A new training strategy for DFE-MLP using Modified BP algorithm and application to channel equalization

SAMIR SAIDANI[1], ABD ELKRIM MOUSSAOUI[1], MOHAMED GHADJATI[2]
[1]Department of Electrical and Automatic Engineering
[2]Department of Electronics and Telecommunications
University 8 mai 1945 - Guelma
BP 401, 24000 Guelma
ALGERIA
saidani.samir@univ-guelma.dz & samirtlc@gmail.com

Abstract: In this work, a new training strategy using a new modified back-propagation (BP) algorithm for a multilayer perceptron (MLP) based upon the previously introduced in [8], [9] is proposed. Its performance is investigated and compared to those of MLP-DFE based on the standard BP algorithm and the previously introduced in [8], [9]. The results show improved performance obtained by the new structure in nonlinear channels.

Key-Words: Multi layer perceptron (MLP), Decision feedback equalizer (DFE), Back-propagation (BP), Channel Equalization, Digital Communication.

1 Introduction

Adaptive channel equalizer plays an important role in digital communication systems. It is used to reduce the effects of channel distortion such as noise, intersymbol interference (ISI), non linear distortions, etc. Nonlinear equalizers are superior to linear ones in applications where the channel distortion is too severe for a linear equalizer to handle [1], [2], [3]. A decision feedback equalizer (DFE) is a nonlinear equalizer that is widely used in situations where the ISI is very large [2], [3].

More recently, artificial neural networks have been used in the field of channel equalization. Neural networks-based equalizers have been proposed as alternatives to classical equalizers and provided significant performance improvement in a variety of communication channels because of their capable performance in handling non-linear problems [4], [5]. These equalizers are employing various neural network structures, such as multi-layer perceptron (MLP) which is one of the most popular neural network used in digital communications [6], and it has been incorporated to the DFE (Decision Feedback Equalizer) to enhance its performance. It is shown that the MLP-based DFE trained with the back-propagation algorithm (BP) give a significantly improved performance over the simple DFE [2]. Back-propagation (BP) algorithm is the best known and one of the most common learning algorithms used in neural networks [4], [7]. For a standard BP algorithm, the error signal which is obtained by comparing the output of the node in the output layer with the desired response [4], is propagated layer by layer from the output layer to the input layer to adaptively adjust all weights in the MLP (hence the name back propagation algorithm for this training process). Therefore all parameters of the MLP are obtained by a single BP algorithm [8].

A new learning scheme named Modified back-propagation algorithm applied to decision-feedback equalization has been proposed for improve the performance of the standard BP algorithm [8], [9]. The Modified back-propagation algorithm (HBP) proposed by [8], [9] Based on the hierarchical approach, from the input layer to the output layer of the MLP, every two layers of neural nodes (with one hidden layer) will be trained with an individual BP algorithm. Where each BP algorithm operates a two-layer sub-MLP. These sub-MLP's are arranged from the input layer to the output layer of the MLP.

In this paper, the MLP is adjusted by two BP algorithms independently, with a new scheme hierarchical BP (NHBP) algorithm. The MLP can be divided into two sub-MLPs, and each sub-MLP is optimised by its own BP algorithm in which one of the BP operates the first layer and the other one operates all the other layers (second layer to output layer). The performance of the new scheme hierarchical BP (NHBP) algorithm is evaluated and it is shown that a great improvement in performance is obtained through the use of this technique over both the proposed by [8], [9] and the MLP-DFE based on the standard BP algorithm.

This paper is organized as follows. Section 2 briefly introduces the decision feedback equalizer. The multilayer perceptron based decision feedback equalizer is considered in section 3. Section 4 gives the derivation of the BP algorithm along with the proposed Architecture. Subsequently, Section 5 provides the simulation results. Finally, our conclusion is presented in Section 6.

2 Decision Feedback Equalizer (DFE)

A decision feedback equalizer (DFE) is a nonlinear equalizer that is broadly used for channels with severe amplitude distortion [10]. Fig.1 shows such a decision feedback equalizer (DFE).

