IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 7, Issue 2, Ver. I (Mar. - Apr. 2017), PP 34-40
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Adaptive Channel Equalization using Multilayer Perceptron Neural Networks with variable learning rate parameter

Emmanuel K. Chemweno[1], Edward. N. Ndung'u[2], Heywood A. Ouma[3]
[1]*(Dept. of Telecommunications and Information Engineering, Jomo Kenyatta University of Agriculture and Technology, Box 62000, Nairobi , Kenya)*
[2]*(Dept. of Telecommunications and Information Engineering, Jomo Kenyatta University of Agriculture and Technology, Box 62000, Nairobi , Kenya)*
[3]*(Dept. of Electrical and Information Engineering, University of Nairobi, Box 30197, Kenya)*

Abstract: *This research addresses the problem inter-symbol interference (ISI) using equalization techniques for time dispersive channels with additive white Gaussian noise (AWGN). The channel equalizer is modelled as a non-linear Multilayer Perceptron (MLP) structure. The Back Propagation (BP) algorithm is used to optimize the synaptic weights of the equalizer during the training mode. In the typical BP algorithm, the error signal is propagated from the output layer to the input layer while the learning rate parameter is held constant. In this study, the BP algorithm is modified so as to allow for the learning rate to be variable at each iteration and this achieves a faster convergence. The proposed algorithm is used to train the MLP based decision feedback equalizer (DFE) for time dispersive ISI channels. The equalizer is tested for a random input sequence of BPSK signals and its performance analysed in terms of the Bit Error Rates and speed of convergence. Simulation results show that the proposed algorithm improves the Bit Error Rate (BER) and rate of convergence.*

Keywords: *Equalization, intersymbol interference, multilayer perceptron, back propagation, variable learning rate, bit error rate (BER)*

I. Introduction

Real physical channels suffer the problem of signal degradation due to AWGN, time dispersive channels and multipath propagation [1], [2]. This causes the received pulses to smear onto each other with the result that they are no longer distinguishable [3].This is known as ISI and has the effect of reducing the rate of data transmission over a communication channel.

Channel equalization is used to mitigate the effects of ISI. In most channels, the characteristics vary with time, and an adaptive equalizer whose coefficients vary automatically and adaptively is desirable [1], [4]. Traditionally, equalizers were modelled as linear equalizers using finite impulse response (FIR) transversal or lattice structure filters. These used the least mean square error (LMS) and recursive least square error (RLS) algorithms to optimise the equalizer coefficients. These algorithms do not perform well for channels with non-linearities and is never used for non-minimum phase channels [5]. Decision feedback equalizer (DFE) is non-linear and yields superior results to a linear equalizer [1] and is used to mitigate ISI in non-linear channels provided the distortion is not too severe. The DFE, however, suffer the problem of error propagation [6].

Neural networks (NN) perform well for non-linear channels [7] and are used for both minimum and non-minimum phase channels and are resistant to additive noise [8]. NN are complex and exhibit slower learning times than linear equalizers [6]. In order to minimize the computational complexity and accelerate the slow learning process (convergence rate), modification of the learning algorithms has been proposed. Lee et al [9] have proposed a hierarchical topology of the MLP, where, each layer has got its own BP algorithm and the results were superior to that of the standard BP algorithm.

II. Multilayer Perceptron Based Decision Feedback Equalizer

The equalizer is modelled as a 4-1 DFE with a 9-3-1 fully connected MLP structure. The equalization scheme shown in Figure 1.

