Monograph:
Mitchinson, B. and Harrison, R.F. (2000) Digital Communications Channel Equalisation Using the Kernel Adaline. Research Report. ACSE Research Report 768 . Department of Automatic Control and Systems Engineering

Digital Communications Channel Equalisation using the Kernel Adaline

Ben Mitchinson, Robert F. Harrison

Department of Automatic Control and Systems Engineering

The University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK

Research Report 768

28 March, 2000

Abstract

For transmission of digital signals over a linear channel with additive white gaussian noise, it has been shown that the optimal symbol decision equaliser is non-linear. The Kernel Adaline algorithm, a non-linear generalisation of Widrow's and Hoff's Adaline, has been shown to be capable of learning arbitrary non-linear decision boundaries, whilst retaining the desirable convergence properties of the linear Adaline. This work investigates the use of the Kernel Adaline as equaliser for such channels. It is shown that the Kernel Adaline performs comparably to the Bayesian optimal equaliser for these channels, and further has something to offer even if the channel noise is non-white.

Section I - Introduction

Distortion and Equalisation

Band-limited communications channels driven at high data rates often display intersymbol interference (ISI) due to a dispersive time response. We consider a discrete time model of a communications system shown in Fig.1, where the channel response is given by the linear transfer function

$$H(z) = Z\{h(k)\} = \sum_{i=0}^{n_h} h_i z^{-i} \tag{1}$$

with channel order n_h. The channel suffers also from additive noise to give

$$y(k) = h(k) * s(k) + q(k) \tag{2}$$

where * represents the operation of convolution.

The transmitted sequence $s(k)$ is an equiprobable and independent binary series with $s(k) \in \{-1,+1\}$. An equaliser may be used to combat the effects of the distortion on the signal due to $H(z)$; the task of the equaliser is to recover an estimate of the transmitted sequence $s(k)$, given the channel output $y(k)$. The optimal receiver for such a system is the maximum likelihood sequence estimator (MLSE) [Forney, 1972] which operates on the entire transmitted sequence, but in practical situations it is often necessary to obtain the estimate $\hat{s}(k)$ in real time, so sub-optimal equalisers which make decisions symbol by symbol are preferred. A common structure for this equaliser is the indirect-modelling equaliser, shown in Fig. 2, in which the channel output $y(k)$ is filtered to give the sequence estimate $\hat{s}(k)$; it is this type of equaliser we will examine. For discrete amplitude signals, this can be viewed as a classification task, where the transmitted symbol $s(k-D)$ is to be estimated from the channel output vector

$$\mathbf{y}(k) = [y(k), y(k-1), \ldots y(k-L)] \tag{3}$$

where D is termed the equaliser delay, and L the equaliser order.

Automatic and Adaptive Equalisation

Many practical transmission channels display different characteristics with each use. As a result, a practical equaliser requires the transmission of a known training sequence through the channel in order to determine optimal filter coefficients (automatic equalisation). The characteristics of some transmission channels may also vary during transmission, such that, to maintain optimality, the coefficients of the equaliser must be continually updated during transmission (adaptive equalisation). Simple least mean square regression may be used to set the coefficients of a linear filter, thus approximating an inverse of the channel (Linear Transversal Equaliser or LTE). This gives the optimum real-time decision equaliser for a continuous amplitude signal, but makes no use of the information that the transmitted sequence is binary.

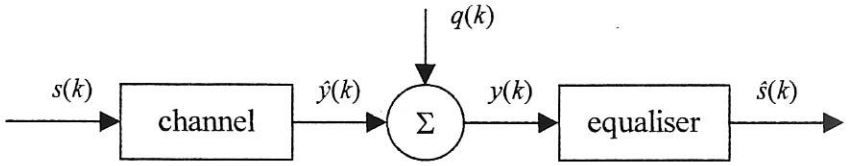

Fig. 1 – Data communications system

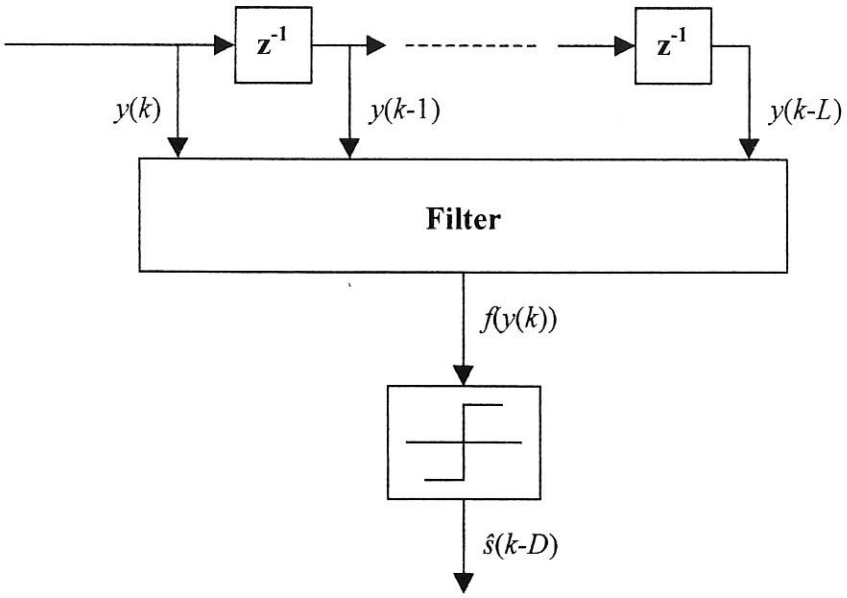

Fig. 2 – Indirect-modelling equaliser structure.