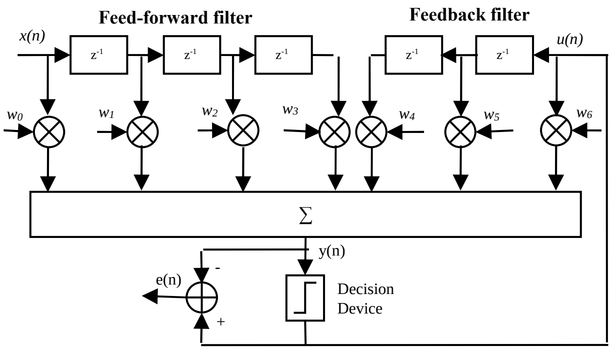

Fig.1 Decision Feedback Equalizer

The structure of the decision feedback equalizer consists of two filters a feed-forward filter (FFF) and a feedback filter (FBF). The input to the FFF is the sequence of the received symbols while the FBF has, as input, the output of the decision device (the hard symbol estimates).
The basic idea is that if the value of the symbols previously found are known, then the ISI contributed by these symbols can be cancelled exactly, by removing past symbols values with suitable weighting from the equalizer output [10].

Consider that the DFE is updated with a recursive algorithm the feed-forward filter weights and feedback filter weights can be jointly adapted by the LMS algorithm on a common error signal *e(n)* as shown in (1).

$$W(n+1)=W(n)+\eta e(n)V(n) \quad (1)$$

Where $e(n)=u(n)-y(n)$ (2)
and

$$V(n)=\left[x(n),x(n-1),..,x(n-k1-1),u(n-k2-1),..,u(n)\right]^T \quad (3)$$

The feed-forward and feedback filter weight vectors are written in a joint vector as:

$$W(n)=\left[w_0(n),w_1(n),....,w_{k1+k2-1}(n)\right]^T \quad (4)$$

k1 and *k2* represent the feed-forward and feedback filter tap lengths respectively.

To further improve the capacities of the DFE, neural networks have been integrated to its fundamental structure.

3 MLP Based DFE

A MLP-based DFE with two hidden layers, as shown in Fig. 2, consists of a feed-forward and feedback section. The input to the feed-forward filter is the sequence of noisy received signal samples. The input to the feedback filter is the output symbol decision sequence from a non-linear symbol detector.

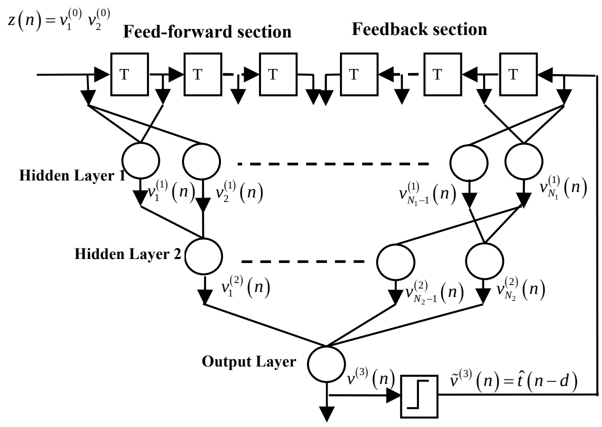

Fig.2 Three-layer MLP based DFE structure.

The basic element of the multilayer perceptron is the neuron. Each neuron in the layer has primary local connections and is characterized by a set of real weights [w_{1j}, w_{2j},…. w_{Nj}] applied to the previous layer to which it is connected and a real threshold level I_j. The *jth* neuron in the *pth* layer accepts real inputs $v_h^{(p-1)}$, from the *(p-1)th* layer and produces an output $v_j^{(p)}$, expressed in the following way:

$$s_j^{(p)}(n)=\sum_{h=1}^{N_{p-1}} w_{hj}^{p}(n)v_h^{(p-1)}(n)+I_j^{(p)}(n) \quad (5)$$

$$v_j^{(p)}(n)=f\left(s_i^{(p)}(n)\right) \quad (6)$$

This output value $v_j^{(p)}$ serves also as input to the *(p+1)th* layer to which the neuron is connected. In the above expressions, *f(.)* represents the activation function. In this paper we are going to use the hyperbolic tangent function (*f(x) =tanh(x)*) in the hidden layer and the linear function in the output layer.

4 Back-propagation algorithm with the new scheme hierarchical (NHBP)

The MLP-based DFE with two hidden layers trained with BP, modified back-propagation (HBP) algorithm and new scheme hierarchical BP algorithm (NHBP) are shown in Figs. 3, 4 and 5, respectively. The back-propagation algorithm has become the standard algorithm used for training multilayer perceptron. It is a generalized LMS algorithm that minimizes the mean-squared error between the desired and actual outputs of the network [11].

