ELSEVIER

Signal Processing 47 (1995) 145–158

SIGNAL PROCESSING

A polynomial-perceptron based decision feedback equalizer with a robust learning algorithm

Ching-Haur Chang[a], Sammy Siu[b], Che-Ho Wei[a,*]

[a] *Institute of Electronics, National Chiao-Tung University, Hsin Chu, Taiwan 300, ROC*
[b] *Telecommunication Laboratories, MOTC, P.O. Box 71, Chung Li, Taiwan 320, ROC*

Received 3 December 1993; revised 1 July 1994, 10 January 1995 and 27 July 1995

Abstract

A new equalization scheme, including a decision feedback equalizer (DFE) equipped with polynomial-perceptron model of nonlinearities and a robust learning algorithm using l_p-norm error criterion with $p < 2$, is presented in this paper. This equalizer exerts the benefit of using a DFE and achieves the required nonlinearities in a single-layer net. This makes it easier to train by a stochastic gradient algorithm in comparison with a multi-layer net. The algorithm is robust to aberrant noise for the addressed equalizer and, hence, converges much faster in comparison with the l_2-norm. A detailed performance analysis considering possible numerical problem for $p < 1$ is given in this paper. Computer simulations show that the scheme has faster convergence rate and satisfactory bit error rate (BER) performance. It also shows that the new equalizer is capable of approaching the performance achieved by a minimum BER equalizer.

Zusammenfassung

Ein neues Entzerrungsverfahren wird in diesem Beitrag vorgestellt. Es schließt einen Entzerrer mit Entscheidungsrückführung ein, der mit einem Polynom-Perceptronmodell für Nichtlinearitäten und einem robusten Lernalgorithmus ausgestattet ist, welcher mit einem l_p-Norm-Kriterium mit $p < 2$ arbeitet. Dieser Entzerrer nutzt den Vorteil der Entscheidungsrückkopplung und erzielt die gewünschten Nichtlinearitäten in einem Netzwerk mit einer Ebene. Gegenüber einem mehrlagigen Netzwerk wird so das Training durch ein stochastisches Gradientenverfahren erleichtert. Der Algorithmus ist beim angesprochenen Entzerrer robust gegenüber Fehlerrauschen und konvergiert daher viel schneller als bei Verwendung der l_2-Norm. Die Leistungsfähigkeit wird bezüglich möglicher numerischer Probleme für $p < 1$ im einzelnen analysiert. Rechnersimulationen zeigen die höhere Konvergenzgeschwindigkeit und zufriedenstellende Bitfehlerrate. Es zeigt sich außerdem, daß der neue Entzerrer fähig ist, die Leistungsfähigkeit einer Entzerrung mit minimierter Bitfehlerrate zu erreichen.

Résumé

Cet article présente un nouveau schéma d'égalisation incluant un équaliseur à décision par retour arrière (DFE) équipé avec une modélisation des non-linéaritées par modèle de perceptron polynomial et un algorithme d'apprentissage robuste utilisant la norme l_p comme critère d'erreur (avec $p < 2$). Cet égaliseur exploite les avantages d'une DFE et remplit les conditions de non-linéarité requisent par un réseau monocouche. Ce dernier point permet un

[*] Corresponding author.

SSDI 0165-1684(95)00103-4

apprentissage par un algorithme de gradient stochastique plus facile par comparaison à un réseau multi-couche. L'algorithme est robuste au bruit aberrant dans le cas de l'équaliseur utilisé, et, de plus, il converge plus rapidement que la norme l_2. Une analyse détaillée des performances tenant compte des problèmes numériques lorsque $p < 1$ est donnée dans cet article. Les résultats de simulation montrent que le schéma a une vitesse de convergence plus rapide et des performances en terme de taux d'erreur par bit satisfaisant. Il montre également que ce nouvel égaliseur est capable d'approcher les performances d'un égaliseur à BER minimum.

Keywords: Robust learning algorithm; Polynomial-perceptron based DFE; l_p-norm error criterion

1. Introduction

Adaptive equalization is an important technique to combat intersymbol interference (ISI) in digital communication systems. The equalizer has the task to recover the transmitted sequences from the received signal. It is well understood that optimal performance can be obtained by detecting the entire transmitted sequence using the maximum likelihood sequence estimator (MLSE) [7, 15]. However, the computation complexity and memory requirement of the MLSE make it impractical in some real-time applications. A more practical approach is to use symbol-by-symbol-decision linear transversal equalizer. However, it is known that in the signal space only linear decision boundaries [14] can be formed by a linear equalizer. An optimal symbol-by-symbol-decision equalizer must realize some nonlinear functions such that it can partition the signal space with nonlinear decision boundaries [4, 5, 9]. Recently, nonlinear equalizers equipped with multi-layer perceptron (MLP) model of nonlinearities were proposed by Gibson et al. [9] and Siu et al. [23]. Their results indicated that the optimal decision boundaries must be nonlinear and the MLP can approach the required nonlinearities. In addition, nonlinear approaches equipped with different models of nonlinearities such as using a polynomial-perceptron structure (PPS) [4], a radial basis function [5] and using a functional-link net [8] were demonstrated to have similar capabilities in partitioning the signal space.

In this paper, we focus on a new nonlinear equalizer in the following two respects. First, a single-layer nonlinear decision feedback equalizer (DFE) equipped with polynomial-perceptron model of nonlinearities is developed, Second, an l_p-norm [21] based learning algorithm suitable for the addressed structure is investigated. The structure exerts the benefit of using a DFE and achieves the required nonlinearities in a single-layer net. This is advantageous since it is much easier to train by a stochastic gradient algorithm. The algorithm using l_p-norm error criterion with $p < 2$ can be robust to aberrant noise when the distribution of error signals is more or less known a priori. The new equalizer has error signals lying in an interval suitable for applying the l_p-norm error criterion. A performance analysis of the l_p-norm back propagation (BP) algorithm [20] with $p \geqslant 1$ has been conducted by Siu et al. [22] for an MLP equalizer. A detailed performance analysis with a consideration on the possible numerical problem arising when $p < 1$ is given in this paper. Computer simulations show that the new equalizer is attractive in both convergence rate and BER performance and has a performance close to that achieved by a minimum BER equalizer.

The paper is organized as follows. Section 2 describes the polynomial-perceptron based DFE. Section 3 presents the l_p-norm tap-weight updating algorithm for the proposed structure. A performance analysis of the l_p-norm algorithm with a numerical stability consideration is given in Section 4. Computer simulation results are given in Section 5. A comparison of the new equalizer with other nonlinear equalizers is given in Section 6.

2. Polynomial-perceptron based DFE

The capability of a single-layer net is limited since only linear decision boundaries can be formed in the pattern space [16]. Although the use of a multi-layer net [20, 24] can achieve the required nonlinearities, the high complexity of the multi-layer net precludes its use in many applications. It is known that functions realized by hidden layer nodes in a multi-layer net can also be realized or approximated by a single-layer net provided that sufficient orders of nonlinear links are incorporated into the node [17]. For example, consider a set of components described by a vector X_1, the required nonlinearities can be achieved by a series of higher-order expansion as

$$X_1 \rightarrow X_2 = [X_1] \times [1, X_1] \rightarrow X_3 = [X_2] \times [1, X_2] \rightarrow \cdots, \tag{1}$$

where × denotes outer product operation. This expansion introduces higher-order terms, together with the first-order terms given by X_1, to represent the information. Some advantages through the use of the outer product model can be found in [17]. A perceptron with nonlinear links obtained by using (1) on the input is called a polynomial perceptron [4].

