Adaptive channel equalisation using a polynomial-perceptron structure

S. Chen, PhD
G.J. Gibson, PhD
C.F.N. Cowan, PhD, CEng, MIEE

Indexing terms: Channel equalisation, Polynomial approximation, Adaptive algorithms

Abstract: The paper investigates the application of a simple nonlinear structure to the problem of adaptive channel equalisation. Based on the Bayes decision rule, it is shown that the optimal equalisation solution is an inherently nonlinear problem and, therefore, it is desired to incorporate some degree of nonlinearity in the design of equaliser structure. The approximate realisation of the optimal equalisation solution is implemented using a polynomial-perceptron architecture and simulation results are included to support the theoretical analysis.

1 Introduction

Communications channel equalisation is concerned with the reconstruction of digital signals that have been passed through a dispersive channel and then corrupted with additive noise. Traditional techniques for solving this equalisation problem are based on linear finite filters. Adaptive linear equalisers are robust and can easily be implemented. The operation of an equaliser at each sample instant is typically based on a finite number of channel observations and decisions are usually made on a symbol-by-symbol base. Even under this classical information constraint, it has been shown that channel equalisation is an inherently nonlinear problem [4] regardless of whether a channel is minimum or nonminimum phase. Nonlinear structures are therefore required to achieve fully or near optimal performance.

Gibson *et al.* [4] proposed a nonlinear equaliser structure based on the multilayer perceptron and demonstrated its superior performance over the linear equaliser. The multilayer perceptron has a very general ability of nonlinear decision making and, theoretically, a multilayer perceptron equaliser with sufficient size can realise the optimal performance. There are, however, some practical difficulties associated with this highly nonlinear structure that require further investigation. The selection of architecture and parameter values for the multilayer perceptron equaliser is mainly by experiment. The training algorithms are usually gradient based algorithms, such as the back propagation algorithm [7], and training times are typically very long. Although the use of recursive Gauss-Newton algorithms [2, 9] can significantly improve the convergence properties of the multilayer perceptron equaliser, these algorithms require more computation at each recursion and will have difficulties in meeting the real-time requirements of high-speed data transmission where adaptive equalisation is mostly needed.

In this paper an alternative nonlinear equaliser structure is examined. Using the Bayes decision rule, it is shown that the optimal equalisation solution is highly nonlinear, a result identical to that derived in [4] by a different approach. An old technique, namely polynomial approximation, is then employed as a means of approximately realising the optimal solution. This leads to a polynomial-perceptron equaliser that is theoretically more tractable compared with the multilayer perceptron equaliser as the filter parameters are almost linear with respect to the output error. Simple simulation examples are included to compare the performance of this polynomial-perceptron equaliser with the optimal one. It is also demonstrated that a direct polynomial approach [6] may converge to a fallacious classification function if polynomial degree is not large enough and a nonlinear perceptron activation is beneficial in such a situation. Further justifications of introducing nonlinear decision making ability into the adaptive equaliser structure are provided by examining the performance of the Wiener filter, which is the performance bound for any linear equaliser.

2 Channel equalisation

The digital communications system considered in this paper is depicted in Fig. 1. A random binary sequence

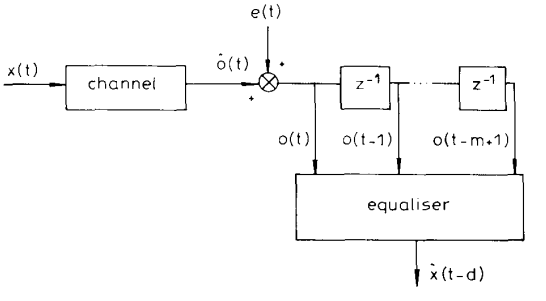

Fig. 1 *Schematic diagram of data transmission system*

Paper 7507I (E7, E8), first received 2nd December 1989 and in revised form 22nd January 1990

Dr. Chen and Dr. Cowan are with the Department of Electrical Engineering, University of Edinburgh, Mayfield Road, Edinburgh EH9 3JL, United Kingdom

Dr. Gibson is at Plessey Electronics Systems Research Ltd., Roke Manor, Romsey, Hants. SO5 0ZN, United Kingdom

$x(t)$ is transmitted through a linear dispersive channel modelled as a finite impulse response filter whose transfer function is given by

$$H_n(z) = \sum_{i=0}^{n} q_i z^{-i} \tag{1}$$

The channel output is corrupted by an additive Gaussian white noise $e(t)$. The task of the equaliser at the sample instant t is to reconstruct the input symbol $x(t-d)$ using the information contained in the channel output observations $o(t), \ldots, o(t-m+1)$, where the integers m and d are known as the order and the delay of the equaliser, respectively. The following assumption on the data sequence $x(t)$ is introduced to simplify the analysis.

Assumption 1: $x(t)$ is an independent sequence taking values of either 1 or -1 with an equal probability.

The above assumption on the channel model and signal conditions is, however, mostly for convenience. In fact, the approach discussed in the present study can directly be applied to the case of nonlinear channel model and additive non-Gaussian correlated noise without any modification [3]. It does not really matter whether $x(t)$ takes value 1 with a higher probability, or vice versa.

The information constraint on the general equaliser structure depicted in Fig. 1 is characterised by the equaliser order m and delay d, and the equaliser makes decisions on a symbol-by-symbol base. Given a channel response and a noise distribution, an important question is: what is the best possible performance, in terms of bit error rate, that an equaliser with fixed m and d can offer? An understanding of this question clearly helps to design better equalisers. As the equalisation of digital communications systems, described in Fig. 1, can be viewed as a two-state classification problem, optimal solution of the two-state classification problem is briefly summarised.