Section II - Bayes Optimal Symbol Decision Equaliser

Taking into account the binary nature of the transmitted sequence, Bayes theory gives us the form of the optimal symbol decision equaliser (Bayesian Equaliser or BEQ). We consider the noiseless channel output vector which, along with some noise, will form the input to the equaliser at sample k

$$\hat{\boldsymbol{y}}(k) = [\hat{y}(k), \hat{y}(k-1), \ldots \hat{y}(k-L)] \tag{4}$$

The vector of channel inputs that affect this channel output vector is dependent on the channel order and the equaliser order and is given by

$$\boldsymbol{s}(k) = [s(k), s(k-1), \ldots s(k-L-n_h)] \tag{5}$$

The vector $\boldsymbol{s}(k)$ can take 2^n different values, where $n = L+n_h+1$, and this gives rise to 2^n different values of the vector $\hat{\boldsymbol{y}}(k)$, which will be termed centres and denoted $\boldsymbol{y}_i$. Of these, half will correspond to $s(k-D) = 1$ and half to $s(k-D) = -1$; these will be denoted by $\boldsymbol{y}_i^+$ and $\boldsymbol{y}_i^-$ respectively. For $q(k)$ assumed white, the noisy channel output vector

$$\boldsymbol{y}(k) = [y(k), y(k-1), \ldots y(k-L)] \tag{6}$$

is a stochastic process having a gaussian density function centred at $\hat{\boldsymbol{y}}(k)$. Hence the instantaneous values of $\boldsymbol{y}(k)$ form clusters centred on $\hat{\boldsymbol{y}}(k)$. Bayes theory then gives us the optimum symbol decision equaliser

$$\hat{s}(k-D) = sign(f_B(\boldsymbol{y}(k))) \tag{7}$$

$$f_B(\mathbf{y}(k)) = \sum_i \exp(-\left\|\mathbf{y}(k) - \mathbf{y}_i^+\right\|^2 / 2\sigma_q^2) - \sum_j \exp(-\left\|\mathbf{y}(k) - \mathbf{y}_j^-\right\|^2 / 2\sigma_q^2) \tag{8}$$

which specifies a decision boundary that is a hypersurface in a space of dimension $L+1$, and is dependent on the variance of the noise σ_q^2 as well as the channel characteristics. Hence the hyperplane described by the LTE will always be suboptimal. Fig. 3 is an example plot of the BEQ decision surface and channel output vectors and centres for the given channel.

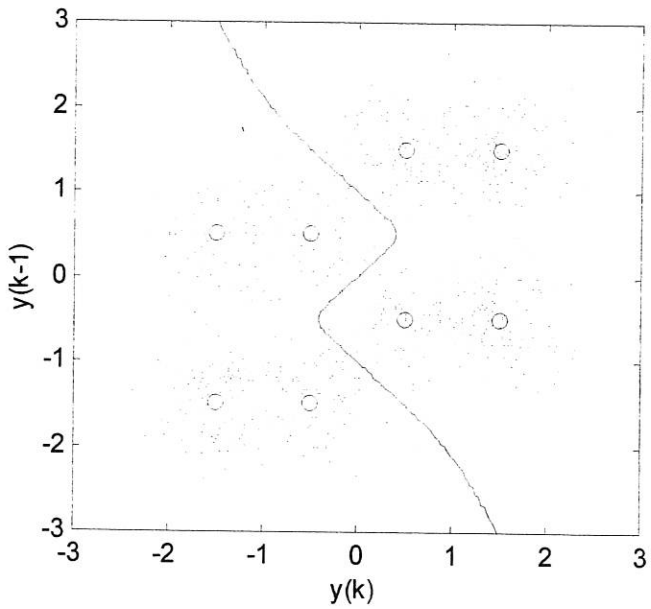

Fig. 3 – Centres y_i, clusters $y(k)$, and BEQ optimal decision boundary. 1000 samples through the channel $H(z) = 0.5 + 1.0\ z^{-1}$ with SNR = 10dB, $L = 1$, $D = 1$. (after Chen 1993)

Section III - Kernel Adaline

The conventional linear Adaline (Eqns. 9) [Widrow & Hoff 1960] has the attractive property of a cost function that is quadratic with respect to the weight vector, which is therefore easily minimised.

$$y_p = < \mathbf{w}, \mathbf{x}_p > \tag{9a}$$

$$J = \sum_{p=1}^{N} (t_p - y_p)^2 \tag{9b}$$

$$\mathbf{w} \leftarrow \mathbf{w} + \eta(t_p - y_p)\mathbf{x}_p \tag{9c}$$

where $\boldsymbol{x}_p$ and t_p are the pth input-output pair from a set of N pairs with $\boldsymbol{x}_p$ a vector, y_p is the output of the Adaline in response to $\boldsymbol{x}_p$, and $\boldsymbol{w}$ is a row-vector of 'weights'. J denotes the 'cost function' of the Adaline, and η the learning rate; the expression $<x,y>$ denotes the scalar product of the vectors x and y.

We can obtain a non-linear Adaline by pre-processing the input to a Linear Adaline using some function $\phi(.)$ (Eqn. 10), but a large number of terms may be required to model strong non-linearity. For example, a d-dimensional pattern undergoing a pth order polynomial expansion requires $(p+d)!/(p!d!)$ terms. Alternatively, we can use a non-linear version of the Adaline (Eqn. 11), but for any significant non-linearity, this may result in a non-convex cost function which has local minima, causing minimisation problems.

$$y_p = < w, \phi(x_p) > \tag{10}$$

$$y_p = f(w, x_p) \tag{11}$$

The Adaline may also be represented in its data-dependent form [Frieß & Harrison 1998a] (Eqns. 12 – Note that these equations describe the same machine as Eqns. 9).

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i \tag{12a}$$

$$y_p = < \sum_{i=1}^{N} \alpha_i \mathbf{x}_i, \mathbf{x}_p > = \sum_{i=1}^{N} \alpha_i < \mathbf{x}_i, \mathbf{x}_p > \tag{12b}$$

$$\alpha_i \leftarrow \alpha_i + \eta(t_p - y_p) \tag{12c}$$

where the α_i are known as the 'multipliers'.

