Signal Processing 20 (1990) 107–119
Elsevier

ADAPTIVE EQUALIZATION OF FINITE NON-LINEAR CHANNELS USING MULTILAYER PERCEPTRONS

S. CHEN, G. J. GIBSON, C. F. N. COWAN and P. M. GRANT
Department of Electrical Engineering, University of Edinburgh, Mayfield Road, Edinburgh EH9 3JL, Scotland

Received 23 August 1989
Revised 22 November 1989

Abstract. Adaptive equalization of channels with non-linear intersymbol interference is considered. It is shown that difficulties associated with channel non-linearities and additive noise correlation can be overcome by the use of equalizers employing a multi-layer perceptron structure. This provides further evidence that the neural network approach proposed recently by Gibson et al. is a general solution to the problem of equalization in digital communications systems.

Zusammenfassung. Betrachtet wird die adaptive Entzerrung von Kanälen mit nichtlinearer Intersymbol-Interferenz. Schwierigkeiten aufgrund von Kanälen mit Nichtlinearitäten und korreliertem additivem Rauschen können überwunden werden, indem Entzerrer mit einer Mehrlagen Perceptron-Struktur verwendet werden. Dies ist ein weiterer Hinweis darauf, daß der kürzlich vorgeschlagene Ansatz auf der Basis neuraler Netzwerke eine allgemeine Lösung der Entzerrungsaufgabe in digitalen Kommunikationssystemen ist (sehe [Gibson et al.]).

Résumé. On considère l'égalisation adaptative de canaux introduisant des interférences intersymboles non-linéaires. On montre qu'en utilisant une une structure de perceptron multi-couche, les non-linéarités du canal et l'influence d'un bruit additif coloré peuvent être corrigées avec succés. Par ailleurs, cela fournit la preuve que l'approche des réseaux de neurones proposée récemment est une solution générale au problème de l'égalisation dans les systèmes de communications numériques (cf. [Gibson et al.]).

Keywords. Adaptive non-linear systems, channel equalization, neutral networks.

1. Introduction

High-speed data transmission over channels having substantial amplitude and delay distortion has become a common practice due to the development of adaptive equalization techniques based on the linear finite impulse response (FIR) channel model. Non-linear distortion is now a significant factor hindering further increase in the attainable data rate. Although sources of channel non-linearity such as non-linearity in data converters may be regarded as memoryless, these non-linear components are connected to or embedded in a linear dynamic network and, consequently, the overall channel response is a non-linear dynamic mapping. That is, the received signal at each sample instant is a non-linear function of the past values of the transmitted symbols. Because non-linear distortion varies with time and from place to place, effective non-linear compensation should be adaptive.

Channel equalization can be considered as a deconvolution problem where an equalizer is constructed such that the impulse response of the channel and equalizer combination is as close to a delta function as possible. This approach is very difficult to extend to correct non-linear intersymbol interference, and so far only a few techniques have been proposed based on a linear equalizer coupled with some non-linear decision feedback or similar

cancellation schemes [1]. By contrast, channel equalization can also be viewed as a classification problem where an equalizer is constructed as a decision-making device to reconstruct the transmitted symbol sequence as accurately as possible. Gibson et al. [6] adopted this second approach and, by formulating the equalization problem in a geometric setting, derived an adaptive equalizer which employed a neural network architecture, namely, that of the multi-layer perceptron (MLP) [7]. Although the original intention of [6] was to equalize FIR channels with additive white noise, the concepts developed are readily applicable to finite non-linear channels with additive coloured noise, and this is demonstrated in the present study.

The paper is organized as follows. Section 2 introduces a general non-linear channel model. In Section 3, the minimum bit error rate (BER) equalizer of [6] is first described. By viewing this optimal-BER equalizer as a classification mapping, the influence of channel non-linearities and coloured noise are then studied. Using simple examples, it is shown that non-linearities and coloured noise only change the location of the classification boundary, and do not fundamentally alter the nature of the equalization problem. The ability of neural networks to realize a wide variety of classification mappings [4, 5, 7] provides the basis for their use as adaptive equalizers and Section 4 briefly summarizes the application of the MLP as a channel equalizer. Some simulation results are shown in Section 5 and final concluding remarks are given in Section 6. Throughout the discussion the transmitted data are assumed to be binary taking values of either 1 or -1. The approach is, however, not restricted to this specific signalling method.

2. Channel model

The digital communications system considered in this paper is shown in Fig. 1, where the 'channel' includes the effects of the transmitter filter, the transmission medium, the receiver matched filter and other components. The transmitted data sequence $x(t)$ is assumed to be an independent sequence taking values from $\{-1, 1\}$ with an equal probability. The channel output $\hat{o}(t)$ is corrupted by an additive noise $e(t)$. The observations $o(t), \ldots, o(t-m+1)$ are passed to a channel equalizer. The task of the equalizer at the sampling instant t is to produce an estimate of the input symbol $x(t-d)$ using the information contained in $o(t), \ldots, o(t-m+1)$, where the integers m and

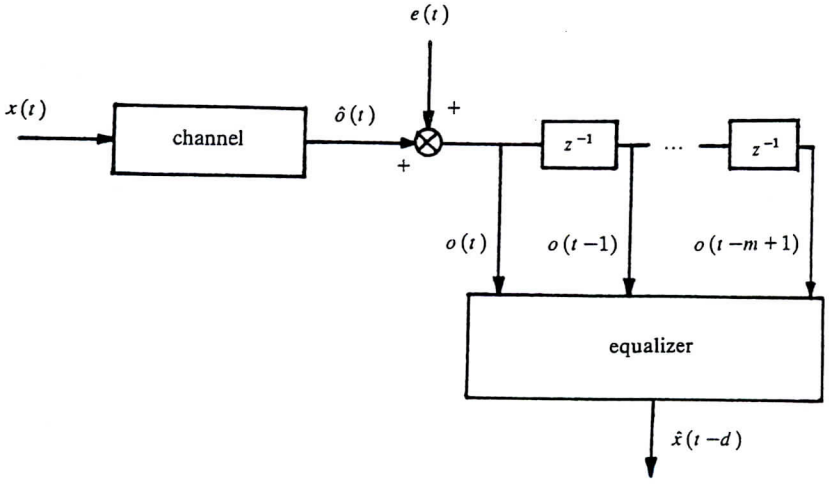

Fig. 1. Schematic of data transmission system.

d are known as the order and the delay of the equalizer, respectively.