In the BP algorithm, the output value is compared with the desired output, resulting in an error signal [2]. This error signal is fed back through the network whose weights are adjusted to minimize a cost function (Fig.3).

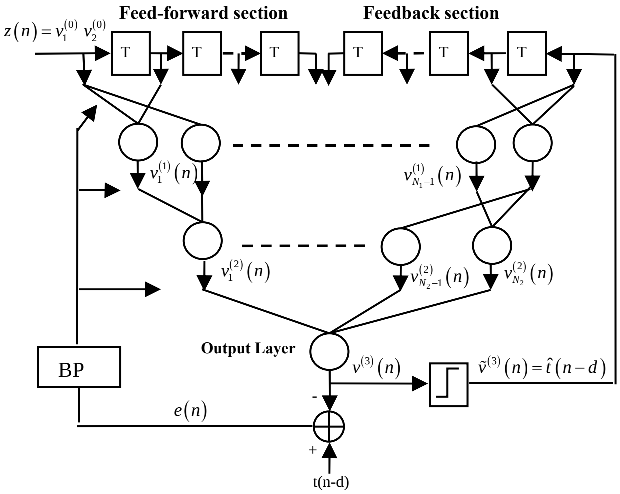

Fig.3 Three-layer MLP-DFE trained with the standard BP algorithm.

The Modified back-propagation algorithm HBP proposed by [8], [9] Based on the hierarchical approach, from the input layer to the output layer of the MLP, every two layers of neural nodes (with one hidden layer) will be trained with an individual BP algorithm. Where each BP algorithm operates a two-layer sub-MLP. These sub-MLP's are arranged from the input layer to the output layer of the MLP. According to figure 4, The BP1 algorithm operates in a two-layer perceptron structure, and the BP2 algorithm operates in a one-layer perceptron structure.

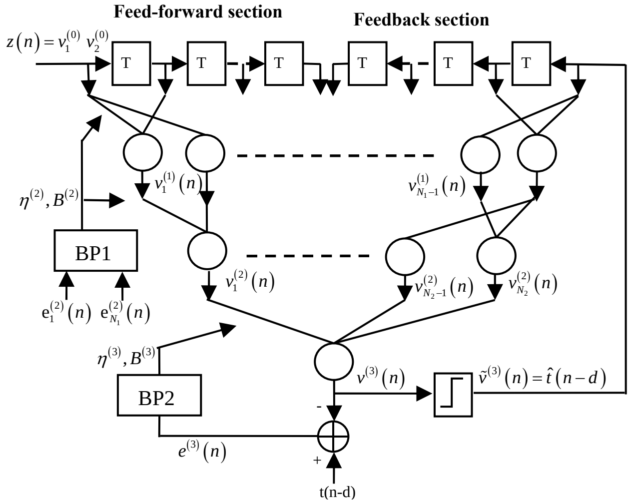

Fig.4 Three-layer MLP-DFE trained with the HBP algorithm.

However, the new scheme hierarchical NHBP algorithm consists of two BP algorithms. The first BP algorithm (BP1) operates the first layer and the second BP algorithm (BP2) operates all the other layers (second layer to output layer).

The MLP-based DFE with two hidden layers trained with new scheme hierarchical BP algorithm (NHBP) as shown in Fig. 5. The BP1 algorithm operates in a one-layer perceptron structure, and the BP2 algorithm operates in a two-layer perceptron structure.

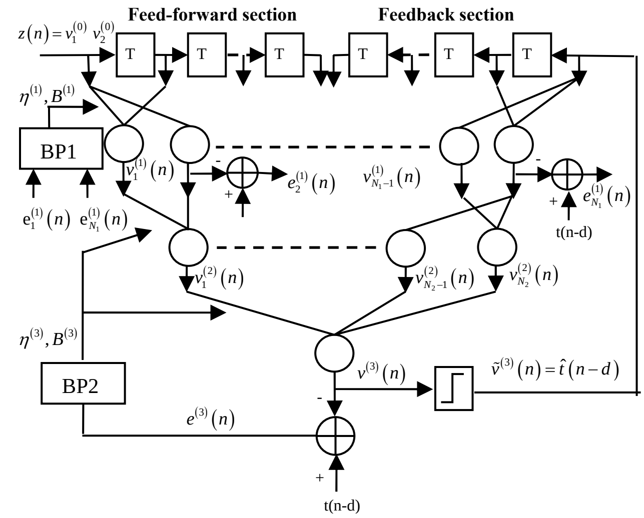

Fig.5 Three-layer MLP-DFE trained with the NHBP algorithm.