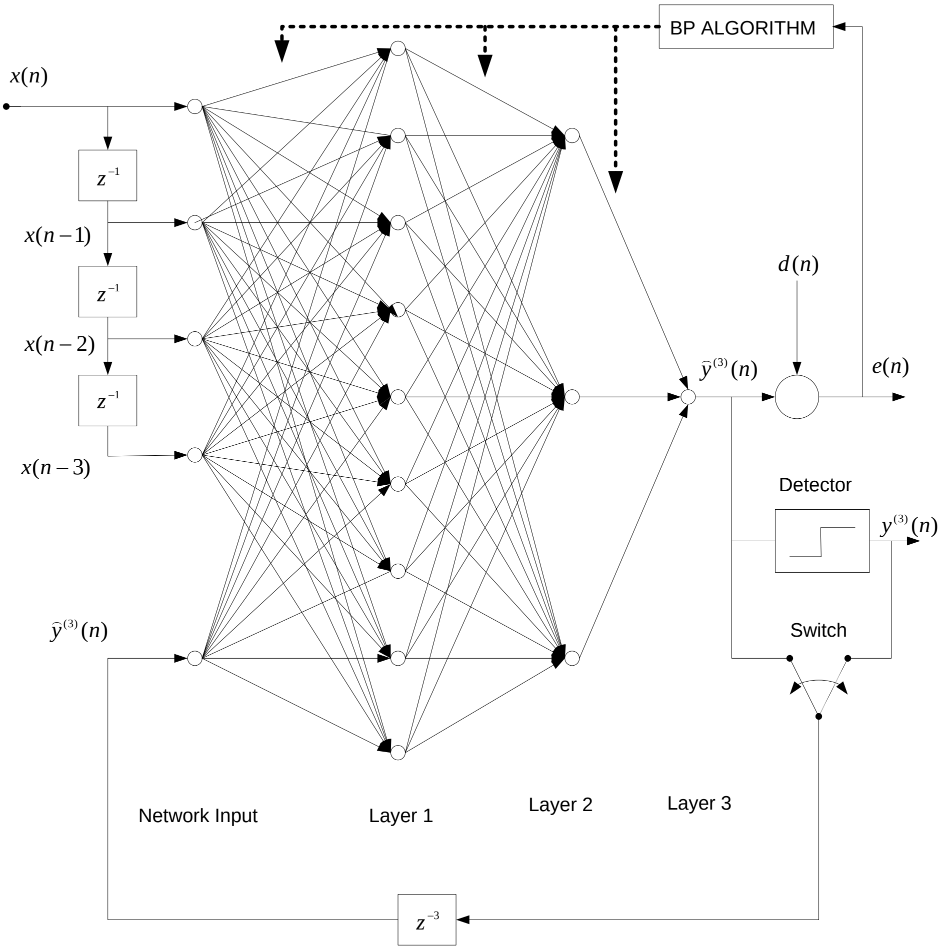

Figure 1: MLP structure for channel equalization

The MLP consists of an input layer, one hidden layer and an output layer.
The detector demodulates the output signal, so as to make a decision of which signal was transmitted.
The switch is used to change from the training phase to the equalization phase after the training input samples have all been presented.
The error signal activates the BP algorithm so as to adjust the synaptic weights of the network.
The structure of the MLP equalizer may be described as:

(i) (4,1) DFE as having an input vector of four feedforward input taps and one feedback tap.
(ii) (9,3,1) MLP as a fully connected MLP structure of 9 input neurons, 3 hidden neurons and one output neuron.
(iii) Decision delay d, as the feedback loop delay. This is to ensure that the equalizer is causal at time $n = 0$, and that the output can be determined after the input sequence has been presented to all the taps.

The following notation is used to describe the MLP network:

n	-	Time index
$d = 3$	-	Decision delay
$L = 3$	-	Number of layers
$l = 0,1,2$	-	Layer index
$N^{(l)}$	-	Number of neurons for layer l.
$x_j^{(l)}$	-	Input associated to neuron j of layer l of the network
$y_k^{(l)}$	-	Output of neuron k of layer l.
$b_k^{(l)}$	-	Bias of neuron k of layer l.