The pre-processing approach to accounting for non-linearity may be applied to the data-dependent form by replacing the scalar product between patterns with a scalar product between patterns transformed into some 'linearisation' space by an operation $\phi(.)$ (Eqn. 13).

$$y_p = \sum_{i=1}^{N} \alpha_i < \phi(\mathbf{x}_i), \phi(\mathbf{x}_p) > \tag{13}$$

The Kernel Adaline (KA) [Friess & Harrison 1998b] has been introduced, using Mercer kernels to compute the dot product between these expanded terms, without explicitly performing the expansions (Eqns. 14).

$$k(\mathbf{x}_i, \mathbf{x}_p) = < \phi(\mathbf{x}_i), \phi(\mathbf{x}_p) > \tag{14a}$$

$$y_p = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}_p) \tag{14b}$$

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i) \tag{14c}$$

where $k(a,b)$ is the Mercer kernel evaluated at a,b.

We note that $\boldsymbol{w}$ given in (Eqn. 14c) is not accessible, i.e. computable, since we do not know the form of the mapping $\phi(.)$. Since the adaptive structure of the Kernel Adaline is the same as that of the linear Adaline, it retains the convex cost function mentioned above.

Thus the Kernel Adaline is able to learn a non-linear mapping between input and output data, and retains the property of a quadratic cost function, without suffering the explosion of terms associated with highly non-linear pre-processing layers. This suggests it might be a 'good' solution to any problem that can be cast as a non-linear mapping problem.

Mercer Kernels

The choice of kernel for use with the Kernel Adaline will be dependent on the nature of the non-linearity that is to be modelled. Some examples of Mercer kernels are

$$k_{POL}(\mathbf{x},\mathbf{y}) = (<\mathbf{x},\mathbf{y}>+1)^d \tag{15}$$

$$k_{RBF}(\mathbf{x},\mathbf{y}) = \exp(-\|\mathbf{x}-\mathbf{y}\|^2 / 2\sigma^2) \tag{16}$$

where k_{POL} is a polynomial basis kernel of order d, and k_{RBF} is a radial basis kernel with gaussian width governed by σ^2. Note that each kernel has a parameter which might loosely be called a 'smoothing' parameter. Note also that use of the polynomial kernel with d set to unity in a Kernel Adaline reduces the system to a conventional linear Adaline with bias. Given that we are trying to approximate the decision surface of the BEQ given in Eqn. 8, the RBF kernel is most appropriate here, and though the appropriate value for σ^2 is not clear, choosing it to be equal to σ_q^2, were this value known, would seem to be a reasonable choice.

Training Particulars

There follows a pseudo-code algorithm for the batch training procedure used in this study, which is a familiar network training algorithm. The available data is first divided into three sections, the first to be used for training (n_{tr} in $\boldsymbol{x}_{tr}$, t_{tr}), the second for regularisation via early stopping (n_{te} in $\boldsymbol{x}_{te}$, t_{te}), and the third to remain unseen for unbiased validation of system performance (n_{va} in $\boldsymbol{x}_{va}$, t_{va}).

LET $\alpha_i = 0 \; \forall$ i, last_MSE = a very large number
DO Calculate each output $y_{tr}(i)$ of KA for each $\boldsymbol{x}_{tr}(i)$ using Eqn. 14b
Update each α_i using Eqn. 12c based on $y_{tr}(i)$ and $t_{tr}(i)$
Calculate each output $y_{te}(i)$ of KA for each $\boldsymbol{x}_{te}(i)$ using Eqn. 14b
Calculate mean square error (MSE) between y_{te} and t_{te}
IF (last_MSE-MSE<thresh) THEN BREAK
last_MSE = MSE
END DO

Section IV – The Radial Basis Function Equaliser

A system has been devised [Chen et al. 1993] which has the same structure as the BEQ and is able to learn the positions of the centres y_i using a κ-means clustering algorithm. The decision function of the system is identical in form to Eqns. 7 and 8, based on the learned centres and a noise estimate σ^2. The simulation work in the current study is arranged so as to be directly comparable with this work, and thus broadly follows the same path and uses the same parameters (channel characteristics, equaliser characteristics, experimental range).

We will refer to this equaliser as the REQ.

Section V – Simulation Results

Throughout we consider the system given in Fig. 1, with $s(k)$ the same equiprobable and independent binary sequence given in Sect. 1. We assume a correct estimate of the noise variance, $\sigma^2 = \sigma_q^2$, for use with all equalisers except as stated in Exps. V and VI, and we employ 640 training samples, 200 testing samples and 100,000 validation samples throughout Exps. III – VI.

Some definitions

Signal to noise ratio (SNR) is calculated according to

$$\mathrm{SNR} = 10\log_{10}\frac{\sum_k (H(z)s(k))^2}{\sum_k (q(k))^2} \tag{17}$$

Bit error rate (BER) is the performance measure that we seek to minimise in a digital communications problem, and is defined as

$$BER_A = \frac{\max(\text{number of errors},1)}{\text{number of samples}} \quad \text{returned by equaliser A} \tag{18}$$

