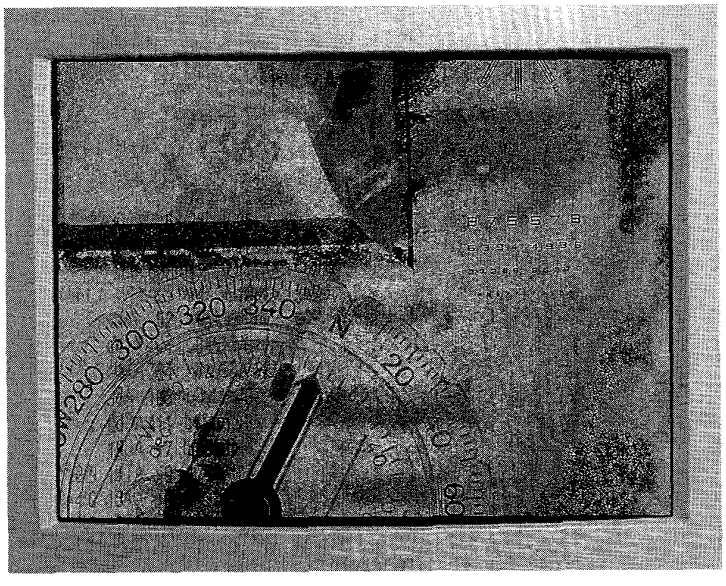
Applying Radial Basis Functions



The Image Bank©A.T. Willet & PhotoDisc, 1993

Neural-network-based algorithms strike a good balance between performance and complexity in adaptive equalization, show promise in spread spectrum systems

BERNARD MULGREW

Adaptive equalization has been an active area of research for many years. Even in 1985 there were a plethora of available solutions [1] whose properties were well understood. Many of the techniques are firmly based on linear adaptive filter algorithms and exhibit the same much-lauded 'learning' property as neural networks. Alternatively, maximum likelihood strategies, which are usually based on the Viterbi algorithm (VA) [2] and its variants, have long been understood to provide the best performance of all equalization techniques. Why then is it worth considering the application of artificial neural networks (NN) to this problem? The answer comes in two parts.

The first is one that has always driven science, and that is curiosity. How well will a neural network perform in this benchmark problem and how will it fare when compared with standard solutions?

Initial work [3] demonstrated that multilayer perceptron (MLP) equalizers were superior to conventional transversal and decision feedback equalizers in terms of the usual measure of equalizer performance, which is bit error rate (BER). On the other hand, the work also highlighted several of the difficulties that are well known in the wider application of MLP's.

These are the extreme length of training times; the indeterminate nature of the training times; the lack of a methodology for architecture selection. These problems are largely unsolved and severely restrict the practical application of MLP's in this area.

1053-5888/96/$5.00©1996IEEE

The second reason for considering neural networks as adaptive equalizers is an engineering one. Each of the conventional solutions has associated with it a measure of computational complexity and a measure of performance. Thus, for example, a transversal equalizer is one of the least computationally demanding techniques and one of the poorest performers, whereas maximum likelihood has theoretically the best performance and can be extremely expensive computationally.

There is always a demand for algorithms that provide alternative compromises between performance and complexity. For example, an algorithm with slightly poorer performance than maximum likelihood but significantly lower complexity would be particularly attractive and provide the practitioner with a useful alternative solution in addressing a range of adaptive equalization problems.

If MLP's are not the solution, why should radial basis function (RBF) networks fare better? The answer to this question lies in the structure of the RBF network and its close relationship to Bayesian methods for channel equalization and interference rejection. This relationship has encouraged the blending of techniques from communications, neural networks, and signal processing. For example, in many applications of neural networks there may not be a universally recognised "best" or "optimal" solution. The network is presented with a training set of input/output pairs and learns a nonlinear mapping from input to output. The network is then tested on unseen data, and if performance goals are met, the operation is deemed a success.

By contrast, in the design of nonlinear adaptive equalizers and receivers, the optimal solution has been clearly defined by the communications community as the Bayesian or one-shot receiver [5-7]. There are several papers in the communications literature that develop this receiver from the perspective of known channel characteristics. It has been compared and contrasted with both linear equalizers and maximum likelihood methods. The radial basis function network, on the other hand, has its origins in the signal processing and neural networks literature [8], and is a general purpose method for approximating nonlinear mappings. Associated with it is a body of techniques for training the network, such as data clustering, gradient descent, and recursive least squares. Further, it has a robust quality, which indicates that it may not be necessary to learn the exact Bayesian solution to achieve satisfactory performance. Thus, the signal processing and neural network perspective on the problem has enabled the development of algorithms that converge reliably in a reasonable timescale for practical application.

The general approach in this article is to cover the application of radial basis function networks, or equivalently, adaptive Bayesian techniques, to a range of equalization and interference rejection problems. In all application areas, the training of the receiver is assumed to be non-blind i.e., a training sequence is available. Simple examples are used extensively to illustrate the methods. Thus, all transmitted data is assumed to be binary and hence the terms *bit rate* and *symbol rate* are used interchangeably. All signal and channels are assumed to be real. The interested reader can find the complex versions developed in [9-11]. The focus of the discussion will be on finite-memory linear channels. Comments on the use of the techniques on nonlinear channels are made occasionally, where appropriate.

Radial Basis Function Networks

Originally, RBF networks were developed for data interpolation in multi-dimensional space [12-14]. The interpolation problem can be stated as follows: given a set of vectors $\{\mathbf{y}_i\}$ and a set of associated scalar data points $\{u_i\}$, find a function $F(\ )$ of the vectors that is constrained to go through all the data points, i.e.,

$$u_i = F(\mathbf{y}_i) \quad \forall i$$

The function can then be used to calculate values for the data points at vectors that are not in the original set. One solution is to choose $F(\mathbf{y})$ such that:

$$F(\mathbf{y}) = \sum_i w_i \, \phi(\|\mathbf{y} - \mathbf{y}_i\|)$$

where $\phi(\|\mathbf{y} - \mathbf{y}_i\|)$ is a radially symmetric scalar function with $\mathbf{y}_i$ at its center. Hence, the vectors $\{\mathbf{y}_i\}$ are frequently referred to as the centers. The vector norm $\|\cdot\|$ is usually an Eucilean one. Possible choices for the radial basis function $\phi(\ )$ include a thin plate spline:

$$\phi(\zeta) = \frac{\zeta}{\sigma^2} \log\left(\frac{\zeta}{\sigma}\right)$$

a multi-quadratic:

$$\phi(\zeta) = \sqrt{\zeta^2 + \sigma^2}$$

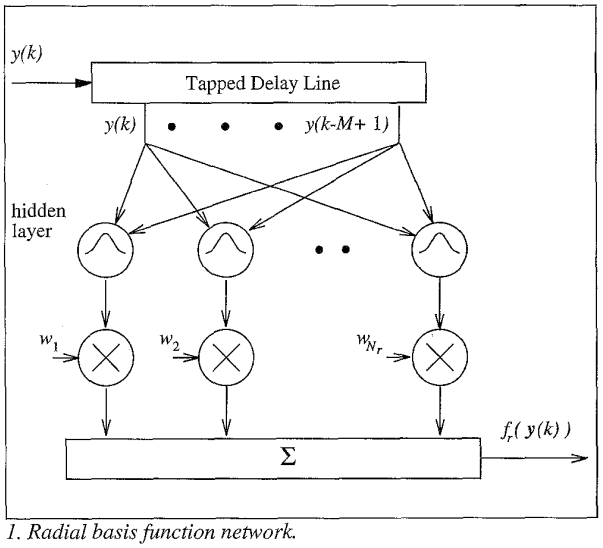
an inverse multi-quadratic:

$$\phi(\zeta) = \frac{1}{\sqrt{\zeta^2 + \sigma^2}}$$

a Gaussian kernel

$$\phi(\zeta) = \exp\left(-\frac{\zeta^2}{2\sigma^2}\right)$$

The parameter, σ, controls the radius of influence of each basis function. This is particularly evident in the case of the inverse multi-quadratic and the Gaussian kernel, as they are both bounded and localised in that $\phi(\zeta) \rightarrow 0$ as $\zeta \rightarrow \infty$. Τηε ραδιυσ, σ, determines how rapidly the basis function goes to zero as $\mathbf{y}$ gets further from the center $\mathbf{y}_i$. Micchelli's results [12] indicate that there are a range of both bounded and unbounded function that are suitable for interpolation in that they lead to linear sets of equations in the weights $\{w_i\}$ for which a unique solution exists.



1. Radial basis function network.

In a signal processing context, interpolation can be problematic. Typically, the amount of data available is larger than the number of basis functions (degrees of freedom) required to give an acceptable approximation i.e. the number of linear equations is greater than the number of unknowns. Further, the data may also be corrupted by noise in which case interpolation or fitting the function through every data point is inappropriate.

To accommodate these situations, Broomhead and Lowe [8] re-interpreted the RBF network as a least squares estimator. This can be achieved by relaxing the requirement for each input vector to have an associated basis function and by removing the requirement for the centers to be drawn from the original set of vectors. The set of linear equations for the weights is then overdetermined with a minimum norm solution provided through the use of a Moore-Penrose pseudo inverse [8]. The approach is justified by universal approximation properties of RBF networks, which were demonstrated first by Hartman, *et al.*, [15] for Gaussian kernels and subsequently for a wider class of basis functions by Park and Sandberg [16, 17]. Essentially, an RBF network with the same radius for each basis function can form an arbitrarily close approximation to any continuous function provided that there are enough basis functions.

Broomhead and Lowe's approach has lead to its widespread application in signal processing problems such as prediction of time series [18], system modelling [19], interference rejection [20], and channel equalization [21]. Figure 1 depicts a feedforward RBF network that is commonly used in such applications. The tapped delay line (TDL) extracts the input M-vector $\mathbf{y}(k)$, where

$$\mathbf{y}(k) = [y(k)\, y(k-1) \ldots y(k-M+1)]^T$$

The network function, $f_r(\mathbf{y}(k))$, is defined as a linear combination of basis functions:

$$f_r(\mathbf{y}(k)) = \sum_{i=1}^{N_r} w_i\, \phi(\|\mathbf{y}(k) - \mathbf{c}_i\|) \tag{1}$$

The vectors $\{\mathbf{c}_i\}$ are the centers and N_r is the number of centers. In general, RBF networks are easier to train than MLP's, primarily because the learning processes for the centers, $\mathbf{c}_i$, the radius parameter, σ, and the weights w_i can be performed in sequence. For example, unsupervised clustering [22, 23] can be used to estimate the centers. An estimate of the variance of the input vector with respect to each center provides the radius. Finally, having fixed the centers and the radii, we calculate the weights as a linear-in-the-parameter problem, for which the least mean squares (LMS) algorithm provides a simple solution. Having performed this initial estimate of the network parameters, we fine tune that estimate using gradient techniques based on all the parameters, rather than the weights alone [11]. An alternative strategy might be to choose a large set of vectors from realizations of $\mathbf{y}(k)$ as candidate centers, and use forward regression techniques such as the orthogonal least squares (OLS) algorithm to select a parsimonious set of centers and associated weights [24]. Indeed, it was OLS training that was used in the initial application of RBF networks to the channel equalization problem [21].