2.1 Two-state Bayes decision rule

Consider the two-state classification problem in which the state s is known to be either s_A or s_B. Based on a measurement $\mathbf{o} = [o_1 \cdots o_m]^T$, a decision is made as to whether $s = s_A$ or $s = s_B$. A common strategy of solving this problem is to minimise the expected or average risk of making a wrong decision and this strategy leads to the following Bayes decision rule [10]:

$$\hat{s}(\mathbf{o}) = \begin{cases} s_A & \text{if } q_A L_A f_{s_A}(\mathbf{o}) > q_B L_B f_{s_B}(\mathbf{o}) \\ s_B & \text{if } q_A L_A f_{s_A}(\mathbf{o}) < q_B L_B f_{s_B}(\mathbf{o}) \end{cases} \tag{2}$$

Here q_A and q_B are the *a priori* probabilities of occurrences of s_A and s_B, respectively, and $q_A + q_B = 1$. L_A is the loss associated with the decision $\hat{s}(\mathbf{o}) = s_B$ when actually $s = s_A$ and, similarly, L_B is the loss associated with $\hat{s}(\mathbf{o}) = s_A$ when $s = s_B$. $f_{s_A}(\mathbf{o})$ and $f_{s_B}(\mathbf{o})$ are the conditional density functions of $\mathbf{o}$ given $s = s_A$ and $s = s_B$, respectively. Eqn. 2 can be rewritten as

$$\hat{s}(\mathbf{o}) = \begin{cases} s_A & \text{if } f_{de}(\mathbf{o}) > 0 \\ s_B & \text{if } f_{de}(\mathbf{o}) < 0 \end{cases} \tag{3}$$

where

$$f_{de}(\mathbf{o}) = f_{s_A}(\mathbf{o}) - k f_{s_B}(\mathbf{o}),\ k = \frac{q_B L_B}{q_A L_A} \tag{4}$$

is known as the decision function and the set of points $\mathbf{o}$ that satisfy

$$f_{de}(\mathbf{o}) = 0 \tag{5}$$

is often referred to as the decision boundary, which partitions the m-dimensional Euclidean space $\mathbf{R}^m$ into two disjoint sets D_A and D_B. The decision making process, eqn. 3, can alternatively be stated as

$$\hat{s}(\mathbf{o}) = \begin{cases} s_A & \text{if } \mathbf{o} \in D_A \\ s_B & \text{if } \mathbf{o} \in D_B \end{cases} \tag{6}$$

When a measurement $\mathbf{o}$ satisfies eqn. 5, making the decision either way has a same expected risk and we may then arbitrarily decide $\hat{s}(\mathbf{o}) = s_A$ in this situation. D_A plus the decision boundary, denoted as $\bar{D}_A$, will be called the decision region.

2.2 Optimal equalisation solution

The optimal equalisation solution can be directly obtained from the above Bayes decision rule. The state concerned here is the transmitted data symbol $x(t-d)$ with two possible values $s_A = 1$ and $s_B = -1$. According to *assumption 1*, $x(t-d) = 1$ and $x(t-d) = -1$ have the same probability 0.5. The estimate of $x(t-d)$ is denoted as $\hat{x}(t-d)$. Mistakes $\hat{x}(t-d) = -1$ when $x(t-d) = 1$ and $\hat{x}(t-d) = 1$ when $x(t-d) = -1$ cause equal damage and, therefore, each case should be assigned with a same loss level. It is clear, under these conditions, that $k = 1$, and this results in the following minimum error-probability or bit-error-rate equaliser

$$\hat{x}(t-d) = \text{sgn}\,(f_{de}(\mathbf{o}(t))) = \text{sgn}\,(f_1(\mathbf{o}(t)) - f_{-1}(\mathbf{o}(t))) \tag{7}$$

where $\mathbf{o}(t) = [o(t) \cdots o(t-m+1)]^T$ is the channel observation vector, $f_1(\mathbf{o}(t))$ and $f_{-1}(\mathbf{o}(t))$ are the conditional density functions of observing $\mathbf{o}(t)$, given $x(t-d) = 1$ and $x(t-d) = -1$, respectively, and

$$\text{sgn}\,(y) = \begin{cases} 1 & y \geqslant 0 \\ -1 & y > 0 \end{cases} \tag{8}$$

is a slicer. The above result is identical to that derived in Reference 4. The approach used here is more general and can be applied easily to other situations. The noise-free channel output vector $\hat{\mathbf{o}}(t) = [\hat{o}(t) \cdots \hat{o}(t-m+1)]^T$, which is generated from input sequence $x(t), \ldots, x(t-m+1-n)$, can only take finite states or values. These values can be partitioned into two classes:

$$\left.\begin{aligned} P_{m,d}(1) &= \{\hat{\mathbf{o}}(t) \in \mathbf{R}^m \,|\, x(t-d) = 1\} \\ P_{m,d}(-1) &= \{\hat{\mathbf{o}}(t) \in \mathbf{R}^m \,|\, x(t-d) = -1\} \end{aligned}\right\} \tag{9}$$

The task of the equaliser is to decide whether a channel observation vector $\mathbf{o}(t)$ represents a noise corruption of an element in $P_{m,d}(1)$ or $P_{m,d}(-1)$ and thus to determine the input sample $x(t-d)$. The sets $P_{m,d}(1)$ and $P_{m,d}(-1)$ are determined by the channel transfer function $H_n(z)$, the equaliser order m and the delay d. These two sets, together with the distribution of the additive noise $e(t)$, completely specifiy the optimal decision function $f_{de}()$ or the corresponding decision region $\bar{D}_1$.

2.3 Some illustrations

We shall consider the case of equaliser order $m = 2$ simply because graphic display is difficult in higher dimension.

Example 1: Channel transfer function is $H_1(z) = 1.0 + 0.5z^{-1}$ and the equaliser delay $d = 0$.

The elements of the sets $P_{2,0}(1)$ and $P_{2,0}(-1)$ are plotted in Fig. 2 using the symbols 'circle' ○ and 'cross' ×, respectively. If there is no additive noise, the channel

output vector will be either a 'circle' or a 'cross' in this two dimensional space. Each point in $P_{2,0}(1)$ and $P_{2,0}(-1)$ has a same probability of appearance. This channel is minimum phase, therefore the two classes

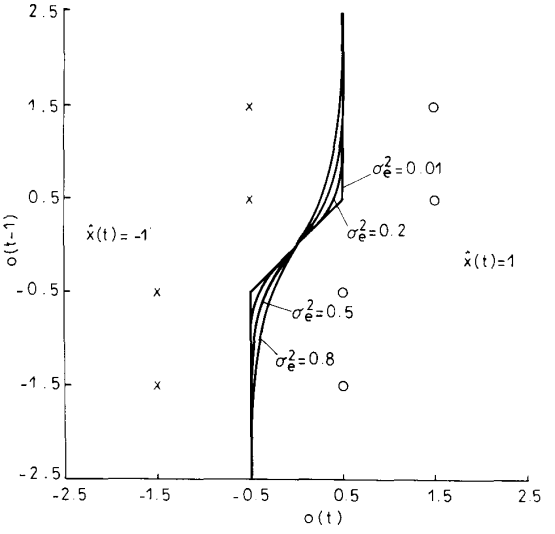

Fig. 2 *Channel output points and optimal decision boundaries*
Channel $1.0 + 0.5z^{-1}$, additive Gaussian white noise with variance σ_e^2, equaliser order $m = 2$ and delay $d = 0$

$P_{2,0}(1)$ and $P_{2,0}(-1)$ are linearly separable and a linear equaliser can perfectly reconstruct input signals in the noise-free case.