A widely used model for a linear dispersive channel is the FIR model, which can be written as

$$o(t)=\sum_{i=0}^{n}\theta_i x(t-i)+e(t), \tag{1}$$

where the θ_i are channel parameters. It is usually assumed that $e(t)$ is a Gaussian distributed white noise with mean zero and variance σ_e^2. The white noise assumption is, however, only an approximation because, although the channel noise itself is generally Gaussian and white, the matched filter at the receiver end will colour the noise.

If non-linear distortions are taken into account, the general channel model should be non-linear and can be represented as

$$o(t)=g(x(t),\ldots,x(t-n);\Theta)+e(t), \tag{2}$$

where $g(\cdot)$ is some non-linear function and

$$\Theta=[\theta_1\cdots\theta_{n_\theta}]^T \tag{3}$$

is a channel parameter vector. Due to the physical restriction, the dynamics of model (2) are always stable. That is, a bounded input $x(t)$ can only lead to a bounded output $\hat{o}(t)$ where

$$\hat{o}(t)=g(x(t),\ldots,x(t-n);\Theta). \tag{4}$$

It is also clear that, if the states of $x(t),\ldots,x(t-n)$ are finite, $\hat{o}(t)$ can only take finite values. Both the functional form $g(\cdot)$ and Θ can be time-varying. The finite non-linear channel model (2) is obviously a generalization of the FIR model (1). The non-linear model considered in [1], for example, can be written in the form of (2) with $g(\cdot)$ chosen to be a degree-3 polynomial of $x(t),\ldots,x(t-n)$.

3. Influence of channel non-linearities and coloured noise

This section describes a geometric formulation of the equalization problem due to [6], and studies the effects of channel non-linearities and coloured noise to the equalization problem. Using similar notation to that in [6], define

$$\begin{aligned}P_{m,d}(1)&=\{\hat{\boldsymbol{o}}(t)\in\mathbb{R}^m\,|\,x(t-d)=1\},\\ P_{m,d}(-1)&=\{\hat{\boldsymbol{o}}(t)\in\mathbb{R}^m\,|\,x(t-d)=-1\},\end{aligned} \tag{5}$$

where $\mathbb{R}^m$ is the m-dimensional Euclidean space and

$$\hat{\boldsymbol{o}}(t)=[\hat{o}(t)\cdots\hat{o}(t-m+1)]^T, \tag{6}$$

$P_{m,d}(1)$ and $P_{m,d}(-1)$ represent the two sets of possible channel noise-free output vectors $\hat{\boldsymbol{o}}(t)$ that can be produced from sequences of channel inputs containing $x(t-d)=1$ and $x(t-d)=-1$, respectively. The equalizer can be characterized by the function

$$h:\mathbb{R}^m\to\{-1,1\}, \tag{7}$$

with

$$\hat{x}(t-d)=h(\boldsymbol{o}(t)), \tag{8}$$

where $\boldsymbol{o}(t)=[o(t)\cdots o(t-m+1)]^T$ is the observed channel output vector. Notice that given values of m and d and knowledge of the channel $g(\cdot)$ and Θ, the sets $P_{m,d}(1)$ and $P_{m,d}(-1)$ are known. If the distribution of the noise $e(t)$ is further provided, the conditional density functions of observing the channel output vector $\boldsymbol{o}(t)$ given $\hat{\boldsymbol{o}}(t)\in P_{m,d}(1)$ and $\hat{\boldsymbol{o}}(t)\in P_{m,d}(-1)$, respectively, are completely specified. Denote these two conditional density functions as f_1 and f_{-1}, respectively. It has been shown in [6] that the equalizer which is defined by

$$\begin{aligned}h_0(\boldsymbol{o}(t))&=\operatorname{sgn}(f_{de}(\boldsymbol{o}(t)))\\ &=\operatorname{sgn}(f_1(\boldsymbol{o}(t))-f_{-1}(\boldsymbol{o}(t)))\end{aligned} \tag{9}$$

achieves the minimum BER for the given order m and lag d, where

$$\operatorname{sgn}(y)=\begin{cases}1, & y\geq 0,\\ -1, & y<0\end{cases} \tag{10}$$

represents a slicer. The set

$$\{\boldsymbol{o}(t)\in\mathbb{R}^m\,|\,f_{de}(\boldsymbol{o}(t))\geq 0\} \tag{11}$$

is known as the decision region of the optimal

BER equalizer and the decision boundary of this equalizer consists of the set of points

$$\{\boldsymbol{o}(t)\in\mathbb{R}^m \,|\, f_{de}(\boldsymbol{o}(t))=0\}. \tag{12}$$

Consider the linear channel model $\hat{o}(t)=x(t)+0.5x(t-1)$ with the equalizer order $m=2$ and lag $d=0$. The elements of the sets $P_{2,0}(1)$ and $P_{2,0}(-1)$ are illustrated in Fig. 2 using the symbols 'diamond' $\diamond$ and 'cross' $\times$, respectively. For a Gaussian noise $e(t)$ with

$$\tilde{\Sigma}=\begin{bmatrix} E[e^2(t)] & E[e(t)e(t-1)] \\ E[e(t-1)e(t)] & E[e^2(t-1)] \end{bmatrix} = \sigma_e^2\begin{bmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{bmatrix}, \tag{13}$$

where $E[\cdot]$ is the expectation operator, the points $\boldsymbol{o}$ of the optimal decision boundary satisfy

$$f_{de}(\boldsymbol{o})=\alpha\Big(\sum \exp\big(-\tfrac{1}{2}(\boldsymbol{o}-\hat{\boldsymbol{o}}_+)^{\mathrm{T}}\tilde{\Sigma}^{-1}(\boldsymbol{o}-\hat{\boldsymbol{o}}_+)\big) - \sum \exp\big(-\tfrac{1}{2}(\boldsymbol{o}-\hat{\boldsymbol{o}}_-)^{\mathrm{T}}\tilde{\Sigma}^{-1}(\boldsymbol{o}-\hat{\boldsymbol{o}}_-)\big)\Big) = 0. \tag{14}$$

Here α is a constant, the first sum is over all the points $\hat{\boldsymbol{o}}_+\in P_{2,0}(1)$ and the second sum over all $\hat{\boldsymbol{o}}_-\in P_{2,0}(-1)$. The optimal boundary for Gaussian white noise with $\sigma_e^2=0.2$ is illustrated in Fig. 2, where the region on the left half plane of the boundary (including the boundary) is the optimal decision region. For the same linear channel with Gaussian coloured noise $\sigma_e^2=0.2$ and $\rho_1=0.48$, the optimal boundary under the same constraints $m=2$ and $d=0$ is also shown in Fig. 2. It is seen that coloured noise clearly affects the decision region.