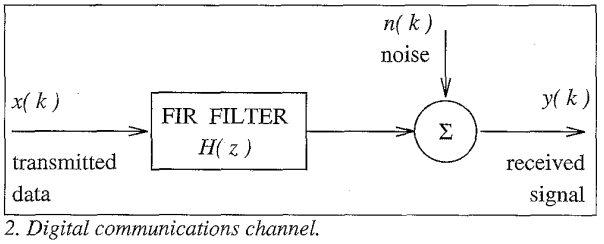
Bayesian Equalizers and RBF Networks

Many digital communications channels are subject to intersymbol interference (ISI). This interference is usually a result of the restricted bandwidth allocated to the channel and/or the presence of multipath distortion in the medium through which the information is transmitted. Many such channels can be characterized by a finite impulse response digital filter and an additive noise source [25]. The appropriate channel model is depicted in Fig. 2. The digital data sequence $\{x(k)\}$ is passed through a linear dispersive channel of finite impulse response (FIR). The observed sequence, $\{y(k)\}$, is formed by adding Gaussian random noise $\{n(k)\}$ to the output of the FIR filter. The relationship between the channel input, $x(k)$, and the channel output, $y(k)$, can be summarized in the following equation.

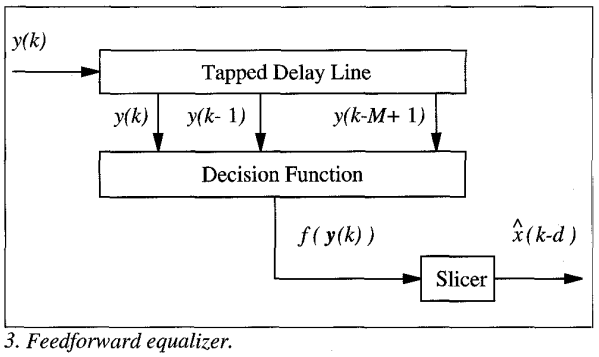
$$y(k) = \sum_{i=0}^{N-1} h_i x(k-i) + n(k)$$

The transfer function of the FIR filter is

$$H(z) = \sum_{i=0}^{N-1} h_i z^{-i}$$



2. Digital communications channel.



3. Feedforward equalizer.

where N is the length of the impulse response.

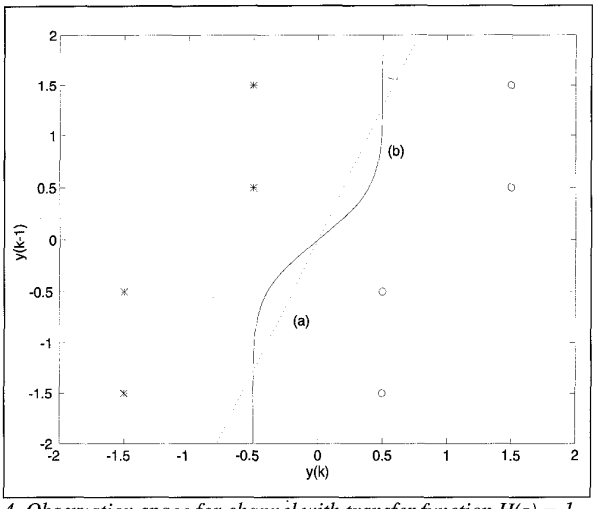
In symbol-by-symbol equalization, the problem is that of using the information present in the observed channel output M-vector $\mathbf{y}(k)$ to produce an estimate, $\hat{x}(k-d)$, as in Fig. 3, of the channel input, $x(k-d)$. The device or algorithm that performs this function is the equalizer. Such equalizers operate in two stages: (i) a scalar function $fd(\ )$ of the vector $\mathbf{y}(k)$ is evaluated— the decision function; (ii) a slicer or memoryless decision device that selects the element of the transmitted alphabet that is closest to $fd(\mathbf{y}(k))$. For binary data, the slicer is the sign function i.e., $\text{sgn}(\zeta) = 1$, if $\zeta \geq 0$, and 0 otherwise. Such an equalizer has order, M, and operates with lag or decision delay, d.

One method of adaptively compensating for the ISI is to restrict $fd(\ )$ to be a linear function of $\mathbf{y}(k)$. The function of the linear adaptive filter is to construct a causal approximation to the inverse of the channel i.e., $H^{-1}(z)$. The output of the adaptive filter is then applied to the slicer to form a reconstruction of the transmitted sequence. This approach has both an intuitive and practical appeal: equalization has been reduced to an inverse filtering or deconvolution problem; there are many linear adaptive filter algorithms available with which to train the equalizer [26]. The disadvantages of this technique are well known [1]. In attempting to compensate for the frequency response of the channel, the adaptive filter can enhance the additive noise component; the adaptive filter does not exploit the fact that the transmitted sequence is drawn from a finite alphabet.

In order to exploit the characteristics of the transmitted sequence more fully, it is appropriate to highlight the finite state nature of the channel itself. To simplify the development it is convenient to assume that the transmitted sequence is chosen from {-1, +1} with equal probability and that the sequence is i.i.d. The channel $H(z)$ is assumed to be purely real rather than the more general complex case. For the simple case where the channels has two coefficients and the equalizer vector $\mathbf{y}(k)$ has two elements, all possible channel output vectors can be summarized by the following equation:

$$\begin{bmatrix} y(k) \\ y(k-1) \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & 0 \\ 0 & h_0 & h_1 \end{bmatrix} \begin{bmatrix} x(k) \\ x(k-1) \\ x(k-2) \end{bmatrix} + \begin{bmatrix} n(k) \\ n(k-1) \end{bmatrix}$$

which can be written more compactly as



4. Observation space for channel with transfer function $H(z) = 1 + 0.5z^{-1}$ ($M = 2$; $N = 2$; $d = 0$): (a) linear decision boundary; (b) Bayesian decision boundary.

$$\mathbf{y}(k) = \mathbf{H}\,\mathbf{x}(k) + \mathbf{n}(k) \tag{2}$$

Equation 2 is fundamental to the development of many equalization and interference rejection techniques. More generally for the case where $\mathbf{y}$ has M elements and the ISI extends over N symbols, the "impulse response matrix" $\mathbf{H}$ has M rows and $M + N - 1$ columns. It is a Toeplitz matrix and has the following form:

$$\begin{bmatrix} h_0 & h_1 & h_2 & \cdots & h_{N-1} & 0 & \cdots & 0 \\ 0 & h_0 & h_1 & \cdots & h_{N-2} & h_{N-1} & \cdots & 0 \\ \cdot & & & & \cdot & \cdot & & \cdot \\ \cdot & & & & \cdot & \cdot & & \cdot \\ 0 & 0 & 0 & \cdots & 0 & h_0 & \cdots & h_{N-1} \end{bmatrix}$$

Returning to the simple example, the vector $\mathbf{y}'(k) = \mathbf{H}\mathbf{x}(k)$, contains channel output values before noise has been added. Because the channel input vector, $\mathbf{x}(k)$, contains 3 elements, each with 2 possible values, there are a total of 2^3 states for $\mathbf{x}(k)$, and hence 2^3 states for the channel output vector $\mathbf{y}(k)$. Table 1 shows all possible channel output vectors for a simple 2 coefficient channel.

Figure 4 shows each of these 8 possible points plotted as either a * to indicate that the output vector represents an input $x(k) = -1$, or a ○ (circle) to represent an input $x(k) = +1$. As noise is added to the vector $\mathbf{y}'(k)$, the 8 points become 8 clusters and the 8 possible values of $\mathbf{y}'(k)$ are the 8 means or centers of the clusters. Thus, the equalization problem is now one of classification, i.e., the task is to assign regions within the observation space spanned by the noisy vector $\mathbf{y}(k)$ to represent inputs of either $x(k) = +1$, or $x(k) = -1$.

A linear equalizer performs such a classification in conjunction with a decision device or slicer. The decision boundary is the locus of all values of $\mathbf{y}(k)$ for which the output of the linear filter is zero. The decision device in this simple case is a sign function. Thus, zero is the point where the slicer

decides whether +1 or -1 was transmitted. If the equalizer has transfer function $G(z) = \sum_{n=0}^{M-1} g_n z^{-n}$, then the boundary is the solution of

$$\mathbf{g}^T \mathbf{y}(k) = 0 \tag{3}$$

where

$$\mathbf{g} = [g_0\ g_1 \cdots g_{M-1}]^T$$

For the simple example where $\mathbf{y}(k)$ has only two elements, Eq. 3 represents a straight line rather than a hyperplane. The coefficients of the equalizer $\mathbf{g}$ are calculated by minimizing the mean squared error with respect to $\mathbf{g}$, and is thus a Wiener filter. The straight line divides the space into two regions. All observation vectors to the right of line will be classified as indicating that $x(k) = +1$ and all points to the left of the line as $x(k) = -1$. If, however, noise was added to the output of the channel, then output vectors distributed about each of the 8 centers would be observed. Take, for example, the points $\mathbf{y}(k) = [-0.5\ \ 0.5]^T$ and $\mathbf{y}(k) = [1.5\ 0.5]^T$. The point associated with a -1 decision is closer to the boundary than the point associated with a +1 decision. Therefore, there is a higher probability of a channel output centerd on $[-0.5\ \ 0.5]^T$ being incorrectly detected as a +1 than a channel output centerd on $[1.5\ 0.5]^T$ being incorrectly detected as a -1. This is clearly a non-optimum strategy.

This geometrical description of the shortcomings of the linear equalizer leads directly to a Bayesian approach [3, 5]. Having observed the vector $\mathbf{y}(k)$, we decide in favor of +1 rather than -1, if the probability that it was caused by $x(k-d) = +1$ exceeds the probability that it was caused by $x(k-d) = -1$, and vice versa. The optimal decision boundary is the locus of all values of $\mathbf{y}(k)$ for which the probability $x(k-d) = +1$ is equal to the probability that $x(k-d) = -1$, given the same value of $\mathbf{y}(k)$. More formally, the decision boundary is defined as the solution to a conditional probability equation.

$$P_{x|\mathbf{y}}(\{x(k-d)) = +1\}|\mathbf{y}(k) = P_{x|\mathbf{y}}(\{x(k-d)) = -1\}|\mathbf{y}(k)) \tag{4}$$

The boundary defined by this equation can be expressed in a slightly more useful manner by applying Bayes rule and using the common assumption that the probability of transmitting a +1 is equal to the probability of transmitting a -1.

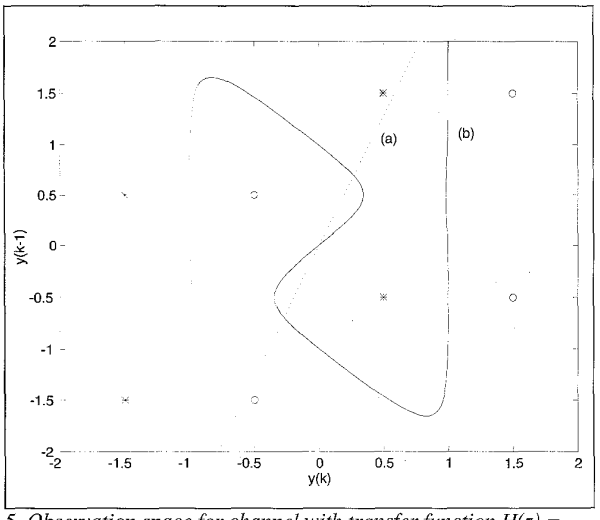
This defines the Bayesian decision function:

$$f_b = \mathbf{y}(k)) = p_{\mathbf{y}|x}(\mathbf{y}(k)|\{x(k-d) = +1\}) - p_{\mathbf{y}|x}(\mathbf{y}(k)|\{x(k-d) = -1\}) \tag{5}$$

with associated boundary

$$f_b(\mathbf{y}(k)) = 0$$

where $p_{\mathbf{y}|x}$ is the condition density function of $\mathbf{y}(k)$ given $x(k-d)$. This decision boundary is also depicted in Fig. 4. Another shortcoming of linear equalizers can also be demonstrated using this geometrical presentation. If the channel is non-minimum phase it may not be possible to separate the two classes at all using a straight line. This situation is illustrated in Fig. 5.