Because of additive Gaussian white noise, the channel observation vector is actually a random variable having a Gaussian probability distribution centred at one of the points of $P_{2,0}(1)$ and $P_{2,0}(-1)$. The lines in Fig. 2 are the optimal decision boundaries corresponding to different noise variances. We observe that the optimal boundary is always nonlinear. If a channel observation vector lands on the left-hand side of the optimal boundary, the optimal equaliser will produce the estimate $\hat{x}(t) = -1$, otherwise it gives $\hat{x}(t) = 1$. In this way the equaliser makes least possible mistakes. Because a linear equaliser can only generate a linear decision boundary the bit error rate of the linear equaliser will be considerably larger than that of the optimal equaliser.

Example 2: The channel transfer function is $H_1(z) = 0.5 + 1.0z^{-1}$, and the additive Gaussian white noise has a variance 0.2.

The elements of the sets $P_{2,0}(1)$ and $P_{2,0}(-1)$ are shown in Fig. 3, and the shaded region in Fig. 3 is the optimal decision region $\bar{D}_1$ under the constraint $d = 0$. Notice that $P_{2,0}(1)$ and $P_{2,0}(-1)$ are not linearly separable because the channel is nonminimum phase, and a linear equaliser with a zero delay is incapable of reconstructing input signals even in the noise-free case.

We now examine the case of nonzero delay. $P_{2,1}(1)$ and $P_{2,1}(-1)$ are given in Fig. 4. Although these two classes are linearly separable, the optimal classification boundary is nonlinear and a linear equaliser will not be able to realise such a boundary, as can be seen clearly from Fig. 4.

In general, the optimal boundary is a hypersurface in the m-dimensional space and can be highly nonlinear. The decision boundary of a linear equaliser is a hyperplane in the same space. Therefore, any linear equaliser structure is inherently suboptimal and this motivates the investigation of nonlinear architectures capable of realising highly nonlinear boundaries.

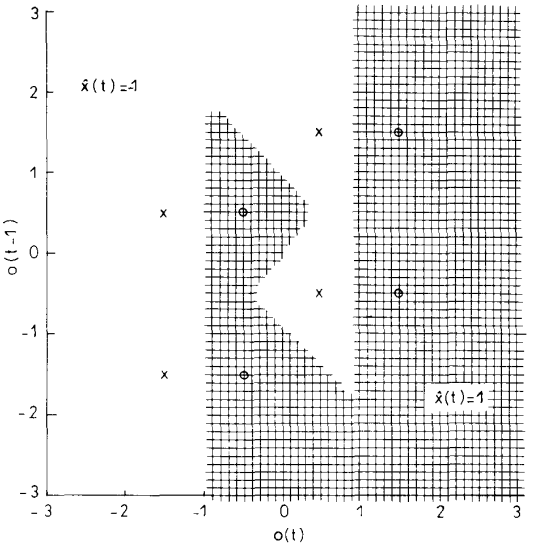

Fig. 3 *Channel output points and optimal decision region*
Channel $0.5 + 1.0z^{-1}$, additive Gaussian white noise with variance 0.2, equaliser order $m = 2$ and delay $d = 0$

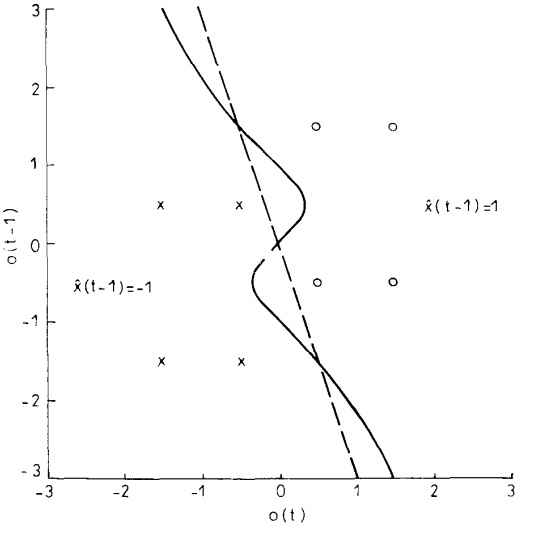

Fig. 4 *Channel output points and decision boundaries*
Channel $0.5 + 1.0z^{-1}$, additive Gaussian white noise with variance 0.2, equaliser order $m = 2$ and delay $d = 1$
——— optimal boundary
– – – – linear boundary

3 Polynomial approximation of optimal decision function

As the optimal decision function $f_{de}()$ for a communications channel is generally not available and can be time varying, a means of adaptively approximating this function or generating the corresponding decision region is essential to realise the optimal equaliser solution. We shall assume that $f_{de}()$ is continuous and this requires that the noise distribution satisfies the following condition.

Assumption 2: The distribution of $e(t)$ is continuous.

The usual Gaussian distribution satisfies this requirement.

The use of a polynomial function to approximate a continuous function is an old but effective technique and is widely applied to the identification of nonlinear systems [1]. Let $\mathbf{Z}$ be a compact subset of $\mathbf{R}^m$ and denote $\mathbf{C}^0(\mathbf{Z})$ as the space of all continuous functions from $\mathbf{Z}$ into $\mathbf{R}$. With the help of the Stone-Weierstrass theorem [8], it can be shown that the set of all polynomial functions from $\mathbf{Z}$ into $\mathbf{R}$ is dense in $\mathbf{C}^0(\mathbf{Z})$. This means that any continuous function can be approximated to within an arbitrary accuracy by a polynomial function with a sufficient size.