$w_{kj}^{(l)}$ - The synaptic weight connecting to the input of neuron k of layer l to output j of layer $l-1$ to
$d_k^{(2)}$ - Target output k of the output layer.
$\varphi(\bullet)$ - Activation function

III. Proposed Algorithm

To increase the speed of the LMS algorithm without using estimates of the input signal correlation matrix, variable convergence factor is desirable. The proposed algorithm is developed from the Normalised Least Mean Square (NLMS) algorithm which is expressed by equation (1):

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \eta \frac{\mathbf{x}(n)}{\|\mathbf{x}(n)\|^2} e(n) \tag{1}$$

Where:
$\hat{\mathbf{w}}(n)$ = Filter weight vector at time n
$\mathbf{x}(n)$ = Input signal vector
η = Learning rate parameter
$e(n) = d(n) - y(n)$; The error between the desired output and the actual output

In order to make the learning rate parameter adaptive, the NLMS weight update equation given by equation (1) is re-written as equation (2):

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \eta(n) \frac{\mathbf{x}(n)}{\|\mathbf{x}(n)\|^2} e(n) \tag{2}$$

Where:
$\eta(n)$ = The learning rate parameter at time n
Defining the weight error vector as the difference between the unknown true filter weight vector $\mathbf{w}_t$ and the filter weight at given time $\hat{\mathbf{w}}(n)$:

$$\boldsymbol{\varepsilon}(n) = \mathbf{w}_t - \hat{\mathbf{w}}(n) \tag{3}$$

$$\boldsymbol{\varepsilon}(n+1) = \mathbf{w}_t - \hat{\mathbf{w}}(n+1) = \boldsymbol{\varepsilon}(n) - \eta(n) \frac{\mathbf{x}(n)}{\|\mathbf{x}(n)\|^2} e(n) \tag{4}$$

To minimize the mean square error (MSE), the expectations of the squared norms of equation (4) is taken:

$$E\{\|\boldsymbol{\varepsilon}(n+1)\|^2\} = E\{\left\|\boldsymbol{\varepsilon}(n) - \eta(n) \frac{\mathbf{x}(n)}{\|\mathbf{x}(n)\|^2} e(n)\right\|^2\}$$

$$= E\{\|\boldsymbol{\varepsilon}(n)\|^2\} - \Delta\eta(n) \tag{5}$$

Where:

$$\Delta\eta(n) = -\eta^2(n) E\{\left\| e(n) \frac{\mathbf{x}^T(n)\mathbf{x}(n)}{\|\mathbf{x}^T(n)\mathbf{x}(n)\|^2} e(n) \right\|\} + 2\eta(n) E\{\left\| e(n) \frac{\mathbf{x}^T(n)}{\|\mathbf{x}^T(n)\mathbf{x}(n)\|} \boldsymbol{\varepsilon}(n) \right\|\} \tag{6}$$

To maximize $\Delta\eta(n)$, we make it the objective function and differentiate with respect to η , and equate to zero.

$$\frac{d\Delta\eta(n)}{d\eta} = -2\eta(n) E\{\left\| e(n) \frac{\mathbf{x}^T(n)\mathbf{x}(n)}{\|\mathbf{x}^T(n)\mathbf{x}(n)\|^2} e(n) \right\|\} + 2E\{\left\| e(n) \frac{\mathbf{x}^T(n)}{\|\mathbf{x}^T(n)\mathbf{x}(n)\|} \boldsymbol{\varepsilon}(n) \right\|\} = 0 \tag{7}$$

And
$$\eta_o(n) = \frac{E\{\left\| e(n) \frac{\mathbf{x}^T(n)}{\|\mathbf{x}^T(n)\mathbf{x}(n)\|} \boldsymbol{\varepsilon}(n) \right\|\}}{E\{\left\| e(n) \frac{\mathbf{x}^T(n)\mathbf{x}(n)}{\|\mathbf{x}^T(n)\mathbf{x}(n)\|^2} e(n) \right\|\}} \tag{8}$$

Where:
$\eta_o(n)$ = The optimum step size at time n

To make the computation of $\eta_o(n)$ tractable, the fluctuations of the input signal energy $\mathbf{x}^T(n)\mathbf{x}(n)$ is assumed to be small from one iteration to the next such that equation (8) may be simplified to [3] :