Let L and M denote, respectively, the number of taps in the feedforward and the feedback filters of a DFE. The signal vector in a DFE can be represented as

$$X(n) = [X_{\mathrm{F}}(n), X_{\mathrm{B}}(n)]^{\mathrm{T}}, \tag{2}$$

with

$$X_{\mathrm{F}}(n) = [x_{\mathrm{f}}(n), x_{\mathrm{f}}(n-1), \ldots, x_{\mathrm{f}}(n-L+1)], \tag{3}$$

$$X_{\mathrm{B}}(n) = [x_{\mathrm{b}}(n-1), x_{\mathrm{b}}(n-2), \ldots, x_{\mathrm{b}}(n-M)], \tag{4}$$

where $X_{\mathrm{F}}(n)$ and $X_{\mathrm{B}}(n)$ denote, respectively, the feedforward and the feedback signal vectors. In a DFE, the feedforward and feedback signals can be highly correlated [2]. This fact can be seen from the cross-correlation between $X_{\mathrm{F}}(n)$ and $X_{\mathrm{B}}(n)$ given as

$$E[X_{\mathrm{F}}^{\mathrm{T}}(n)X_{\mathrm{B}}(n)] = E\begin{bmatrix} x_{\mathrm{f}}(n)x_{\mathrm{b}}(n-1) & x_{\mathrm{f}}(n)x_{\mathrm{b}}(n-2) & \cdots & x_{\mathrm{f}}(n)x_{\mathrm{b}}(n-M) \\ x_{\mathrm{f}}(n-1)x_{\mathrm{b}}(n-1) & x_{\mathrm{f}}(n-1)x_{\mathrm{b}}(n-2) & \cdots & x_{\mathrm{f}}(n-1)x_{\mathrm{b}}(n-M) \\ \vdots & \vdots & \vdots & \vdots \\ x_{\mathrm{f}}(n-L+1)x_{\mathrm{b}}(n-1) & x_{\mathrm{f}}(n-L+1)x_{\mathrm{b}}(n-2) & \cdots & x_{\mathrm{f}}(n-L+1)x_{\mathrm{b}}(n-M) \end{bmatrix}. \tag{5}$$

Notice that the elements in (4) are related to the equalizer delay d by

$$x_{\mathrm{b}}(n) = s(n-d) + v(n), \tag{6}$$

where $s(n)$ is the transmitted signal and $v(n)$ is the decision error. Moreover, it is reasonable to assume that the decision error is independent of the feedforward signal. Consequently, the elements in (5) become

$$E[x_{\mathrm{f}}(n-i)x_{\mathrm{b}}(n-j)] = E[x_{\mathrm{f}}(n-i)s(n-d-j)], \tag{7}$$

with $i = 0, 1, \ldots, L-1$ and $j = 1, 2, \ldots, M$. We can see from (7) that the elements in (5) depend solely on the channel response. The polynomial-perceptron based DFE shown in Fig. 1(a) is denoted as PPDFE-I, where the threshold term for the perceptron is treated as an input of value unity ($u_0 = 1$) with tap-weight $w_0(n)$. The DFE shown in Fig. 1(b), denoted as PPDFE-II, uses the same higher-order expansion but only on the feedforward part of $X(n)$. The number of taps needed for an ith-order expansion can be found by

$$N = \sum_{k=0}^{i} n_k, \tag{8}$$

with $n_0 = 1$ and $n_k = n_{k-1}(L + M + k - 1)/k$, $k = 1, 2, \ldots, i$ [4], for PPDFE-I, while

$$N = \sum_{k=0}^{i} n_k + M, \tag{9}$$

with $n_0 = 1$ and $n_k = n_{k-1}(L + k - 1)/k$, $k = 1, 2, \ldots, i$, for PPDFE-II. Given the same L and M, the number of taps in PPDFE-I is larger than that of the PPDFE-II for a given order of expansion. For example, if $L + M = 5$ and $i = 3$, the PPDFE-I has 56 taps, while the PPDFE-II has

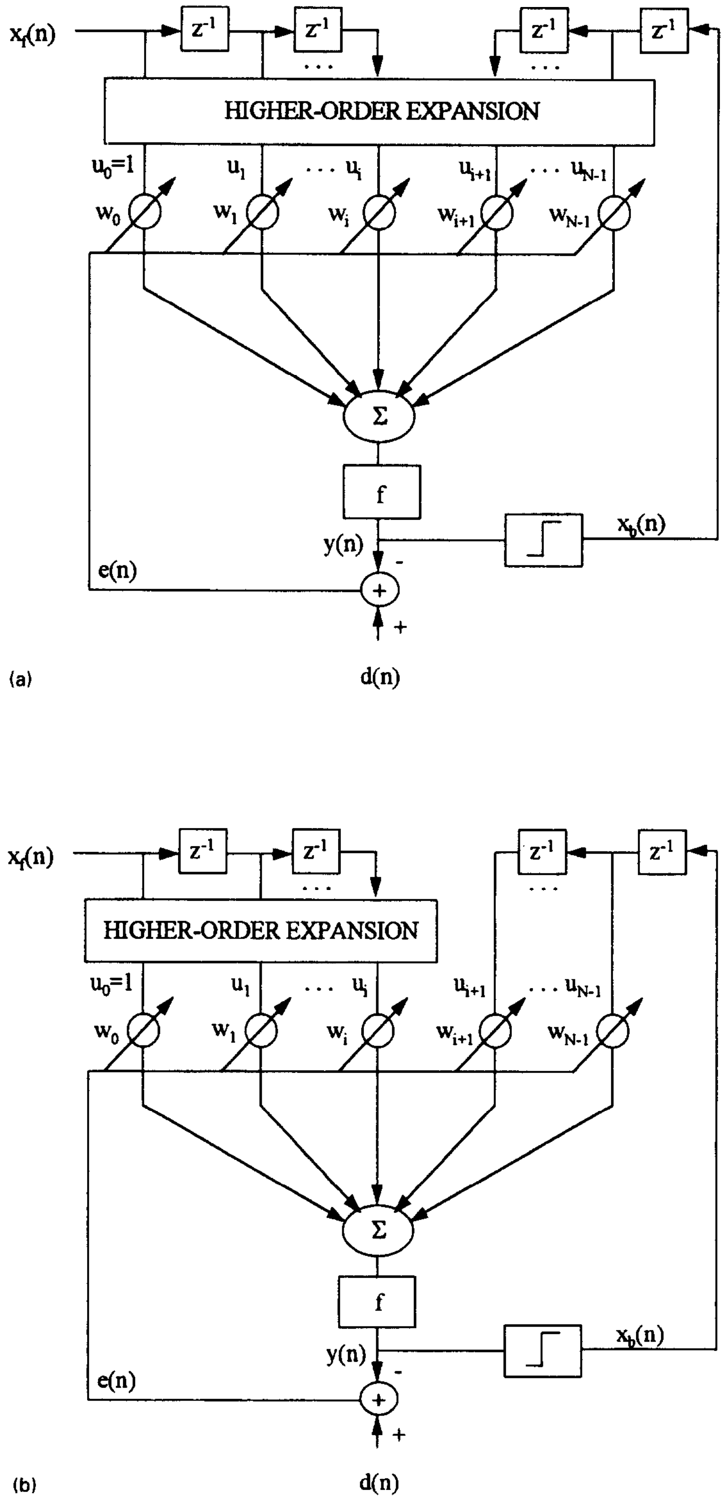

Fig. 1. Two polynomial-perceptron based DFEs: (a) PPDFE-I; (b) PPDFE-II.