In order to derive the NHBP algorithm, we define the error cost function relative to the *i*th node of the *pth* layer as :

$$\zeta_i^{(p)}(n) = \left|e_i^{(p)}(n)\right|^2 \quad (6)$$

where $e_i^{(p)}(n) = t(n-d) - v_i^{(p)}(n)$ (7)

The BP algorithm performs a gradient descent on this function the adjustments of the weight $\Delta w_{ij}^{(p)}(n)$ and threshold level $\Delta I_i^{(p)}(n)$ in each BP algorithm can be derived as :

$$\Delta w_{ij}^{(p)}(n) = -\eta^{(p)}.\nabla_{\Delta w_{ij}^{(p)}(n)}\left(\zeta_i^{(p)}(n)\right) + \Delta w_{ij}^{(p)}(n-1)$$

$$= \eta^{(p)}.\delta_i^{(p)}(n).v_j^{(p-1)}(n) + \Delta w_{ij}^{(p)}(n-1) \quad (8)$$

$$\Delta I_i^{(p)}(n) = -B^{(p)}.\nabla_{\Delta I_i^{(p)}(n)}\left(\zeta_i^{(p)}(n)\right) = B^{(p)}.\delta_i^{(p)}(n) \quad (9)$$

Where $\nabla_x(.)$ is the gradient of $\zeta_i^{(p)}(n)$ with respect to *x*. $\eta^{(p)}$ and $B^{(p)}$ are the learning-rate parameters of the weights and threshold levels, respectively. The delta function $\delta_i^{(p)}(n)$ in (8) and (9) can be defined as :

$$\delta_i^{(p)}(n) = -\nabla_{s_i^{(p)}(n)}\left(\zeta_i^{(p)}(n)\right) = e_i^{(p)}(n).\left(1-\left(v_i^{(p)}(n)\right)^2\right) \quad (10)$$

Furthermore, the formula for the *(p-1)th* layer can be obtained recursively as

$$\delta_i^{(p-1)}(n) = \left(\sum_{j=1}^{N_p}\delta_j^{(p)}(n)w_{ji}^{(p)}(n)\right).v_i^{'(p-1)}(n) \quad (11)$$

Where $v_i^{'(p-1)}(n)$ is the derivative of $v_i^{(p-1)}(n)$.

$$\Delta w_{ij}^{(p-1)}(n) = \eta^{(p)}.\delta_i^{(p-1)}(n).v_j^{(p-2)}(n) + \Delta w_{ij}^{(p-1)}(n-1) \quad (12)$$

$$\Delta I_i^{(p-1)}(n) = B^{(p)}.\delta_i^{(p-1)}(n) \quad (13)$$

The formula of NHBP algorithm in Fig. 5 can be derived as follows.
For the first BP algorithm (BP1): The adjustments of the weights $\Delta w_{ij}^{(1)}(n)$ and the threshold-level $\Delta I_i^{(1)}(n)$ can be written as:

$$\Delta w_{ij}^{(1)}(n) = -\eta^{(1)}.\nabla_{w_{ij}^{(1)}(n)}\left(\xi_i^{(1)}(n)\right) + \Delta w_{ij}^{(1)}(n-1)$$

$$= -\eta^{(1)}.\nabla_{s_i^{(1)}(n)}\left(\xi_i^{(1)}(n)\right).\frac{\partial s_i^{(1)}(n)}{\partial w_{ij}^{(1)}(n)} + \Delta w_{ij}^{(1)}(n-1)$$

$$= \eta^{(1)}.\delta_i^{(1)}(n).v_j^{(0)}(n) + \Delta w_{ij}^{(1)}(n-1) \quad (14)$$

$$\Delta I_i^{(1)}(n) = -B^{(1)}.\nabla_{I_i^{(1)}(n)}\left(\xi_i^{(1)}(n)\right)$$

$$= -B^{(1)}.\nabla_{s_i^{(1)}(n)}\left(\xi^{(1)}(n)\right).\frac{\partial s_i^{(1)}(n)}{\partial I_i^{(1)}(n)}$$

$$= B^{(1)}.\delta_i^{(1)}(n) \quad (15)$$

Where the delta function $\delta_i^{(1)}(n)$ can be evaluated as

$$\delta_i^{(1)}(n) = -\nabla_{s_i^{(1)}(n)}\left(\xi_i^{(1)}(n)\right) = e_i^{(1)}(n).\left(1-\left(v_i^{(1)}(n)\right)^2\right) \quad (16)$$