The following polynomial decision function can therefore be employed as an approximate realisation of $f_{de}(\,)$:

$$p_\theta^l(\mathbf{o}(t)) = c_0 + \sum_{i_1=1}^{m} c_{i_1}\, o(t - i_1 + 1) + \sum_{i_1=1}^{m}\sum_{i_2=i_1}^{m} c_{i_1 i_2}\, o(t - i_1 + 1)o(t - i_2 + 1) + \cdots + \sum_{i_1=1}^{m}\cdots\sum_{i_l=i_{l-1}}^{m} c_{i_1 \cdots i_l}\, o(t - i_1 + 1) \cdots o(t - i_l + 1) = \sum_{i=1}^{n_\theta} \theta_i\, y_i(t) \qquad (10)$$

where l is the polynomial degree, the $y_i(t)$ are monomials of $o(t), \ldots, o(t - m + 1)$ from degree-0 (constant 1) up to degree-l ($o(t - i_1 + 1) \cdots o(t - i_l + 1)$) and the θ_i are the corresponding coefficients c_0 to $c_{i_1 \cdots i_l}$. The number of all the coefficients is given by

$$n_\theta = \sum_{i=0}^{l} n_i,\ n_0 = 1,\ n_i = n_{i-1}(m + i - 1)/i,\ i = 1, \ldots, l \qquad (11)$$

The polynomial expansion, eqn. 10, is also known as the Volterra series.

4 Polynomial-perceptron equaliser

The polynomial decision function (eqn. 10) can be implemented by first expanding the input space into an extended nonlinear space and then employing a linear combiner structure on this space. Notice that what really matters is the sign of $p_\theta^l(\,)$. If $p_\theta^l(\,)$ can always realise the same sign of $f_{de}(\,)$, the optimal performance is achieved. Based on this observation the following polynomial-perceptron equalser is introduced:

$$\hat{x}(t - d) = \mathrm{sgn}\,(g_s(p_\theta^l(\mathbf{o}(t)))) = \mathrm{sgn}\left(g_s\left(\sum_{i=1}^{n_\theta} \theta_i\, y_i(t)\right)\right) \qquad (12)$$

where

$$g_s(y) = \tanh\,(\alpha y/2) = \frac{1 - \exp\,(-\alpha y)}{1 + \exp\,(-\alpha y)} \quad \alpha > 0 \qquad (13)$$

Notice that $g_s(\sum \theta_i\, y_i(t))$ has the structure of a single perceptron [5] with a sigmoid activation function given in eqn. 13. The need to include such a nonlinear activation function is explained in Section 5 and the particular choice of the sigmoid function (eqn. 13) reflects the bipolar nature of the transmitted signal $x(t)$.

The structure of the polynomial-perceptron equaliser is specified by the equaliser order m and the polynomial degree l. Fig. 5 shows a detailed implementation for $m = 2$ and $l = 3$. From eqn. 11 it is seen that the number of parameters increases exponentially as l increases. Our experience suggests that, in practice, restricting $l = 3$ or 5 is often adequate, and this is also supported by the other results of the authors in the field of nonlinear systems identification. The selection of the equaliser order is to a large extent influenced by a phenomenon called noise enhancement. In a high noise level situation it is preferred to employ a low equaliser order. More quantified discussion is given in Section 6.

The polynomial-perceptron equaliser is computationally more demanding compared with a simple linear structure. Increasing computation complexity and dimensionality is a common price for employing a nonlinear

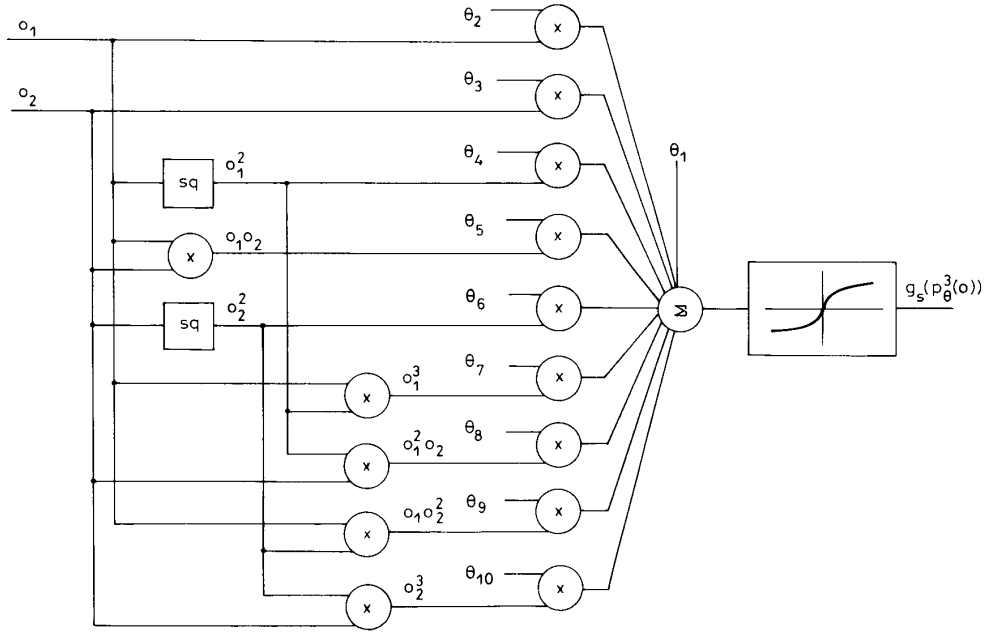

Fig. 5 *Implementation of polynomial-perceptron structure*
Input order $m = 2$ and polynomial degree $l = 3$

architecture. The structure and operation of the polynomial-perceptron equaliser are, however, simpler than those of the multilayer perceptron equaliser [3].