36 taps. This is the price paid by the PPDFE-I for its better performance. The number of taps grows exponentially with increasing order of expansion. However, a small order such as 3 or 5 is often adequate to achieve required nonlinearities encountered in practical applications [4]. Furthermore, higher-order terms for which the correlation is insignificant can be dropped. This makes the proposed structure more feasible.

3. l_p-norm learning algorithm

The lack of robustness in the least-mean square (LMS) algorithm is attributed to its overweighing of aberrant noise in the error signal. Therefore, for the LMS algorithm to be robust, an appropriate error suppressor must be equipped prior to evaluating the increments of the tap-weights. For example, a robust LMS algorithm can be of the form

$$W(n+1) = W(n) + \eta z[e(n)]\frac{\partial y(n)}{\partial W(n)}, \tag{10}$$

where η is the learning gain and $z(\cdot)$ is an error suppressor. If, for example, the error signal has a probability density function (PDF) that resembles a logistic density, then a function of the form $z[e(n)] = \tanh[e(n)/2]$ can be a robust error suppressor [13]. However, the PDF of an error signal is generally not known a priori and hence the determination of an error suppressor is subject to trial and error. An alternative way to achieve this goal is to use the l_p-norm error criterion with $p < 2$. The l_p-norm error function [21] is defined by

$$\varepsilon_p(n) = \frac{1}{p}E[|e(n)|^p], \tag{11}$$

where p denotes the power metric and the factor of p^{-1} is padded simply for mathematical convenience [17, 20]. For $p = 2$, (11) becomes the mean squared error (MSE) or the l_2-norm error criterion. Similar to the rule of LMS algorithm, the negative gradient, $-\nabla_p(n)$, in the p-power error surface can be estimated as

$$-\hat{\nabla}_p(n) = |e(n)|^{p-1}\frac{\partial y(n)}{\partial W(n)}\operatorname{sgn}[e(n)], \tag{12}$$

where $\operatorname{sgn}[e(n)]$ denotes a sign function of $e(n)$. We can see from (12) that a nonlinear error function of the form

$$z[e(n)] = \operatorname{sgn}[e(n)]\,|e(n)|^{p-1} \tag{13}$$

is now inherently built in the tap-weight updating equation.

The factor $|e(n)|^{p-1}$ in (13) rescales $|e(n)|$ to some extent if $p \neq 2$. Fig. 2 shows how $|e(n)|$ is rescaled by $|e(n)|^{p-1}$ using different values of p. The value of $|e(n)|$ is assumed to be in the interval $[0,2]$. For $p > 2$, $|e(n)|^{p-1}$ scales up large values of $|e(n)|$ and scales down small values of $|e(n)|$. This tells why the use of $p \geqslant 2$ cannot be robust when possible statistical outliers [10, 12] are located close to the upper bound of $|e(n)|$. For $p < 2$, $|e(n)|^{p-1}$ scales down large values of $|e(n)|$ and scales up small values of $|e(n)|$ to some extent. This indicates that the use of $p < 2$ can provide a robust error suppressor if possible statistical outliers are located close to the upper bound of $|e(n)|$. However, for $p < 1$, a numerical stability problem arises whenever $|e(n)|$ is close to zero. This is the issue that must be resolved when using $p < 1$. Notice that using l_p-norm with $p < 2$ could throw some information away if the distribution of $|e(n)|$ is not known completely.

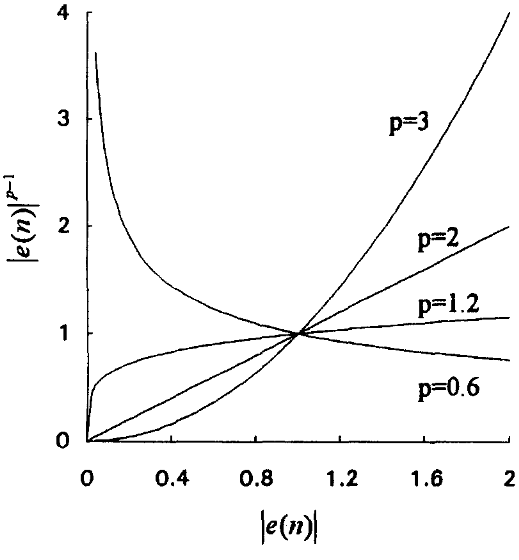

Fig. 2. $|e(n)|^{p-1}$ versus $|e(n)|$ for different values of p.

The tap-weight updating equation for the proposed DFE is derived as follows. The equalizer output $y(n)$ in Fig. 1 can be written as

$$y(n) = f[W^{\mathrm{T}}(n)U(n) + w_0(n)], \tag{14}$$

where $f(\cdot)$ is the activation function, $w_0(n)$ is the threshold level for the perceptron, $W(n) = [w_1(n), w_2(n), \dots, w_{N-1}(n)]^{\mathrm{T}}$ is the tap-weight vector, and $U(n) = [u_1(n), u_2(n), \dots, u_{N-1}(n)]^{\mathrm{T}}$ is the input vector comprising components to be supplied to the perceptron. If the activation function is of the form $f(x) = (1 - \mathrm{e}^{-x})/(1 + \mathrm{e}^{-x})$, then the resulting tap-weight and threshold level updating equations can be written, respectively, as

$$W(n + 1) = W(n) + \eta \operatorname{sgn}[e(n)]|e(n)|^{p-1}f'(n)U(n), \tag{15}$$

$$w_0(n + 1) = w_0(n) + \beta \operatorname{sgn}[e(n)]|e(n)|^{p-1}f'(n), \tag{16}$$

where β is the threshold level adaptation gain and $f'(n)$ is the derivative of the activation function with respect to its argument, that is, $f'(n) = [1 - y^2(n)]/2$. It can be seen from (15) or (16) that the l_2-norm yields a standard LMS algorithm and the l_1-norm yields a sign algorithm. Table look-up method can be employed to implement the factor $|e(n)|^{p-1}$ in the algorithm.

4. Performance analysis of the l_p-norm algorithm

The following analyses will focus on the tap-weight updating equation given by (15) since the same results can be applied directly to the threshold level update. To see the effect of p on the learning gain, we rewrite (15) as

$$W(n + 1) = W(n) + \frac{\bar{\eta}}{|e(n)|^{2-p}}\bar{\delta}(n)U(n), \tag{17}$$

where $\bar{\delta}(n) = e(n)f'(n)$ is the change of error at $p = 2$ and $\bar{\eta}$ is the learning gain at $p = 2$. Having this, the effective learning gain at arbitrary values of p can be written as $\eta_p(n) = \bar{\eta}/|e(n)|^{2-p}$. Further investigation of this requires a priori knowledge of $|e(n)|$. For the assumed activation function f, the output will be limited by the interval $[-1, 1]$ and most of it will be located close to its steady states, -1 or 1 in our case, due to the nonlinearities of f. If the desired signal has an alphabet of $\{-1, 1\}$, then $|e(n)|$ will be distributed in $[0, 2]$ and most of them will be located close to its lower bound under correct decisions. Consequently, using l_p-norm with $p < 2$ can treat $|e(n)|$ in a 'robust' manner as discussed in the last section. Without loss of generality, the output is assumed to be uniformly distributed in $[-1, 1]$ for simplicity [19] in the following analyses. Having this, $|e(n)|$ will be uniformly distributed in $[0, 2]$.