Log error rate (LER) is defined as

$$LER_A = \log_{10}(BER_A) \tag{19}$$

Relative bit error rate (RBER) is used to compare the performance of two equalisers and is defined as

$$RBER_{A,B} = \frac{BER_A}{BER_B} \tag{20}$$

Relative log error rate (RLER) is defined as

$$RLER_{A,B} = \log_{10}(RBER_{A,B}) \tag{21}$$

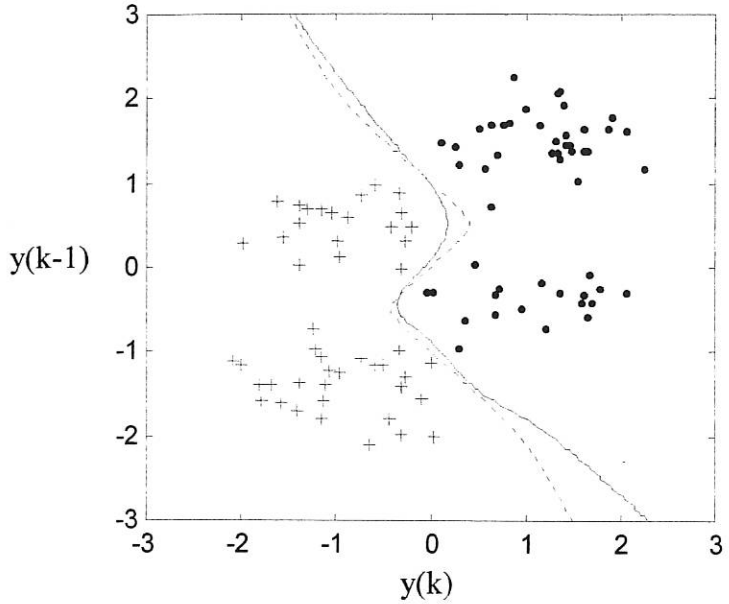

Fig. 4 – 100 training vectors (+, •), BEQ-DS (dotted), KA-DS (solid).

Experiment I

To illustrate the action of the equaliser we consider the channel given in Fig. 3 and Eqns. 1 and 2 with

$$H(z) = 0.5 + 1.0z^{-1} \tag{22}$$

and SNR = 10dB. 100 input-output samples are used to train the KA. Fig. 4 shows the decision surface learned by the KA, along with the BEQ decision surface. The training vectors that correspond to $s(k\text{-}D) = -1$, $s(k\text{-}D) = 1$, are plotted as crosses and dots respectively.

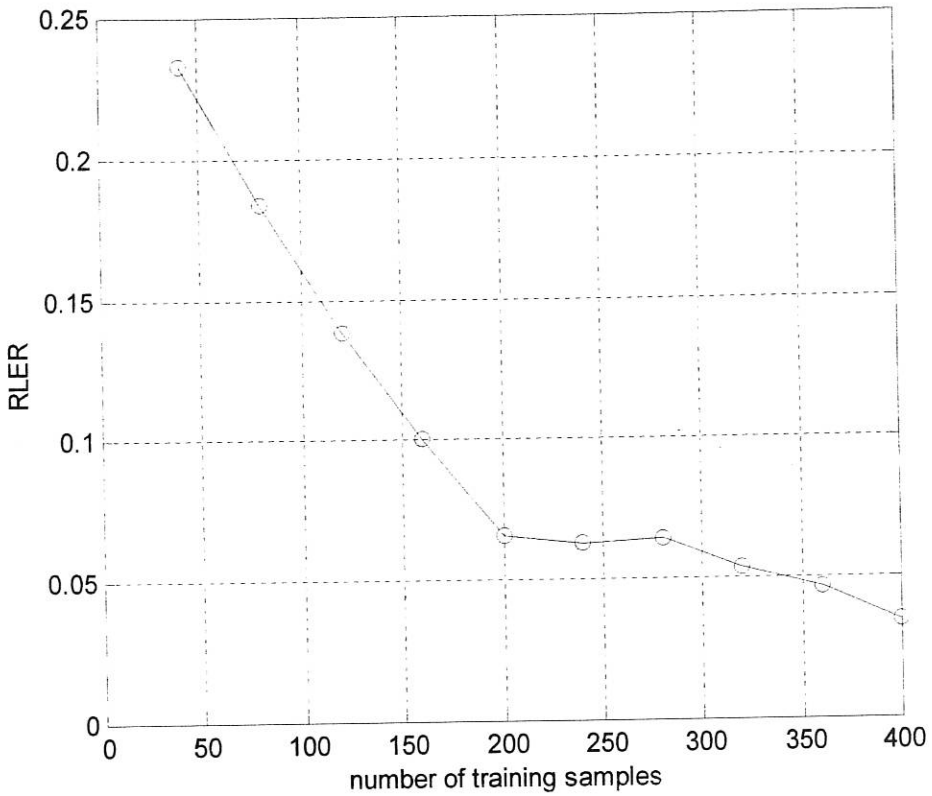

Fig. 5 – $RLER_{KA,BEQ}$ against number of training samples.

Experiment II

To investigate the effect of the number of training samples used to train the KA, Exp. 1 is repeated here while varying the size of the training set. For each training set size $RBER_{KA,BEQ}$ is calculated. The RLER is calculated from an average RBER taken over 10 repetitions of the experiment, and is plotted against training set size in Fig. 5.