4.1 Training algorithms

The training of the equaliser (eqn. 12) can be carried out either by the stochastic gradient algorithm

$$\theta_i(t+1) = \theta_i(t) + \beta\bar{\varepsilon}(t)\frac{\alpha}{2}(1 - z^2(t))y_i(t) \quad 1 \leqslant i \leqslant n_\theta \tag{14}$$

or by the smoothed stochastic gradient algorithm

$$\left.\begin{aligned}\Delta_i(t+1) &= \gamma\Delta_i(t) + \beta\bar{\varepsilon}(t)\frac{\alpha}{2}(1 - z^2(t))y_i(t)\\ \theta_i(t+1) &= \theta_i(t) + \Delta_i(t+1)\end{aligned}\right\} 1 \leqslant i \leqslant n_\theta \tag{15}$$

where

$$z(t) = g_s\left(\sum_{i=1}^{n_\theta}\theta_i(t)y_i(t)\right)$$

and

$$\bar{\varepsilon}(t) = x(t-d) - z(t) \tag{16}$$

β and γ are the adaptive gain and momentum constant, respectively, $\bar{\varepsilon}(t)$ is the error signal and $0.5\alpha(1 - z^2(t))y_i(t)$ is the gradient of $z(t)$ with respect to $\theta_i(t)$. Eqn. 15 is referred to as the back propagation algorithm in the neural network context [7]. Using a smoothed stochastic gradient usually improves the performance at the cost of more computation in each recursion. During data transmission, $x(t-d)$ is substituted by its estimate $\hat{x}(t-d)$ and the algorithm of eqn. 14 or eqn. 15 can continuously be employed to track a time varying environment. The computational complexity of the algorithm of eqn. 14 or eqn. 15 can be shown to be an order of n_θ.

4.2 Simulation results

In all the cases, the algorithm of eqn. 15 was used in the training and the adaptive gain and momentum constant were set to $\beta = 0.001$ and $\gamma = 0.8$.

The ability of the polynomial-perceptron equaliser to form nonlinear decision regions is illustrated using examples 1 and 2. The parameter α for the sigmoid function (eqn. 13) was chosen to be $\alpha = 1.0$, and the equaliser order was given as $m = 2$.

For the channel of example 1 with a noise variance 0.2, the equaliser has the structure of $l = 3$ ($n_\theta = 10$) and $d = 0$. Fig. 6 gives the decision region formed by this equaliser after training, where it is seen that the decision region shows a close correspondence with the optimal equaliser.

For example 2, a trained polynomial-perceptron equaliser of zero delay and $l = 5$ ($n_\theta = 21$) produces the decision region depicted in Fig. 7. By introducing a delay $d = 1$ into this equaliser, it generates the decision region given in Fig. 8 after training.

A third example is given to compare the bit error rates achieved by the optimal and polynomial-perceptron equalisers for different signal-to-noise ratios.

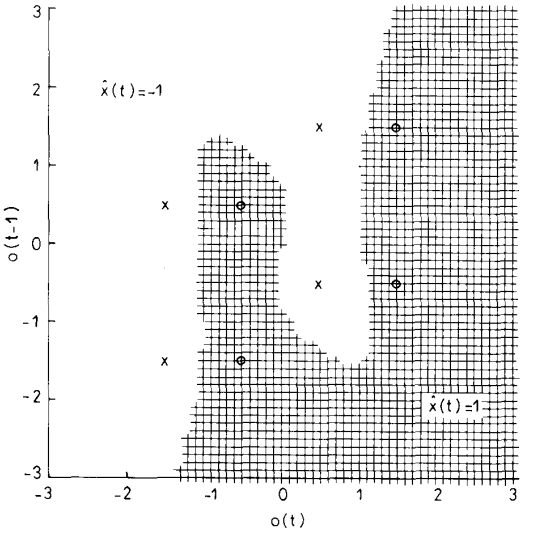

Fig. 7 *Decision region formed by polynomial-perceptron equaliser*
Channel $0.5 + 1.0z^{-1}$; additive Gaussian white noise with variance 0.2; equaliser order $m = 2$; polynomial degree $l = 5$ and delay $d = 0$

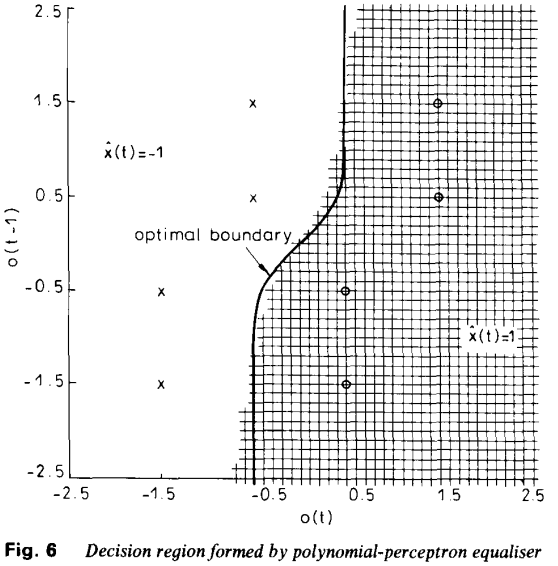

Fig. 6 *Decision region formed by polynomial-perceptron equaliser*
Channel $1.0 + 0.5z^{-1}$; additive Gaussian white noise with variance 0.2; equaliser order $m = 2$; polynomial degree $l = 3$ and delay $d = 0$

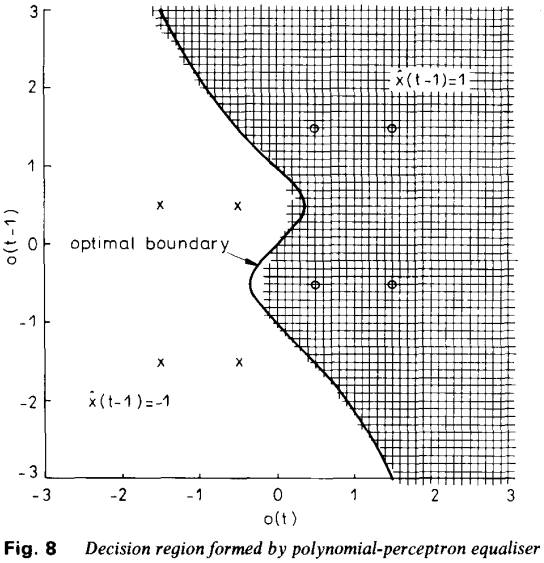

Fig. 8 *Decision region formed by polynomial-perceptron equaliser*
Channel $0.5 + 1.0z^{-1}$; additive Gaussian white noise with variance 0.2; equaliser order $m = 2$; polynomial degree $l = 5$ and delay $d = 1$

For Example 3, the channel model is $H_2(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$. The equaliser employed the structure of order $m = 4$, polynomial degree $l = 3$ ($n_\theta = 35$) and delay $d = 1$.