The influence of channel non-linearities is now investigated. The decision boundary of a non-

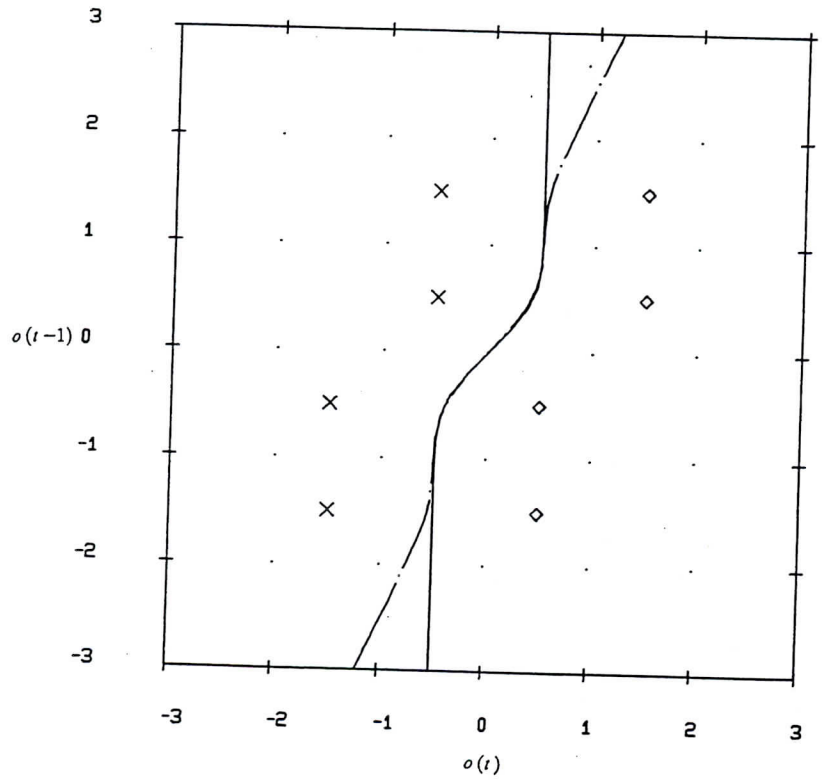

Fig. 2. Channel output points and optimal decision boundary. Channel $\hat{o}(t)=x(t)+0.5x(t-1)$, equalizer order $m=2$ and lag $d=0$; solid line: Gaussian white noise with $\sigma_e^2=0.2$; dashed line: Gaussian coloured noise with $\sigma_e^2=0.2$ and $\rho_1=0.48$.

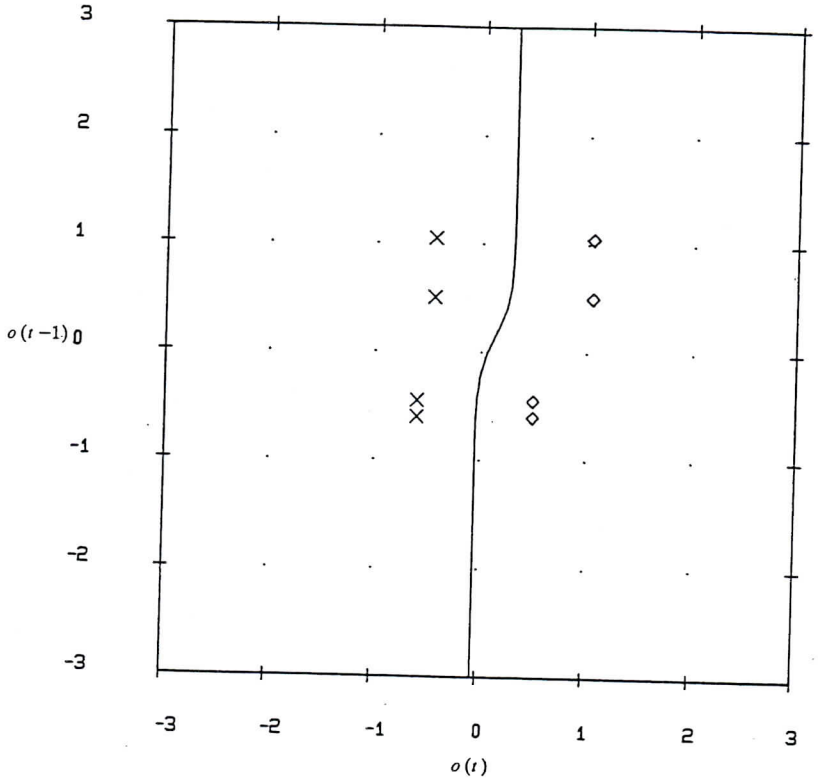

Fig. 3. Channel output points and optimal decision boundary. Channel $\hat{o}(t)=\tilde{o}(t)+0.1\tilde{o}^2(t)-0.2\tilde{o}^3(t)$ and $\tilde{o}(t)=x(t)+0.5x(t-1)$, Gaussian white noise with $\sigma_e^2=0.2$; equalizer order $m=2$ and lag $d=0$.

linear channel $\hat{o}(t)=\tilde{o}(t)+0.1\tilde{o}^2(t)-0.2\tilde{o}^3(t)$, where $\tilde{o}(t)=x(t)+0.5x(t-1)$, with Gaussian white noise is plotted in Fig. 3. In this case, channel non-linearities only slightly influence the decision region. This is however not always the case and a severe example is shown in Fig. 4 where the shaded region is the optimal decision region. It is seen that channel non-linearities dramatically change the optimal decision region. In particular the sets $P_{2,0}(1)$ and $P_{2,0}(-1)$ are no longer linearly separable. It is known that a nonzero lag d generally permits a better equalization performance. This becomes apparent by examining the sets $P_{m,d}(1)$ and $P_{m,d}(-1)$ and the optimal equalizer boundary for different values of d. For the same channel described in Fig. 4, Fig. 5 shows $P_{2,1}(1)$, $P_{2,1}(-1)$ and the corresponding optimal boundary. Theoretically non-linearities may make the elements of $P_{m,d}(1)$ and $P_{m,d}(-1)$ so close or even overlapped for certain values of d that reconstruction of the channel input sequence in a noisy situation becomes virtually impossible and this is demonstrated by Fig. 6.