4.1. l_p-norm for $p < 1$

The possible numerical problem encountered when using $p < 1$ can be solved by replacing $|e(n)|$ with a small positive number θ whenever $|e(n)| \leqslant \theta$. Although alternative method, such as switching p from $p < 1$ to $p \geqslant 1$ when $|e(n)| \leqslant \theta$, could be feasible, only the former is focused in this paper. The following results are valid if $|e(n)|$ is bounded in the interval $[0, 2]$ and θ is limited by $\theta < 1$.

In the training mode, a correct decision makes $|e(n)|$ to be distributed in $[\theta, 1]$, whereas an incorrect decision makes $|e(n)|$ to the distributed in $[1, 2]$. Let $P(e)$ denote the error probability in making the decisions. The expectation value of $|e(n)|^{2-p}$ can be found by assigning a weight of $P(e)$ to the expectation value of $|e(n)|^{2-p}$ for $|e(n)|$ lying in the interval $[1, 2]$ and a weight of $[1 - P(e)]$ to that for $|e(n)|$ lying in the interval $[\theta, 1]$, i.e.,

$$\begin{aligned} E[|e(n)|^{2-p}] &= [1 - P(e)]E\{|e(n)|^{2-p} \| e(n)| \in [\theta, 1]\} \\ &\quad + P(e)E\{|e(n)|^{2-p} \| e(n)| \in [1, 2]\}. \end{aligned} \tag{18}$$

The two expectation values in the right-hand side of (18) can be obtained as

$$\begin{aligned} E\{|e(n)|^{2-p} \| e(n)| \in [\theta, 1]\} \\ = (1 - \theta)^{-1}(3 - p)^{-1}(1 - \theta^{3-p}), \end{aligned} \tag{19}$$

$$E\{|e(n)|^{2-p} \| e(n)| \in [1, 2]\} = (3 - p)^{-1}(2^{3-p} - 1). \tag{20}$$

By substituting (19) and (20) into (18), we obtain

$$E[|e(n)|^{2-p}] = \frac{[1-P(e)](1-\theta^{3-p}) + P(e)(1-\theta)(2^{3-p}-1)}{(3-p)(1-\theta)}. \tag{21}$$

The average learning gain is defined by $\eta_{\text{av}} = E[\eta_p(n)] = \bar{\eta}/E[|e(n)|^{2-p}]$. This is in turn found as

$$\eta_{\text{av}} = \frac{(3-p)(1-\theta)}{[1-P(e)](1-\theta^{3-p}) + P(e)(1-\theta)(2^{3-p}-1)}\bar{\eta}. \tag{22}$$

The result indicates that η_{av} increases with decreasing p and decreases with increasing θ and/or $P(e)$. A large θ can counteract the effect of p on η_{av}. If $P(e) \approx 0$ and $\theta^{3-p} \ll 1$, (22) can be simplified as

$$\eta_{\text{av}} \approx (1-\theta)(3-p)\bar{\eta}. \tag{23}$$

Defining $\eta_{\text{av}}/\bar{\eta}$ as the learning gain enhancement relative to the l_2-norm, Fig. 3 shows $\eta_{\text{av}}/\bar{\eta}$ as a function of p under $P(e) \approx 0$. It can be seen that $\eta_{\text{av}}/\bar{\eta}$ increases with decreasing θ. However, small values of θ may result in a numerical problem and hence θ cannot be too small. Some empirical values of θ will be used in our later simulations.

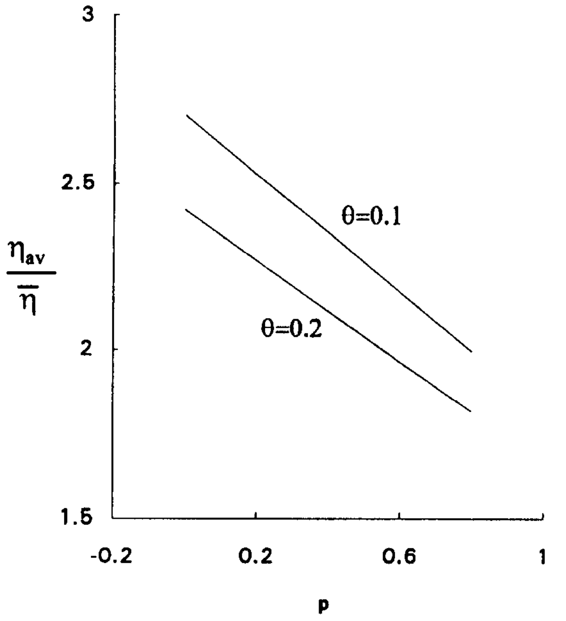

Fig. 3. Learning gain enhancement ($\eta_{\text{av}}/\bar{\eta}$) as a function of p under $P(e) \approx 0$.

In the decision-directed mode, $|e(n)|$ will be distributed in $[\theta, 1]$. Therefore, η_{av} can be obtained directly from (19) as

$$\eta_{\text{av}} = (1-\theta)(3-p)(1-\theta^{3-p})^{-1}\bar{\eta}. \tag{24}$$

If $\theta^{3-p} \ll 1$, (24) can be simplified as

$$\eta_{\text{av}} \approx (1-\theta)(3-p)\bar{\eta}. \tag{25}$$

4.2. l_p-norm for $1 \leqslant p \leqslant 2$

Since no numerical problem occurs for $1 \leqslant p \leqslant 2$, η_{av} can be obtained directly from (22) and (24) by setting $\theta = 0$. Therefore, in the training mode, η_{av} becomes

$$\eta_{\text{av}} = (3-p)[1 + P(e)(2^{3-p}-2)]^{-1}\bar{\eta}. \tag{26}$$

If $P(e) \approx 0$, (26) yields

$$\eta_{\text{av}} \approx (3-p)\bar{\eta}. \tag{27}$$

Finally, in the decision-directed mode, η_{av} is given by

$$\eta_{\text{av}} = (3-p)\bar{\eta}. \tag{28}$$

The same results can also be found in [22]. Eq. (27) or (28) tells that η_{av} can be, at most, enhanced by a factor of $(3-p)$. It is worth noting that the η_{av} for the l_1-norm is twice of that for the l_2-norm. This property is attractive since using l_1-norm can increase the convergence rate at a reduced computational complexity.

