Edition 2001
[2] Simon Haykin, "*Adaptive Filter Theory*", fourth edition 2002
[3] Theodore S. Rappaport, "*Wireles Communications principles & practice Article*", Second Edition.
[4] John G.Proakis, Masoud salehi, "*Contemporary Communication Systems Using MATLAB*", Second Edition, 2004

Papers from Conference Proceedings (Published):

[5] Shahid U.H.Qureshi, "Adaptive Equalization", Vol-73, Issue-9, IEEE Proceedings-Sep 1985
[6] Yaser Ahmad, Farooq jawed, Sohaib Zia, Muhammad Aga, "Real-time Implementation of Adaptive Channel equalization Algorithms on TMS320C6x DSP Processor", IEEE Proceedings-2004
[7] E. Kalpana, O. Uma Maheswari, "Architecture Design for an Adaptive Equalizer", National Conference, Pune.
[8] Moshe Tarrab, Arie Feuer, "Convergence and Performance of Normalized LMS Algorithm with Uncorrelated Gaussian Data", IEEE transactions on Information Theory, Vol-34, No. 4, July 1988
[9] John. J Shynk, "Frequency-Domain and Multirate Adaptive Filtering", IEEE Transactions on Signal Processing, Jan-1992
[10] Gregory A. Clark, Sydney R. Parker, Sanjit K. Mitra, "A Unified Approach to Time and Frequency-Domain Realization of FIR Adaptive Digital Filters", IEEEtransactions on Acoustics, speech, and Signal Processing, vol. ASSP31, No., October 1983
[11] Domnique N. Godard, "Self-Recovering Equalization and carrier tracking in Two-Dimensional Data Communication Systems", IEEE transactions on Communications, vol. COM-28, No.11, Nov.1980

VII. BIOGRAPHIES

Minal Gonsalves is working as Lecturer in St. Francis Institute f Technology, Mount Poinsur, Borivali (w), Mumbai 400 103 and also a P.G Student of Department of Electronics Engineering, Fr.Conceicao Rodrigues College of Engineering, Bandra (w), Mumbai – 400 050 (e-mail: minal_ag@rediffmail.com)

Srija Unnikrishnan is HOD of Department of Electronics Engineering, Fr.Conceicao Rodrigues College of Engineering, Bandra (w), Mumbai – 400 050 (e-mail: srija@frcrce.ac.in)