Further illustration is given in Figs. 7 and 8, where the effects of channel non-linearities and coloured noise are vividly demonstrated. Although polynomial channel models and Gaussian noise are used in the above demonstration, the results are valid for the general model (2) with non-Gaussian additive noise. Some general observations can be summarized. For given equalizer order m and lag d, the equalization solution is determined by the channel model and noise statistics. Specifically $P_{m,d}(1)$ and $P_{m,d}(-1)$ are determined by the channel model $g(\cdot)$ and Θ. This channel knowledge together with the distribution of the

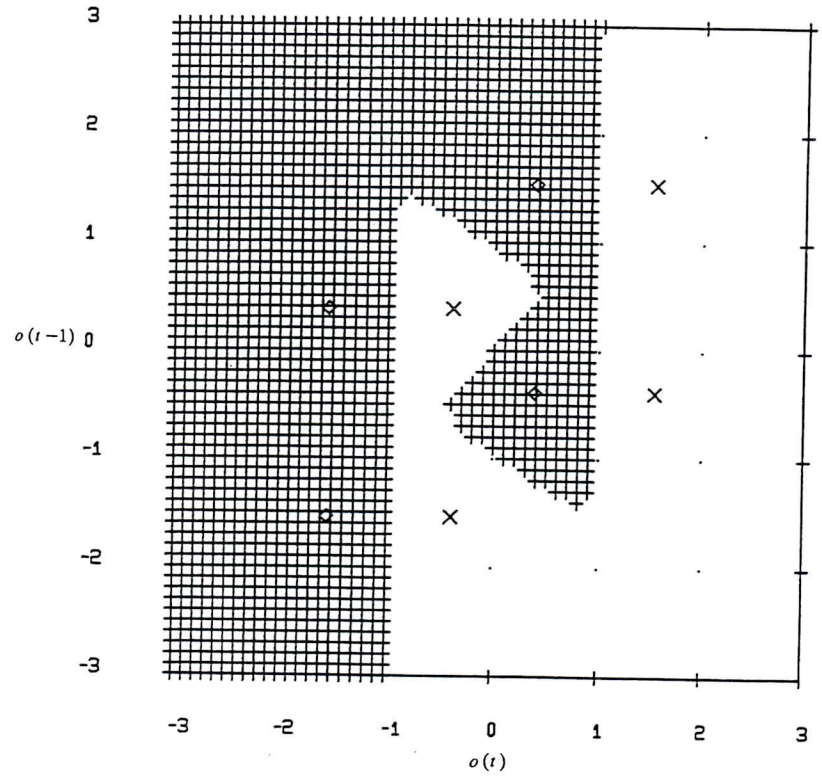

Fig. 4. Channel output points and optimal decision region. Channel $\hat{o}(t)=\tilde{o}(t)-0.9\tilde{o}^3(t)$ and $\tilde{o}(t)=x(t)+0.5x(t-1)$, Gaussian white noise with $\sigma_e^2=0.2$; equalizer order $m=2$ and lag $d=0$.

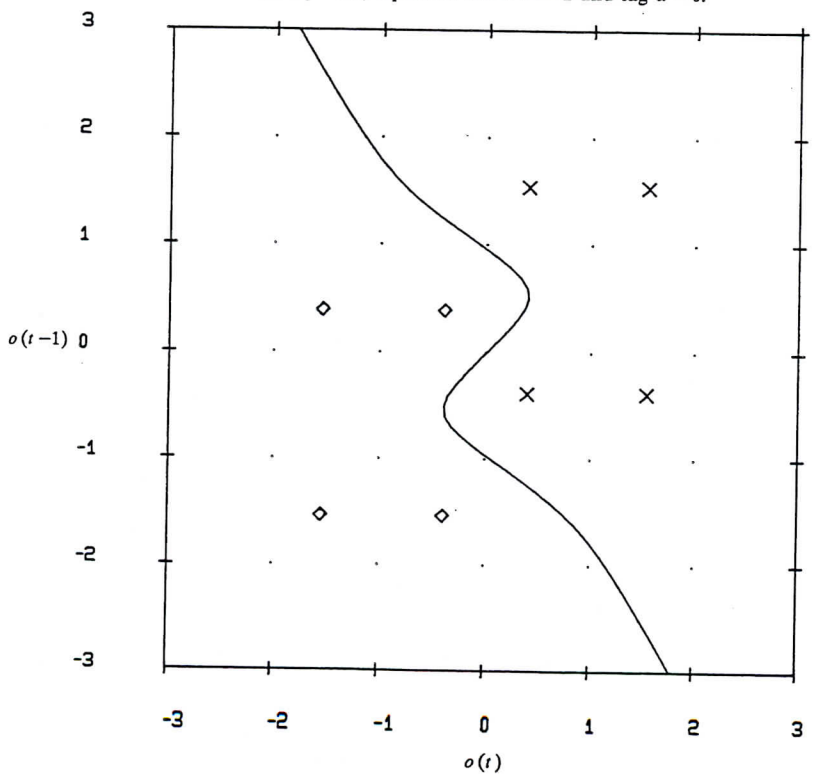

Fig. 5. Channel output points and optimal decision boundary. Channel $\hat{o}(t)=\tilde{o}(t)-0.9\tilde{o}^3(t)$ and $\tilde{o}(t)=x(t)+0.5x(t-1)$, Gaussian white noise with $\sigma_e^2=0.2$; equalizer order $m=2$ and lag $d=1$.

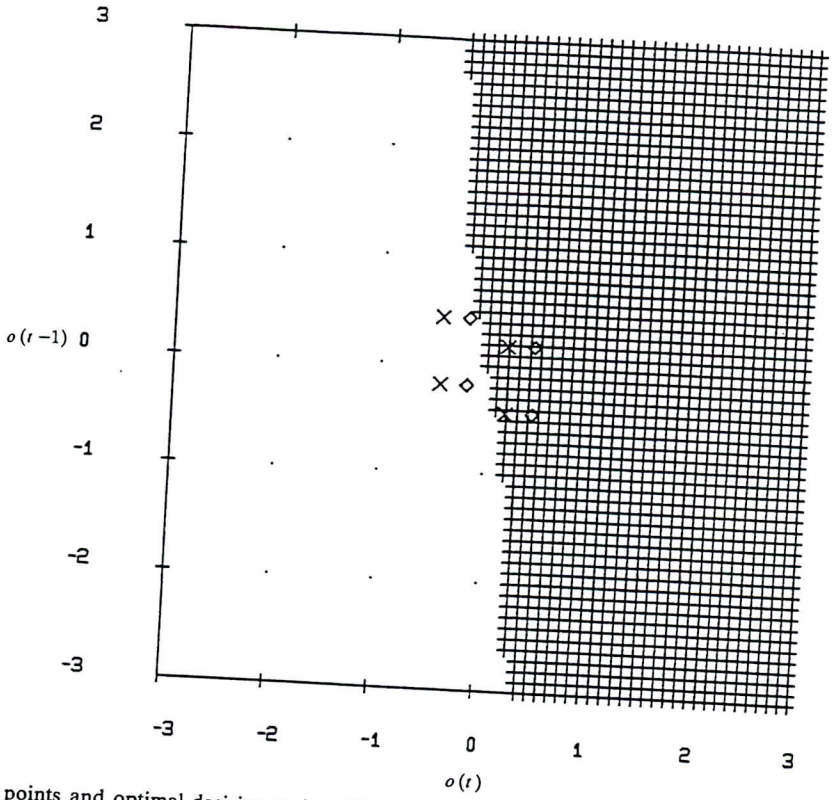

Fig. 6. Channel output points and optimal decision region. Channel $\hat{o}(t)=\tilde{o}(t)-0.5\tilde{o}^3(t)$ and $\tilde{o}(t)=x(t)+0.5x(t-1)$, Gaussian white noise with $\sigma_e^2=0.2$; equalizer order $m=2$ and lag $d=0$.