5. Observation space for channel with transfer function $H(z) = 0.5 + 1.0z^{-1}$. ($M = 2$; $N = 2$; $d = 0$): (a) linear decision boundary; (b) Bayesian decision boundary.

For the signal generation process defined by Eq. 2, the observation vector $\mathbf{y}(k)$ is drawn from a Gaussian mixture, with means given by the values of $\mathbf{y}' = \mathbf{H}\,\mathbf{x}$. The conditional densities that constitute the decision function are straightforward to evaluate. The set of noise free output states of $\mathbf{y}'(k)$ is partitioned into two sets conditioned on the transmitted symbol of interest:

$$S^+ = \{\mathbf{y}'(k)|x(k-d) = +1\}$$

and

$$S^- = \{\mathbf{y}'(k)|x(k-d) = -1\}$$

Thus, the Bayesian decision function becomes

$$f_b(y(k)) = \sum_{\mathbf{y}'_i \in S^+} \frac{\exp\left(-\frac{\|\mathbf{y}(k) - \mathbf{y}'_i\|^2}{2\sigma_n^2}\right)}{(2\pi\sigma_n^2)^{N_r/2} N_r} - \sum_{\mathbf{y}'_j \in S^-} \frac{\exp\left(-\frac{\|\mathbf{y}(k) - \mathbf{y}'_i\|^2}{2\sigma_n^2}\right)}{(2\pi\sigma_n^2)^{N_r/2} N_r} \tag{6}$$

Since the decision function is applied to a slicer, the scaling factor, $(2\pi\sigma_n^2)^{N_r/2} N_r$, is irrelevant as far as the decisions are concerned and can be normalized to unity. The relationship between the Bayesian decision function and the RBF of Eq. 1 is clear.

The centers of Eq. 1 are the noise free states of Eq. 6, and the weights of Eq. 1 are +1 or -1, depending on whether a +1 or -1 was transmitted. This relationship between the Bayesian

decision function and the RBF network provides many insights that assist with the formulation of training algorithms.

Training the Equalizer

Many communications systems that exploit adaptive receivers or equalizers are designed to include a training signal which is transmitted at regular intervals. The receiver has a replica of this training signal and can use it for supervised training of the receiver subsystems. The simplest example is a linear adaptive equalizer [1]. The actual data is transmitted between the bursts of training data. If the channel characteristics change slowly with time, there may be a requirement for the receiver to track changes in the channel between the training bursts. One method of doing this is known as decision directed mode where the decisions from the equalizer are assumed to be correct and used to continue the training process when actual data is being transmitted. In common with many bootstrap techniques, it works well when the decisions are correct but can produce bursts of errors or error propagation in response to one incorrect decision. Training of RBF equalizers is usually a two stage process consisting of estimation of the centers or clustering and learning of the weights in the output layer [27]. When selecting a clustering algorithm there are three design choices to be made. Should it be supervised or unsupervised; explicit or implicit; vector or scalar?

When the receiver has access to a replica of the transmitted signal $x(k)$ for a limited period, clustering can be supervised during that period. For example, if the transmitted data is available to the receiver, it is possible to uniquely identify which row of Table 1 is appropriate and update an estimate of that center. If training data is not available or is interrupted, then an unsupervised clustering algorithm such a κ-means algorithm [28] or adaptive κ-means algorithm [23] can be used. The convergence performance of the supervised techniques is superior to the unsupervised ones, but unsupervised may be more convenient if there is a requirement to track changes in a slowly varying channel without allowing the possibility of error propagation to the centers through decision errors.

Explicit clustering algorithm operate directly on the received vector $\mathbf{y}(k)$ to produce estimates of the centers or means of the distribution. Alternatively, if the channel is linear, an adaptive filter algorithm such as the least mean squares (LMS) or recursive least squares (RLS) can be used to estimate the channel impulse response; implicit estimates of the centers are generated using all possible realisations of $\mathbf{x}(k)$ and the identity $\mathbf{y}' = \mathbf{H}\,\mathbf{x}$.

Generally, the number of channel coefficients is smaller than the number of centers and hence the implicit method requires a shorter training period than the explicit one. It also tends to perform better on time-varying channels. However, if the channel is subject to nonlinear distortion, the implicit method based on Eq. 2 would either produce biased estimates of the centers or would have to be modified to include a nonlinear system identification algorithm. The explicit method is immune to nonlinear distortion.

Table 1:
Possible Noise-Free Output Vectors $\mathbf{y}'(k)$ for $H(z) = 1 + 0.5z^{-1}$

$x(k)$	$x(k\text{-}1)$	$x(k\text{-}2)$	$y(k)$	$y(k\text{-}1)$
-1	-1	-1	-1.5	-1.5
-1	-1	1	-1.5	-0.5
-1	1	-1	-0.5	0.5
-1	1	1	-0.5	1.5
1	-1	-1	0.5	-1.5
1	-1	1	0.5	-0.5
1	1	-1	1.5	0.5
1	1	1	1.5	1.5

An examination of Table 1 reveals that there are only 4 possible values for $y(k)$: either ± 1.5, or ± 0.5. These 4 values alone could be used to reconstruct all eight entries for $y(k)$ and $y(k - 1)$. This is not surprising, since both $y(k)$ and $y(k - 1)$ are just consecutive outputs from a FIR filter driven by a sequence drawn from a finite alphabet i.e., $x(k)$. Thus, as in, [29], vector clustering is not necessary. Rather, it is sufficient to perform scalar clustering on $y(k)$ and use the results to regenerate all of the entries in Table 1. By its very nature scalar clustering will converge more rapidly than vector clustering.

Training the output layer is a supervised training process. The fastest method (in terms of learning rate) is to use the relationship with the Bayesian equalizer and simply assign values to the weights. Thus, if a particular center $\mathbf{y}'_j$ was an element of S_+, then $w_j = +1$. Likewise, if $\mathbf{y}'_j$, then $w_j = -1$. Alternatively, the weights could be trained using an LMS or RLS algorithm e.g.,

$$w_j(k+1) = w_j(k) + 2\mu\phi(\|\mathbf{y}(k) - \mathbf{y}'_j\|)(x(k-d) - f_r(\mathbf{y}(k)))$$

where μ is the step size. If the estimates of the centers are accurate then clearly the assignment technique is to be preferred. However, if the exact number of centers is not known precisely or if there is a deliberate choice to use a reduced set of centers, it may be more prudent to use the LMS or RLS techniques as they will make best use of the actual centers that have been provided.

Similarly in noisy environments, where the clustering techniques may only provide fairly crude estimates of the centers, an LMS or RLS algorithm will make best use of available centers. It is tempting to combine the two techniques i.e., assign the weights to ± 1 and then use these values as the starting weights for the LMS algorithm. However, caution is advised. As noted in [30], the Bayesian decision function of Eq. 6 can be scaled arbitrarily, whereas the LMS

will find a specific set of weights that minimizes the MSE $E[\{x(k-d)-f_r(\mathbf{y}(k))\}^2]$. Assigning +1 or -1 to the weights will not necessarily provide a solution that is close to that which minimizes the MSE. Thus, initializing in this manner may not necessarily speed up the convergence of the LMS algorithm. A possible solution is to follow Cha and Kassam [20] and construct a Bayesian estimate of the transmitted data, i.e.,

$$f_{be}(\mathbf{y}(k)) = E[x(k-d)|\mathbf{y}(k)]$$

This estimate is minimal in a MSE sense and the decision boundary is identical to that implied by Eq. 6. It is straightforward to show that:

$$f_{be}(\mathbf{y}(k)) = \sum_{\mathbf{y}'_j \in S^+} \phi_j^{norm}(\mathbf{y}(k)) - \sum_{\mathbf{y}'_j \in S^-} \phi_j^{norm}(\mathbf{y}(k))$$

where the normalized basis function is:

$$\phi_j^{norm}(y(k)) = \frac{\exp\left(-\frac{\|\mathbf{y}(k)-\mathbf{y}'_j\|^2}{2\sigma_n^2}\right)}{\sum_{\mathbf{y}'_i \in S} \exp\left(-\frac{\|\mathbf{y}(k)-\mathbf{y}'_i\|^2}{2\sigma_n^2}\right)}$$

Since the decision function is also the MMSE estimator, there is no longer any ambiguity about the weights.

Recurrent Networks

The Bayesian decision boundary defined by Eq. 4 has finite memory—in other words, all decisions are conditioned on the M channel output samples contained in the vector $\mathbf{y}(k)$. The quality of the decisions naturally improves as the number of elements in $\mathbf{y}(k)$ is increased. The price to be paid for this improvement is an increase in complexity. The solution, of course, is to seek a recursion for the conditional densities of Eq. 5, which updates them as the data record grows. Such a method was provided in [5] and recently given a radial basis function interpretation in [31]. The starting point is to redefine $\mathbf{y}(k)$ as a growing vector that contains all the available data from time $t = 0$. Thus:

$$\mathbf{y}(k) = [y(k)\ y(k-1)\cdots y(1)\ y(0)]^T$$

The Bayesian decision boundary in common with Eq. 5 is based on conditional densities of the form $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}(k)|\mathbf{x}(k))$. To simplify the presentation it is assumed that the equalizer is operating with a lag, $d = 0$. In general, it can accommodate lags up to N conveniently. Applying total probability theory [32] and using standard probability techniques leads us to the following recursion for the probability density of $\mathbf{y}(k)$ conditioned on the last N - 1 channel inputs:

$$f_{\mathbf{y}}(k) = \tfrac{1}{2} f_{n|+}(k) f_{\mathbf{y}|+}(k-1) + \tfrac{1}{2} f_{n|-}(k) f_{\mathbf{y}|-}(k-1) \qquad (7)$$

where

$$f_{\mathbf{y}}(k) = \tfrac{1}{2} p_{\mathbf{y}(k),N-1}(\mathbf{y}(k)|x(k),\ldots x(k-N+2))$$
$$f_{n|+}(k) = p_n(y(k) - [x(k),\ldots,x(k-N+2),+1]\mathbf{h})$$

and

$$f_{\mathbf{y}|+}(k-1) = p_{\mathbf{y}(k-1),N-1}(\mathbf{y}(k-1)|x(k-1),\ldots,x(k-N+1)=+1)$$