4.3. l_p-norm in a higher signal-to-noise ratio environment

When the signal-to-noise ratio (SNR) is high, $P(e)$ tends to be negligible and $|e(n)|$ can be bounded by a small value of $\alpha \leqslant 1$ in some cases. Accordingly, for $p < 1$, $E[|e(n)|^{2-p}] \approx E\{|e(n)|^{2-p} \| e(n)| \in [\theta, \alpha]\}$ with $\theta \leqslant \alpha \leqslant 1$, while for $1 \leqslant p \leqslant 2$, $E[|e(n)|^{2-p}] \approx E\{|e(n)|^{2-p} \| e(n)| \in$

$[0, \alpha]\}$ with $0 < \alpha \leqslant 1$. Then we obtain

$$E[|e(n)|^{2-p}] \approx \begin{cases} (\alpha-\theta)^{-1}(3-p)^{-1}(\alpha^{3-p}-\theta^{3-p}), & p < 1, \\ \alpha^{2-p}(3-p)^{-1}, & 1 \leqslant p \leqslant 2. \end{cases} \quad (29)$$

Therefore, in both training mode and decision-directed mode, η_{av} becomes

$$\eta_{\mathrm{av}} \approx \begin{cases} (\alpha-\theta)(3-p)(\alpha^{3-p}-\theta^{3-p})^{-1}\bar{\eta}, & p < 1, \\ (3-p)\alpha^{p-2}\bar{\eta}, & 1 \leqslant p \leqslant 2. \end{cases} \quad (30)$$

5. Computer simulations

A random sequence with alphabet $\{-1, 1\}$ is used as the input to the channel for computer simulations. A zero-mean white Gaussian noise is used as the additive interference. A linear channel of the form

$$r(n) = 0.348s(n) + 0.870s(n-1) + 0.348s(n-2) \quad (31)$$

is first chosen to justify the effect of p on the performance results obtained above. The equalizer delay chosen is $d = 2$. Unless otherwise indicated, the order of polynomial expansion used is $i = 3$ for all the polynomial-perceptron based equalizers. The SNR is defined by SNR $= 10\log(\sigma_{\mathrm{s}}^2/\sigma_{\mathrm{n}}^2)$, where σ_{s}^2 is the signal power at the output of the channel and σ_{n}^2 is the noise power. For the simplicity in representation, the notation (L, M) DFE is used to denote a DFE having L taps in the feedforward filter and M taps in the feedback filter. Also, the notation (N_1, N_2, N_3) MLP is used to denote an MLP having N_1 neurons in the hidden layer 1, N_2 neurons in the hidden layer 2, and N_3 neurons in the output layer. The learning curves are determined by an average of 800 individual trials with each comprising different input sequences and different sets of random initial tap-weights. The BER performances are evaluated by an average of 800 independent trials with each comprising 10000 bits in length. In each trial, the first 1800 bits are used for training, which is considered sufficient for a nearly complete training. Furthermore, in the case of $p < 1$, some empirical values of θ are used.

5.1. Convergence properties

In these simulations, the MSE is used as a measure of the convergence performance and SNR = 20 dB is used in all simulations. Fig. 4 shows the learning curves of the (4, 1) PPDFE-I, the (4, 1) PPDFE-II and a PPS equalizer of length 5. Here, the l_2-norm algorithm with $\bar{\eta} = 0.1$ and $\beta = 0.05$ is used for all cases. The results indicate that the proposed structure (PPDFE-I) converges much faster than the PPDFE-II as well as the PPS equalizer. The former justifies the prediction in Section 2 and the latter demonstrates the benefit of using a DFE in such a single-layer net. Fig. 5 compares the learning curves between the (4, 1) PPDFE-I and a (6, 3, 1) MLP (4, 1) DFE. Here, a (6, 3, 1) MLP is chosen such that the number of tap-weights needed are comparable with the (4, 1) PPDFE-I. The l_2-norm BP algorithm with $\bar{\eta} = 0.1$ and $\beta = 0.05$ is used to train the MLP and the same parameters as above are again used in (4, 1) PPDFE-I. It is seen that the (4, 1) PPDFE-I converges much faster than the MLP.

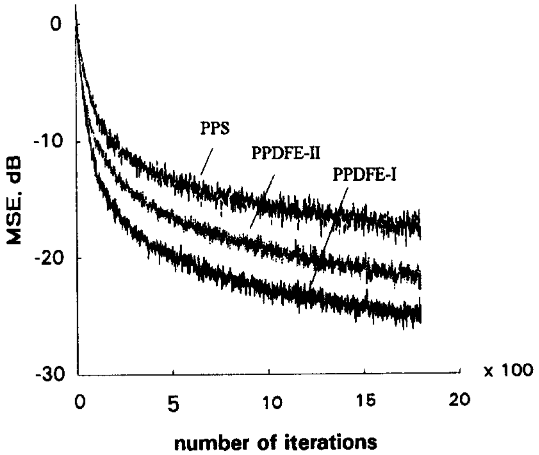

Fig. 4. Comparison of convergence rate for different polynomial-perceptron based equalizers: PPS of length 5, (4, 1) PPDFE-I and (4, 1) PPDFE-II.

Fig. 6 shows the effect of p on the convergence rate for the (4, 1) PPDFE-I. Here, different values of p, 2, 1.2 and 0.6, are used to illustrate their effect on the convergence rate. Other parameters used are: $\bar{\eta} = 0.1$ and $\theta = 0.2$ for $p = 0.6$. The results indicate that significant improvement in convergence rate can be achieved by using smaller values of p. The dependence of convergence rate on p, with noise floor at -20 dB, is shown in Fig. 7. It can be seen that the convergence rate increases with decreasing p for $p < 2$. However, the convergence rate approaches a limit as further decreasing p from certain values of p (say, $p = 0.6$). This is largely caused by the constant θ as well as the error probability $P(e)$. Another reason might be resorted to the potential limit of the algorithm with stochastic gradient estimate. Fig. 8 shows the dependence of

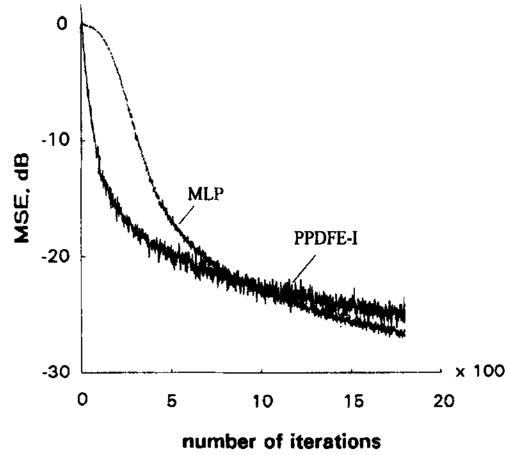

Fig. 5. Comparison of convergence rate for the (4, 1) PPDFE-I using $i = 3$ and the (6, 3, 1) MLP (4, 1) DFE.

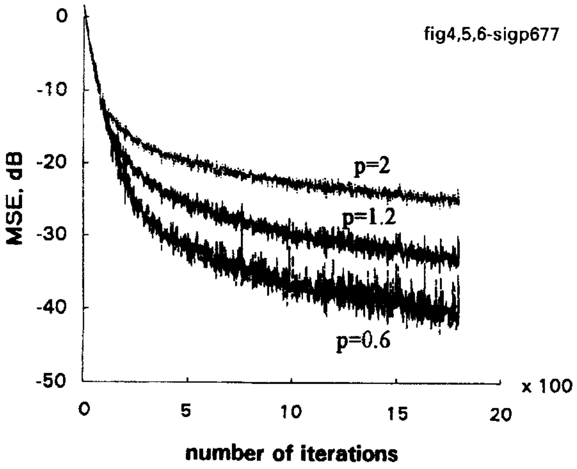

Fig. 6. Learning curves of the (4, 1) PPDFE-I at different values of p for SNR = 20 dB.