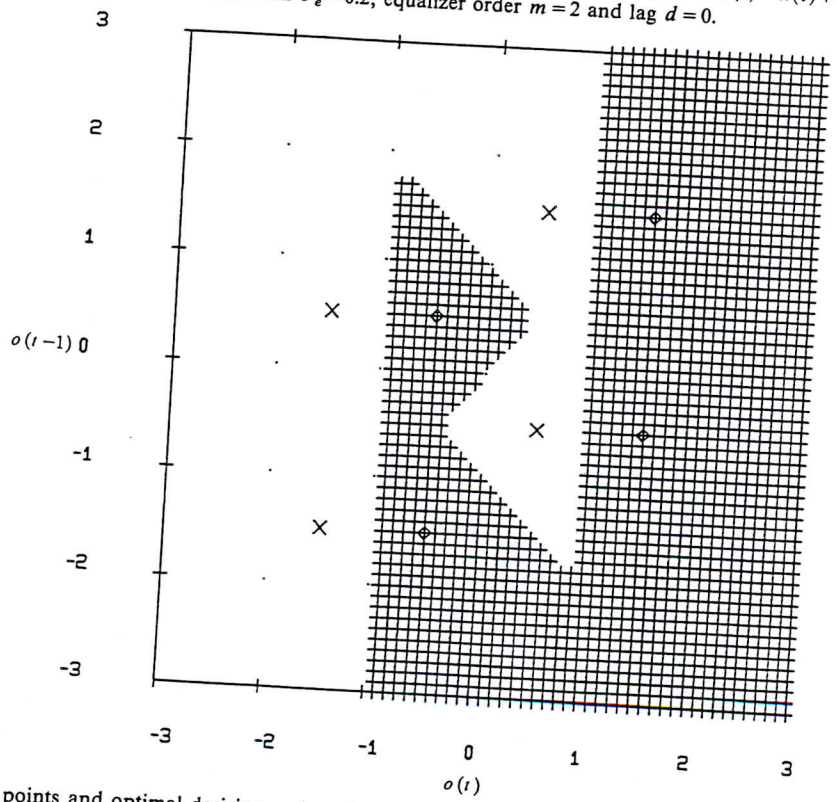

Fig. 7. Channel output points and optimal decision region. Channel $\hat{o}(t)=0.5x(t)+x(t-1)$, Gaussian white noise with $\sigma_e^2=0.2$; equalizer order $m=2$ and lag $d=0$.

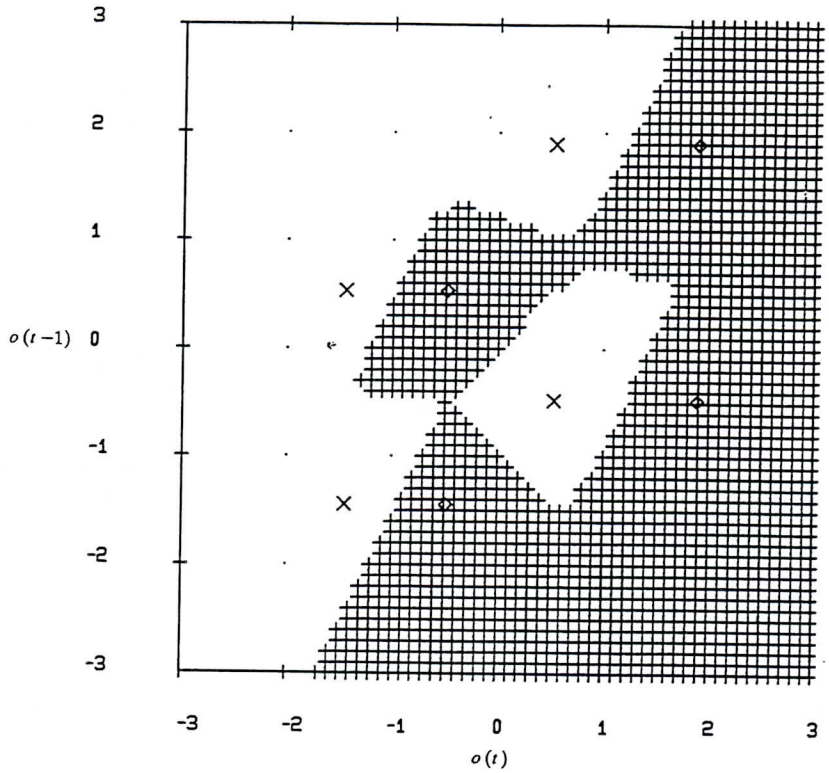

Fig. 8. Channel output points and optimal decision region. Channel $\hat{o}(t)=\tilde{o}(t)+0.1\tilde{o}^2(t)+0.05\tilde{o}^3(t)$ and $\tilde{o}(t)=0.5x(t)+x(t-1)$, Gaussian coloured noise with $\sigma_e^2=0.2$ and $\rho_1=0.48$; equalizer order $m=2$ and lag $d=0$.

additive noise completely specify the optimal decision region or the mapping $f_{de}(\cdot)$. Finally we emphasize that the optimal boundary can be highly non-linear so that any linear equalizer structure is inherently suboptimal and this motivates the investigation of non-linear architectures capable of realizing such a boundary.