The results obtained are displayed in Fig. 9, where the bit error rate was computed over 500 000 points of different realisations of stochastic processes $x(t)$ and $e(t)$. α was set to 1.0 when noise level was high and was gradually increased to 8.0 as the signal-to-noise ratio improved.

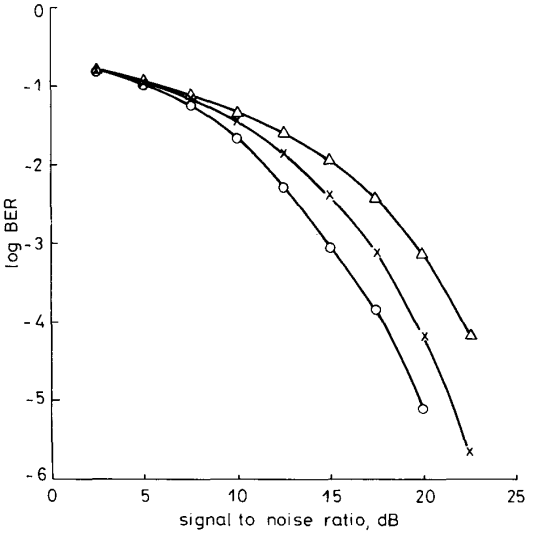

Fig. 9 *Performance comparison*
Channel $0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$; equaliser order $m = 4$; polynomial degree $l = 3$ and delay $d = 1$
—○— optimal equaliser
—×— polynomial-perceptron equaliser
—△— polynomial equaliser (without sigmoid activation)

A similar simulation study was given by Gibson *et al.* [4] for the multi-layer perceptron equaliser and the results were very close to the present simulation study. The training of a polynomial-perceptron equaliser is, however, much easier compared with that of a multilayer perceptron equaliser.

5 The need for a sigmoid activation

It may be asked whether it is really necessary to introduce the sigmoid activation (eqn. 13). A direct minimisation of the mean square error

$$E[\varepsilon^2(t)] = E\left[(x(t-d) - \sum_{i=1}^{n_\theta} \theta_i y_i(t))^2\right] \tag{17}$$

where $E[\,]$ is the expectation operator, would appear to be a better approach because eqn. 17 is quadratic in the parameters θ_i. The least mean square algorithm

$$\theta_i(t+1) = \theta_i(t) + \beta\tilde{\varepsilon}(t)y_i(t) \quad 1 \leqslant i \leqslant n_\theta \tag{18}$$

or its momentum version

$$\left.\begin{aligned}\Delta_i(t+1) &= \gamma\Delta_i(t) + \beta\tilde{\varepsilon}(t)y_i(t)\\ \theta_i(t+1) &= \theta_i(t) + \Delta_i(t+1)\end{aligned}\right\} 1 \leqslant i \leqslant n_\theta \tag{19}$$

would be capable of achieving the single global minimum of eqn. 17, where

$$\tilde{\varepsilon}(t) = x(t-d) - \sum_{i=1}^{n_\theta} \theta_i(t)y_i(t) \tag{20}$$

This approach is suggested in [6] as a viable alternative to the multilayer perceptron structure.

In the channel equalisation setting of Fig. 1, unless $p_\theta^l(\,)$ is closely matched to $f_{de}(\,)$, that is unless a very high polynomial degree l is used, the single global minimum of eqn. 17 may correspond to a bit error rate far away from the optimal bit error rate. This is because the minimum mean-square-error solution of eqn. 17 does not necessarily correspond to the best classification accuracy and may even produce a fallacious decision function if l is not large enough. Because an analytical solution is very difficult, if not impossible, to obtain even for a simple channel equalisation example, we shall use a relevant two-state classification problem to illustrate this aspect.

The example considered is a simple classifer, the input of which is a scalar x uniformly distributed within the interval $[-1, 1]$. The desired output is given as

$$d(x) = \begin{cases} 1 & x \in [0, 0.5] \\ -1 & \text{otherwise} \end{cases}$$

The classifier function is chosen to be a quadratic function

$$p_\theta^2(x) = \theta_1 + \theta_2 x + \theta_3 x^2 \tag{21}$$

which classifies an input x according to

$$\text{sgn}\,(p_\theta^2(x)) = \begin{cases} 1 & p_\theta^2(x) \geqslant 0 \\ -1 & p_\theta^2(x) < 0 \end{cases}$$

It is straightforward to show that the minimum mean-square-error solution is

$$p_\theta^2(x) = (-3 + 36x - 135x^2)/96 \tag{22}$$

which gives the minimum mean-square error -2.779 dB. This is, however, a fallacious classification function having 25% misclassification as can be seen from Fig. 10.

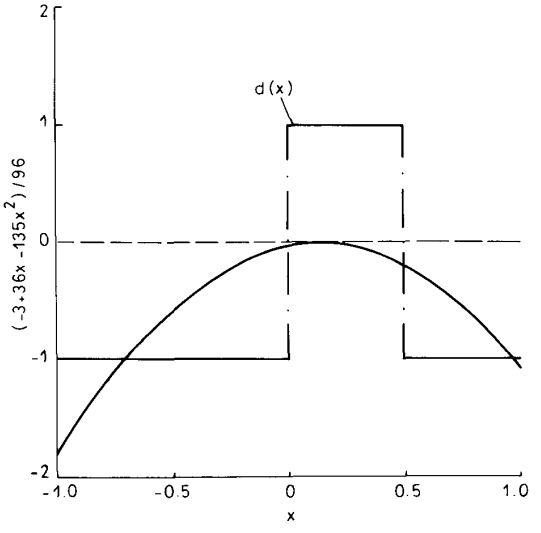

Fig. 10 *Minimum mean-square-error quadratic classifier*

Notice that $p_\theta^2(x) < 0$ for all $x \in [-1, 1]$ and, therefore, all $x \in [0, 0.5]$ are misclassified. Because the mean-square-error surface contains the single minimum of eqn. 22, the quadratic classifier of eqn. 21, trained by gradient-based algorithms, will converge to this solution and this has been confirmed in our simulation study. The correct quadratic classifier for this problem does exist and in fact gives 0% misclassification. It can easily be written down

as

$$p_B^2(x) = \rho(x - 2x^2) \quad \rho > 0 \tag{23}$$

This class of classifiers all produce a mean-square error larger than -2.779 dB. The case of $\rho = 20$, for example, gives a huge mean-square error of 26.294 dB.