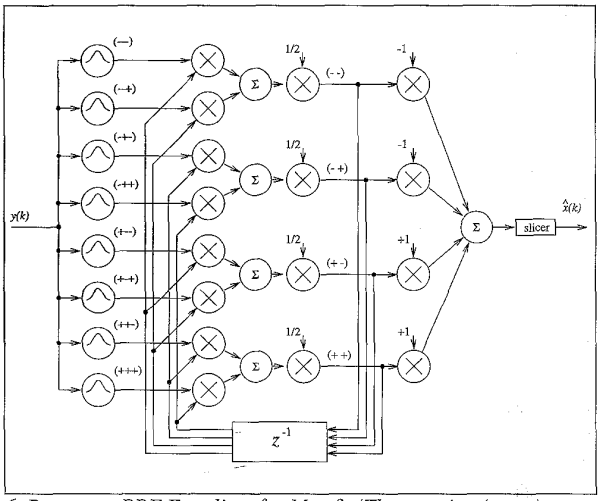
are defined in an analogous manner. $p_n(\ )$ is the probability density function associated with the noise $n(k)$, and the argument is the difference or error between the scalar channel output $y(k)$ and the scalar noise-free channel output $[x(k)\ x(k\text{-}1)\ldots x(k\text{-}N\text{+}1)]\mathbf{h}$, where $\mathbf{h}$ is the channel impulse response vector, i.e.,

$$\mathbf{h} = [h_0\ h_1 \ldots h_{N-1}]^T$$

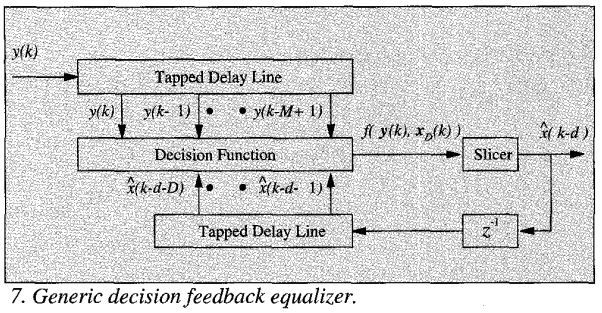
Thus, for Gaussian noise this density function provides a scalar radial basis function expansion with single input $y(k)$, and a set of centers given by all possible values of $[x(k)\ x(k\text{-}1)\ldots x(k\text{-}N\text{+}1)]\mathbf{h}$. The outputs from this expansion provide the inputs to a recurrent network defined by Eq. 7. The desired decision function $p_{\mathbf{y}|x}(\mathbf{y}(k)|x(k))$ can be obtained from $p_{\mathbf{y}(k),N+1}(\mathbf{y}(k)|x(k),\ldots,x(k\text{-}N\text{+}2))$ by repeated application of the total probability theorem, e.g.,

$$p_{\mathbf{y}|x}(y(k)|x(k)) = \tfrac{1}{2} p_{\mathbf{y}|x}(\mathbf{y}(k)|x(k), x(k-1)=+1) + \tfrac{1}{2} p_{\mathbf{y}|x}(\mathbf{y}(k)|x(k), x(k-1)=-1)$$

Thus, the final layer in the network is a simple linear combiner. The network architecture for a simple case where $M = 3$ is illustrated in Fig. 6. The network is trained by estimating the channel impulse response or equivalently the appropriate mapping for finite memory nonlinear channels. MLP recurrent networks have also been applied to the channel equali-



*6. Recurrent RBF Equalizer for M = 3. (The notation (+++) on the layer of basis functions indicates the scalar center which has been used, e.g., (+++) implies the center [+1,+1,+1]**h**. The notation (+ -) indicates the values of the channel inputs upon the densities are conditioned, e.g., (+ -) indicates densities conditioned on x(k) = +1 and x(k - 1) = -1.)*



7. Generic decision feedback equalizer.

zation problem [33]. However, these are even more difficult to train than their feedforward counterparts.

Decision Feedback Equalizers

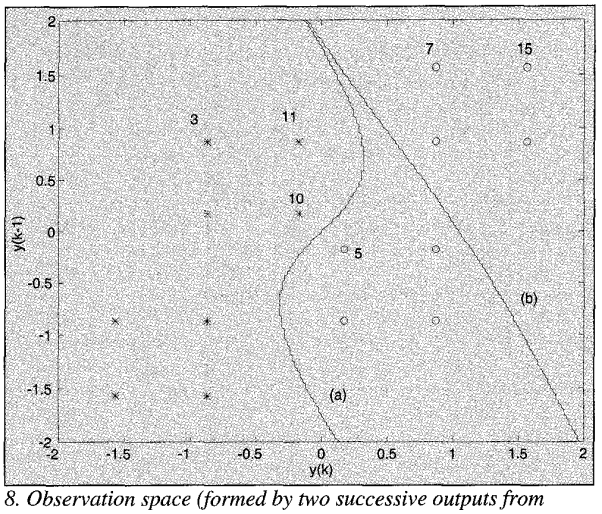
Decision feedback has been used both in linear and nonlinear equalizers to improve the performance. The general structure of a DFE is illustrated in Fig. 7. In contrast to the feedforward equalizer of Fig. 3, the decision function is dependent on previous decisions contained in the vector $\mathbf{x}_D(k)$, where

$$\mathbf{x}_D(k) = [\hat{x}(k-d-1)...\hat{x}(k-d-D)]^T$$

as well as the vector of received samples $\mathbf{y}(k)$. In conventional DFE's, the decision function would be a linear combination of $\mathbf{y}(k)$ and $\mathbf{x}_D(k)$. Such equalizers can be trained with linear adaptive filter algorithms. Care must be exercised in the use of the word 'nonlinear.' In the communications literature, a conventional DFE is described as nonlinear because the decision function can be viewed a a nonlinear function of $\mathbf{y}$ rather than a linear function of $\mathbf{y}$ and $\mathbf{x}_D$. For the remainder of this article, the conventional DFE will be referred to as a linear equalizer to differentiate it from MLP and RBF DFE architectures.

Initial investigations [3,21] into the application of MLP's and RBF's to the DFE problem used a neural network to approximate the decision function, with the aggregate vector $[\mathbf{y}^T\mathbf{x}_D^T]^T$ as the input to the network. Although this work demonstrated that neural networks could provide superior performance to conventional DFE's, the structures employed did not fully exploit the links with Bayesian decision theory. The operation of a Bayesian DFE is most readily described through an example.

Consider a channel with transfer function $H(z) = 0.3482 + 0.8704\ z^{-1} + 0.3482\ z^{-2}$. For the DFE the vector of received samples $\mathbf{y}(k)$ contains two samples i.e., $M = 2$; the feedback vector $\mathbf{x}_D(k)$, contains 2 elements, i.e., $D = 2$; the estimation lag, $d = 1$. Proceeding in a similar manner to the example of Table 1, we construct a state table (Table 2) that relates the input signal states of channel input vector $\mathbf{x}(k)$ to $\mathbf{y}(k)$ in absence of noise. The states have been numbered to aid interpretation. At a particular sample, k the contents of $\mathbf{x}_D$ reduce the number of possible output states of the vector $\mathbf{y}'$ from 16 to 4. For example, if



8. Observation space (formed by two successive outputs from channel with transfer function $H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$: (a) feedforward decision boundary; (b) decision-feedback decision boundary.

$$\mathbf{x}_D(k) = [\hat{x}(k-2)\hat{x}(k-3)]^T = [1\ 1]^T$$

then only states 3, 7, 11, or 15 could have been received, and hence only the centers associated with these states should be used in the Bayesian or RBF network. Thus, the role for the vector $\mathbf{x}_D$ in the decision feedback structure is to select a subset of centers for a particular decision, rather than providing additional terms for the vector input to a neural network [29].

The superior performance of the DFE structure in comparison to the feedforward structure is suggested in Fig. 8. Decision errors are a function of the distance of the centers to the decision boundary. The further the centers are from the boundary, the lower the probability of misclassification. States 5 and 10 are the closest centers to the feedforward boundary, and hence will heavily influence the probability of misclassification. The optimal boundary for the DFE is determined by centers 3, 7, 11, and 15 alone. Of these, 7 and 11 are the closest to the boundary. These states are further from the feedforward boundary than 5 and 10 are from the decision feedback boundary, which suggests that the performance of the DFE will be superior to the feedforward equalizer.

In addition to improving the performance the equalizer decision, feedback reduces the complexity, in that at any time period only a subset of the basis functions is used to form the decision function. To further improve the computational complexity, a technique originally proposed in [34] and extended in the block Bayesian equalizer of [6] can be used. Consider the example of Table 2. Again assume that the decisions $\hat{x}(k-2)$ and $\hat{x}(k-3)$ are correct. The state equation that relates the received signal vector $\mathbf{y}(k)$ to the vector of transmitted symbols is:

$$\mathbf{y}(k) = \mathbf{H}\,\mathbf{x}(k) + \mathbf{n}(k)$$

Table 2:
Possible Noise Free Output Vectors for $H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$ (M = 4, D = 1)*

#	$x(k)$	$x(k-1)$	$x(k-2)$	$x(k-3)$	$y(k)$	$y(k-1)$
0	-1	-1	-1	-1	-1.57	-1.57
1	-1	-1	-1	1	-1.57	-0.87
2	-1	-1	1	-1	-0.87	0.17
3	-1	-1	**1**	1	-0.87	0.87
8	1	-1	-1	-1	-0.87	-1.57
9	1	-1	-1	1	-0.87	-0.87
10	1	-1	1	-1	-0.17	0.17
11	1	-1	**1**	1	-0.17	0.87
4	-1	1	-1	-1	0.17	-0.87
5	-1	1	-1	1	0.17	-0.17
6	-1	1	1	-1	0.87	0.87
7	-1	1	**1**	1	0.87	1.57
12	1	1	-1	-1	0.87	-0.87
13	1	1	-1	1	0.87	-0.17
14	1	1	1	-1	1.57	0.87
15	1	1	**1**	1	1.57	1.57

*"Decision" vector $x_D(k)=[1\ 1]^T$ and associated states of $\mathbf{y}(k)$ are shown in bold typeface.

$$\begin{bmatrix} y(k) \\ y(k-1) \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & | & h_2 & 0 \\ 0 & h_0 & | & h_1 & h_2 \end{bmatrix} \begin{bmatrix} x(k) \\ x(k-1) \\ - \\ \hat{x}(k-2) \\ \hat{x}(k-3) \end{bmatrix} + \begin{bmatrix} n(k) \\ n(k-1) \end{bmatrix} \tag{8}$$

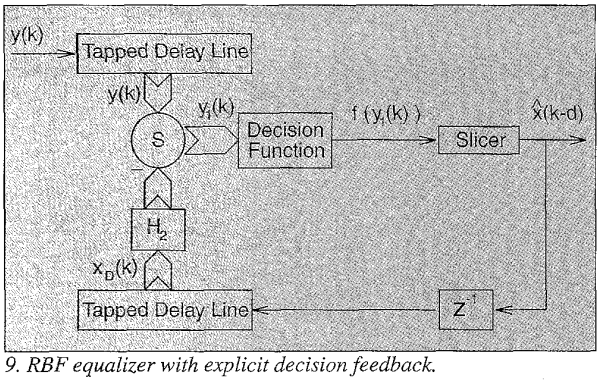
The partitioning of $\mathbf{H}$ and $\mathbf{x}(k)$ highlights the contribution of previous decisions on the observed vector $\mathbf{y}(k)$. Defining the submatrices in a fairly obvious manner, we rewrite Eq. 8 as:

$$\mathbf{y}(k) = \begin{bmatrix} \mathbf{H}_1 | \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(k) \\ - \\ \mathbf{x}_D(k) \end{bmatrix} + \mathbf{n}(k)$$

The effect of the decisions contained in $\mathbf{x}_D(k)$ can then be removed from the observation vector $\mathbf{y}(k)$:

$$\begin{aligned} \mathbf{y}_1(k) &\stackrel{\Delta}{=} \mathbf{y}(k) - \mathbf{H}_2\mathbf{x}_D(k) \\ &= \mathbf{H}_1\mathbf{x}_1(k) + \mathbf{n}(k) \end{aligned} \tag{9}$$

Equation 9 has exactly the same form as Eq. 2, and so we can apply a Bayesian decision function to $\mathbf{y}_1(k)$ rather than $\mathbf{y}(k)$.