4. Multi-layer perceptrons as adaptive channel equalizers

Section 3 demonstrates the need to use non-linear structures. We now examine a particular non-linear structure, namely, MLP. First its architecture and classification capabilities are briefly described. An MLP and the structure of its basic computing unit or node are shown in Fig. 9. A node computes the weighted sum of the inputs, adds a node threshold and passes the result through a non-linear function $a(\cdot)$ called the node activation function. An MLP consists of layered nodes. All the nodes in a layer are fully connected to the nodes in adjacent layers, but there is no connection between the nodes within the same layer and no bridging layer connection. The architecture of an MLP can be summarized as $m-n_1-\cdots-n_l$, where m denotes the dimension of the input space and n_i, $i=1,\ldots,l$, are the numbers of nodes in the respective layers. The input-output relationship of an MLP can therefore be described by a non-linear function $f:\mathbb{R}^m\rightarrow\mathbb{R}^{n_l}$. It can be easily verified that the computation of the classification mapping f requires N_1 multiplications, where

$$N_1=mn_1+\sum_{i=1}^{l-1}n_in_{i+1}. \tag{15}$$

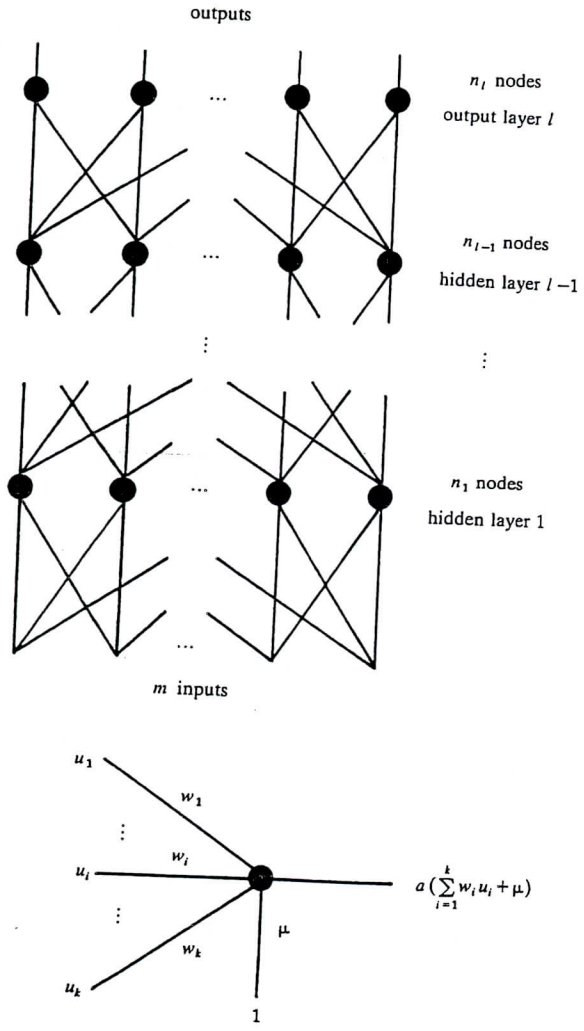

Fig. 9. Multilayer perceptron architecture and node structure.

For our channel equalization application, the output layer requires only one node. Such an MLP produces a mapping $f: \mathbb{R}^m \to \mathbb{R}$ and its decision region is the set of points

$$\{\boldsymbol{o} \in \mathbb{R}^m \,|\, f(\boldsymbol{o}) \geq 0\}. \tag{16}$$

In the simplest case, that is a single-node perceptron, the decision region is the half space bounded by the linear boundary $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{o} + \mu = 0$, where $\boldsymbol{w}$ and μ are the vector of weights and the node threshold, respectively. It is clear that the classification capability of a single-layer perceptron is the same as that of a linear equalizer and will not meet the requirement of realizing non-linear boundaries. If however the number of layers is increased to two or more, we are able to generate very complicated decision regions with highly non-linear boundaries [4, 5, 7]. This capability of the MLP provides the justification for using (16) to realize the optimal equalizer decision region (11). Although any bounded decision region in $\mathbb{R}^m$ can be realized to within any specified accuracy by a two-layer perceptron [4], a very large number of hidden-layer nodes may be required to do so. A three-layer perceptron can achieve the same classification accuracy with less hidden nodes compared with a two-layer perceptron. We therefore employ the three-layer architecture in our application.

Figure 10 illustrates the schematic of using a three-layer perceptron as an adaptive channel equalizer, where in the training mode $d(t) = x(t-d)$ and during the data transmission $d(t) =$

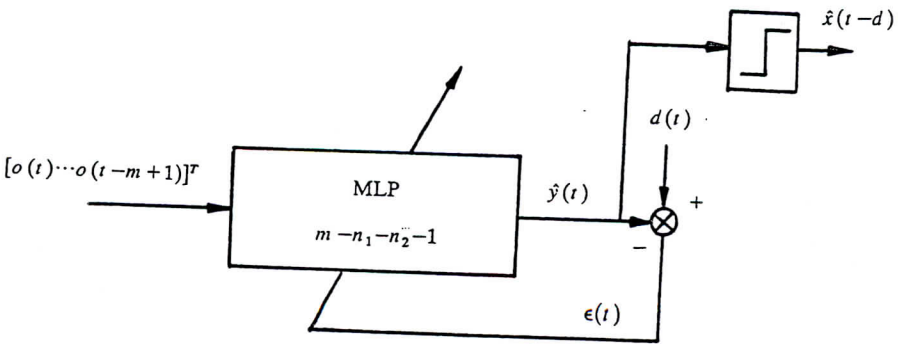

Fig. 10. Three-layer perceptron as adaptive equalizer.

$\hat{x}(t-d)$. The structure of a three-layer perceptron equalizer will be represented as $(m-n_1-n_2-1, d)$. The selection of a node activation function is not critical and a wide variety of functions can be used [4]. A practical constraint is that $a(\cdot)$ should be differentiable in order to use gradient-descent training algorithms, and a common choice of $a(\cdot)$ is the sigmoid function

$$a(y)=\frac{1}{1+\exp(-y)}. \tag{17}$$

Because in our application the transmitted data take binary values 1 or -1, it is appropriate to choose the node activation function as

$$a(y)=\frac{1-\exp(-y)}{1+\exp(-y)}. \tag{18}$$

Training an MLP to perform a particular task can be achieved either using a smoothed stochastic gradient algorithm commonly known as the back-propagation algorithm [8] or using a recursive Gauss-Newton algorithm called the recursive prediction error algorithm [2, 3]. For a feedforward type of neural network such as an MLP, both algorithms utilize the parallel structure of the network to perform two sweeps through the network at each recursion t. The input signals $o(t)$ are propagated forward on the first sweep and, as the signals pass through a layer in the network, all the nodes in that layer compute their own outputs simultaneously. The error signal $\varepsilon(t)$ obtained at the top of the network is then propagated down the network on the second sweep. As this signal, accompanied by the gradient information, propagates back through a layer, all the nodes in that layer update their own weights and threshold simultaneously. Let Ξ be the vector of all the weights and thresholds inside an MLP and denote $\nabla\varepsilon^2(t)$ as the gradient of $\varepsilon^2(t)$ with respect to Ξ. The operation of the back-propagation algorithm is as follows:

$$\begin{aligned}\Delta(t)&=\gamma\Delta(t-1)+\beta(-\nabla\varepsilon^2(t)),\\ \Xi(t)&=\Xi(t-1)+\Delta(t),\end{aligned} \tag{19}$$

where β and γ are the adaptive gain and a momentum parameter respectively, and $\Xi(t)$ is the estimate of Ξ at t. It can be shown that the computational complexity of the algorithm is an order of N_2 for an $m-n_1-\cdots-n_l$ MLP, where

$$N_2=(m+1)n_1+\sum_{i=1}^{l-1}(n_i+1)n_{i+1} \tag{20}$$

is the dimension of Ξ. Although the recursive prediction error algorithm achieves faster convergence and is less sensitive to the initial values of network weights compared with the back-propagation algorithm, it is computationally more complex. The detailed description of this algorithm can be found in [3] and therefore will not be repeated here. The development of new training algorithms is actively pursued within the neural network community. The back-propagation algorithm may be the only one that can meet the real-time requirements of high-speed data transmission at the present time.