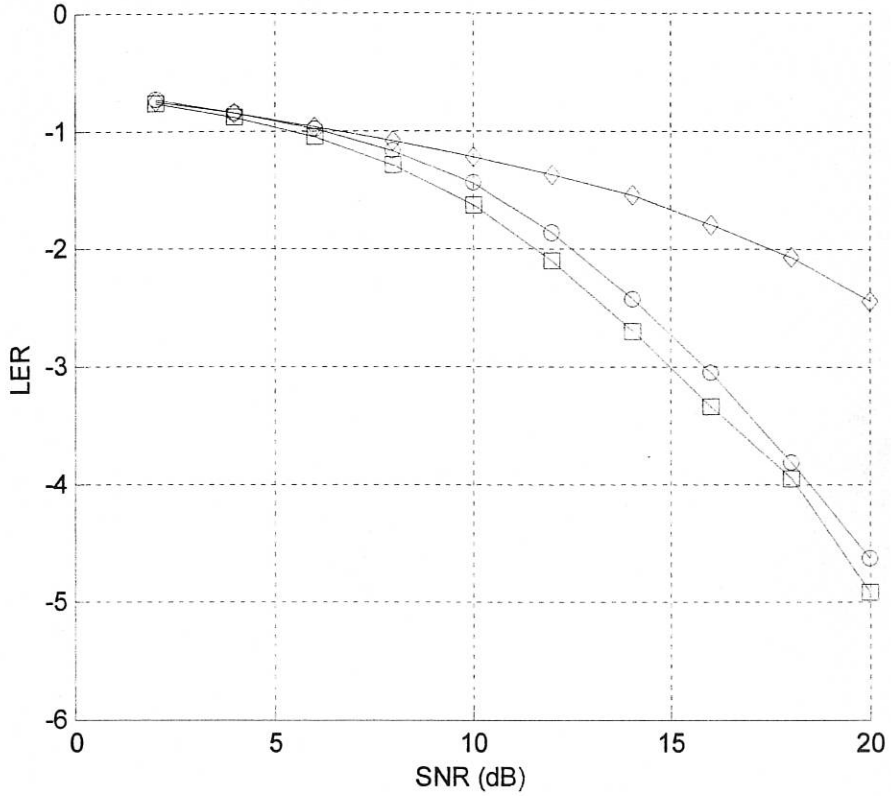

Fig. 6 – LER performance of equalisers on linear channel. Linear Adaline (diamond), Kernel Adaline (circle), BEQ (square).

Experiment III

In order to test error-rate performance with a more realistic equaliser order we consider another channel, given by

$$H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2} \qquad (23)$$

with the equaliser parameters $L = 3$, $D = 1$. For each of a range of values of SNR, the trained KA was tested on 100,000 validation samples and the number of misclassifications recorded. Each of these train-validate runs was repeated 10 times with newly generated signals and the total number of errors used to calculate the log error rate (LER), which is plotted against SNR in Fig. 6, alongside results returned by the optimal BEQ and the linear Adaline for the same signals.

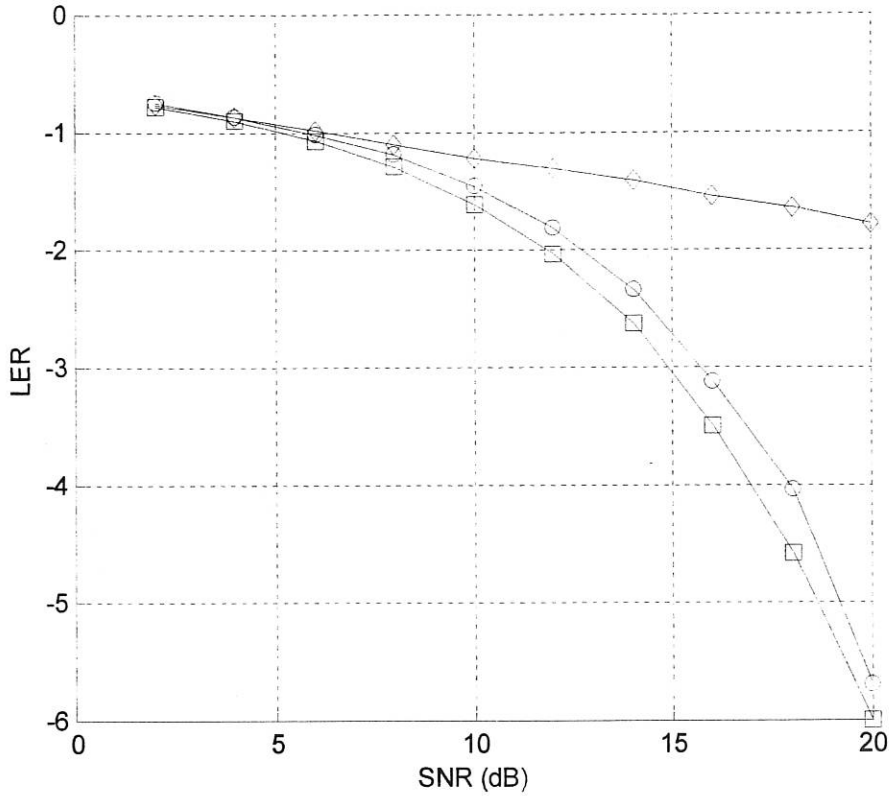

Fig. 7 – LER performance of equalisers on nonlinear channel. Linear Adaline (diamond), Kernel Adaline (circle), BEQ (square).

Experiment IV

The aim of this experiment is to see how the KA performs when the channel is nonlinear. Note that although the nonlinearity is memoryless, the combined channel consisting of Eqns. 24 and 25 taken together is nonlinear with memory.

The procedure and equaliser for this experiment are the same as those for Exp. III, but the channel is now given by

$$y(k) = x(k) + 0.2x^2(k) - 0.1x^3(k) + q(k) \qquad (24)$$

$$X(z)/S(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2} \qquad (25)$$

The results of this experiment are plotted in Fig. 7, again alongside similar results for the BEQ and the linear Adaline.