For the second BP algorithm (BP2): The adjustments of the weights $\Delta w_j^{(3)}(n)$ and the threshold-level $\Delta I^{(3)}(n)$ can be written as :

$$\Delta w_j^{(3)}(n) = -\eta^{(3)}.\nabla_{w_j^{(3)}(n)}\left(\xi^{(3)}(n)\right) + \Delta w_j^{(3)}(n-1)$$

$$= -\eta^{(3)}.\nabla_{s^{(3)}(n)}\left(\xi^{(3)}(n)\right).\frac{\partial s^{(3)}(n)}{\partial w_j^{(3)}(n)} + \Delta w_j^{(3)}(n-1)$$

$$= \eta^{(3)}.\delta^{(3)}(n).v_j^{(2)}(n) + \Delta w_j^{(3)}(n-1) \quad (17)$$

$$\Delta I^{(3)}(n) = -B^{(3)}.\nabla_{I^{(3)}(n)}\left(\xi^{(3)}(n)\right)$$

$$= -B^{(3)}.\nabla_{s^{(3)}(n)}\left(\xi^{(3)}(n)\right).\frac{\partial s^{(3)}(n)}{\partial I^{(3)}(n)}$$

$$= B^{(3)}.\delta^{(3)}(n) \quad (18)$$

The delta function $\delta^{(3)}(n)$ can be evaluated as :

$$\delta^{(3)}(n) = -\nabla_{s^{(3)}(n)}\left(\xi^{(3)}(n)\right) = e^{(3)}(n).\left(1-\left(v^{(3)}(n)\right)^2\right) \quad (19)$$

The adjustments of the weights $\Delta w_{ij}^{(2)}(n)$ and the threshold-level $\Delta I_i^{(2)}(n)$ can be written as :

$$\Delta w_{ij}^{(2)}(n) = \eta^{(3)}.\delta_i^{(2)}(n).v_j^{(1)}(n) + \Delta w_{ij}^{(2)}(n-1) \quad (20)$$

and

$$\Delta I_i^{(2)} = B^{(3)}.\delta_i^{(2)}(n) \quad (21)$$

Where the delta function $\delta_i^{(2)}(n)$ can be evaluated as:

$$\delta_i^{(2)}(n) = \delta^{(3)}(n)w_i^{(3)}.v_i^{'(2)}(n) \quad (22)$$

Where $v_i^{'(2)}(n)$ is the derivative of the $v_i^{(2)}(n)$ with respect to $s_i^{(2)}(n)$.

The simulation results will demonstrate that the NHBP algorithm can improve the performance of the HBP and the standard BP algorithm.

5 Simulation results

The performance of new scheme hierarchical BP (NHBP) algorithm is now examined and compared with classical BP and HBP proposed in [8, 9]. The MLP-DFE structure uses five samples in the input layer, four samples in the feed-forward section and one sample in the feedback section, the number of neurons in the first and second hidden layer and in the output layer is 9, 3, and 1, respectively {(4,1)DFE with (9,3,1)MLP Structure} [8].

The digital message applied to the channel is made of uniformly distributed bipolar random numbers {-1, 1}. The channel noise is taken to be additive white Gaussian noise with a signal to noise ratio (SNR) of 25dB. The learning parameters η=0.07, B =0.05 are used for the BP algorithm, and the learning parameters $\eta^{(2)}$=0.07, $B^{(2)}$=0.05 and $\eta^{(3)}$ =0.07, $B^{(3)}$=0.05 are used for the NHBP and HBP algorithm. Since the three algorithms above are used in the same MLP structure.

Two channel models are used in the simulation and are described by their transfer functions:

$$H_1(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}, \quad (23)$$

and

$$H_2(z) = 0.408 + 0.816z^{-1} + 0.408z^{-2}. \quad (24)$$

The eigenvalue spreads for first and second channel are 25 and 81, respectively. These channels are widely used in literature to evaluate the performance of equalizers [2], [3], [4], [8].

The mean squared error (MSE) is one of the most useful measures for the performance of an equalizer. Here, the MSE performance is determined by taking an average of 600 individual runs, each of which involves a different random sequence and has a length of 3000 symbols.