In general there should be enough hidden nodes in order to generate a sufficiently complex classification mapping. On the other hand, if too many hidden nodes are employed, the convergence of the training process can be very slow. Because a priori knowledge of the optimal classification mapping to be realized is not available, it is very difficult, if not impossible, to work out precisely the optimal numbers of nodes required at each hidden layer. In the following simulation study, n_1 and n_2, the numbers of nodes in the first and second layers, were chosen after some experimentation. The number of adjustable parameters (weights and thresholds) in a three-layer perceptron equalizer is usually quite large. The high dimensionality is a common price for employing a non-linear architecture.

5. Simulation results

In the first example, the channel is given in Fig. 4. A data sequence of 500 points was generated

using simulation to train a (2-9-3-1, 0) perceptron equalizer. The training algorithm used was the recursive prediction error algorithm and training was repeated several times using the same data sequence. The decision region formed by the trained perceptron equalizer is shown in Fig. 11, where it is seen that the set $P_{2,0}(1)$ is correctly within the decision region and the decision region is close to the optimal one.

The channel for the second example is as that given in Fig. 8 and an identical procedure as that for the first example was used to train a (2-9-3-1, 0) perceptron equalizer. The decision region obtained by the trained perceptron equalizer is given in Fig. 12. Again the set $P_{2,0}(1)$ lies correctly within the decision region and this region is very close to the optimal one shown in Fig. 8.

The aim of the third example is to compare the bit error rates achieved by the optimal and perceptron equalizers for different signal-to-noise ratios. The channel model is

$$\hat{o}(t) = \tilde{o}(t) + 0.2\tilde{o}^2(t),$$

$$\tilde{o}(t) = 0.3482x(t) + 0.8704x(t-1) + 0.3482x(t-2).$$

The additive noise is given by $e(t) = 0.8\xi(t) + 0.6\xi(t-1)$ where $\xi(t)$ is a Gaussian white sequence. For $m = 3$ and $d = 1$, the optimal BER was computed according to (9) using simulated data. The performance was averaged over 10 runs and each run had 9000 points of different realizations of stochastic processes $x(t)$ and $e(t)$. The performance of a (3-9-5-1, 1) perceptron equalizer was next investigated. Each run started from different random initial weights and the data sequence was first used to train the perceptron equalizer by the back-propagation algorithm. The

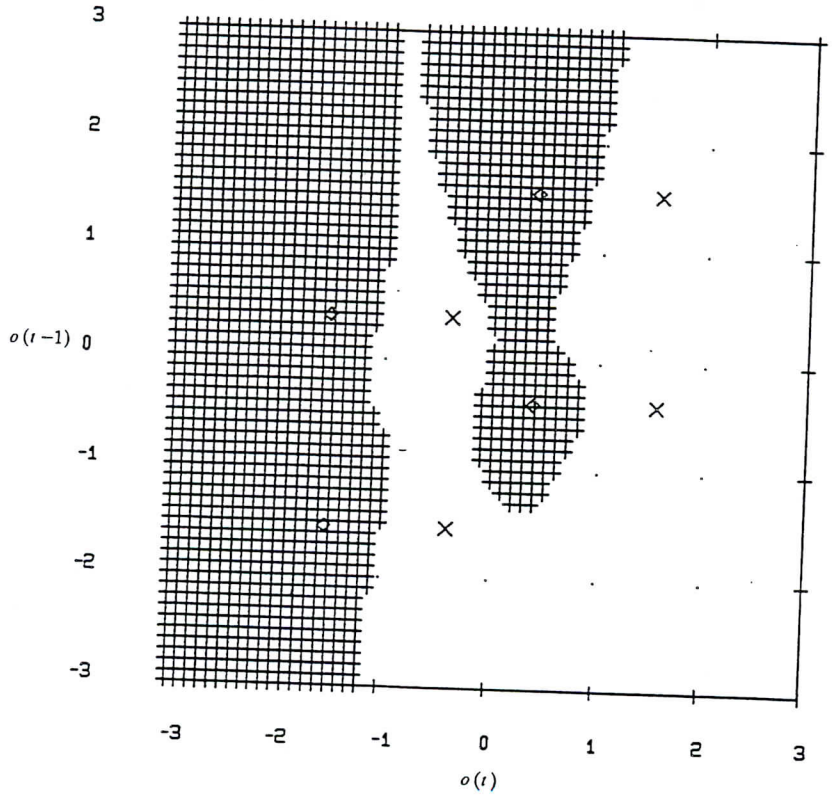

Fig. 11. Decision region formed by perceptron equalizer. Channel $\hat{o}(t) = \tilde{o}(t) - 0.9\tilde{o}^3(t)$ and $\tilde{o}(t) = x(t) + 0.5x(t-1)$, Gaussian white noise with $\sigma_e^2 = 0.2$; perceptron equalizer: (2-9-3-1, 0).