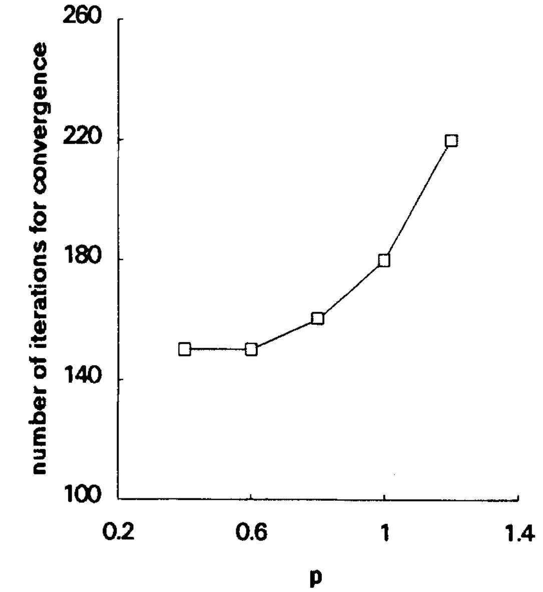

Fig. 7. Convergence rate of the (4, 1) PPDFE-I versus p.

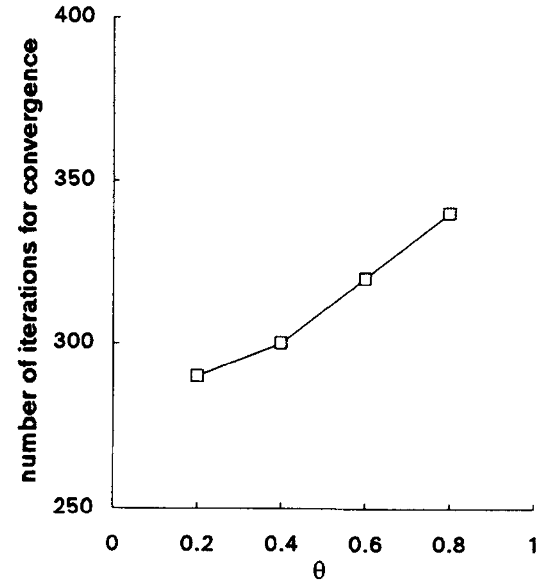

Fig. 8. Convergence rate of the (4, 1) PPDFE-I as a function of θ.

convergence rate on θ with noise floor at -20 dB. As we have expected, the convergence rate decreases with increasing θ.

5.2. BER performances

Unless otherwise indicated, detected-bit feedback is assumed in the following simulations of DFE. Fig. 9 gives a BER performance comparison for the (4, 1) PPDFE-I, the (4, 1) PPDFE-II, and the PPS equalizer of length 5. The result indicates that the proposed structure (PPDFE-I) outperforms the other two, especially in a higher SNR environment. Parameters used here are the same, except $\bar{\eta} = 0.03$, as that used in Fig. 4. Fig. 10 compares the BER performance between the (4, 1) PPDFE-I and the (6, 3, 1) MLP (4, 1) DFE. The simulation result indicates that the proposed structure is capable of achieving performance similar to that achieved by the MLP. Fig. 11 gives the BER performance of the (4, 1) PPDFE-I as a function of p. The result indicates that the case using $p = 0.6$ outperforms those using larger values of p. The improvement becomes more significant when the SNR is high. This is because the equalizer achieves a better tracking capability, as described by (30), in

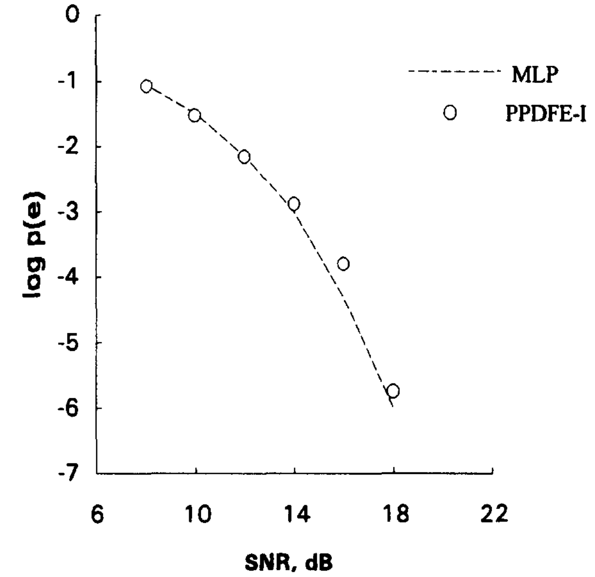

Fig. 10. Comparison of BER performance for the (4, 1) PPDFE-I using $i = 3$ and the (6, 3, 1) MLP (4, 1) DFE.

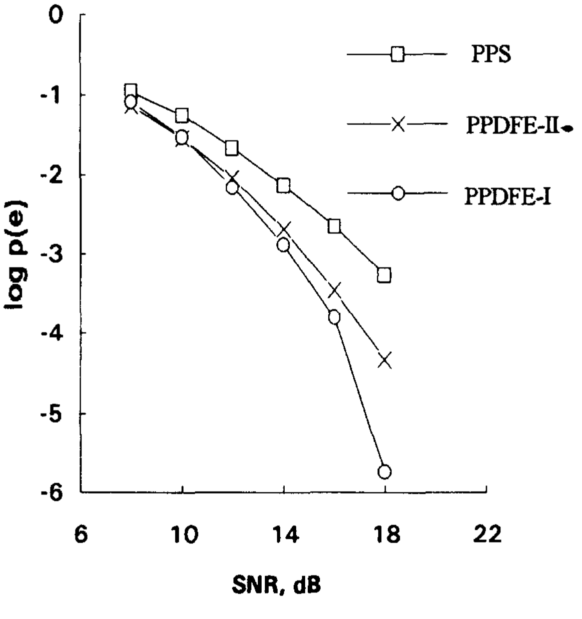

Fig. 9. BER performance for different polynomial-perceptron based equalizers: PPS of length 5, (4, 1) PPDFE-I and (4, 1) PPDFE-II.

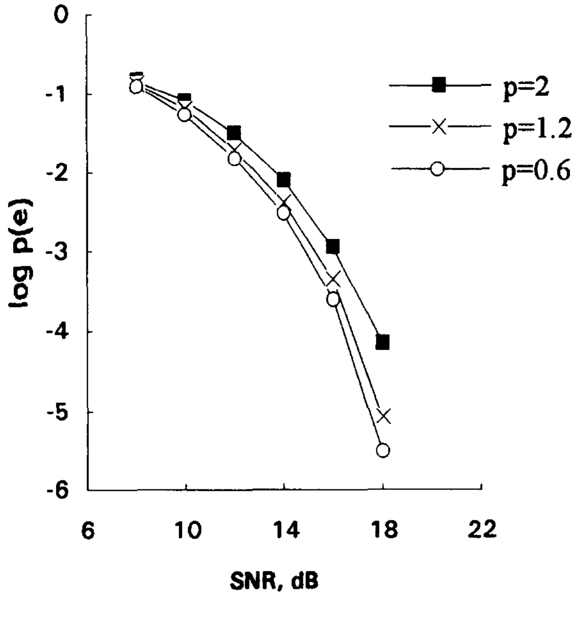

Fig. 11. BER performance for the (4, 1) PPDFE-I at different values of p.