9. RBF equalizer with explicit decision feedback.

Although $\mathbf{y}_1(k)$ has the same number of elements as $\mathbf{y}(k)$, the number of noise free states, and hence the number of centers in the RBF network, will be smaller. In this example, the feedforward equalizer $\mathbf{x}(k)$ has 4 elements, and hence 2^4 centers are required, whereas in the DFE, $\mathbf{x}_1(k)$ has 2 elements and hence 2^2 centers are required. The explicit DFE equalizer is illustrated in Fig. 9. The decision function is implemented using a radial basis function network. For this form of equalizer, direct estimation of the channel impulse response is required to form the feedback matrix $\mathbf{H}_2$, and hence this estimate is also used to calculate the centers for the RBF. The weights in the output are assigned at the end of the training period.

It is reasonable at this stage to ask what advantages if any a Bayesian/RBF structure offers over conventional solutions. There are two standard benchmarks for adaptive equalization: (i) a conventional linear DFE; (ii) a maximum likelihood equalizer based on a Viterbi trellis—the maximum likelihood Viterbi algorithm (MLVA). For fixed channels, the bit error rate (BER) performance of the Bayesian DFE is superior to the conventional DFE but inferior to the MLVA. However, in a fading mobile radio environment, where the channel impulse response is time-varying, the Bayesian DFE can out perform MLVA as well [35]. This result may seem surprising at first, since it is well known that the MLVA algorithm provides the best attainable equalization performance. This is indeed the case if the channel impulse response is known to the equalizer and the decision lag, d, is long in comparison to the length of the impulse response.

In a fading channel environment, both the Bayesian equalizer and the MLVA use an estimate of the channel impulse response—usually provided by an LMS algorithm. Since this estimate is never perfect, neither the Bayesian DFE nor the MLVA will operate as well as theory predicts. It is then a question of how robust they are to inaccuracies in the estimate of the impulse response. The second cause of degradation in the performance of the MLVA has to do with the lag, d. Some finite value of d must be chosen at which a hard decision is made in order to provide an input to the system identification algorithm. This exposes an inherent compromise in all adaptive equalization schemes for fading channels: the longer the estimation lag, the better the performance of the equalization process tends to be; the shorter the lag, the better will be the

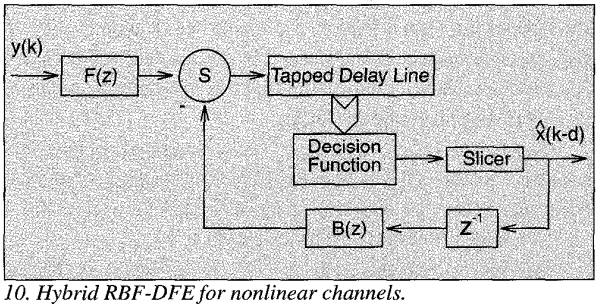
performance of the system identification. The trade-off is a difficult one, since the two phenomena interact. The lag that produces the best overall performance from the system identification does not necessarily produce optimum performance from the MLVA.

In terms of complexity, the Bayesian DFE is more expensive than conventional DFE solutions and less expensive than standard MLVA solutions. As in conventional DFE's, the pragmatic choice for estimation lag is to set it equal to the length of the channel i.e., $d = N$. In [29], it is also shown that there is a simple relation between the number of elements, M, in the received signal vector $\mathbf{y}(k)$, and the estimation lag i.e., $M = d + 1$. This is also a common choice for a conventional DFE. With these choices for M and d, the computational complexity of the Bayesian DFE for 4-QAM signalling is of order 4^N. The complexity of the Bayesian DFE will thus grow rapidly both with the size of the signal alphabet and the length of the channel impulse response. Thus, it is particularly suited to applications such as mobile radio, where the signal alphabet is of moderate size and the ISI typically extends over 4 or 5 symbols. Detailed comparisons in terms of relative performance and complexity are provided in [29] and [35].

The explicit decision feedback architecture of [6], unlike the structure of [36], cannot be used on nonlinear channels because the separation of contributions from detected and undetected symbols inherent in Eq. 9 is not possible. However, the desire to reduce the number of centers in the RBF is still attractive.

Theodoridis, *et al*, [37] developed the architecture of Fig. 10. The structure is essentially a conventional DFE, with the slicer replaced with an RBF decision function in cascade with a slicer. The philosophy of the design is that the conventional DFE removes most of the ISI leaving the RBF to remove the residual ISI due to the nonlinear effects in the channel. The equalizer is trained using a gradient-like algorithm.

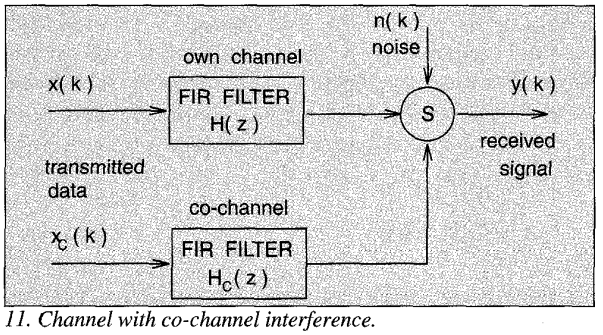
At first sight the presence of the feedback path might suggest that the MSE cost function would be multimodal for this architecture. In fact the training of the conventional DFE consisting of the feedforward FIR filter $F(z)$ and feedback FIR filter $B(z)$ is unaffected by the presence of the RBF network, provided the decisions are correct—which will always be the case during supervised training. The RBF is trained after the conventional DFE using a combination of clustering for the centers and the LMS algorithm for the weights in the output layer.



10. Hybrid RBF-DFE for nonlinear channels.

Co-channel Interference

Although the primary reason for using an equalizer on a communications channel has been to mitigate the effects of intersymbol interference (ISI), more recently it has been demonstrated that conventional equalizers can exploit the cyclostationary nature of the received signal and reduce the distortion due to both co-channel and adjacent channel interference. Radial basis function network can also be applied to this problem without the need to exploit the cyclostationary characteristics of the received signal.

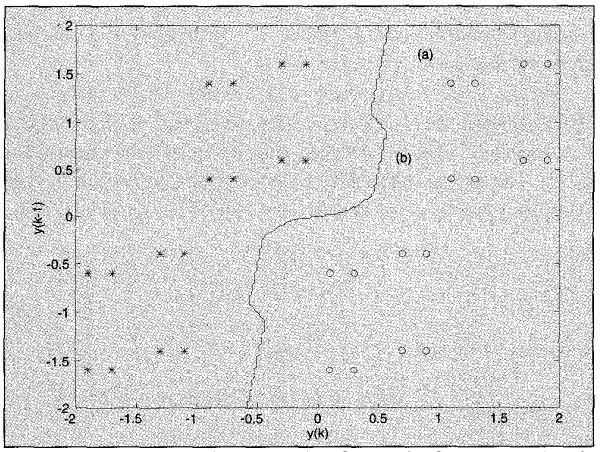


11. Channel with co-channel interference.

Figure 11 illustrates a simple case of channel of interest, the "own channel," and one interfering channel, the "co-channel." The received signal vector $\mathbf{y}(k)$ is now composed of three rather than two terms: an own-channel component $\mathbf{Hx}(k)$ as Eq. 2; a similar co-channel component $\mathbf{H}_c\mathbf{x}_c(k)$; and a noise term, $n(k)$. Thus:

$$\mathrm{y}(k) = [\mathrm{H} \mid \mathrm{H}_c] \begin{bmatrix} \mathrm{x}(k) \\ - \\ \mathrm{x}_c(k) \end{bmatrix} + \mathrm{n}(k) \tag{10}$$

The effect of the co-channel is to increase the number of centers, because the aggregate vector $[\mathrm{x}^T(k)\mathrm{x}_c^T(k)]^T$ has by definition more elements than $\mathbf{x}(k)$. Figure 12 illustrates the states and Bayesian decision boundary for a simple example.



12. Observation space for channel with transfer function H(z) = 1 + 0.5z-1 and co-channel with transfer function Hc(z) = 0.1 + 0.3z -1: (a) linear decision boundary; (b) Bayesian decision boundary.

The co-channel interference has altered the Bayesian boundary from that of Fig. 4 for the single "own-channel" case. The number of centers or noise free states can increase dramatically, with a consequent increase in complexity for the receiver. In [38], it is demonstrated that near to optimal performance can be obtained using a reduced set of centers. The network is trained using a combination of supervised and unsupervised clustering. Supervised clustering alone cannot be used because the receiver does not usually have access to training data from the co-channel i.e., $x_c(k)$. A two stage learning process is adopted: (i) the 'own-channel' impulse response matrix **H** is estimated using the LMS or RLS algorithms; (ii) the effect of the "own-channel" is removed from received signal to form $\mathbf{y}''(k) = \mathbf{y}(k) - \mathbf{H}\mathbf{x}(k)$; (iii) unsupervised clustering is applied to $\mathbf{y}''(k)$ to obtain the centers due to the co-channel interference $y_c''(k)$; (iv) the complete set of centers is reconstituted using every possible combination of $\mathbf{x}(k)$ *and* $y_c''(k)$.

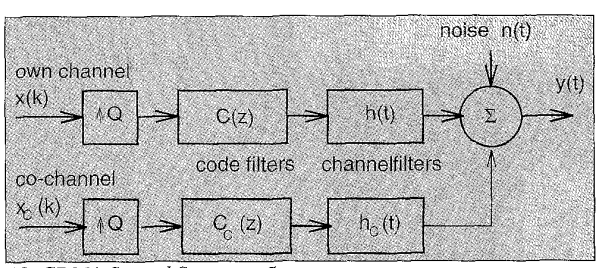
$$\mathbf{y}_c(k) = \mathbf{y}_c''(k) + \mathbf{H}\mathbf{x}(k) \tag{11}$$

Equation 11 also divides the set of all possible centers in two sets: one with $x(k - d) = +1$, and one with $x(k - d) = -1$. The weights on the output layer of the RBF can then be simply assigned to +1 or -1, depending on the originating center. At the unsupervised clustering step (iii), the complexity of the network can be reduced by using a smaller number of centers than that indicated by the degrees of freedom in the system. While this degrades the performance of the equalizer, the degradation can be compensated for by increasing the value of σ of radius parameter of the RBF. A decision feedback equalizer can also be constructed that compensates for co-channel interference. Details are available in [39].

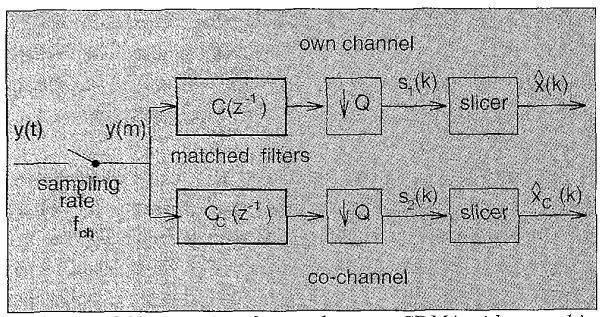
Spread Spectrum

Code division multiple access (CDMA) spread spectrum (SS) communications systems has been an active area for many years [40]. Such systems are characterized by many users simultaneously transmitting over the same bandwidth. A variety of methods are available for combatting the inherent multiple access interference [41-42].

Figure 13 is a schematic of a simple CDMA system. In this example, there are two users, an "own-channel" and a "co-channel." The binary data, $\mathbf{x}(k)$ and $\mathbf{x}_c(k)$, is generated at f_b Hz. The action of the SS transmitter is effectively to up the sampling rate by a factor Q to the chip rate $f_{ch} = Qf_b$ Hz. The



13. CDMA Spread Spectrum System



14. Matched filter receiver for synchronous CDMA without multipath.

high sampling rate signal is then applied to the code filter such as $C(z)$. This finite impulse response of this filter is limited to Q sampling periods at the high sampling rate. The effects of a multipath channel and the analog transmit and receive filters are combined in the channel filters with impulse response such as $h(t)$.