By introducing the nonlinearity tanh ( ), the landscape of the mean square error surface is changed dramatically. The classifier tanh $(20(x - 2x^2))$, depicted in Fig. 11, for

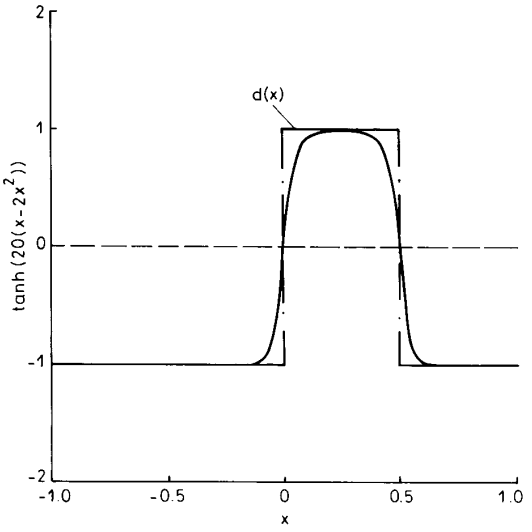

Fig. 11 *Quadratic tanh classifier*

instance, produces an extremely small mean-square error of -14.069 dB. The mean-square-error surface for the quadratic-tanh classifier function

$$z(x) = \tanh(\theta_1 + \theta_2 x + \theta_3 x^2) \tag{24}$$

may contain many minima and their analytical solutions are not easily obtained. Moreover, it can be shown that the mean-square error for tanh $(\rho(x - 2x^2))$ will tend to zero as ρ tends to infinity. We shall not address the detailed analysis of the mean-square-error surface for this example in the present study. Rather we point out that, as long as the absolute values of the initial parameters are not chosen to be too large, the quadratic tanh classifier (eqn. 24), trained by gradient-based algorithms will converge to tanh $(\rho(x - 2x^2))$, where the particular value of ρ depends on the chosen initial parameter values. This has been observed during an intensive simulation study. The reason for not choosing too large initial parameter values is because tanh function may otherwise become saturate over the whole interval $[-1, 1]$ or part of it. Two such examples are $(\theta_1, \theta_2, \theta_3) = (10, 10, 20)$ and $(\theta_1, \theta_2, \theta_3) = (0, 10, 20)$. In the former case, the gradient component $(d(x) - z(x))(1 - z^2(x))$ is virtually zero over $[-1, 1]$, and thus no training will actually take place. In the latter case, the gradient is virtually zero for $x \geqslant 0.5$, and training will not take place in this part of the interval.

It is seen that, although the sigmoid function does complicate the mean-square-error surface, at least the classifier of eqn. 24 will converge to the correct solution when initial parameters are inside a certain sphere. This is in contrast to the pure quadratic classifier of eqn. 21 which always converges to the wrong solution (eqn. 22).

We emphasise that the real criterion is the classification accuracy and the mean square error criterion is only a too for training a classifier to obtain, hopefully, an acceptable level of misclassification. Multiminima of the mean square error, introduced by the inclusion of the tanh function, may not always be bad and they may actually improve the flexibility of the classifier, as shown here. The alternative to this is to increase the polynomial degree sufficiently and to suffer the consequence of filter-dimension explosion (terms increase exponentially as l increases). Our experience shows that, by introducing the tanh function, we can restrict l to be 3 or 5. The resulting classifier or equaliser is able to realise complicated decision regions, such as the one shown in Fig. 3.

A second difficulty associated with the direct polynomial approach is that the gain β in eqn. 18 or eqn. 19 often has to be restricted to an extremely small value in order to guarantee convergence. This can easily be understood because $E[\mathbf{y}(t)\mathbf{y}^T(t)]$, where $\mathbf{y}^T(t) = [y_1(t) \cdots y_{n_0}(t)]$, is often very ill-conditioned and has a large range of eigenvalues. For Example 3, to guarantee convergence for all the signal to noise ratios tested, the gain in eqn. 19 had to be reduced to $\beta = 0.0001$ and the performance achieved using this algorithm is also given in Fig. 9. It is seen that the sigmoid activation introduced in the equaliser of eqn. 12 is indeed required.

6 Performance of the linear equaliser

For the channel and equaliser delay specified in Example 3, the performance of the linear equaliser of order 4 is plotted in Fig. 12 where it is also shown that the

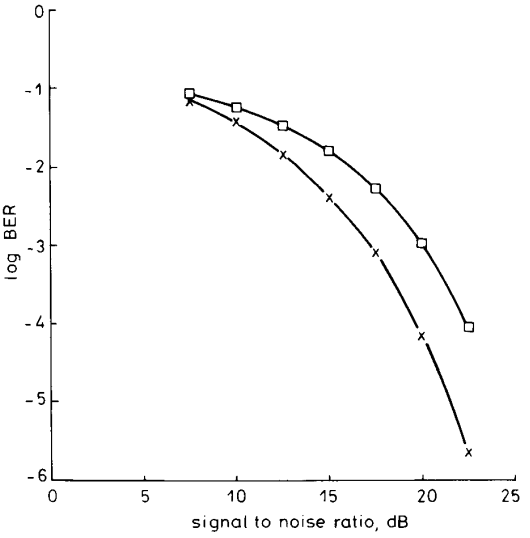

Fig. 12 *Performance comparison*
Channel $0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$, equaliser order $m = 4$ and delay $d = 1$
—×— polynomial-perceptron equaliser ($l = 3$)
—□— linear equaliser

polynomial-perceptron equaliser of the same order significantly improves the bit error rate over the linear equaliser. It might be argued that such a comparison is unfair since a linear equaliser of order 4 only has four tap weights compared with 35 tap weights for the polynomial-perceptron equaliser of $l = 3$. We now examine whether we can improve the performance of the linear equaliser by simply increasing its order.

