Available online at www.sciencedirect.com

ScienceDirect

Digital Signal Processing 18 (2008) 307–320

Digital Signal Processing

www.elsevier.com/locate/dsp

Performance analysis of a RLS-based MLP-DFE in time-invariant and time-varying channels

Kashif Mahmood, Abdelmalek Zidouri, Azzedine Zerguine *

Electrical Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Available online 27 April 2007

Abstract

In this work, a recently derived recursive least-square (RLS) algorithm to train multi layer perceptron (MLP) is used in an MLP-based decision feedback equalizer (DFE) instead of the back propagation (BP) algorithm. Its performance is investigated and compared to those of MLP-DFE based on the BP algorithm and the simple DFE based on the least-mean square (LMS) algorithm. The results show improved performance obtained by the new structure in both time-invariant and time-varying channels. As will be detailed in this work, the newly proposed structure is a compromise between complexity and performance.

Keywords: Multi layer perceptron (MLP); Decision feedback equalizer (DFE); Least-mean square (LMS); Recursive least-square (RLS)

1. Introduction

A serious limitation in attempting to achieve a high transmission rate through a particular band-limited channel is the time dispersion suffered by the signal at the receiving end of this channel [1]. In data transmission, the time dispersion imparted on the transmitted signal results in a time overlap between successive symbols, known as intersymbol interference (ISI). Equalizers have been used to describe filters used to compensate for such distortions in the amplitude and delay characteristics of the channel.

Nonlinear equalizers [1,2] are superior to linear ones in applications where the channel distortion is too severe for a linear equalizer to handle. In particular, a linear equalizer does not perform well on channels with deep spectral nulls in their amplitude characteristics or with nonlinear distortion.

A decision feedback equalizer (DFE) is a nonlinear equalizer that is widely used in situations where the ISI is very large. It has been proved theoretically and experimentally that the DFE performs significantly better than a linear equalizer of equivalent complexity [1]. The basic idea of DFE is that if the values of the symbols already detected are assumed to be correct, then the ISI contributed by these symbols can be canceled exactly by subtracting past symbol values with appropriate weighting from the equalizer output [2].

To further enhance the performance of the DFE, the multilayer perceptron (MLP) has been incorporated to the DFE. It is shown that the MLP-based DFE (MLP DFE) [3] and the MLP DFE with lattice structure [4], using the

* Corresponding author.
E-mail addresses: kashif_mahmood@hotmail.com (K. Mahmood), malek@kfupm.edu.sa (A. Zidouri), azzedine@kfupm.edu.sa (A. Zerguine).

1051-2004/$ – see front matter
doi:10.1016/j.dsp.2007.04.006

back propagation algorithm (BP) [5], give a significantly improved performance over the simple DFE [2]. Moreover, the MLP DFE with lattice structure [4] gives better performance over the MLP DFE, however, its high computational complexity still greatly limits the applications.

Since the back propagation algorithm is no more than a generalized least-mean squares (LMS) algorithm [6], it then suffers from the same problems that the LMS algorithm exhibits, particularly the relatively slow rate of convergence when applied to channels with spectral nulls in their frequency responses. These channels are known to yield a large eigenvalue spread (ES) of the autocorrelation matrix of the signal at their outputs.

To extend the applicability of the MLP DFE to areas involving fast time-varying channels, e.g., mobile communication channels, the recursive least squares (RLS) algorithm [6] rather than the back propagation algorithm is a necessary tool for the MLP DFE to be able to track variations in these environments. Moreover the effect of the eigenvalue spread, encountered in the BP algorithm, will be reduced substantially in both time-invariant and time-varying channels as the RLS algorithm is unaffected by this factor. Also, it is known in adaptive filtering that the RLS algorithm is typically an order of magnitude faster than the LMS algorithm [6]. Eventually this will have a great impact on the convergence behavior of the MLP DFE.

A couple of RLS adaptive algorithms [7–9] designed for the learning of MLP were proposed. These two algorithms were later investigated by Peng [10] in equalization where she reported the drawbacks of these algorithms. Among them is, e.g., the RLS adaptive algorithm proposed by Scalero [9] for the learning of MLP requires finding the inverse of activation function for each neuron which is a real heavy load on the algorithm.

Recently, a new RLS algorithm [11] is proposed in the learning of the MLP but so far it has only been applied for *XOR* function approximation problems. This algorithm, however, does not require finding the inverse of activation function and moreover approximate the values of true symbols. Consequently, the RLS algorithm proposed by [11], rather than the BP algorithm, is used here to update the MLP DFE even though the latter has a lower complexity than the former one.

One of the most powerful algorithms for the training of feedforward networks is undoubtedly the Levenberg–Marquardt (LM) method [12] which combines the excellent local convergence properties of Gauss–Newton method near a minimum with the consistent error decrease provided by (a suitably scaled) gradient descent far away from a solution. The Levenberg–Marquardt optimisation algorithm [13] applies, similarly as the conjugate gradient method [13], numerically estimated information from the second derivative of the cost function. A disadvantage of the LM method, however, is its increased memory requirements arising from the demand to calculate the Jacobian matrix of the error function and the need to invert matrices with dimensions equal to the number of the weights of the neural network. Another disadvantage originates from the fact that, since LM is an unconstrained optimization method, it is not guaranteed to converge to the global minimum of the cost function, but it is globally convergent in the sense that it is guaranteed to converge to a minimizer (local or global) of the cost function where the necessary and sufficient conditions for optimality hold. The increased memory requirements of the LM algorithm however render such a practice clearly unacceptable. Other newly derived techniques to enhance further the performance of the back propagation algorithm are found in [14,15].

In this work, the performance of the MLP DFE using the RLS algorithm [11] is evaluated and it will be called MLP(RLS)-DFE algorithm. It is shown that a great improvement in performance is obtained through the use of this technique over both the simple DFE based on the LMS algorithm and the MLP DFE in both time-invariant and time-varying channels.

The rest of the paper is organized as follows. In Section 2, a brief review of artificial neural networks is given. Section 3 reports the RLS algorithm [11] along with the proposed algorithm and the computational complexity of the proposed algorithm, while its performance is demonstrated by the simulation results of Section 4. Section 5 is concerned with the discussion of the results and conclusions.

2. Artificial neural networks

Because of the capabilities of artificial neural networks in efficiently modeling arbitrary nonlinearities, there has been recent interest in employing them in adaptive equalization for data communication channels [3,16,17]. In this case, the linear adaptive filter is replaced by a neural network. Different artificial neural network architectures such as multilayer perceptron, radial basis functions, and recurrent neural networks have all been proposed in the literature for channel equalization [18]. Among all these structures, the most commonly and widely-used is the MLP structure. The popularity of MLP-based equalizers is due in part to their computational simplicity, finite parameterization, stability, and smaller structure size for a particular problem as compared to other structures.

A multilayer perceptron consists of several hidden layers of neurons that are capable of performing complex, nonlinear mappings between the input and output layer. The hidden layers provide the capability to use the nonlinear sigmoid function to create intricately-curved partitions of space with complex nonlinear decision boundaries [19]. Furthermore, it has been shown that only three layers are needed by the MLP to generate these boundaries [20].