$$\frac{E\{\left\| e(n)\frac{\mathbf{x}^T(n)}{\left\|\mathbf{x}^T(n)\mathbf{x}(n)\right\|}\boldsymbol{\varepsilon}(n)\right\|\}}{E\{\left\| e(n)\frac{\mathbf{x}^T(n)\mathbf{x}(n)}{\left\|\mathbf{x}^T(n)\mathbf{x}(n)\right\|^2}e(n)\right\|\}} \approx \frac{\dfrac{E\{e(n)\mathbf{x}^T(n)\boldsymbol{\varepsilon}(n)\}}{E\{\left\|\mathbf{x}^T(n)\mathbf{x}(n)\right\|\}}}{\dfrac{E\{e(n)\mathbf{x}^T(n)\mathbf{x}(n)e(n)\}}{E\{\left\|\mathbf{x}^T(n)\mathbf{x}(n)\right\|^2\}}} \approx \frac{E\{e(n)\mathbf{x}^T(n)\boldsymbol{\varepsilon}(n)\}}{E\{e(n)^2\}} \tag{9}$$

We introduce *apriori* error vector such that e_a is given by[10]:

$$e_a(n) = d(n) - \mathbf{x}^T(n)\mathbf{w}(n) \quad = d(n) - \mathbf{x}^T(n)\mathbf{w}_t(n) + \mathbf{x}^T(n)\boldsymbol{\varepsilon}(n) \tag{10}$$

The desired signal $d(n)$ is given by equation (10).

$$d(n) = \mathbf{x}^T(n)\mathbf{w}_t(n)\ , \tag{11}$$

Therefore:

$$e_a(n) = \mathbf{x}^T(n)\boldsymbol{\varepsilon}(n) \tag{12}$$

The error vectors $e(n)$ are related to the *apriori* error vector by:

$$e(n) = e_a(n) + v(n) \quad = \mathbf{x}^T(n)\boldsymbol{\varepsilon}(n) + v(n) \tag{13}$$

Where:

$v(n)$ = Noise

Replacing equation (13) in into equation (9), and neglecting the dependency of $\boldsymbol{\varepsilon}(n)$ on the past noise [3],[10], reduces equation (9) to:

$$\eta_o(n) = \frac{E\{\boldsymbol{\varepsilon}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\boldsymbol{\varepsilon}(n)\} + E\{v(n)\mathbf{x}^T(n)\boldsymbol{\varepsilon}(n)\}}{E\{\boldsymbol{\varepsilon}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\boldsymbol{\varepsilon}(n) + 2\mathbf{x}^T(n)\boldsymbol{\varepsilon}(n)v(n) + v(n)^2\}} \tag{14}$$

For Gaussian signals $E[v(n)] = 0$ and $E[v(n)^2] = \sigma^2$

$$\eta_o(n) = \frac{E\{\boldsymbol{\varepsilon}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\boldsymbol{\varepsilon}(n)\}}{E\{\boldsymbol{\varepsilon}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\boldsymbol{\varepsilon}(n) + \sigma^2\}} \tag{15}$$

The term $E[\boldsymbol{\varepsilon}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\boldsymbol{\varepsilon}(n)]$ in equation (15) may be further simplified as:

$$E[\boldsymbol{\varepsilon}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\boldsymbol{\varepsilon}(n)] = E[\boldsymbol{\varepsilon}^T(n)E\{\mathbf{x}(n)\mathbf{x}^T(n)\}\boldsymbol{\varepsilon}(n)] \quad E[\boldsymbol{\varepsilon}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\boldsymbol{\varepsilon}(n)] = E[\boldsymbol{\varepsilon}^T(n)\mathbf{R}\boldsymbol{\varepsilon}(n)] \tag{16}$$

Where:

$\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$ the correlation matrix

By using the diagonalized form of $\mathbf{R} = \mathbf{Q}\Lambda\mathbf{Q}^T$ (17)

$$E[\boldsymbol{\varepsilon}^T(n)\mathbf{R}\boldsymbol{\varepsilon}(n)] = E[\boldsymbol{\varepsilon}^T(n)\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T\boldsymbol{\varepsilon}(n)] = tr\left\{E[\boldsymbol{\varepsilon}'^T(n)\boldsymbol{\Lambda}\boldsymbol{\varepsilon}'(n)]\right\} \tag{18}$$

Where:

$\mathbf{Q}$ = Eigen vector matrix of the correlation $\mathbf{R}$

$\boldsymbol{\Lambda}$ = Eigen values of $\mathbf{R}$

$tr[\bullet]$ = Trace operation

$$\boldsymbol{\varepsilon}'(n) = \mathbf{Q}^T\boldsymbol{\varepsilon}(n) \tag{19}$$

The effect of the transformation $\boldsymbol{\varepsilon} = \mathbf{w}_t - \widehat{\mathbf{w}}(n)$ shifts the error performance contours from the $\widehat{\mathbf{w}}$ axes to the $\boldsymbol{\varepsilon}$ axes. The transformation $\boldsymbol{\varepsilon}'(n) = \mathbf{Q}^T\boldsymbol{\varepsilon}(n)$ rotates the contours such that the principal axes of the ellipses are aligned with the $\boldsymbol{\varepsilon}'$ axes without altering the shape or eccentricity of the performance surface.

The solution for equation (18) is given by [3]:

$$tr\left\{E[\boldsymbol{\varepsilon}'^T(n)\boldsymbol{\Lambda}\boldsymbol{\varepsilon}'(n)]\right\} = \sum_{k=0}^{N-1}\lambda_k \varepsilon'_{kk}{}^2(n) \tag{20}$$

Where:
ε'_{kk} are the diagonal elements of the matrix $\boldsymbol{\varepsilon}'(n)$
λ_k are the corresponding eigen values
In developing the proposed algorithm, the following assumptions are made:
(i) The assumption made in equation (9) that the input data signal matrix is stationary in the wide sense is extended here. Therefore, the changes in $\mathbf{Q}$ and $\boldsymbol{\Lambda}$ are negligible over a block of data input. This aims at minimizing computational complexity.
(ii) To recursively compute equation (20), the following approximation is made:

$$tr\left\{E[\boldsymbol{\varepsilon}'^T(n+1)\Lambda\boldsymbol{\varepsilon}'(n+1)]\right\} \approx \sum_{k=0}^{N-1}\lambda_k\varepsilon'_{kk}{}^2(n) \pm \sum_{k=0}^{N-1}\lambda_k\Delta\varepsilon'_{kk}{}^2(n) \qquad (21)$$

Where:
$\Delta\varepsilon(n)$ = Change in the weight error vector

The choice of the sign will depend on the error gradient $e(n)x(n-1)$. Successive changes in sign of this quantity indicates that the algorithm is close to its optimal and must the learning rate parameter should be reduced.
(i) After the first transformation, the MSE contours are aligned on the principal axes such that $\boldsymbol{\varepsilon}'(n) \approx \boldsymbol{\varepsilon}(n)$. Successive contours are very close to each other along the main axis. The rotation transformations of the weight error matrix change $\Delta\boldsymbol{\varepsilon}(n)$ are avoided for the subsequent data inputs in a block.

The signal power is normalized and noise variance σ^2 in equation (15) be approximated as $\frac{L}{SNR}$ [10]. (22)

Where:
L = Number of taps of a discrete filter model of the channel
The variable step size algorithm of equation (15) may therefore re-written as:

$$\eta^o(n) = \frac{\sum_{k=0}^{N-1}\lambda_k\varepsilon'^2_{kk}}{\sum_{k=0}^{N-1}\lambda_k\varepsilon'^2_{kk} + \frac{L}{SNR}} \quad n = 0; \qquad (23)$$

$$\eta^o(n+1) = \frac{\sum_{k=0}^{N-1}\lambda_k\varepsilon'^2_{kk}(n) \pm \sum_{k=0}^{N-1}\lambda_k\Delta\varepsilon^2_{kk}(n)}{\sum_{k=0}^{N-1}\lambda_k\varepsilon'^2_{kk}(n) \pm \sum_{k=0}^{N-1}\lambda_k\Delta\varepsilon^2_{kk}(n) + \frac{L}{SNR}} \quad n > 0; \qquad (24)$$

IV. Equalization Process

A random input of BPSK modified information sequence $I(n)$ is passed through a time dispersive channel with AWGN. The equalization process takes place in two phases: the training phase and the decision directed phase. During the training phase the input to the MLP network is a vector consisting of a time delayed input sequence and the undetected feedback signal defined as:

$$\mathbf{x}(n) = [x(n), x(n-1), x(n-2), x(n-3), \hat{y}^{(3)}(n-3)]^T \qquad (25)$$

During this phase, the output of the MLP network is computed layer by layer from the input to the output. The error is backpropagated from the output layer to the input layer, for the adjustments of the weights.
After the training phase, the equalizer switches to the decision directed mode, whereby the feedback signal is the detected version of the output $y(n)$.