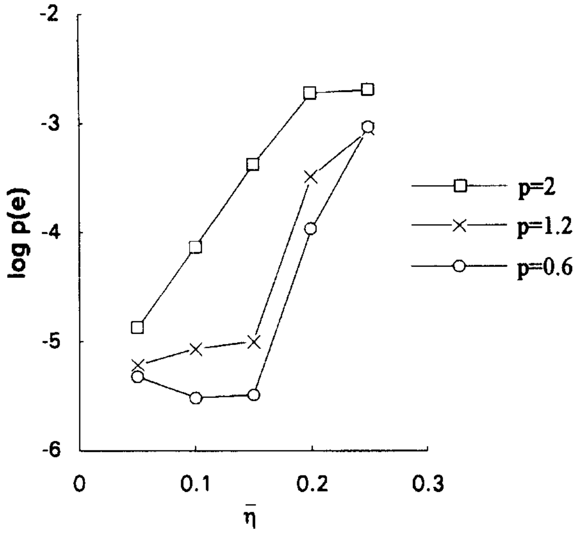

Fig. 12. BER performance as a function of $\bar{\eta}$ for the (4, 1) PPDFE-I.

such a condition. It is worth noting that, besides a convergence rate improvement as indicated in Fig. 6, a BER improvement can also be achieved by using smaller values of p. This simultaneous improvement cannot be achieved by an l_2-norm. Fig. 12 shows the BER performance of the (4, 1) PPDFE-I as a function of $\bar{\eta}$. It can be seen that the performance using $p = 2$ is quite sensitive to the noise caused by larger values of $\bar{\eta}$ [11]. However, this is less serious when using $p < 2$.

To see the ability of the new equalizer to cope with a nonlinear channel, a channel of the form

$$t(n) = 0.408s(n) + 0.816s(n-1) + 0.408s(n-2), \tag{32}$$

$$r(n) = t(n) + 0.2t^2(n) - 0.1t^3(n) \tag{33}$$

is chosen [3, 6, 18] for evaluating the performance of our approach. The linear part in (32) results in severe intersymbol interference, and the nonlinear part in (33) guarantees poor performance for the conventional DFE (CDFE). Combining (32) and (33) yields the Volterra series representation [1] of the form

$$r(n) = \sum_{i=0}^{2} c_i s(n-i) + \sum_{i=0}^{2}\sum_{j=i}^{2} c_{ij} s(n-i)s(n-j)$$
$$+ \sum_{i=0}^{2}\sum_{j=i}^{2}\sum_{k=j}^{2} c_{ijk} s(n-i)s(n-j)s(n-k), \tag{34}$$

with coefficients c_i, c_{ij} and c_{ijk} given in Table 1. The binary nature of $s(n)$ makes it impossible to identify a nonlinear channel by using a direct modeling [6]. Consequently, an adaptive equalizer requiring a channel estimator, such as the adaptive MLSE, cannot perform well in such a situation.

Fig. 13 shows the BER simulation results of the (4, 1) Bayesian DFE [3–5, 9], the (4, 1) PPDFE-I,

Table 1
Volterra coefficients for the nonlinear channel given by (32) and (33)

Linear part		
$c_0 = 0.408$	$c_1 = 0.816$	$c_2 = 0.408$
Second-order nonlinearities		
$c_{00} = 0.033$	$c_{01} = 0.067$	$c_{11} = 0.133$
$c_{02} = 0.033$	$c_{12} = 0.067$	$c_{22} = 0.033$
Third-order nonlinearities		
$c_{000} = -0.007$	$c_{001} = -0.041$	$c_{011} = -0.082$
$c_{111} = -0.054$	$c_{002} = -0.020$	
$c_{012} = -0.082$	$c_{112} = -0.082$	$c_{022} = -0.020$
$c_{122} = -0.041$	$c_{222} = -0.007$	

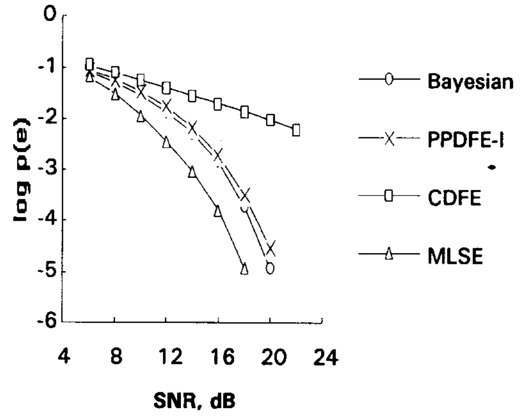

Fig. 13. Comparison of BER performance for the Bayesian (4, 1) DFE, the (4, 1) CDFE, the (4, 1) PPDFE-I and the MLSE under a nonlinear channel.

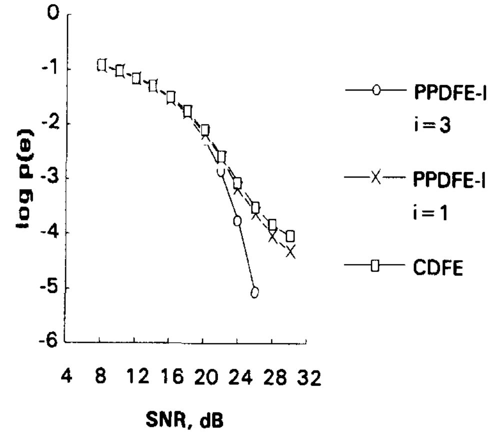

Fig. 14. Comparison of BER performance for the (2, 3) PPDFE-I and the (2, 3) CDFE under channel (35).

the (4, 1) CDFE, and the MLSE under this nonlinear channel. Here, correct-bit feedback is assumed for all cases. Other parameters used are: l_p-norm algorithm with $p = 1.2$ and $\bar{\eta} = \beta = 0.1$ for the PPDFE-I and LMS algorithm with $\bar{\eta} = 0.01$ for the CDFE. Note that, in the LMS, $\bar{\eta} = 0.01$ is deliberately chosen so that the CDFE can approximately achieve its best possible performance. A correct estimate of the channel response is assumed for both the Bayesian criterion and the MLSE. The result demonstrates that our equalizer has a performance close to that offered by the Bayesian criterion, which is optimal in a minimum BER sense for symbol-by-symbol-decision equalizer. Also, it shows that the CDFE performs very poorly under the channel considered. This is due to its limited capabilities in partitioning the signal space.

Finally, to illustrate the advantage of higher-order expansion used in our structure, a linear channel with severe interference [18] as described by

$$r(n) = 0.227s(n) + 0.460s(n-1) + 0.688s(n-2) + 0.460s(n-3) + 0.227s(n-4) \qquad (35)$$

is chosen to evaluate the performance. Fig. 14 compares the performance of a (2, 3) PPDFE-I using different orders of polynomial expansion ($i = 1$ and $i = 3$) and a (2, 3) CDFE. Note that correct-bit feedback is again assumed for all cases. It is worth noting that the third-order PPDFE-I can equalize such a difficult channel, while the CDFE and the first-order PPDFE-I cannot perform well, under this condition. This justifies the benefit of using higher-order expansion over the signal vector presented in a DFE.