In the simplest spread spectrum system, the impulse response $h(t)$ and $h_c(t)$ reduce to scaling factors. This is known as a synchronous CDMA system without multipath. A conventional receiver is illustrated in Figure 14. The analog receive signal $y(t)$ is sampled at the chip rate to form the discrete time signal $y(m)$, which is applied to a filter matched to the code filter, e.g., $C(z^{-1})$. The output of the matched filter is down sampled to the symbol rate before being applied to a slicer to produce the estimate of the transmitted bit.

As an example, consider the case where the code length $Q = 4$ and the two spreading codes are $\mathbf{c} = [1\ \ 1\ \ -1\ -1\ ]^T$ and $\mathbf{c}_c = [-1\ 1\ -1\ 1\ ]^T$. The code filters are related to the codes in a simple manner, e.g.,

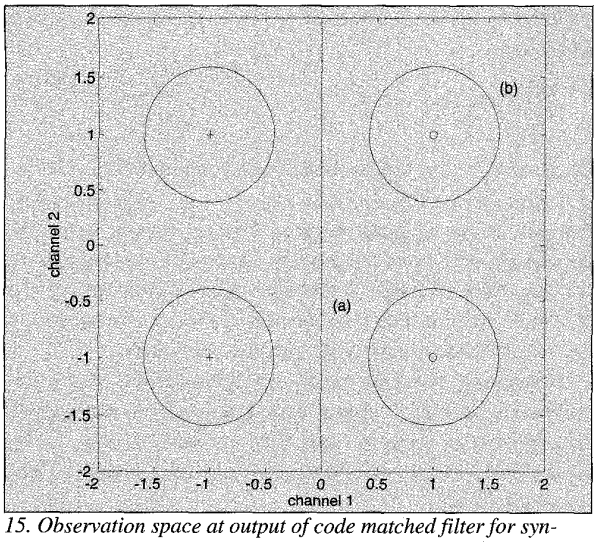
$$C(z) = [1\ \ z^{-1}\ \ z^{-2}\ \ z^{-3}]\mathbf{c}$$

In this case, the codes are orthogonal since $\mathbf{c}^T\,\mathbf{c}_c = 0$. Since the combination of the matched filters and the downsampling process does not destroy any information in the received signal, it is sufficient to base any decision upon the signal vector $\mathbf{s} = [s_1(k)\ s_2(k)]^T$. For future reference, $s_1(k)$ and $s_2(k)$ will be known as the channel 1, and channel 2 signals, respectively.

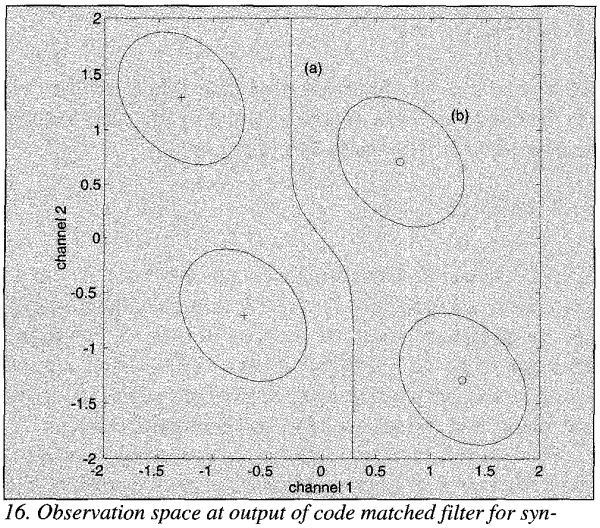
Figure 15 illustrates the Bayesian decision boundary for recovering data on the "own channel." A small number of contours of the decision function are also illustrated. The first point to note is that the decision boundary in this case is linear and is dependent only on channel 1. Thus, it can be implemented with a slicer. Hence, the lower "co-channel" path of Fig. 14 is redundant in this case. The contours of the decision function are circular, indicating that the noise components on $s_1(k)$ and $s_2(k)$ are uncorrelated.

The nature of the decision function and boundary can change dramatically if the codes are not orthogonal. For example, if the "co-channel" code is modified to $\mathbf{c}_c = [0\ 1\ -1\ 1\ ]^T$. This situation is illustrated in Fig. 16.

The non-orthogonal nature of the codes has two effects: (i) the noise free states of $\mathbf{s}(k)$ are no longer on a rectangular



15. Observation space at output of code matched filter for synchronous CDMA (orthogonal codes) (a) Bayesian decision boundary; (b) contour of decision function.



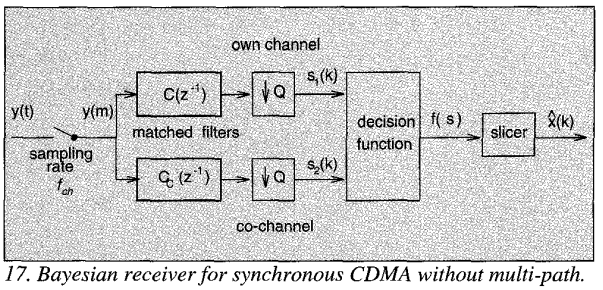
16. Observation space at output of code matched filter for synchronous CDMA (non-orthogonal codes); (a) Bayesian decision boundary; (b) contour of decision function.

grid; (ii) the noise component in $s_1(k)$ and $s_2(k)$ are correlated as indicated by the elliptical nature of the contours of the decision function. The overall effect is that the decision boundary is now a nonlinear function of both s_1 and s_2.

An appropriate receiver structure [43] is illustrated in Fig. 17. There are several ways the decision function could be implemented. The nonlinear decision boundary could be approximated by a linear one, e.g.,

$$f_l(\mathbf{s}(k)) = \mathbf{g}^T\mathbf{s}(k)$$

The weights $\mathbf{g} = [g_0 \;\; g_1]^T$ could be trained using for example an LMS algorithm. Alternatively, the weights could be combined with the matched filters to give one linear filter $G(z)$ operating at the chip rate:



17. Bayesian receiver for synchronous CDMA without multi-path.

$$G(z) = g_1 C(z) + g_2 C_c(z)$$

The complete filter could be trained using any of the adaptive filter methods without prior knowledge of the spreading codes. This filter is often term as an 'equalizer' in this application, as using an adaptive filter removes the restriction on $G(z)$ to be a linear combination of the matched filter responses. Hence, it will have the capability to remove multipath distortion.

The second possible solution is to train an MLP to realize the nonlinear decision boundary as in [43-44]. The difficulties with this approach lie with the training times and architecture selection of the MLP. Third, a radial basis function network, using a Mahalanobis distance metric rather than a Euclidean distance, will reflect the correlated nature of the noise components. Thus,

$$f(\mathbf{s}(k)) = \sum_j w_j \exp\left(\frac{(\mathbf{s}(k)-\mathbf{s}_j)^T \Phi_{ss}^{-1} (\mathbf{s}(k)-\mathbf{s}_j)}{2}\right)$$

where

$$\Phi_{ss} = E[\mathbf{s}(k)-\mathbf{s}_j)(\mathbf{s}(k)-\mathbf{s}_j)^T]$$

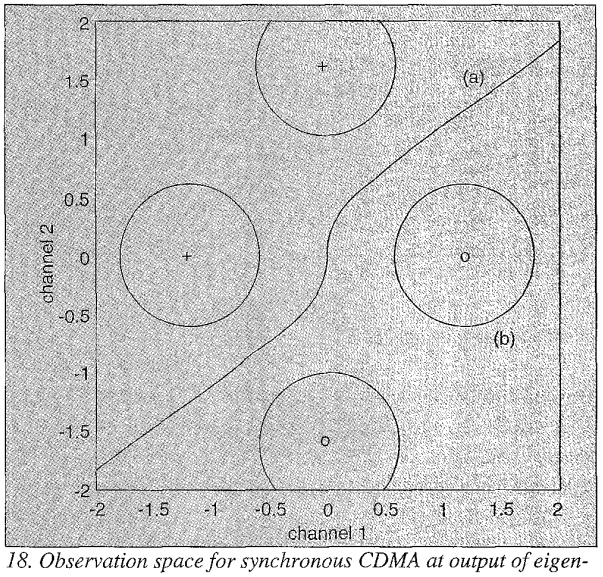
and $\mathbf{s}_j$ is a noise free state of $\mathbf{s}(k)$. The correlated nature of the noise components also means that any unsupervised clustering used to train the network would exploit Mahalanobis rather the Euclidean distance in calculating distances from centers.

Mitra and Poor in [30] adopted a different strategy. They dispense with the matched filters and operate directly on the chip rate discrete sequence $y(m)$. Because this sequence is cyclostationary with period Q, it is convenient to collect together Q consecutive signal samples into an Q-vector $\mathbf{y}(k)$. Thus, for example:

$$\begin{bmatrix} y(kQ+3) \\ y(kQ+2) \\ y(kQ+1) \\ y(kQ) \end{bmatrix} = \begin{bmatrix} \alpha\mathbf{c} & \alpha_c\mathbf{c}_c \end{bmatrix} \begin{bmatrix} x(k) \\ x_c(k) \end{bmatrix} + \begin{bmatrix} n(kQ+3) \\ n(kQ+2) \\ n(kQ+1) \\ n(kQ) \end{bmatrix} \tag{12}$$

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{n}(k)$$

where α and α_c are constants that reflect the different gains experienced by the two channels. The similarity to Eq. 2 is clear. Since there are 2 users, there will be 2^2 noise free states



18. Observation space for synchronous CDMA at output of eigenfilters: (a) Bayesian decision boundary (b) contour of decision function.

of the observation vector $\mathbf{y}(k)$. Since the noise components are uncorrelated, a Euclidean distance can be used for both unsupervised clustering and in forming the RBF network. The disadvantages are: (i) the input vectors to the RBF are now of length Q (4 in this example) as opposed to 2 (the number of users); (ii) any possible noise enhancement provided by the matched filters has been lost, and hence the clustering algorithms may have to operate in a poorer signal/noise environment than the receiver of Figure 17. Note here that, from the perspective of the clustering algorithm, the signal is the sequence of noise free vectors $\mathbf{Hx}(k)$, which includes components from both the "own channel" and the "co-channel" -the noise is the sequence of vectors $\mathbf{n}(k)$.

Even for a code length of $Q = 4$, it is not possible to visualise the centers and decision boundaries in the 4-space spanned by $\mathbf{y}(m)$. However, the noise free states are linear combinations of the two code vectors and hence span a 2-space. It is only the noise component that generates the extra degrees of freedom to span the whole 4-space.

A similar concept is exploited in the MUSIC algorithm [45] for direction finding and spectral analysis. Eigenvalue decomposition of the covariance matrix $\Phi_{yy} = E[\mathbf{y}(k)\mathbf{y}^T(k)]$ provides a means of identifying a set of orthogonal vectors which span the same 2-space as the noise free states. Thus, $\Phi_{yy}=\mathbf{V}\Lambda\mathbf{V}^T$, where Λ is a diagonal matrix of eigenvalues, and the columns of $\mathbf{V}$ are the associated eigenvectors. If the elements of Λ are arranged in order of decreasing size, the last 2 elements are the variance of the noise. This gives the usual partition of the columns of $\mathbf{V}$ into signal and noise eigenvectors i.e., $\mathbf{V} = [\mathbf{V}_s|\mathbf{V}_n]$, where the column of $\mathbf{V}_s$ are the eigenvectors associated with the two largest eigenvalues. The matrix $\mathbf{V}_s$ is (4 x 2) and provides an orthonormal transformation for mapping the 4-vector $\mathbf{y}(k)$ to the 2-space spanned by the noise free states of $\mathbf{y}(k)$. This gives a 2-vector $\mathbf{s}(k)$ where

$$\mathbf{s}(k) = \mathbf{V}_s^T\mathbf{y}(k)$$

Because $\mathbf{V}_s^T$ is an orthonormal transformation (i.e., a series of rotations or reflections) it preserves all the geometrical features of the original 4-space. Figure 18 illustrates the decision boundary in the 2-space spanned by $\mathbf{s}(k)$, showing the Bayesian decision boundary and the contours of the decision function. The contours are again circular indicating the uncorrelated nature of the noise components.