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 $[w_{1j}, w_{2j}, \ldots, w_{Nj}]$ 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 $v_h^{(p-1)}$ $(h = 1, 2, \ldots, N)$ from the $(p-1)$th layer and produces an output $v_j^{(p)}$, which is also a real scalar, expressed in the following way:

$$v_j^{(p)} = f_j\left(\sum_{h=1}^{N} w_{hj} v_h^{(p-1)} + I_j^{(p)}\right). \tag{1}$$

This output value $v_j^{(p)}$ serves also as input to the $(p+1)$th layer (next layer) to which the neuron is connected. In the above expression, $f_j(\cdot)$ represents the nonlinearity function. The most commonly used one in the perceptron is of the sigmoid type, defined as [20]

$$f_j(x) = \frac{1 - e^{-x}}{1 + e^{-x}}, \tag{2}$$

where $f_j(x)$ is always in the range $[-1, 1]$, $\forall x \in \mathbb{R}$ (the set of real numbers). The weights $\{w_{hj}\}$ and thresholds levels $\{I_j\}$ are updated during training [3].

The MLP did not receive much attention in applications until the introduction of the BP algorithm [12]. The BP algorithm was used in both linear equalizers [21] and nonlinear equalizers (DFE) [3], and it was found that in both cases, the neural network-based configuration outperformed its nonneural network-based counterpart. In this work, however, only the MLP DFE [3] will be considered as it is more advantageous than its linear counterpart.

2.1. Learning phase

In the BP algorithm, the output value is compared with the desired output, resulting in an error signal. This error signal is fed back through the network whose weights are adjusted to minimize this error. The increments used in updating the weights, Δw_{hj}, and threshold levels, ΔI_j, of the pth $(p \in [1, 2, \ldots, P])$ layer are updated, respectively, according to the following relations:

$$\Delta w_{hj}^{(p)}(i+1) = \eta \delta_j^{(p)} v_j^{(p-1)} + \alpha \Delta w_{hj}^{(p)}(i) \tag{3}$$

and

$$\Delta I_j^{(p)}(i+1) = \beta \delta_j^{(p)}, \tag{4}$$

where η is the learning gain, α is the momentum parameter, and β is the threshold level adaptation gain.

The error signal $\delta_j^{(p)}$ for layer p is calculated starting from the output layer P, as

$$\delta_j^{(P)} = \frac{(z_j - v_j^{(P)})(1 - v_j^{2(P)})}{2} \tag{5}$$

and is then recursively back propagated to lower layers $(p \in [1, 2, \ldots, P-1])$ according to

$$\delta_j^{(p)} = (1 - v_j^{2(p)}) \sum_l \frac{\delta_l^{(p+1)} w_{jl}^{(p+1)}}{2}, \tag{6}$$

where l is over all neurons above the neuron j in the $(p+1)$st layer and z_j is the desired output. To allow for a rapid learning, a momentum term, $\Delta w_{hj}^{(p)}(i)$, scaled by α, is used to filter out high frequency variations of the weight vector. Consequently, the convergence rate is much faster and the fast weight changes are smoothed out.

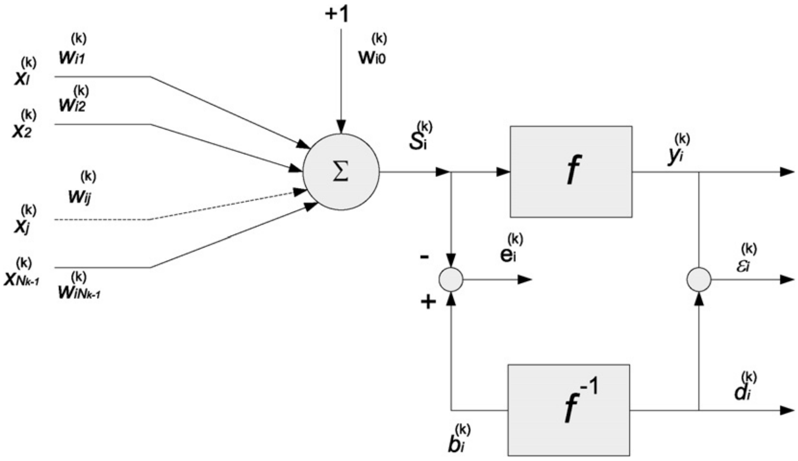

Fig. 1. Model of the ith neuron in the kth layer.

3. RLS algorithm

In the ensuing, a detailed description of the RLS algorithm developed in [11] used for the learning of the MLP in the context of nonlinear equalization (DFE) is reported in this work. For more details about this algorithm, the reader may refer to [11].

The error at the output layer of the neural network is calculated as

$$\varepsilon_i^L(n) = d_i^L(n) - y_i^L(n), \tag{7}$$

where $y_i^L(n)$ is the output of the output layer L, $d_i^L(n)$ and is the desired output or the target for the particular output i. For the rest of the layers, $1 \leqslant k \leqslant L-1$, the error is calculated in the following way:

$$\varepsilon_i^{(k)}(n) = \sum_{j=1}^{N_{k+1}} f'\big(s_j^{(k+1)}(n)\big) w_{ji}^{(k+1)}(n) \varepsilon_j^{(k+1)}(n), \tag{8}$$

where N_k is the number of neurons in the kth layer, f' is the derivative of the activation function already defined in (2) and reproduced here for more clarification:

$$f(x) = \frac{1-e^{-x}}{1+e^{-x}}, \tag{9}$$

where $s_j^{(k)}(n)$ is the linear output of the jth neuron in the kth layer, $w_{ji}^{(k)}(n)$ is the weight of ith neuron of the kth layer connecting this neuron with the jth input. Figure 1 depicts the model of the ith neuron in the kth layer. The weight vector of the ith neuron, $\mathbf{w}_i^{(k)}(n)$, is updated according to the following recursion:

$$\mathbf{w}_i^{(k)}(n) = \mathbf{w}_i^{(k)}(n-1) + \mathbf{g}_i^{(k)}(n)\varepsilon_i^k(n), \tag{10}$$

where $\mathbf{g}_i^{(k)}(n)$ is the Kalman gain and is calculated in the following way:

$$\mathbf{g}_i^{(k)}(n) = \frac{f'(s_j^{(k)}(n))\mathbf{P}_i^{(k)}(n-1)\mathbf{x}^{(k)}(n)}{\lambda + f'^2(s_j^{(k)}(n))\mathbf{x}^{(k)T}(n)\mathbf{P}_i^{(k)}(n-1)\mathbf{x}^{(k)}(n)}. \tag{11}$$