All the symbols are used for training in the evaluation of the MSE. During this part of the simulations, the performance measure is obtained through the use of learning curves, and BER curves.

We consider a nonlinear channel composed of a linear channel followed by a memory less nonlinearity defined by:

$$z(n) = y(n) + 0.1y^2(n) + 0.05y^3(n) + v(n) \quad (25)$$

Where *y(n)* is output of the linear part of the channel, *z(n)* is the nonlinear channel output and *v(n)* is the additive Gaussian noise.

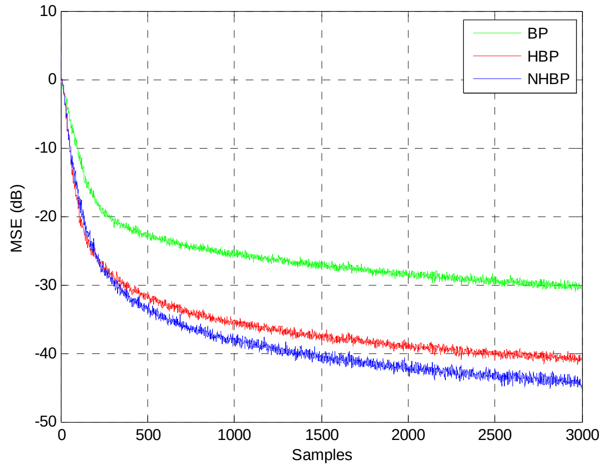

Fig.6 Learning curves for the different algorithms with channel $H_1(z)$

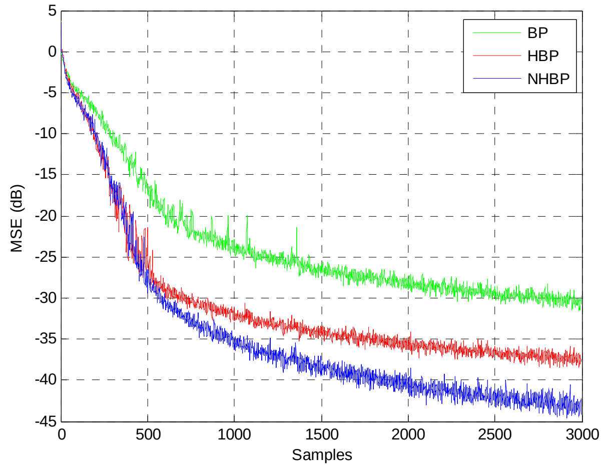

Fig.7 Learning curves for the different algorithms with channel $H_2(z)$.

Fig.6 shows the convergence curves of the three algorithms for the first channel, these curves shows a clear improvement in both the convergence time and the steady state MSE when using the proposed hierarchical. It is also clearly shown that NHBP algorithm performs better than HBP algorithm. As shown in Fig.7, a similar improvement is also obtained for the second channel despite its larger eigenvalue spreads.

Fig.8 and Fig.9 shows the BER performance with various algorithms for the channel (23) and channel (24), respectively. The BER performance is determined by taking an average of 100 individual runs, each of which involves a different random sequence and has a length of 10000 symbols. The first 2000 symbols are used for training and the rest are used for testing. The results indicate that the proposed NHBP algorithm yields a much lower BER than the standard BP and the HBP algorithm.