In general, a spread spectrum system can be both asynchronous and subject to multipath distortion. Asynchronous transmission is characterized in the model of Fig. 13 by letting the channel filters have delay as well as gain. Thus, for example, let

$$h(t) = \alpha\delta(t-\tau)$$

and

$$h_c(t) = \alpha_c\,\delta(t-\tau_c)$$

where τ and τ_c are the respective time delays on the two channels, and $\delta(t)$ is an impulse. For multipath distortion, the channel filter will typically have the form:

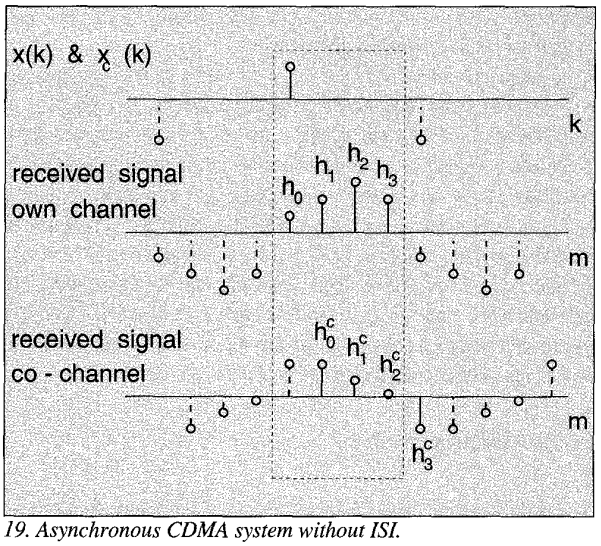
$$h(t) = \sum_j \alpha_j\,\delta(t-\tau_j)$$

Such a system is both multirate and multichannel. It is multirate in the sense that the data rate at the transmitter is f_b Hz, and the sampling rate in the receiver is Q times that at the chip rate of f_{ch} Hz. It is multichannel in that there can be several users, each transmitting at different phases of the bit rate clock over channels with different impulse responses.

The system can be characterised in discrete-time by sampling the complete impulse response of each channel at the chip rate. This complete impulse response is the convolution of the impulse response of the code filter, the analogue transmit and receive filters, and the multipath channel.

A simple example of a 2-user asynchronous system might have an own channel sampled impulse response $H(z) = h_0 + h_1z^{-1} + h_2z^{-2} + h_3z^{-3}$, and a co-channel sampled impulse response $H_c(z) = h_0^cz^{-1} + h_1^cz^{-2} + h_2^cz^{-3} + h_3^cz^{-4} + h_4^cz^{-5} + h_5^cz^{-6}$. In this case, the length of the sampled impulse response is the same as the code length i.e., $Q = 4$. There is no intersymbol interference present in either channel. Typical transmitted and received sequences in Fig. 19 depict the signal components at the receiver due to both the own channel and the co-channel. In this case, the Q-vector of received samples then becomes:

$$\mathbf{y}(k) = \begin{bmatrix} h_3 & h_2^c & 0 \\ h_2 & h_1^c & 0 \\ h_1 & h_0^c & 0 \\ h_0 & 0 & h_3^c \end{bmatrix} \begin{bmatrix} x(k) \\ x_c(k) \\ x_c(k-1) \end{bmatrix} + \mathbf{n}(k) \qquad (14)$$



19. Asynchronous CDMA system without ISI.

Three points are worth noting: (i) even if the impulse responses were simply orthogonal code sequences the columns of **H** would no longer be orthogonal because of the relative delay between the two channels; (ii) the rank of the space spanned by the noise free states is now 3 rather than 2; (iii) the number of centers is 2^3 rather than 2^2. In the presence of multipath, the sampled impulse responses grow beyond the code length $Q = 4$. For example, if the impulse response of the two channels are extended by two samples so that $H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} + h_4 z^{-4} + h_5 z^{-5}$ and $H_c(z) = h_0^c z^{-1} + h_1^c z^{-2} + h_2^c z^{-3} + h_3^c z^{-4} + h_4^c z^{-5} + h_5^c z^{-6}$ the observation vector becomes:

$$\mathbf{y}(k) = \begin{bmatrix} h_3 & 0 & h_2^c & 0 \\ h_2 & 0 & h_1^c & h_5^c \\ h_1 & h_5 & h_0^c & h_4^c \\ 0 & h_4 & 0 & h_3^c \end{bmatrix} \begin{bmatrix} x(k) \\ x(k-1) \\ x_c(k) \\ x_c(k-1) \end{bmatrix} + \mathbf{n}(k) \tag{15}$$

The rank is now 4 and the number of noise free states is 2^4. From Eqs. 14 and 15, it is evident than an RBF network could implement a Bayesian receiver using a combination of supervised/unsupervised clustering and LMS learning as in [30].

To improve the signal/noise conditions prior to clustering, a matched filter strategy similar to Fig. 17 could be employed. The appropriate vector matched filter is simply the transpose of **H**, which is applied to $\mathbf{y}(k)$ as a preprocessor of the decision function. Thus,

$$\mathbf{s}(k) = \mathbf{H}^T \mathbf{y}(k) \tag{16}$$

Unlike Figure 17, the matched filters are not derived from the spreading codes but from the columns of **H**. In the example of Eq. 14, there would be three matched filters, whereas in the example of Eq. 15, there would be four. The matched filters would improve the signal/noise ratio prior to clustering. The practical difficulty with this approach lies in estimating the impulse response.

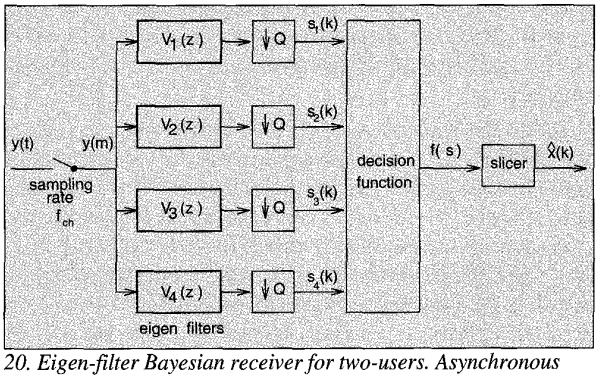
While it may be reasonable to estimate the impulse response of the own channel using a training signal, it is unlikely that similar training signals will be available for the co-channel. Further, in common with the non-orthogonal synchronous case, the noise components on the elements of $\mathbf{s}(k)$ will be mutually correlated necessitating a Mahalanobis distance metric in the RBF.

Matched filters are not the only filters that are useful in this context. The rows of the eigenvector matrix $\mathbf{V}_s^T$ provide an alternative filter network. The action of $\mathbf{V}_s^T$ is well known is the signal processing literature. It is an operator that projects the vector $\mathbf{y}(k)$ onto a subspace spanned by the noise free states of $\mathbf{y}(k)$, and hence reduces the noise. Since the rank of the subspace is less than Q, the vector $\mathbf{s}(k)$ will have a smaller number of components than $\mathbf{y}$.

The projection operator is also an orthogonal transformation, and hence uncorrelated noise components on $\mathbf{y}(k)$ will result in uncorrelated noise components on $\mathbf{s}(k)$. Hence, a Euclidean distance metric is still appropriate. For the example of Eq. 14, $\mathbf{V}_s^T$ would have three rows corresponding to the rank of the signal subspace. The elements of each row would be the coefficients of a Q-tap FIR filter. For the example of Eq. 15, there would be four rows and four Q-tap filters. The complete receiver for a 2-user asynchronous system with multipath is illustrated in Fig. 20 [46-47], showing the eigenfilters $V_j(z)$, corresponding to each row of $\mathbf{V}_s^T$. In general, there will be two filters per user, provided the sampled impulse response of each channel has less than $2Q$ elements. The projection matrix $\mathbf{V}_s^T$ can be estimated directly from the received data alone, without access to transmitted training data. This is usually a three-step process: (i) estimation of the covariance matrix, e.g., a suitable estimate would be

$$\hat{\Phi}_{yy} = \sum_{k=1}^{K} \mathbf{y}(k)\mathbf{y}^T(k)$$

(ii) eigenvalue decomposition; (iii) rank determination using Akaike information criterion. The rank determination step also provides an indirect estimate of the number of users and hence provides a mechanism for allocating the appropriate number of centers is the RBF approximation to the decision function.



20. Eigen-filter Bayesian receiver for two-users. Asynchronous CDMA with multi-path.

Concluding Remarks

At the beginning of this article, we introduced the motives for applying neural networks in general and radial basis functions in particular to adaptive equalization and interference rejection problems. These were scientific curiosity and a desire to provide alternative engineering solutions to the problems. The former is a matter of personal taste—neural networks have a tendency to evoke either extreme positive or negative reactions in many researchers. For the latter, more specific comment can be made. The most encouraging result is the application of an adaptive Bayesian DFE to fading mobile radio channels.

It is the author's opinion that this goes beyond the objectives implied in the introduction in that the adaptive Bayesian DFE can outperform an adaptive MLVA in both computational complexity and bit error rate performance. Such a statement must, however, carry with it certain caveats. In particular, it does not take into account the body of work on reduced state Viterbi algorithms. A more cautious commentator might conclude that the adaptive Bayesian DFE meets the original objectives in that it provides a useful alternative to both linear and maximum likelihood equalizers in terms of the complexity/performance trade-off. The application of RBF and adaptive Bayesian methods to CDMA systems is at a much earlier stage than its ISI counterpart. Current results indicate that they do provide performance somewhere between linear and maximum likelihood methods. The use of the RBF has provided receivers with more controllable training characteristics than MLP receivers. However, the length of the training period is still too long for practical consideration.

There is, of course, much pertinent work that has been omitted for reasons of space and time. Blind equalization, in particular, is a demanding problem that currently receives a great deal of attention. While many techniques have been applied, RBF/Bayesian methods [48-51] have a unique contribution to play in this area, inasmuch as they explicitly exploit the finite nature of the transmitted alphabet. This is unlike, for example, techniques based on higher order statistics or the cyclostationary nature of the received signal. Another topical area is the use of array processing techniques in both conventional and CDMA based communications systems. Arnott [52] has demonstrated how RBF techniques can be used to exploit spatial diversity to overcome multipath interference.

It has not been the author's intention to place the nail in the coffin of MLP's in this application. As indicated in the introduction, significant problems presently preclude their use. However, that is not to say that future theoretical developments will not provide solutions. It is appropriate to re-iterate the major difficulty with RBF/Bayesian solutions, which is the growth in complexity with both the size of the transmit alphabet, P, and the length, N, of the channel. In particular, for an explicit DFE, the complexity is of order P^N. Hence, its application is typically restricted to small signal alphabets such as 4-QAM and channels where the ISI extends over four or five symbol periods.