Finally, $\mathbf{x}^{(k)}(n)$ and λ are the vector of input signals for the kth layer and the forgetting factor in the RLS algorithm, respectively, and $\mathbf{P}_i^{(k)}(n)$ is the inverse of the correlation matrix defined by

$$\mathbf{P}_i^{(k)}(n) = \lambda^{-1}\big[\mathbf{I} - f'\big(s_j^{(k)}(n)\big)\mathbf{g}_i^{(k)}(n)\mathbf{x}^{(k)T}(n)\big]\mathbf{P}_i^{(k)}(n-1). \tag{12}$$

3.1. *Proposed scheme*

Because the BP algorithm is no more than a generalized least-mean squares (LMS) algorithm [6], it then suffers from the same problems as the LMS algorithm, particularly the slow rate of convergence when applied to channels with spectral nulls in their frequency responses. These channels are known to yield a large eigenvalue spread of the autocorrelation matrix of the signal at their outputs. This kind of channel characteristics is often encountered in time-variant channels. Moreover, as shown in [4], the BP-based MLP DFE has an equal performance to that of the simple DFE in time varying channels. The MLP DFE equalizer based on the BP algorithm can then hardly be justified for the equalization of such channels. Eventually, a moderate configuration in complexity and performance must be proposed to improve the performance of the MLP DFE based on the BP algorithm and as well having less complexity than that of MLP-DFE with lattice structure. Therefore, a compromise between complexity and performance must be reached in order to solve this conflict.

The RLS algorithm proposed by Bilski and Rutkowski [11] has not been used so far in the field of equalization. In this work, the RLS algorithm [11] is used to update the MLP instead of the BP algorithm. As assessed in the simulation results section, great improvement is obtained through the use of this technique in time-invariant and time-varying channels. Figure 2 details the MLP-based DFE where the RLS instead of the BP is used to train the MLP.

3.2. *Computational complexity of the algorithms*

Most common operations used in adaptive filtering algorithms are additions, multiplications and divisions. When digitally implemented, the last two operations are known to be more computationally costly than the first one and are very prone to causing instability specially in fixed-point computations. Even in systems equipped with floating-point arithmetic, these two operations can sometime lead to overflow, hence causing the algorithm to diverge. To avoid this problem of overflow, floating-point arithmetic is preferred. From a computational viewpoint and whenever possible, algorithms involving fewer multiplications and divisions should always be sought after, especially in applications involving tracking in fast-changing environments, e.g., wireless communication.

Table 1 gives the computational complexity of the three algorithms used in this work, namely, the simple DFE, the MLP DFE, and the MLP(RLS)-DFE. The notations used in Table 1 are defined as follows: N_1 is the number of feedforward taps, N_2 is the number of feedback taps, $(N_1 + N_2)$ is the number of inputs to the MLP, L_1 and L_2 are the numbers of neurons in the first and second hidden layers of the MLP, respectively, and finally $L_3 = 1$ is the number of neurons in the output layer of the MLP.

Table 1
Computational load for different adaptive DFE algorithms used in this work

	DFE	MLP DFE	MLP(RLS)-DFE
Total number of additions and multiplications	$2(N_1+N_2)+1$	$N_1+N_2+11L_1+17L_2$ $+5(N_1+N_2)\times L_1+6L_1\times L_2+9$	$16N_1+32N_2+11L_1+17L_2$ $+5(N_1+N_2)\times L_1+6L_1\times L_2-18$
Divisions	0	$2L_1+2L_2+2$	$(N_1+N_2)\times L_1+2L_1\times L_2$

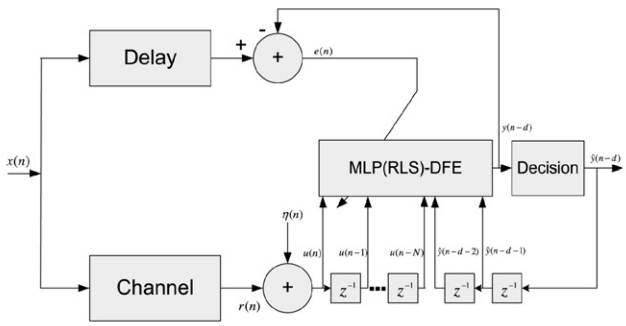

Fig. 2. Block diagram of MLP(RLS)-DFE.

From Table 1, we see that the LMS algorithm is the simplest of all and by using the RLS algorithm instead of the BP algorithm for the MLP DFE, a load of $(15N_1 + 31N_2 - 27)$ computations is added.

4. Simulation results

The performance of the new structure MLP(RLS)-DFE is compared to those of the LMS DFE (using LMS algorithm) and MLP DFE. Both the LMS DFE and the MLP DFE structures use four samples in the feedforward section and one sample in the feedback section. For the latter structure, this results in five input samples in its input layer. For the MLP DFE the number of neurons in the first and second hidden layer, and the output layer are 9, 3, and 1, respectively, whereas for the MLP-RLS DFE the number of neurons in the first and the output layers are 9 and 1, respectively. This is made so that both structures have comparable complexity. The back propagation algorithm is used to update the MLP DFE where the learning gain parameter η, momentum parameter α, and threshold level adaptation gain β, have been chosen as in [3], namely 0.07, 0.3, and 0.05, respectively. The step size for the LMS algorithm used in the LMS DFE is 0.035. The forgetting factor in the RLS algorithm, λ, is chosen to be 0.98. The digital message applied to the channel is made of uniformly distributed bipolar random numbers $(-1, 1)$. The channel noise is taken to be