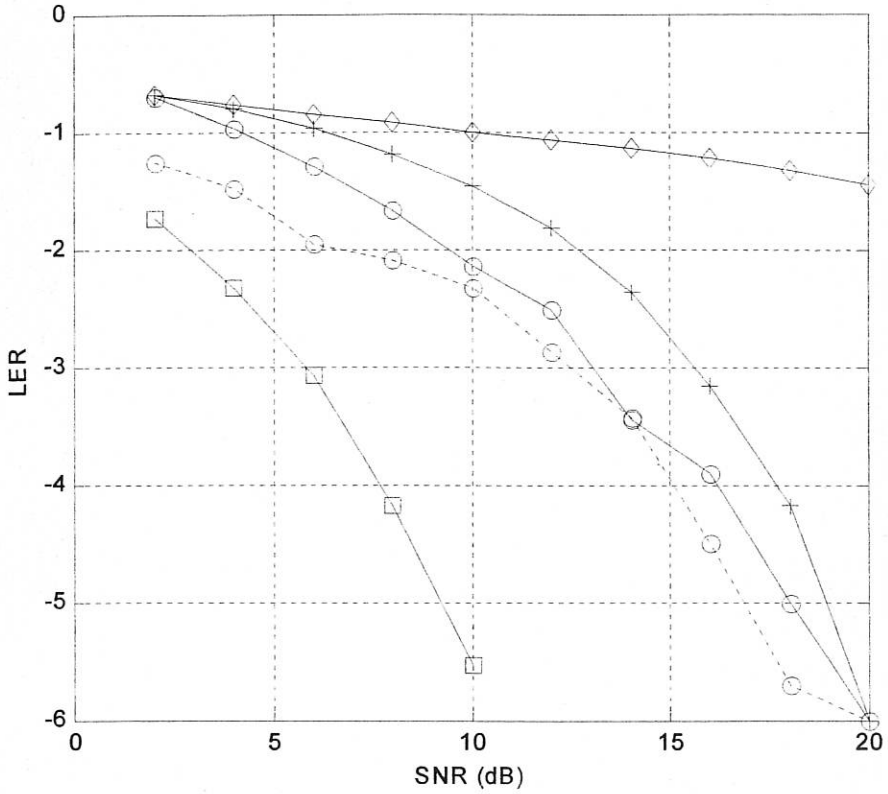

Fig. 8 – LER performance of equalisers on linear channel with coloured noise. Linear Adaline (diamond), REQ (cross), Kernel Adaline (circle, solid line), Kernel Adaline with revised kernel parameter (circle, dotted line), Bayesian Optimal (square).

Experiment V

The aim of this experiment is to see how non-white channel noise affects the performance of the KA and the REQ [Chen et al. 1993]. The channel used is given by

$$y(k) = H(z)s(k) + N(z)q(k) \tag{26}$$

$$H(z) = 1 - 0.7z^{-1} \tag{27}$$

$$N(z) = \frac{0.1411}{1 + 0.99z^{-1}} \tag{28}$$

and the equaliser parameters are $L = 1$ and $D = 1$; again 10 repetitions are made.

The results of this experiment are plotted in Fig. 8, alongside similar results for the linear Adaline and the REQ. The BEQ given earlier was derived under the assumption of white channel noise, and is thus not appropriate for this channel; results for the Bayesian optimal equaliser with the appropriate form for this problem are plotted here. Fig. 9 is a plot of the decision surfaces learned by the KA and the REQ for the case SNR = 10dB.

Also plotted are the results returned by the KA using the revised kernel parameter given by $\sigma^2 = \sigma_q^2/3$, which is a better choice (based on performance) for this problem than $\sigma^2 = \sigma_q^2$.

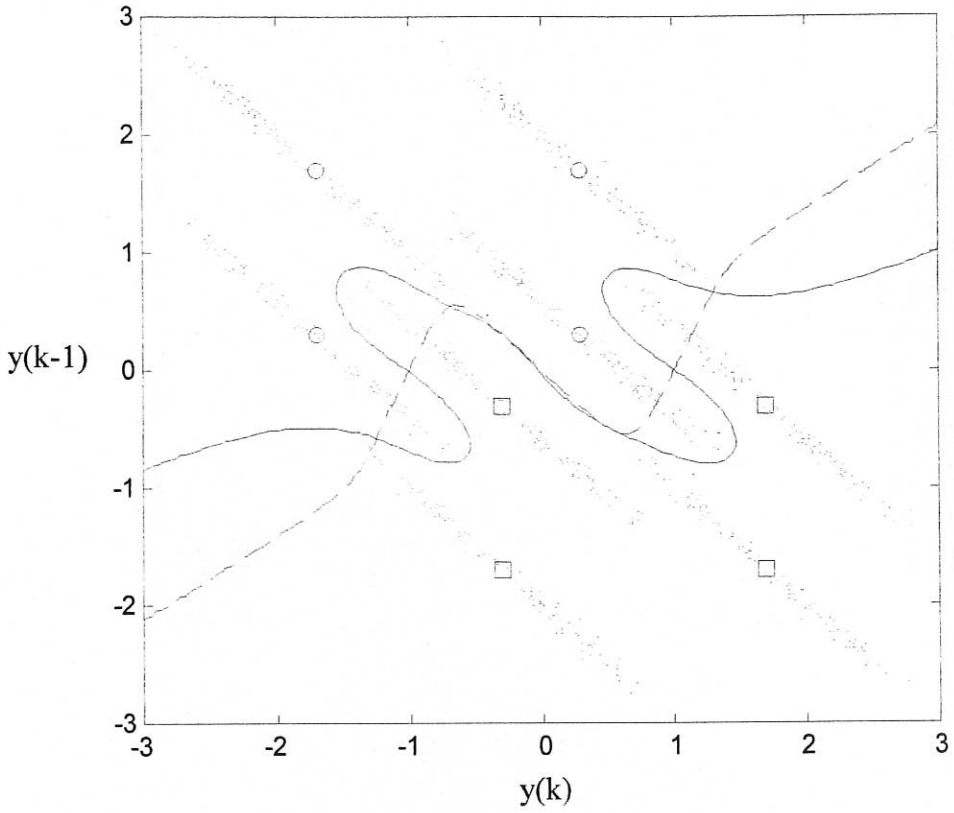

Fig. 9 – Decision surfaces learnt by REQ (dashed) and KA (solid) for linear channel with coloured noise, clusters (training set) and centres (theoretical), $s(k\text{-}D) = 1$ (circles), $s(k\text{-}D) = -1$ (squares).