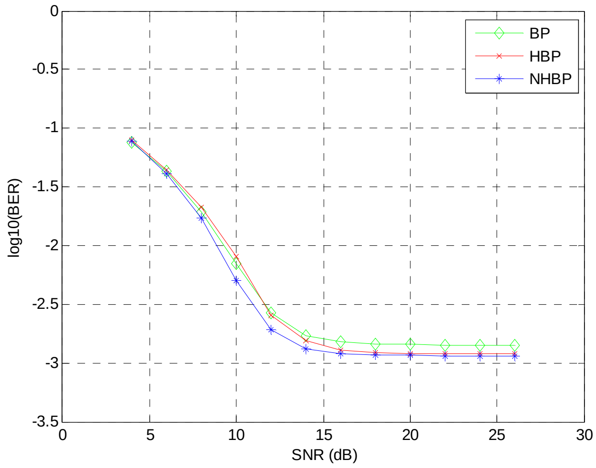

Fig.8 BER curves for the three algorithms with channel $H_1(z)$

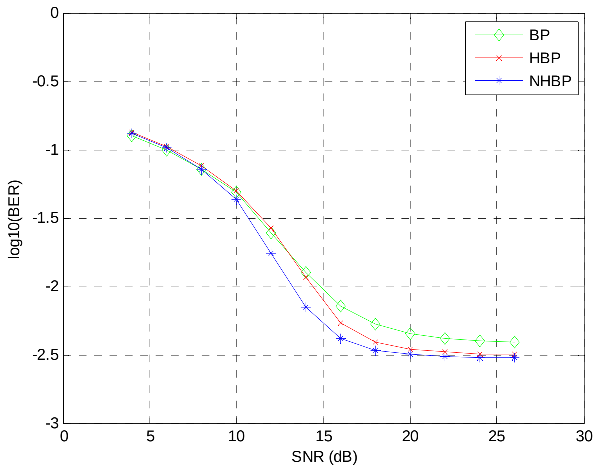

Fig.9 BER curves for the three algorithms with channel $H_2(z)$

6 Conclusion

In this paper, we have used a new scheme hierarchical BP (NHBP) algorithm to update the multilayer perceptron based decision feedback equalizer (MLP-DFE). The performance of the new hierarchical BP (NHBP) algorithm is compared to that of the MLP-DFE based on the standard BP algorithm, and the previously introduced in [8], [9].
The difference between the algorithm proposed in this paper and the algorithm proposed in [8], [9] is that in this latter, the MLP is divided into several sub-MLPs. These sub-MLPs are arranged from the input layer to the output layer of the MLP and each one contains two layers of neural. Every two layers of neural will be trained with an individual BP algorithm. Whereas, in the proposed algorithm (NHBP), the MLP is adjusted by two BP algorithms independently in which one of the BP operates the first layer of the MLP and the other one operates all the other layers of the MLP (from the second layer to the output layer). The simulation has demonstrated the effectiveness of the proposed scheme in terms of the steady state mean square error (MSE) attainable and BER curves for non-linear channels.

References:

[1] M. N. Tehrani, M. Shakhsi, H. Khoshbin, Kernel Recursive Least Squares-Type Neuron for Nonlinear Equalization, *IEEE 21st Conference on Electrical Engineering*, Mashhad, Iran, May 14-16, 2013.

[2] K. Mahmood, A. Zidouri, A. Zerguine, Performance analysis of a RLS-based MLP-DFE in time-invariant and time-varying channels, *Digital Signal Processing*, Vol.18, No.3, 2008, pp. 307-320.

[3] A. Zerguine, A. Shafi, M. Bettayeb, Multilayer perceptron based DFE with lattice structure, *IEEE Transaction on Neural Network*, Vol.12, No.3, 2008, pp. 532-545.

[4] P. Sivakumar, L. Arthi, Enhancement of New Channel Equalizer Using Adaptive Neuro Fuzzy Inference System, *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*, Vol.6, No.1, 2013, pp. 34-43.

[5] S. Panda, P.K. Mohapatra, S.P. Panigrahi, A new training scheme for neural networks and application in non-linear channel equalization, *Applied Soft Computing*, Vol.27, 2015,pp.47-52.

[6] M. Ibnkahla, Applications of neural networks to digital communication-Asuvey, *Signal Processing*, Vol.80, No.7, 2000, pp. 1185-1215.

[7] S. Panda, A. Sarangi, S. P. Panigrahi, A new training strategy for neural network using shuffled frog-leaping algorithm and application to channel equalization, *AEU - International Journal of Electronics and Communications*, Vol.68, No.11, 2014, pp. 1031-1036.

[8] C. M. Lee, S. S. Yang, C.L. Ho, Modified back-propagation algorithm applied to decision-feedback equalisation, *Image Signal Process*, Vol.153, No.6, 2006, pp. 805-809.

[9] S.S. Yang, C.L. Ho, C.M. Lee, HBP: Improvement in BP Algorithm for an Adaptive MLP Decision Feedback Equalizer, *IEEE Transactions On Circuits And Systems*, Vol.53, No.3, 2006, pp. 240-244.

[10] S. Qureshi, Adaptive equalization, *Proceeding of IEEE*, Vol.73, No.9, 1985, pp. 1349-1387.

[11] R. S. Scalero, N. Tepedelenlioglu, A fast new algorithm for training feedforward neural networks, *IEEE Transactions on Signal Processing*, Vol.40, No.1, 1992, pp. 202-210.