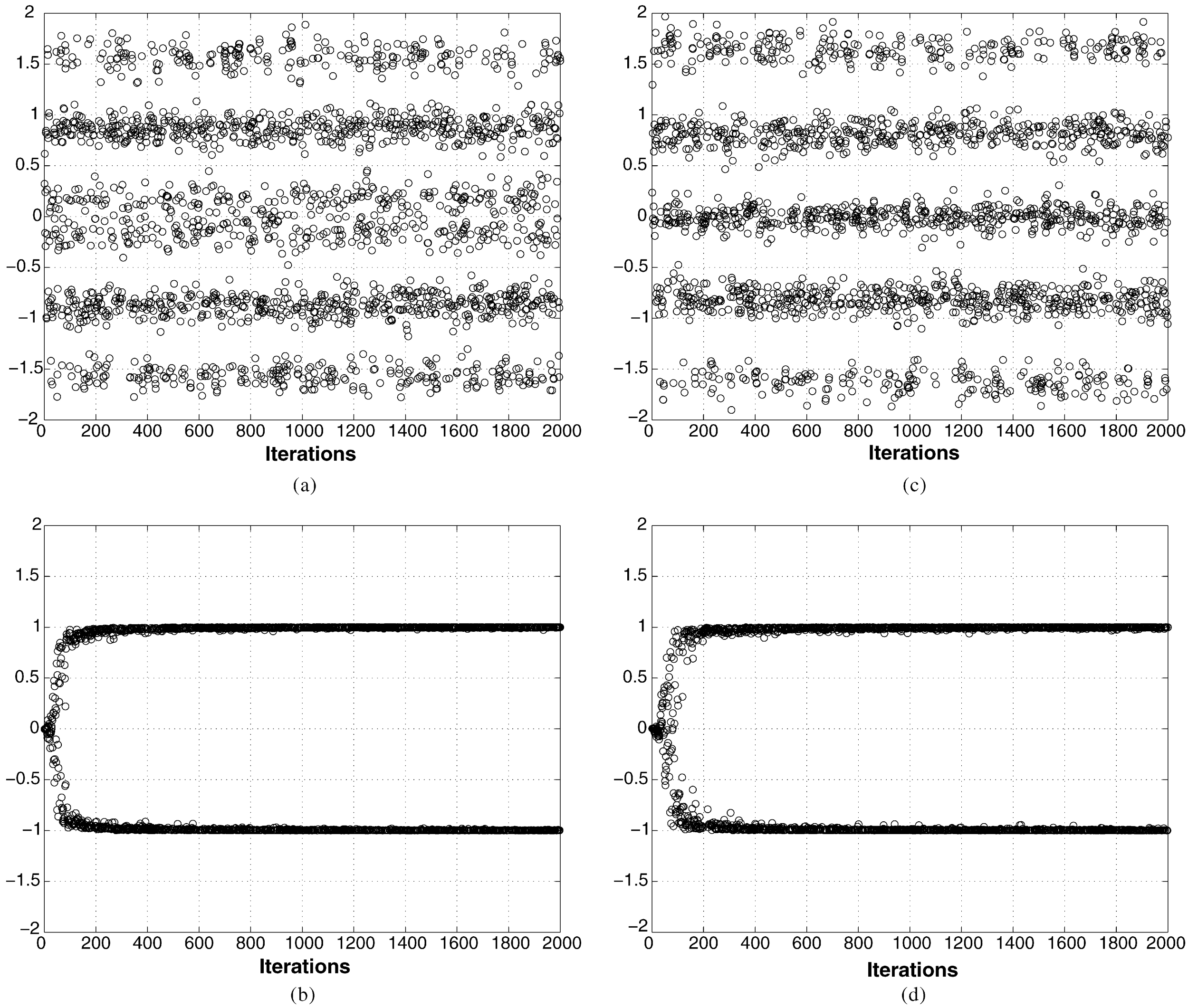

Fig. 3. Signal constellation: (a) at the input of the equalizer for channel $H_1(z)$ under SNR = 10 dB; (b) after equalization for channel $H_1(z)$; (c) at the input of the equalizer for channel $H_2(z)$ under SNR = 10 dB; (d) after equalization for channel $H_2(z)$.

additive white Gaussian noise. The system performance will be evaluated using both time-invariant and time-varying channels, as detailed next.

4.1. System performance in time-invariant channels

To assess the performance of the proposed structure, two time-invariant channel models are used in the simulation and are described by their transfer functions $H_1(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$ and $H_2(z) = 0.408 + 0.816z^{-1} + 0.408z^{-2}$ with eigenvalue spreads of 25 and 81, respectively. Also, it should be pointed out here that the first channel model matches the one used in [3], while the second channel matches one of the time-invariant channel models used in [22]. During this part of the simulations, the performance measure is obtained through the use of signal constellations, eye diagrams and learning curves.

Signal constellations and eye diagrams are plotted for channel $H_1(z)$ and $H_2(z)$. Figures 3 and 4 show these diagrams for the MLP(RLS)-DFE equalizer for first channel and second channel, respectively, and with a signal to noise ratio (SNR) of 10 dB. In Fig. 3, parts (a) and (c) show the unequalized data for channels $H_1(z)$ and $H_2(z)$, respectively, while parts (b) and (d) depict their respective equalized versions. A huge difference in performance, as can be observed from these diagrams, is obtained through the use of the proposed algorithm. Figure 4 depicts the performance of the MLP (RLS)-DFE as far as the eye diagram is concerned. In summary, the equalizer's performance

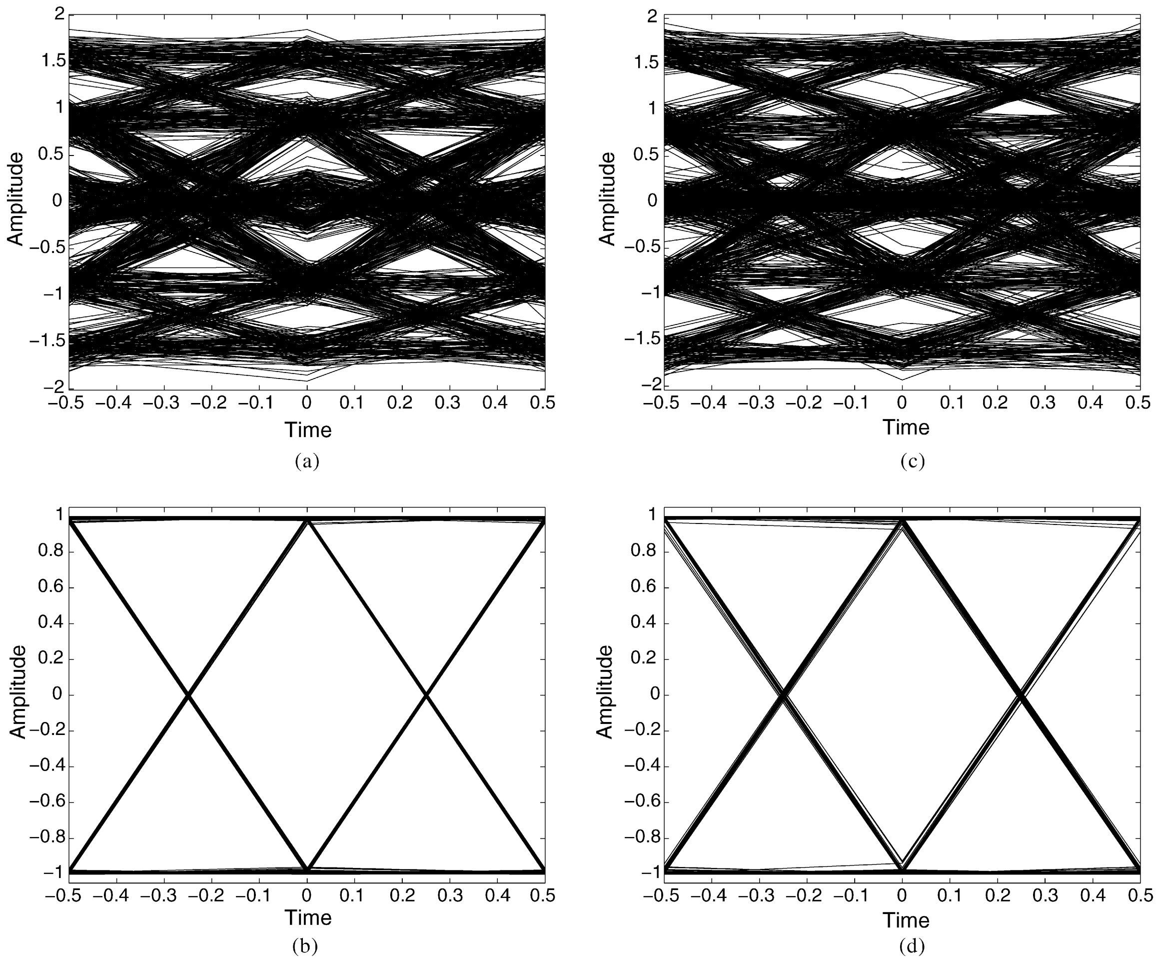

Fig. 4. Eye diagram: (a) at the input of the equalizer for channel $H_1(z)$ under SNR = 10 dB; (b) after equalization for channel $H_1(z)$; (c) at the input of the equalizer for channel $H_2(z)$ under SNR = 10 dB; (d) after equalization for channel $H_2(z)$.