Because the weight vector of the linear equaliser after convergence should approximate the Wiener optimal filter of same order, the bit error rate achievable by the linear equaliser can therefore be predicted from that of

the Wiener filter. Under the conditions given in Section 2, the Wiener filter weight vector

$$\hat{w} = [\hat{w}_1 \quad \cdots \quad \hat{w}_m]^T \tag{25}$$

can be easily obtained. The bit error rate of the Wiener filter is defined by

$$\text{Prob}\ \{\hat{\mathbf{w}}^T\mathbf{o}(t) < 0 \,|\, \hat{\mathbf{o}}(t) \in P_{m,d}(1)\}$$

or

$$\text{Prob}\ \{\hat{\mathbf{w}}^T\mathbf{o}(t) > 0 \,|\, \hat{\mathbf{o}}(t) \in P_{m,d}(-1)\} \tag{26}$$

It is not difficult to compute the probability of eqn. 26 because $\hat{\mathbf{w}}^T\mathbf{o}(t)$ is Gaussian distributed with mean $\hat{\mathbf{w}}^T\hat{\mathbf{o}}(t)$ and variance $\sigma_e^2\hat{\mathbf{w}}^T\hat{\mathbf{w}}$, where σ_e^2 is the variance of additive noise $e(t)$. For the channel and equaliser delay defined in Example 3, Fig. 13 shows the relationship between the

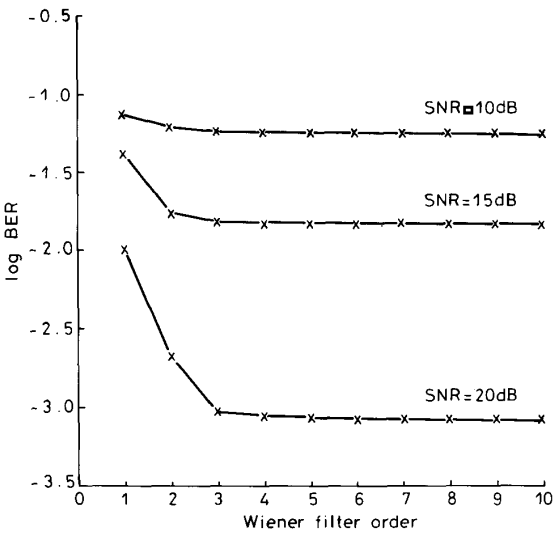

Fig. 13 *Bit error rate versus Wiener filter order*
Channel $0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$ and equaliser delay $d = 1$

theoretical bit error rate (eqn. 26) and the Wiener filter order in a variety of noise conditions. It is clear that the performance achievable by simply increasing the order of the linear equaliser cannot match the polynomial-perceptron equaliser of low order. Furthermore, little advantage can be gained in a noisy environment by employing a linear equaliser which has an order greater than 4 for this example.

The phenomenon shown in Fig. 13 is known as the noise enhancement. As the order of the equaliser increases, the total noise power on the equaliser input is also increased and this tends to diminish any advantage gained by increasing the equaliser order. On the contrary, it could be argued that increasing the order may only lead to an increase in complexity, training time and misadjustment, and ultimately a decrease in efficiency, in a high noise environment. The above results provide further justification for considering nonlinear equalisers of low order in high noise conditions.

7 Conclusions

By viewing the communications channel equalisation as a classification problem, the optimal equalisation solution has been derived, based on the Bayes decision rule. It has been shown that an equaliser which incorporates some degree of nonlinear decision making ability can achieve a bit error rate superior to that offered by linear equalisers. A polynomial-perceptron structure employing a sigmoid activation has been considered as an adaptive equaliser which is capable of approximating the optimal equaliser solution.

The complexity of the polynomial-perceptron equaliser is determined by the two structure parameters, namely, equaliser order and polynomial degree. Practical selection of polynomial degree has been discussed and it has been shown that employing a low equaliser order is justified in poor signal to noise ratio conditions.

8 Acknowledgments

This work was supported by the UK Science and Engineering Research Council (Grant GR/E/10357). The authors wish to thank the referees for their valuable comments on the manuscript.

9 References

1 CHEN, S., and BILLINGS, S.A.: 'Representation of non-linear systems: the NARMAX model', *Int. J. Control*, 1989, **49**, (3), pp. 1013–1032
2 CHEN, S., GIBSON, G.J., COWAN, C.F.N., and GRANT, P.M.: 'Recursive prediction error algorithm for training multilayer perceptrons'. Proceedings of IEEE Colloquium on Adaptive Algorithms for Signal Estimation and Control, September 13th 1989, Edinburgh, Scotland
3 CHEN, S., GIBSON, G.J., COWAN, C.F.N., and GRANT, P.M.: 'Adaptive equalisation of finite non-linear channels using multilayer perceptrons', *Signal Processing*, 1990, **20**, (2), pp. 107–119
4 GIBSON, G.J., SIU, S., and COWAN, C.F.N.: 'Application of multilayer perceptrons as adaptive channel equalisers'. Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing, May 23–26 1989, Glasgow, Scotland, pp. 1183–1186
5 LIPPMANN, R.P.: 'An introduction to computing with neural nets', *IEEE ASSP Magazine*, **4**, 1987
6 RAYNER, P.J.W., and LYNCH, M.R.: 'A new connectionist model based on a non-linear adaptive filter'. Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing, May 23–26 1989, Glasgow, Scotland, pp. 1191–1194
7 RUMELHART, D.E., HINTON, G.E., and WILLIAMS, R.J.: 'Learning internal representations by error propagation', *in*: RUMELHART, D.E., and McCLELLAND, J.L. (Eds.): 'Parallel distributed processing: exploration in the microstructure of congnition' (MIT Press, 1986), pp. 318–362
8 SIMMONS, G.F.: 'Introduction to topology and modern analysis' (McGraw-Hill, New York, 1963)
9 SINGHAL, S., and WU, L.: 'Training feed-forward networks with the extended Kalman algorithm'. Proc. of IEEE Int. Conference on Acoustics, Speech and Signal Processing, May 23–26 1989, Glasgow, Scotland, pp. 1187–1190
10 SPECHT, D.F.: 'Generation of polynomial discriminant functions for pattern recognition', *IEEE Trans.*, 1967, **EC-16**, (3), pp. 308–319