It is reasonable to ask if it is possible to form approximately the same boundary using a network defined by a smaller number of coefficients. Initial results indicate that there are two classes of problem where this may be possible: (i) if the states are linearly separable it is possible to adaptively design a linear equalizer with better BER performance than the Weiner solution [53]; (ii) if the states are not linearly separable, a McCulloch-Pitts network can be designed in a deterministic manner in a finite number of steps [54]. Both methods require knowledge of the centers, but the resultant equalizer is less complex than an RBF.

Acknowledgements

The author gratefully acknowledges the contributions of coworkers Sheng Chen, Stephen McLaughlin, Peter Grant, and Edward Warner. Many thanks also to Urbashi Mitra, Colin Cowan, Sergios Theodoridis, and Aníbal Figueiras-Vidal for their comments on the original version of the manuscript.

Bernard Mulgrew is Senior Lecturer of Electrical Engineering at the University of Edinburgh. Address for correspondence: Department of Electrical Engineering, The King's Buildings, Mayfield Road, Edinburgh EH93JL, Scotland, UK. E-mail: B.Mulgrew@ee.ed.ac.uk

References

1. S.U.H. Qureshi, "Adaptive equalization," *Proceedings IEEE,* vol. 73, no.9, pp. 1349-1387, 1985.

2. G.D. Forney, "Maximum likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Transactions Information Theory*, vol. IT-18, pp. 363-378, May 1972.

3. S. Siu, GJ. Gibson, and C.F.N. Cowan, "Decision feedback equalisation using neural network structures and performance comparison with the standard architecture," *IEE Proccedings* Part I, vol. 137, no. 4, pp.221-225, Aug. 1990.

4. G.J. Gibson, S. Siu, and C.F.N. Cowan, "The application of nonlinear structures to the reconstruction of binary signals," *IEEE Trans Signal Processing*, vol. 39, no. 8, pp. 1877-1884, Aug. 1991.

5. K. Abend and B.D. Fritchman, "Statistical detection for communications channels with intersymbol interference," *Proccedings IEEE*, vol. 58, no. 5, pp. 779-785, May 1970.

6. D. Williamson, R.A. Kennedy, and G.W. Pulford, "Block decision feedback equalization," *IEEE Transactions on Communications*, vol. 40, no.2, pp. 255-264, Feb. 1992.

7. H.V. Poor and S. Verdu, "Single user detectors for multi-user channels," *IEEE Trans Communications*, vol. 36, no.I, pp. 50-60, Jan 1988.

8. D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321-355, 1988.

9. S Chen, S McLaughlin, and B Mulgrew, "Complex valued radial basis function networks: network architecture and learning algorithms (part I)," *EURASIP Signal Processing Journal*, vol. 25, no. 1, pp. 19-31. January 1994.

10. S Chen, S McLaughlin, and B Mulgrew, "Complex valued radial basis function network: application to digital communications channel equalisation (part II)," *EURASIP Signal Processing Journal*, vol. 36, no. 2, pp. 175-188, March 1994.

11. I. Cha and S.A. Kassam, "Channel equalization using adaptive complex radial basis function networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 1, pp. 122-131, Jan 1995.

12. C.A. Micchelli, "Interpolation of scatter data: distance matrices and conditionally positive definite functions," *Constructive Approximation*, vol. 2, pp. 11-22, 1986.

13. M.J.D. Powell, "Radial basis functions for multivariable interpolation: a review," *Algorithms for Approximation*, pp. 143-167, Oxford, 1987.

14. N. Dyn, "Interpolation of scattered data by radial functions," *Topics in Multivariate Approximation*, Academic Press. New York. 1987.

15. E.J. Hartman, J.D. Keeler, and J.M. Kowalski, "Layered neural networks with Gaussian hidden units as universal approximations," *Neural Computation*, vol. 2, no. 2, pp. 210-215, 1990.

16. I. Park and I.W. Sandberg, "Universal approximation using radial basis function networks," *Neural Computation*, vol. 3, pp. 247-257, 1991.

17. J. Park and I.W. Sandberg, "Approximation and radial basis function networks," *Neural Computation*, vol. 5, pp. 305-316, 1993.

18. T.J. Shepherd and D.S. Broomhead, "Nonlinear signal processing using radial basis functions," *SPE Advanced Signal Processing Algorithms, Architectures, and Implementations*, vol. 1348, pp. 51-61, 1990.

19. S. Chen, S.A. Billings, C.F.N. Cowan, and P.M. Grant, "Non-linear Systems Identification using radial basis functions," *International Journal Systems Science*, vol. 21, no. 12, pp. 2513-2539, December 1990.

20. I. Cha and S.A. Kassam, "Interference cancellation using radial basis function networks," *EURASIP Signal Processing*, vol 147, no. 3, Dec. 1995, p. 247-268.

21. S. Chen, G.J. Gibson, C.F.N. Cowan, and P.M. Grant, "Reconstruction of binary signals using an adaptive radial basis function equaliser," *EURASIP Signal Processing*, vol. 22, no. 1, pp. 77-93, 1991.

22. J. Moody and C.J. Darken, "Fast learning in networks of locally tuned processing units," *Neural Computation*, vol. 1, no.2, pp. 281-294, 1989.

23. C. Chinrungrueng and C.H. Sequin, "Optimal adaptive κ-means algorithm with dynamic adjustment of learning rate," *IEEE Trans Neural Networks*, vol. 6, no. 1, pp. 157-169, 1995.

24. S. Chen, C.F.N. Cowan, and P.M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans Neural Networks*, vol. 2, pp. 302-309, 1991.

25. J.G. Proakis, *Digital Communications*, McGraw-Hill, 1984.

26. S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, N.J., 1992.

27. S. Chen, S. McLaughlin, P.M. Grant, and B. Mulgrew, "Reduced-complexity multistage blind clustering equaliser," *IEEE International Communications Conference (ICC) Proceedings*, pp. 1149-1153, May 1993.

28. R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley, New York. 1973.

29. S. Chen, B. Mulgrew, and S McLaughlin, "Adaptive Bayesian equaliser with decision feedback," *IEEE Trans Signal Processing*, vol. 41, no. 9, pp. 2918-2927, September 1993.

30. U. Mitra and H. V. Poor, "Neural network techniques for adaptive multiuser demodulation," *IEEE Journal on Selected Areas in Communications*, vol. 12, no.9, pp. 1460-1470, Dec. 1994.

31. J. Cid-Sueira, A. Artes-Rodriguez, and A.R. Figueiras-Vidal, "Recurrent radial basis function networks for optimal symbol-by-symbol equalisation," *EURASIP Signal Processing*, vol. 40, pp. 53-63, 1994.

32. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McOraw-Hill, 1984. 2nd ed.

33. G. Kechriotis, E. Zervas, and E.S. Manolakis, "Using reccurrent neural networks for adaptive communications channel equalization," *IEEE Trans Neural Networks*, vol. 5, no. 2, pp. 267-278, Mar 1994.

34. A.P. Clark, L.H. Lee, and R.S. Marshall, "Developments of the conventional nonlinear equaliser," *IEE Proccedings* Part F, vol. 129, no.2, pp. 85-94, 1982.

35. S. Chen, S. McLaughlin, B. Mulgrew, and P.M. Grant, "Adaptive Bayesian decision feedback equaliser for dispersive mobile radio channels," *IEEE Transactions on Communications*, vol. 43, no. 5, pp. 1937-1946, May 1995.

36. S. Chen, S. McLaughlin, P.M. Grant, and B. Mulgrew, "Joint channel estimation and data detection using a blind decision feedback equaliser," *Proceedings IEEE Globecom'93*, pp. 2017-2021, November 1993.

37. S. Theodoridis, C.F.N. Cowan, C.P. Callender, and C.M.S. See, "Schemes for equalisation of communications channels with nonlinear impairments," *IEE Proccedings Communications*, vol. 142, no.3, pp. 165-171, June 1995.

38. S. Chen and B. Mulgrew,, "Overcoming co-channel interference using an adaptive radial basis function equaliser," *EURASIP Signal Processing Journal*, vol. 28, no. 1, pp. 91-107, 1992.

39. S. Chen, S. McLaughlin, B. Mulgrew, and P.M. Grant, "Adaptive Bayesian decision feedback equaliser incorporating co-channel interference suppression," *Proccedings IEEE International Conference on Communications*, vol. 1, pp. 530-533, May 1994.

40. A.J. Vlterbi, "The orthogonal-random wave form dichotomy for digital personal communications," *IEEE Personal Comrnunications Magazine*, pp. 18-24, first quarter 1994.

41. A. Duel-Hallen, J. Holtzman, and Z. Zvonar, "Multiuser detection for CDMA systems," *IEEE Personal Communications*, pp. 46-58, April 1995.

42. S. Verdu, "Adaptive multiuser detection." *Proceedings ISSSTA* 94, pp. 43-50, Oulu, Finland, July 1994.

43. B. Aazhang, B.-P. Paris, and G. C. Orsak, "Neural network for multiuser detection in code division multiple access communications," *IEEE Trans Communications*, vol. 40, no. 7, pp. 1212-1222, 1992.

44. U. Mitra and H.V. Poor, "Adaptive receiver algorithms for near-far resistant CDMA," *IEEE Transactions Commanications*, vol. 43, no. 2/3/4, Part III, pp. 1713-1724, Feb/Mar/Apr 1995.

45. R.O. Schmidt, "A signal subspace approach to multiple emitter location and spectral estimation," Ph.D. Dissertation, Stanford University, Stanford, 1981.

46. E.S. Warner, B. Mulgrew, and P.M. Grant, "Adaptive receivers for synchronous and asynchronous CDMA communications systems," *Proc COST 229*, Vigo, Spain, October 1994.

47. B. Mulgrew, E.S. Warner, and P.M. Grant, "A Bayesian receiver for asynchronous code division multiple access communications," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications* (PIMRC 95), vol. 3, pp. 985-989, Toronto, Sept 1995.

48. I.C. Cha and S.A. Kassam, "Blind equalisation using radial basis function networks," *Proc. Canadian Conference Elec Eng. and Comp Sci.*, vol. 1, pp. TM.3.13.14, Sept 1992.

49. S. Chen, S. McLaughlin, P.M. Grant, and B. Mulgrew, "Fast blind equalisation based on a Bayesian decision feedback equaliser," *Electronics Letters*, vol. 29, no. 10, pp. 891-3, May 1993.

50. G. Lee, S.B. Gelfand, and M.P. Fitz, "Bayesian decision feedback techniques for deconvolution," *IEEE Journal on Selected Areas in Communications*, vol. 13. no. 1, p. 155, Jan 1995.

51. I. Cid-Sueiro and A. Figueiras-Vidal, "Recurrent radial basis function networks for optimal blind equalisation," *Proceedings IEEE Workshop on Neural Networks for Signal Processing*, pp. 562-571, Baltimore, MD, June 1993.

52. R. Arnott, "An adaptive radial basis function diversity combiner," *IEE Electronics Letters*, vol. 29, no. 12, pp. 1092-1094, 10th June 1993.

53. E.S. Chng, S. Chen, B. Mulgrew, and G.J. Gibson, "Realising the Bayesian decision boundary for channel equalisation using radial basis function network and linear equaliser," *Mathematics for Neural Nerworks and Applications*, Oxford, July 1995.

54. C.Z.W. Hassell Sweatman, G.J. Gibson, and B. Mulgrew, "Constructive neural network design for the solution of two-state classifications: problems and application to channel equalisatioo," *IEEE International Workshop on Neural Networks for Signal Processing*, Cambridge, Mass, September 1995.