6. Comparison with other nonlinear equalizers

It is known that the performance of the equalizer with detection based on the entire sequence can be better than those based on a symbol-by-symbol basis, provided that the characteristic of the channel is known. However, if the channel response is not known a priori, a procedure of channel estimation is necessary [15]. An inaccurate estimate may degrade the performance. The new equalizer does not require a channel estimate yet achieves a performance comparable to the Bayesian criterion. Furthermore, the implementation complexity of the MLSE may preclude its use in some real-time applications. Although the use of Viterbi algorithm [7] may reduce the complexity significantly, the requirement of computations and memory is still demanding. The new approach has much less demanding in computation and no memory requirement.

In [6], Chen et al. showed that the radial basis function (RBF) equalizer and the optimal Bayesian solution are equivalent in structure. The result indicated that the RBF equalizer has similar performance to the Bayesian approach given a known channel order and noise variance. However, performance degradation occurs when the above parameters are not known a priori. Our approach achieves similar performance without any knowledge of the channel and the noise.

One disadvantage of our structure is its large number of tap-weights needed in some practical applications. Fortunately, a number of higher-order terms that tend to be inconsequential can be neglected. A good example for this can be found in [1]. It indicated that a significant MSE improvement can be achieved even using only a few nonlinear terms for the equalization over a nonlinear satellite channel.

7. Conclusions

A new equalizer, including a DFE equipped with polynomial-perceptron model of nonlinearities and an l_p-norm based robust learning algorithm, is presented in this paper. The structure exerts the benefit of using a DFE with a polynomial-perceptron structure. The algorithm is robust in the sense of dealing with aberrant noise by the effect of $p < 2$ on the error signal. A detailed performance analysis of the algorithm including a consideration on the possible numerical problem arising when $p < 1$ is given in this paper. The algorithm is advantageous due to its robustness as well as its simplicity. Computer simulation results show that the proposed equalizer simultaneously accomplishes faster convergence rate and satisfactory BER performance. Particularly, it is shown that our equalizer can approach the performance offered by the Bayesian criterion and achieves significant improvement in BER performance for channels having severe interference.

Acknowledgements

The authors would like to thank the anonymous referees for their valuable comments to improve the manuscript.

Notation

β	threshold level adaptation gain
d	equalizer delay
∇_p	gradient vector in the p-power error surface
$\hat{\nabla}_p$	gradient vector estimate in the p-power error surface
$\bar{\delta}$	change of error at $p = 2$
e	error signal
E	expectation operator
ε_p	l_p-norm error function
η	learning gain
$\bar{\eta}$	learning gain at $p = 2$
η_{av}	average learning gain
η_p	effective learning gain at arbitrary values of p
$\eta_{\mathrm{av}}/\bar{\eta}$	learning gain enhancement
f	activation function
f'	first derivative of f
i	order of polynomial expansion
L	number of taps in the feedforward part of a DFE
l_p	p-power error metric
l_2	mean squared error metric
M	number of taps in the feedback filter of a DFE
n	time index
N	number of weights needed for the perceptron
$P(e)$	error probability in making decisions
p	power of error metric
r	output of a channel
s	transmitted signal
sgn	sign operator
σ_{n}^2	noise power
σ_{s}^2	signal power
θ	small positive number
U	perceptron input vector
u_0	input of unity value
v	decision error
W	perceptron weight vector
w_0	threshold level for the perceptron
X	input vector represented by a DFE
X_{F}	feedforward signal vector in X
X_{B}	feedback signal vector in X
y	perceptron output
z	error suppressor function

References

[1] S. Benedetto, E. Biglieri and V. Castellani, *Digital Transmission Theory*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[2] C.H. Chang, S. Siu and C.H. Wei, "A decision feedback equalizer utilizing higher-order correlation", *Proc. 1993 IEEE ISCAS*, Chicago, IL, May 1993, pp. 707–710.

[3] S. Chen, G.J. Gibson, C.F.N. Cowan and P.M. Grant, "Adaptive equalization of finite non-linear channels using multilayer perceptrons", *Signal Processing*, Vol. 20, No. 2, June 1990, pp. 107–119.

[4] S. Chen, G.J. Gibson and C.F.N. Cowan, "Adaptive channel equalization using a polynomial-perceptron structure", *IEE Proc.*, Pt. I, Vol. 137, No. 5, 1990, pp. 257–264.

[5] S. Chen, G.J. Gibson, C.F.N. Cowan and P.M. Grant, "Reconstruction of binary signals using an adaptive radial-basis-function equalizer", *Signal Processing*, Vol. 22, No. 1, January 1991, pp. 77–93.

[6] S. Chen, B. Mulgrew and P.M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks", *IEEE Trans. Neural Networks*, Vol. 4, No. 4, 1993, pp. 570–579.

[7] G.D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference", *IEEE Trans. Inform. Theory*, Vol. IT-18, 1972, pp. 363–378.

[8] W.S. Gan, J.J. Soraghan and T.S. Durrani, "New functional-link based equalizer", *Electron. Lett.*, Vol. 28, No. 17, August 1992, pp. 1643–1645.

[9] G.J. Gibson, S. Siu and C.F.N. Cowan, "The application of nonlinear structures to the reconstruction of binary signals", *IEEE Trans. Signal Process.*, Vol. 39, No. 8, 1991, pp. 1877–1884.

[10] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw and W.A. Stahel, *Robust Statistics: The Approach based on Influence Functions*, Wiley, New York, 1986.

[11] S. Haykin, *Adaptive filter theory*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[12] P.J. Huber, *Robust Statistics*, Wiley, New York, 1981.

[13] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[14] R.P. Lippmann, "An introduction to computing with neural nets", *IEEE ASSP Magazine*, April 1987, pp. 4–22.

[15] F.R. Magee and J.G. Proakis, "Adaptive maximum-likelihood sequence estimation for digital signaling in the presence of intersymbol interference", *IEEE Trans. Inform. Theory*, Vol. IT-19, 1973, pp. 120–124.

[16] M. Minsky and S. Papert, *Perceptron: An Introduction to Computational Geometry*, MIT Press, Cambridge, MA, 1988.

[17] Y.H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, Reading, MA, 1989.

[18] J.G. Proakis, *Digital Communications*, McGraw-Hill, New York, 1989.

[19] A.K. Rigler, J.M. Irvine and T.P. Vogl, "Rescaling of variables in back propagation learning", *Neural Networks*, Vol. 4, 1991, pp. 225–229.

[20] D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press, Cambridge, MA, 1986, Vol. 1, Chapter 8.

[21] J. Schroeder, R. Yarlagadda and J. Hersey, "l_p normed minimization with applications to linear predictive modeling for sinusoidal frequency estimation", *Signal Processing*, Vol. 24, No. 2, August 1991, pp. 193–216.

[22] S. Siu and C.F.N. Cowan, "Performance analysis of the l_p-norm back propagation algorithm for adaptive equalization", *IEE Proc.*, Pt. F, Vol. 140, No. 1, 1993, pp. 43–47.

[23] S. Siu, G.J. Gibson and C.F.N. Cowan, "Decision feedback equalization using neural network structures and performance comparison with standard architecture", *IEE Proc.*, Pt. I, Vol. 137, No. 4, August 1990, pp. 221–225.

[24] B. Widrow, R.G. Winter and R.A. Baxter, "Layered neural nets for pattern recognition", *IEEE Trans. Acoust. Speech Signal Process.*, July 1988, pp. 1096–1099.