V. Simulation Results

The proposed algorithm was simulated for a channel model given by the impulse response:

$$H(z) = 0.3482 + 0.8704z^{-1} + 0.3842z^{-2} \qquad (26)$$

This channel is non linear in nature and has been cited extensively in literature and is recommended by International Telecommunication Union (ITU) for testing equalizers
The performance criteria used to evaluate the performance of the algorithm was the quality of convergence and the bit error rate. In the quality of convergence criteria, an average of 5000 individual runs, each having a sequence of 2500 random signals was used. Figure 2 shows the simulation results plotted against the standard back propagation algorithm (BP) and the modified back propagation (HBP) algorithm [9].

The MATLAB® BER template is used to determine the BER, whereby multiple runs is made with different data inputs. SNR is varied from 0 to 25 dB. At each SNR, the process of equalization is carried out until either a target number of 100 errors are observed, or when the equalizer has processed a maximum of 100,000 bits. The first 1000 symbols were used for training the MLP, with the remaining 1500 to test the performance. Figure 3 shows the BER performance for the BP, HBP [9] and the proposed algorithm, plotted on the same axes.

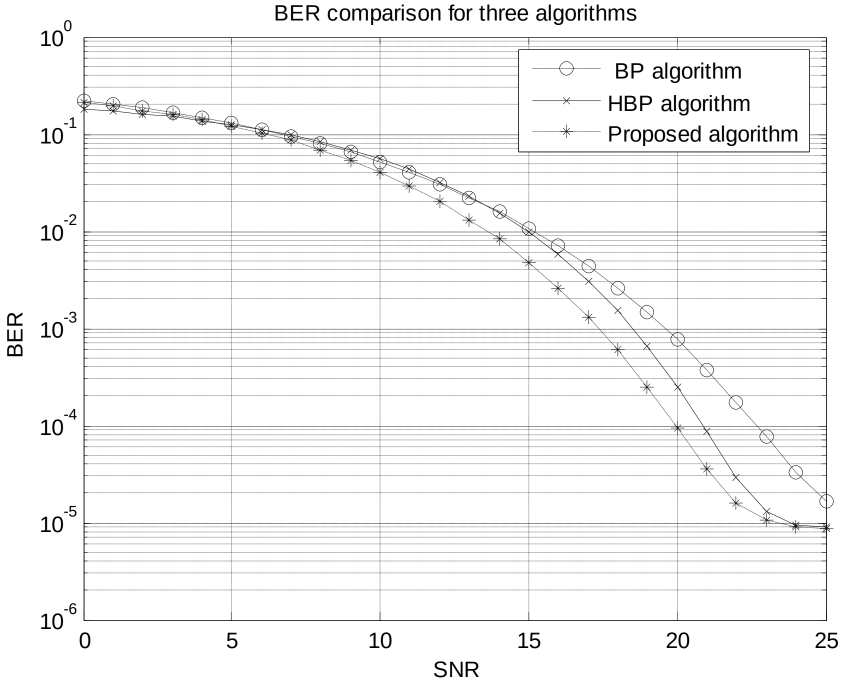

Figure 2: BER Comparison for three algorithms

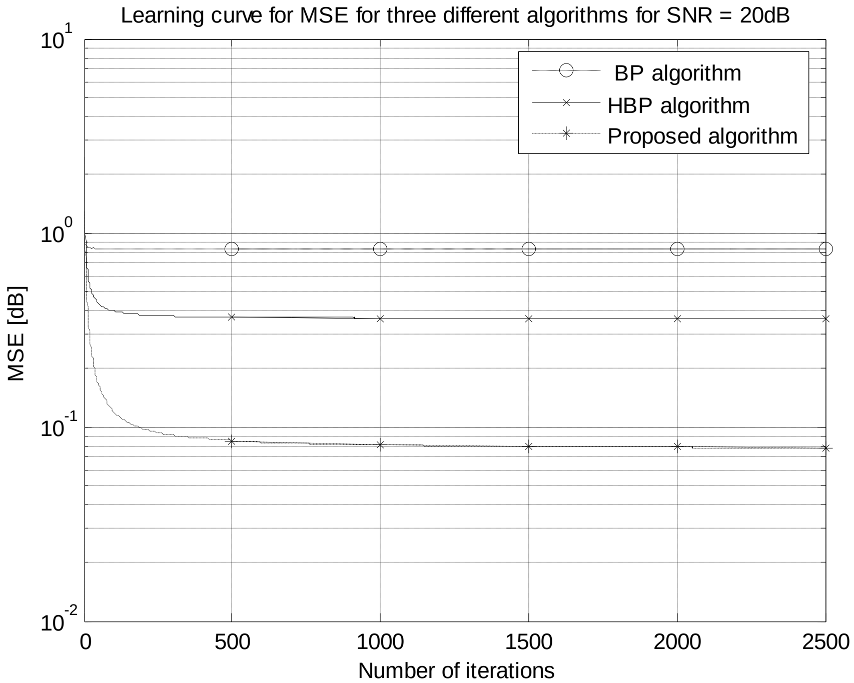

Figure 3: Learning curves for the three algorithms

VI. Simulation Results

This study presents a variable learning rate algorithm for MLP based DFE equalizer. Results for the standard BP, HBP and the proposed algorithm are plotted on the same axes for comparison and inference. At low SNR where the level of corruption in the transmitted signal is very high, the three algorithms yield approximately the same results. BER performance is also very poor owing to high noise power. As the SNR is increased, the proposed algorithm offers an improved performance