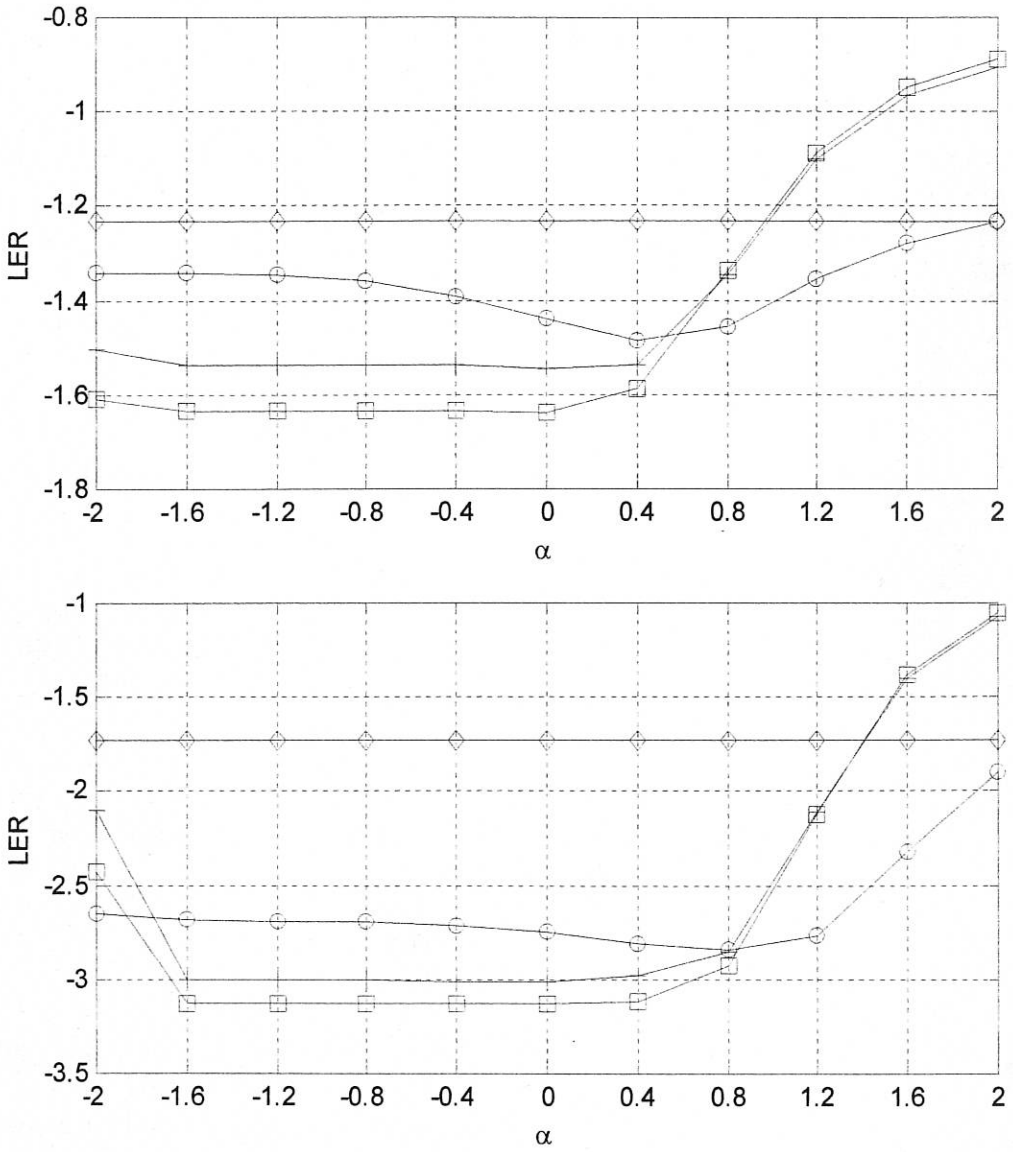

Fig. 10 – Dependency of equaliser performance on accuracy of noise estimate. KA (circles), REQ (crosses), BEQ (squares), Linear Adaline (diamonds). SNR = 10dB (top), SNR = 15dB (bottom).

Experiment VI

The aim of this experiment is to discover how sensitive the KA performance is to the accuracy of the noise estimate used. The channel and equaliser are as used in Exp. III, and the experiment is performed once at SNR = 10dB, and again at SNR = 15dB. The noise variance estimate is adjusted according to

$$\sigma^2 = 10^{\alpha}\sigma_q^{\ 2} \tag{29}$$

The results are plotted in Fig. 10, along with similar results for the BEQ[1], REQ and Linear Adaline for comparison.

---

[1] Note that the BEQ quoted is clearly no longer the optimal solution when σ^2 is incorrectly estimated.

Section VI –Discussion and Conclusions

Given some input-target training pairs, the KA can learn an approximation to the optimal decision surface as given by Bayes theory for the type of problem discussed (Exp. I). The accuracy with which this decision surface is learnt is, as we would expect, dependent on the number of training samples supplied (Exp. II), performance improving as the size of the training set increases. We note though that the computational complexity of the KA also increases with training set size, in terms of both storage and operations.

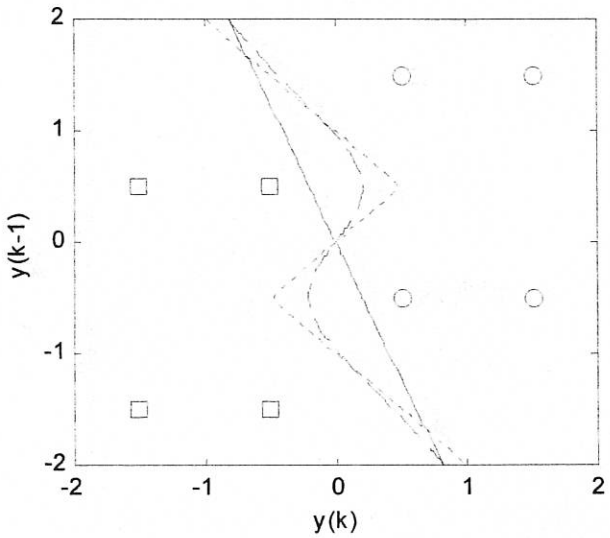

Fig. 11 – BEQ-DS for centres given by squares and circles, with SNR = -5dB (solid), 5dB (dashed) and 15dB (dotted).