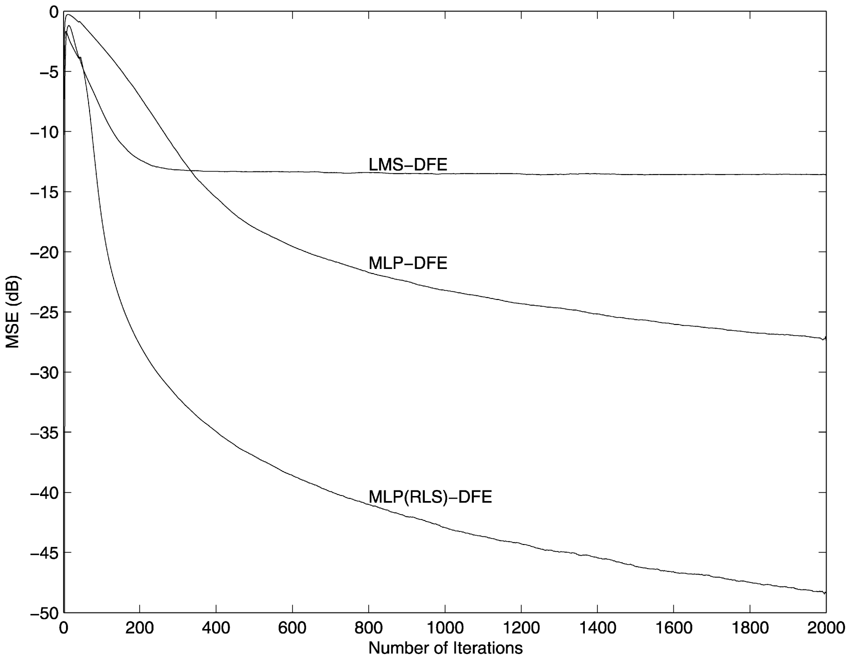

Fig. 5. Learning curves of different types of equalizers with channel $H_1(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$.

is much improved and the symbols are seen to converge closer to their original positions when the MLP (RLS)-DFE is used. The positive effect of the RLS algorithm is clear.

In the case of the learning curves, these are obtained by averaging 600 independent runs. Each run has a different random sequence and random starting weights for the perceptron structure, and an SNR of 20 dB is used. Figure 5 depicts the convergence behavior of the three algorithms for the first channel. This figure shows a clear improvement in both the convergence time and the steady-state MSE when the MLP (RLS)-DFE algorithm is deployed. This result illustrates also that even though the MLP DFE configuration converges more slowly than that of the simple DFE, it nevertheless results in a lower steady-state MSE value than that of the latter. It should also be clear from this figure that the steady-state MSE of both MLP DFE configurations is below the noise level. This results from the nonlinear nature of the equalizer transfer function [3]. Furthermore, the MLP DFE equalizer is capable of generating highly nonlinear decision regions, in contrast to the LMS DFE equalizer which only forms a hyperplane decision boundary [17]. Also, it is generally understood in linear signal processing that the l_2 norm error criterion produces a parabolic error surface with no local minima and has a continuously differential nature. However, the l_2 norm error criterion for perceptrons will not generally produce a parabolic error surface owing to its nonlinear nature [23]. Similarly here, using the same reasoning one can reach the conclusion that the least squares error criterion for perceptrons, which the RLS algorithm is a member, is unlikely to produce a parabolic error surface.

A similar improvement is also obtained for the second channel despite its larger eigenvalue spread, as shown in Fig. 6. The difference in convergence time between the MLP DFE and the MLP DFE using the RLS algorithm is now more pronounced. The insensitivity of the RLS algorithm to the eigenvalue spread is very clear.

To further investigate the consistency in performance of the MLP(RLS)-DFE in the past scenario, its computational complexity is reduced almost by 50%, the number of neurons in the first layer is now 5 instead of 9. Again as depicted in Figs. 7 and 8, the performance of MLP(RLS)-DFE is still better than those of the LMS DFE and the MLP DFE. This is indicated in the figures by the label MLP(RLS)-DFE (5) to differentiate it from the MLP(RLS)-DFE (9) (the original configuration shown in Figs. 5 and 6).

4.2. System performance in time-varying channels

In this second part of the simulations, a time-variant channel is used to evaluate the capability of the equalizer to track the changes in a time-varying dispersive channel. The discrete-time channel model for time-varying channel is

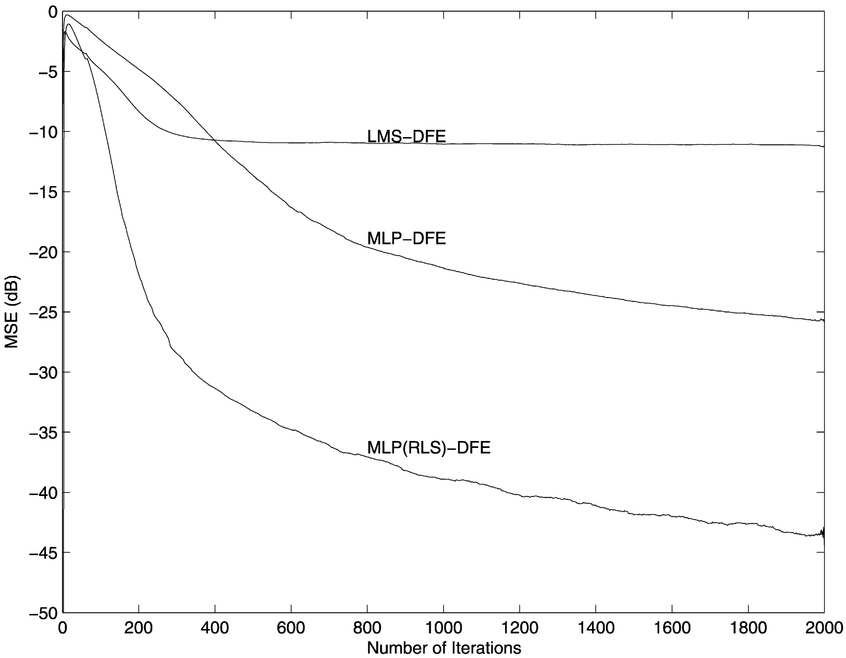

Fig. 6. Learning curves of different types of equalizers with channel $H_2(z) = 0.408 + 0.816z^{-1} + 0.408z^{-2}$.