and the number of errors in the equalized signal is greatly reduced. The algorithm offers a better classification of the noisy input so as correctly make a decision on which symbol had been transmitted. Furthermore, the proposed algorithm offers a minimum MSE compared to the standard BP and HBP algorithms. Simulation results show that the proposed algorithm offers a better performance in both the BER and the quality of convergence criteria as compared to the standard BP and the HBP algorithms.

References

[1]. Proakis J. G. Digital communications, 3rd Edition, McGraw-Hill, Inc. 1995

[2]. Paul J. R, Vladimirova T. "Blind equalization with recurrent neural networks using natural gradient", *ISCCSP symposium*, March 2008, pp 178-183

[3]. Haykin S., Adaptive Filter Theory, Prentice Hall, 4th Edition, 2002

[4]. Bic J. C., Duponteil D. and Imbeaux, J. C. Principles of communication systems, John Wiley & Sons, 1986

[5]. Pichevar R., Vakili V. T., Channel equalization using neural networks, *IEEE*, 1999

[6]. Yuhas B, Ansaris N. Neural Networks in Telecommunications, "Neural network Channel Equalization" Kluwer Academic Press, 1994

[7]. Mo S. and Shafai B. "Blind equalization using higher order cumulants and neural networks", *IEEE transaction on signal processing*. Vol 42 no. 11, 1994, pp 3309 - 3217

[8]. Wang B. Y. and Zheng W. X., "Adaptive blind channel equalization in chaotic communications by using non linear prediction technique", *IEE proceedings on Visual, image and signal processing*, vol 153 no 6, December 2006, pp – 810 - 814

[9]. Lee C. M., Yang S. S. and Ho C. L., "Modified back propagation algorithm applied to Decision feedback equalization", *IEE procedings on visual, image signal processing*, vol 153, no. 6, December 2006, pp 805 - 809

[10]. Hadei S. A., Azmi P., Sonbolestam N., Abbasian J., " Adaptive channel equalization using variable step-size partial rank algorithm based on unified framework", *Proceedings of the ICEE*, May 2010