The KA consistently outperforms the linear Adaline in terms of error rate on the simulated problems (Exps. III, IV, V). This performance gain tends to increase with increasing SNR, which can be understood if we consider the form of the optimal DS for varying amounts of noise, shown in Fig. 11 for the channel given in Exp. III. As the noise becomes very large relative to the signal (solid curve), the optimal DS tends towards linearity, and hence the KA and linear Adaline will perform comparably. As the noise tends to zero power (dotted curve), the DS tends to the piecewise linear surface which is equidistant between the classes $s(k\text{-}1) = -1$ and $s(k\text{-}1) = 1$, and thus the KA can significantly outperform the linear Adaline by learning this non-linear DS.

When these types of problems are approached practically maximum error rate is typically a design parameter, so it is reasonable to compare the performance of two equalisers by comparing the SNR necessary for each to achieve the same error rate. By this measure, to achieve the same error rate as the BEQ on the linear channel (Exp. III) and non-linear channel (Exp. IV) the KA requires a channel SNR up to 0.9dB higher than the BEQ. In both these cases (Exps. III and IV) the KA is marginally outperformed by the REQ, which closely matches the BEQ performance. This is not surprising since the REQ is designed to fit these channels, and takes the form of the BEQ; conversely, the KA contains no implicit knowledge of the form of the DS to be learnt, but still achieves good performance on both of the channels.

To illustrate this point more clearly, Exp. V considers performance on a linear channel with additive non-white noise. The BEQ as given by Eqns. 7 and 8 is derived under the assumption of white additive channel noise, and is thus sub-optimal for this problem. The performance of the REQ, which has the same form given in Eqns. 7 and 8, is not as good as on the previous two problems. The derivation of the KA though contains no assumptions about the nature of the DS, and it outperforms the REQ on this problem significantly; to achieve the same error probability the REQ requires an SNR up to 3.0dB higher than the KA (up to 6.5 dB higher in the case of the revised kernel parameter). Fig. 9 demonstrates how the KA can attain this improved performance by accounting for non-white noise distribution. It should be noted though that the colouring of the simulated noise in this experiment may be far in excess of that which would be encountered in practical equalisation problems.

It is also clear that the performance of the KA is not in line with that of the optimal equaliser for this problem; though the performance does approach optimal with increasing training set size, the computational load also increases. This is due to the strong data-dependency of the decision function learned by the KA with an RBF kernel - in poorly trained regions of input space (where the nearest training vector is several times σ distant), the decision function tends towards nearest neighbour classification, i.e. a presented vector is assigned the same class as the nearest (in the Euclidean sense) training vector.

Exp. VI demonstrates that the KA is relatively insensitive to the noise estimate; with an estimate out by a factor of 10, the LER performance is degraded by no more than 0.1 in either of the cases shown. With an estimate out by a factor of 100, the KA still outperforms the Linear Adaline. Also, in both cases, the performance improves as σ^2 set to several times σ_q^2, and we see in Exp. V that performance was improved significantly by setting σ^2 to a fraction of σ_q^2; this suggests that the overall performance of the system may be improved if a formal method for setting the RBF kernel parameter is devised.

Generally, the performance of the Kernel Adaline on this type of problem is comparable to that of the Bayesian optimal equaliser and slightly less good than that of the REQ except in Exp. V, where the noise is non-white, a situation with which the REQ is not designed to cope. This highlights an important feature of the KA – that it requires very little information regarding the form of the decision surface that is to be learned. In the case of the RBF kernel, the supplied parameter is required to give an estimate of the level of detail or 'scale' of the problem, and the value of this parameter has been shown to be non-critical within an order of magnitude for a typical case. Consequently, we can reasonably infer that the KA has the potential to perform well on similarly posed problems whether or not the form of the required decision surface is known. Also highlighted by the failure of the KA to approach optimal performance in Exp. V is the strong dependency of the learned decision function on the training data supplied. This fact combined with the prohibitive computational complexity of employing much larger training sets imposes a performance limit on the KA in classification problems having additive non-white noise. Currently under investigation is a new version of the Kernel Adaline algorithm that employs set reduction such that it is not necessary to use all the training data as basis patterns for the training or classification. Using this new algorithm, it is expected that it will be possible to train the KA on large data sets without incurring a prohibitive computational load. As a further advantage, it should be possible to use this new version online as an adaptive automatic equaliser.

References

Frieß T.T, Harrison R.F, 'Perceptrons in Kernel Feature Spaces.', Research Report No. 720, Dept. of AC&SE, The University of Sheffield, Sheffield, 1998(a).

Frieß T.T, Harrison R.F, 'The Kernel Adaline: A New Algorithm for Non-linear Signal Processing and Regression.', Research Report No. 731, Dept. of AC&SE, The University of Sheffield, Sheffield, 1998(b).

Forney, G.D, 'Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference.', IEEE Trans. Info. Theory, Vol. IT-18, pp. 363-378, 1972.

Chen S, Mulgrew B, Grant P.M, 'A Clustering Technique for Digital Communications Channel Equalisation Using Radial Basis Function Networks.', IEEE Trans. Neural Networks, Vol. 4, no. 4, July 1993.

Widrow B, Hoff M.E, 'Adaptive Switching Circuits', IRE WESCON Convention Record, New York: IRE, pp. 96-104, 1960.