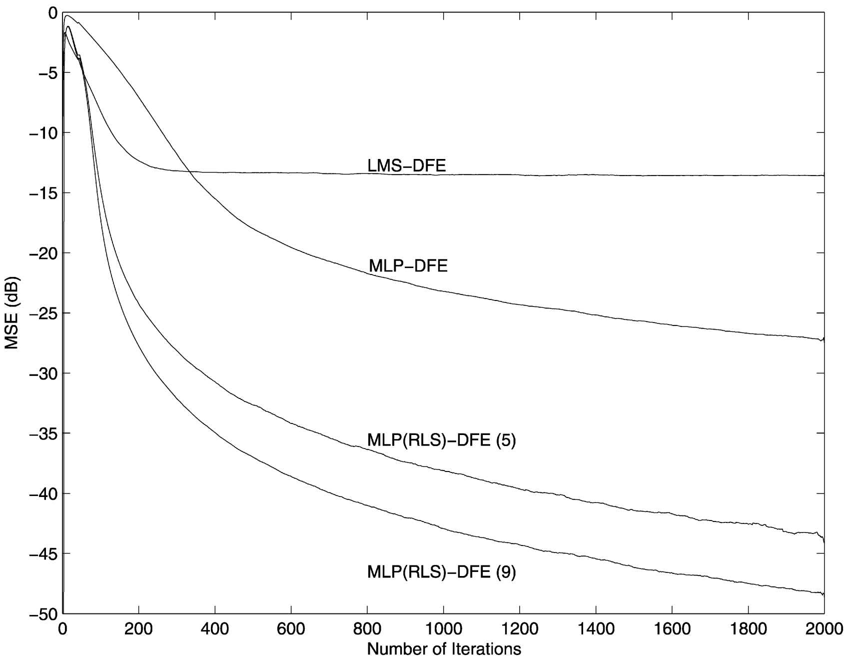

Fig. 7. Learning curves of different types of equalizers with channel $H_1(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$.

described by the following transfer function $H(z) = a_0(t) + a_1(t)z^{-1} + a_2(t)z^{-2}$, where $a_0(t)$, $a_1(t)$, and $a_2(t)$ are the time-varying coefficients of the channel impulse response. These are generated by passing white Gaussian noise through a low-pass filter of a specified bandwidth [24]. If we assume that we have a nominal 3 kHz HF channel, the signaling rate is 2400 symbols/s, and the low-pass filter is a two-pole Butterworth filter, then the 3-dB bandwidth of

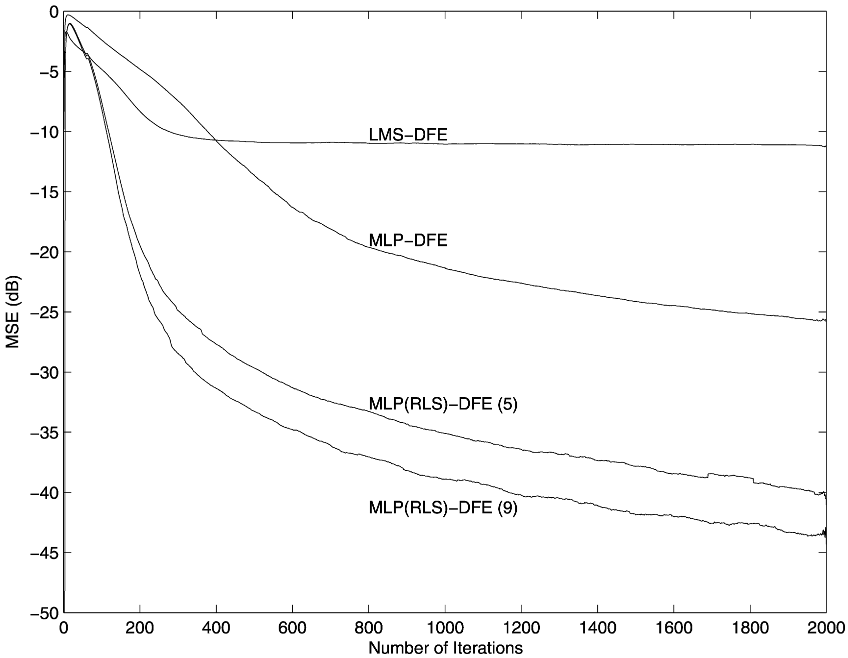

Fig. 8. Learning curves of different types of equalizers with channel $H_2(z) = 0.408 + 0.816z^{-1} + 0.408z^{-2}$.

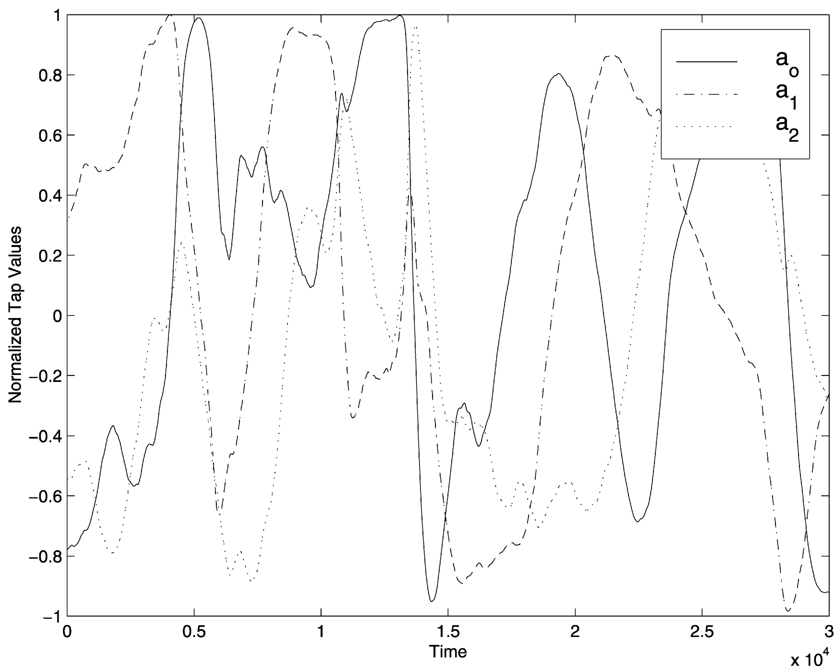

Fig. 9. Tap coefficients of time-variant channel with lowpass filter bandwidth of 0.5 Hz.

the low-pass filter can be used as a parameter to control the rate of variation of the channel impulse response. The curves representing the time variation of the coefficients are depicted in Fig. 9 for bandwidth of 0.5 Hz.