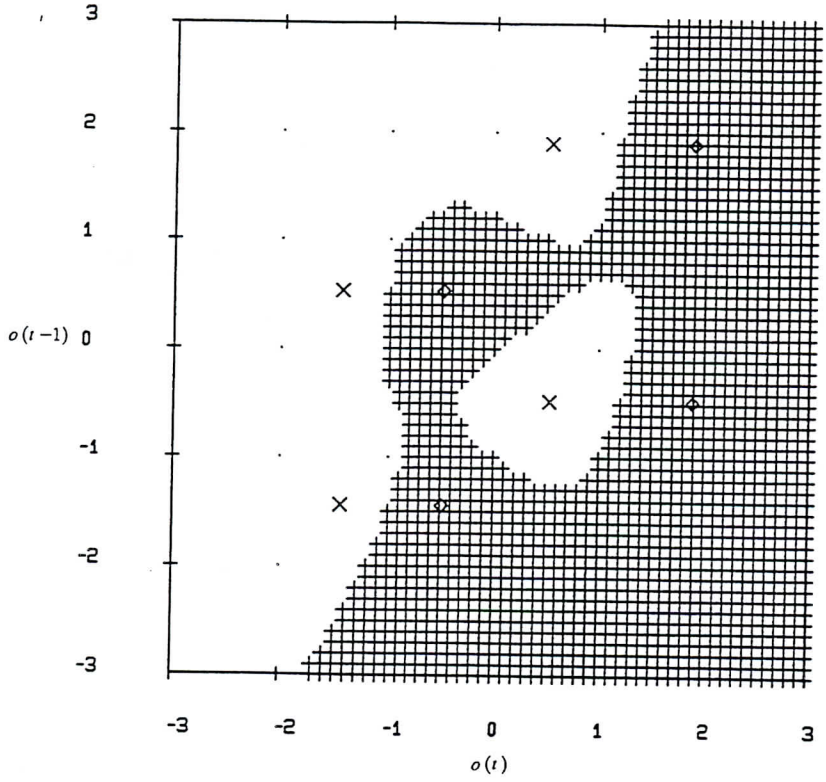

Fig. 12. Decision region formed by perceptron equalizer. Channel $\hat{o}(t)=\tilde{o}(t)+0.1\tilde{o}^2(t)+0.05\tilde{o}^3(t)$ and $\tilde{o}(t)=0.5x(t)+x(t-1)$, Gaussian coloured noise with $\sigma_e^2=0.2$ and $\rho_1=0.48$; perceptron equalizer: (2-9-3-1, 0).

BER was then computed for the same data by the trained perceptron equalizer. The averaged results are shown in Fig. 13 where it is seen that the perceptron equalizer enjoys a performance which is close to that achieved by the optimal equalizer.

6. Conclusions

By viewing the equalization problem as a classification problem, the influence of channel non-linearities and additive coloured noise to the optimal bit-error-rate solution has been investigated. It has been shown that the neural network approach offers equal effectiveness for adaptively equalizing linear or non-linear channels with white or coloured additive noise and it provides a general solution to the problem of channel distortion in digital communications systems. Simulation results have been included to support the analysis.

However, at present there are some practical difficulties associated with this neural network approach, notably, the danger of being trapped in local minima of the mean square error surface and the slow convergence using the back propagation algorithm during the training process. These and other issues such as how to select hidden layer nodes have to be resolved before the real-time implementation can be addressed. These existing problems point to various areas which are worth further study. Using alternative non-linear architectures such as a polynomial-perceptron structure to approximate the optimal equalizer solution is another area currently under investigation.

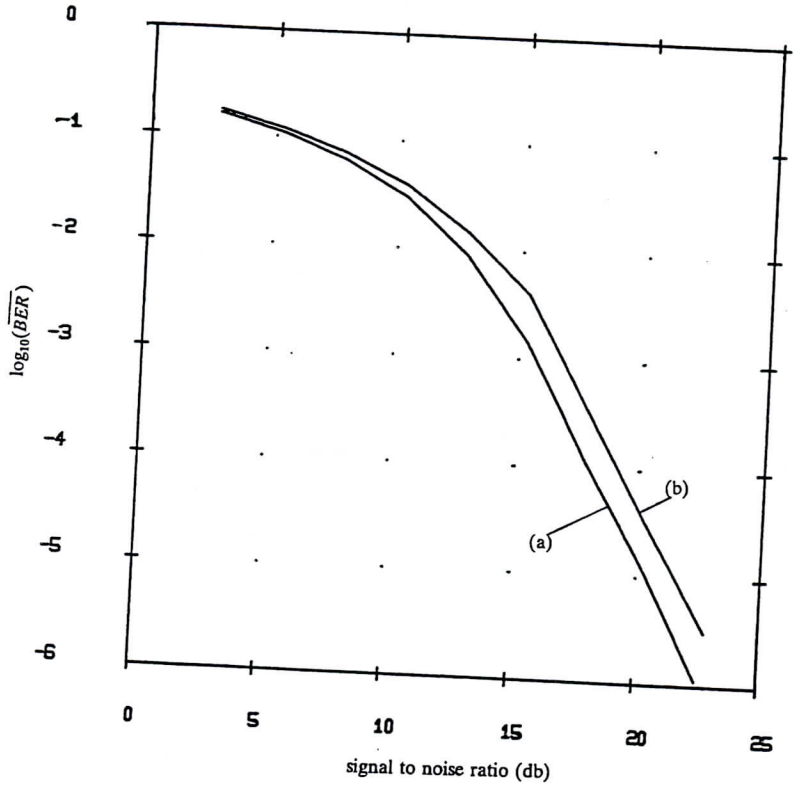

Fig. 13. Comparison of bit error rates achieved by optimal and perceptron equalizers. (a) optimal equalizer; (b) perceptron equalizer.

Acknowledgments

This work was supported by the UK Science and Engineering Research Council. The authors wish to acknowledge the contributions from Mr. S. Siu and to thank the referees for their valuable comments on the manuscript.

References

[1] E. Biglieri, A. Gersho, R.D. Gitlin and T.L. Lim, "Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission", *IEEE J. Sel. Areas Commun.*, Vol. SAC-2, No. 5, 1984, pp. 765–777.

[2] S. Chen and S.A. Billings, "Recursive prediction error parameter estimator for non-linear models", *Internat. J. Control*, Vol. 49, No. 2, 1989, pp. 569–594.

[3] S. Chen, C.F.N. Cowan, S.A. Billings and P.M. Grant, "Parallel recursive prediction error algorithm for training layered neural networks", *Internat. J. Control*, 1990, to appear.

[4] G. Cybenko, "Approximations by superpositions of a sigmoidal function", *Math. Control Signals Syst.*, Vol. 2, No. 4, 1989, pp. 303–314.

[5] K. Funahashi, "On the approximate realization of continuous mappings by neural networks", *Neural Networks*, Vol. 2, 1989, pp. 183–192.

[6] G.J. Gibson, S. Siu and C.F.N. Cowan, "Application of multilayer perceptrons as adaptive channel equalisers", *Proc. IEEE Internat. Conf. Acoust. Speech Signal Process.*, Glasgow, Scotland, 23–26 May 1989, pp. 1183–1186.

[7] R.P. Lippmann, "An introduction to computing with neural nets", *IEEE Acoust. Speech Signal Process. Mag.*, Vol. 4, 1987.

[8] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning internal representations by error propagation", in: D.E. Rumelhart and J.L. McClelland, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press, Cambridge, MA, 1986, pp. 318–362.