Figures 10 and 11 show the BER performance of the three equalizers for the time variations of the coefficients for bandwidths of 0.1 and 0.5 Hz, respectively. The results illustrate the superiority of the MLP(RLS)-DFE. In both

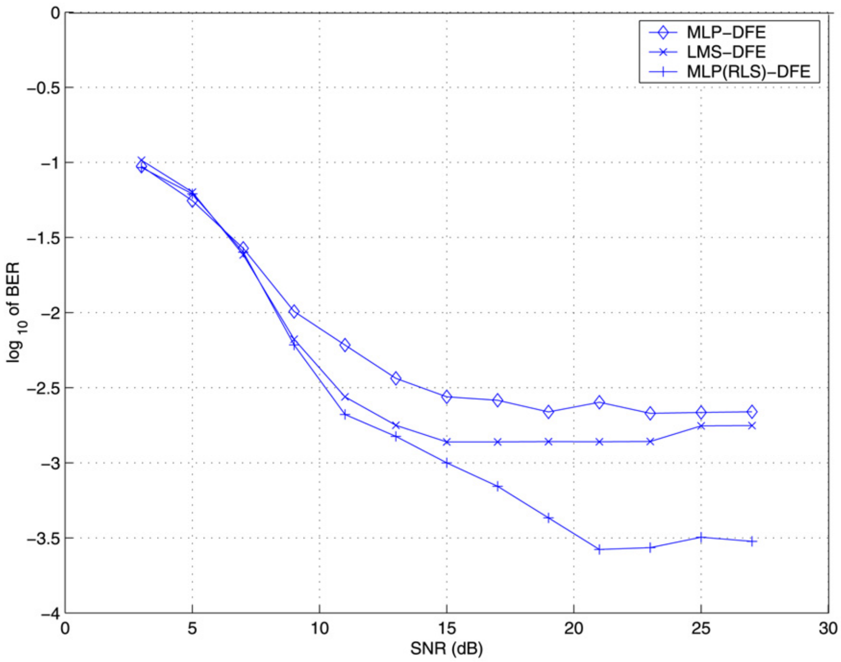

Fig. 10. BER performance of different types of equalizers for time varying channel with lowpass filter bandwidth of 0.1 Hz.

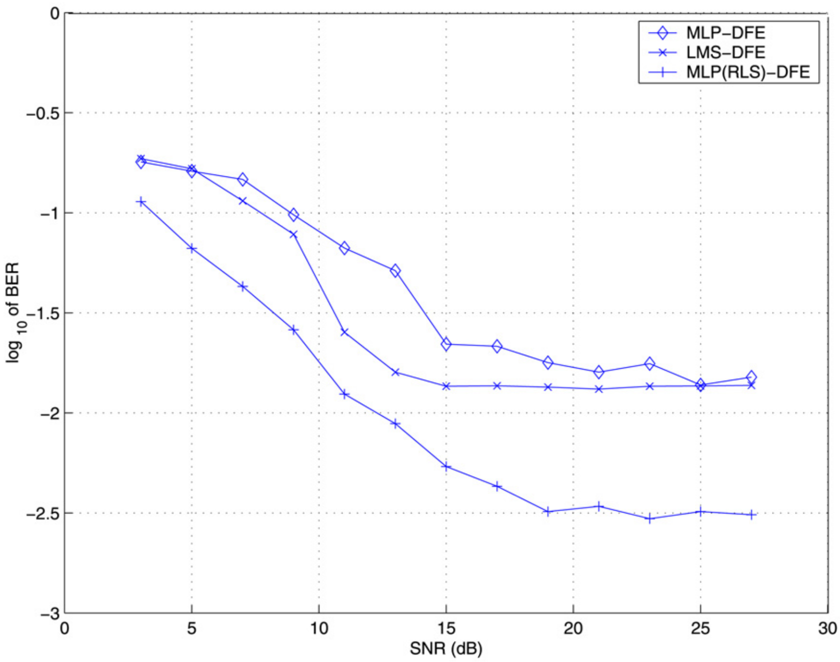

Fig. 11. BER performance of different types of equalizers for time varying channel with lowpass filter bandwidth of 0.5 Hz.

figures, the other two equalizers' performances are not attractive. Moreover, as can be noticed from these figures that comparable performance is obtained for both MLP DFE and LMS DFE configurations.

To further investigate the consistency in performance of the MLP(RLS)-DFE, the rate of variation of the channel impulse response is increased. This is done by increasing the bandwidth of the low-pass filter, e.g., bandwidth of 1.0 Hz. There is a deterioration in the performance of both the LMS DFE and the MLP DFE and the difference

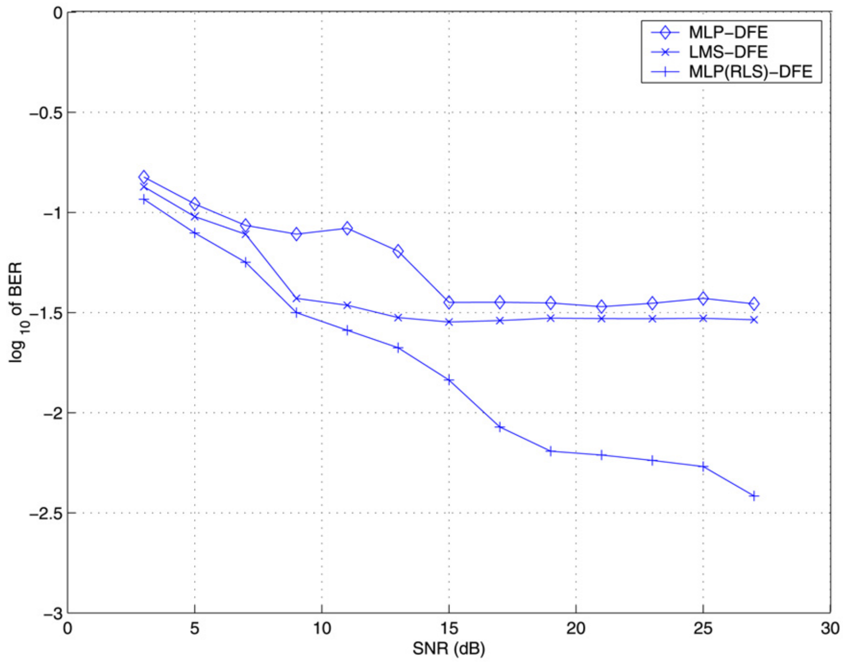

Fig. 12. BER performance of different types of equalizers for time varying channel with lowpass filter bandwidth of 1.0 Hz.

between them and the MLP(RLS)-DFE is increased as clearly shown in Fig. 12. The MLP(RLS)-DFE attains lower error floor than both of the LMS DFE and the MLP DFE. The LMS DFE and the MLP DFE saturate after the SNR of 13 dB to approximately the same BER. Again in this case the MLP DFE and the LMS DFE have comparable performance.

The saturation effect (error-floor) for both LMS DFE and MLP DFE in Figs. 10–12 is due mainly to when SNR is larger than 15 dB, noise has little influence on the BER, because at that time, errors are mainly caused by tap-gain lag due to the variation of channels, this is depicted in Figs. 10–12.

Finally, as was detailed in Section 3.2, the computational load of the MLP(RLS)-DFE is larger by far than those of the LMS DFE and MLP DFE. Therefore, it was made sure that both MLP configurations have similar computational load such that a fair comparison is made. Moreover, even with almost 50% decrease in computational complexity, the performance of the MLP (RLS)-DFE is much better than those of the LMS DFE and MLP DFE.

5. Conclusion

This work has presented the improvements brought about by the RLS algorithm used for the first time in equalization to train the MLP instead of the BP algorithm. The computer simulations have illustrated the better performance of the MLP(RLS)-DFE over both the MLP DFE and the LMS DFE in time-invariant and time-varying channels. The results of our study can be summarized briefly as follows:

1. The use of the RLS algorithm instead of the back propagation algorithm for the MLP DFE results in substantial improvements in terms of convergence rate, steady-state MSE and BER.
2. The convergence rate of the MLP(RLS)-DFE is insensitive of the eigenvalue spread of the channel correlation matrix for time-invariant channels.
3. The proposed MLP(RLS)-DFE has better BER performance than both the LMS DFE and MLP DFE in time-varying channels.
4. The simulation results indicated that the MLP (RLS)-DFE is stable and no sign of instability was shown for both time-invariant and time-varying channels.
5. Further work aimed at applying the proposed scheme with respect to nonlinearities is currently being pursued.

6. Finally, the improved performance of the proposed structure is obtained without any additional increase in complexity.

Acknowledgment

The authors would like to acknowledge the support of KFUPM.

References

[1] S. Qureshi, Adaptive equalization, Proc. IEEE 73 (9) (1985) 1349–1387.
[2] J.G. Proakis, Digital Communications, McGraw–Hill, New York, 1989.
[3] S. Siu, G.J. Gibson, C.F.N. Cowan, Decision feedback equalization using neural network structures and performance comparison with standard architecture, IEE Proc. 137 (1990) 221–225.
[4] A. Zerguine, A. Shafi, M. Bettayeb, Multilayer perceptron-based DFE with lattice structure, IEEE Trans. Neural Networks 12 (6) (2001) 532–545.
[5] R.P. Lippmann, An introduction to computing with neural nets, IEEE ASSP Mag. 4 (1987).
[6] S. Haykin, Adaptive Filter Theory, Prentice Hall, Englewood Cliffs, NJ, 1991.
[7] S. Azimi, R.J. Liou, Fast learning process of multilayer neural networks using recursive least squares method, IEEE Trans. Signal Process. 40 (2) (1992) 446–450.
[8] J. Coloma, R.A. Carrasco, MLP equalizer for frequency selective time-varying channels, Electron. Lett. 30 (6) (1994) 503–504.
[9] R.S. Scalero, A fast new algorithm for training feedforward neural networks, IEEE Trans. Signal Process. 40 (1992) 202–210.
[10] M. Peng, Neural networks applications in linear and nonlinear channel equalization, Ph.D. dissertation, Northeastern University, Boston, MA, June 1994.
[11] J. Bilski, L. Rutkowski, A fast training algorithm for neural networks, IEEE Trans. Circuits Syst. II 45 (6) (1998) 749–753.
[12] D.E. Rumelhart, J.L. McClelland, Parallel Distribution Processing: Explorations in the Microstructure of Cognition, vol. 1, MIT Press, Cambridge, MA, 1986.
[13] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes, second ed., Cambridge Univ. Press, Cambridge, 1992.
[14] S. Halgamuge, L. Wang (Eds.), Computational Intelligence for Modeling and Predictions, Springer-Verlag, Berlin, 2005.
[15] L. Wang (Ed.), Soft Computing in Communications, Springer-Verlag, Berlin, 2004.
[16] S. Chen, G.J. Gibson, C.F.N. Cowan, P.M. Grant, Adaptive equalization of finite nonlinear channels using multilayer perceptrons, Signal Process. 20 (1990) 107–119.
[17] G.J. Gibson, S. Siu, C. Cowan, The application of nonlinear structures to the reconstruction of binary signals, IEEE Trans. Signal Process. 39 (1991) 1877–1884.
[18] B. Mulgrew, Applying radial basis functions, IEEE ASSP Mag. 13 (1996) 50–65.
[19] A. Wieland, R. Leighton, Geometric analysis of neural network capabilities, in: 1st International Conference on Neural Networks, June 1987, pp. 385–393.
[20] S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan Co., New York, 1994.
[21] G.J. Gibson, S. Siu, C. Cowan, Multilayer perceptron structures applied to adaptive equalizers for data communications, IEEE Proc. ICASSP, Glasgow, May 1989, pp. 1183–1186.
[22] F. Ling, J.G. Proakis, Adaptive lattice decision-feedback equalizers: Their performance and application to time-variant multipath channels, IEEE Trans. Commun. COM 33 (1985) 348–356.
[23] S. Siu, C.F.N. Cowan, Performance analysis of the l_p norm back propagation algorithm for adaptive equalisation, IEE Proc. F 140 (1) (1993) 43–47.
[24] F. Ling, J.G. Proakis, Lattice decision feedback equalizers and their application to fading dispersive channels, in: Proc. Int. Conf. Comm., Boston, MA, June 1983, pp. C8.2.1–C8.2.5.

Kashif Mahmood received the B.Sc. degree from Pakistan in 1997, and the M.Sc. degree from King Fahd University of Petroleum & Minerals (KFUPM), Dhahran, Saudi Arabia, in 2000, in electrical engineering. Since 2000, he is a Lecturer at Hafr Al-Batin Community College. Mr. Mahmood research interests include signal processing for communications, neural networks, and local area networks.

Dr. Abdelmalek Zidouri is a Senior IEEE member and an Assistant Professor in the Department of Electrical Engineering at King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. He holds a Master of Science in control engineering from Bradford University, UK, in 1984, and a Doctor of Engineering in applied electronics from Tokyo Institute of Technology Japan, in 1995. His research interests are in the field of signal processing and pattern recognition. In particular character recognition and document image analysis. Dr. Zidouri has published many refereed journal and conference papers. He has supervised many senior projects and M.Sc. theses.

He is currently the head of the Digital Signal Processing Group at the EE Department. He is a member of Engineering Education Society, Signal Processing Society, and Communications Society.

Azzedine Zerguine received the B.Sc. degree from Case Western Reserve University, Cleveland, OH, in 1981, the M.Sc. degree from King Fahd University of Petroleum & Minerals (KFUPM), Dhahran, Saudi Arabia, in 1990, and the Ph.D. degree from Loughborough University, Loughborough, UK, in 1996, all in electrical engineering. From 1981 to 1987, he was working with different Algerian state owned companies. During the period from 1987 to 1990, he was a research and teaching assistant, Electrical Engineering Department, KFUPM. Dr. Zerguine is presently an Associate Professor, Electrical Engineering Department, KFUPM, working in the areas of signal processing and communications. Dr. Zerguine is a Senior Member of IEEE and currently he is serving as an associate editor with the EURASIP Journal on Advances in Signal Processing.

Dr. Zerguine research interests include signal processing for communications, adaptive filtering, neural networks, multiuser detection, and interference cancellation.