# A Clustering Technique for Digital Communications Channel Equalization Using Radial Basis Function Networks

Sheng Chen, *Member, IEEE,* Bernard Mulgrew, *Member, IEEE,* and Peter M. Grant, *Senior Member, IEEE*

*Abstract*—**The paper investigates the application of a radial basis function network to digital communications channel equalization. It is shown that the radial basis function network has an identical structure to the optimal Bayesian symbol-decision equalizer solution and, therefore, can be employed to implement the Bayesian equalizer. The training of a radial basis function network to realize the Bayesian equalization solution can be achieved efficiently using a simple and robust supervised clustering algorithm. During data transmission a decision-directed version of the clustering algorithm enables the radial basis function network to track a slowly time-varying environment. Moreover, the clustering scheme provides an automatic compensation for nonlinear channel and equipment distortion. This represents a radically new approach to the adaptive equalizer design. Computer simulations are included to illustrate the analytical results.**

## I. Introduction

HIGH speed communications channels are often impaired by channel intersymbol interference and additive noise. Adaptive equalizers are required in these communications systems to obtain reliable data transmission. A discrete time model of a digital communications system is depicted in Fig. 1, where a digital sequence $s(t)$ is transmitted through a dispersive channel with transfer function

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

The transmitted symbol sequence $s(t)$ is assumed to be an equiprobable and independent binary sequence taking values from $\{\pm 1\}$. The channel output is corrupted by an additive white Gaussian noise $e(t)$. The task of the equalizer is to recover the transmitted symbols based on the channel observation $y(t)$.

From estimation theory, it is known that the best performance is obtained by detecting the entire transmitted sequence using the maximum likelihood sequence estimator (MLSE) [1], [2]. Adaptive MLSE is implemented in the form of a channel estimator and a Viterbi algorithm. High complexity and deferring decisions associated with the MLSE are however often unacceptable in many practical communications systems. Most of the practical equalizers therefore employ an architecture of making decisions symbol by symbol. Symbol-decision equalizers can further be classified into two categories according to whether they estimate a channel model explicitly. In a typical direct-modeling equalizer, a channel model is identified explicitly. The transmitted symbols are treated as states and a Kalman filter is used to estimate these states [3]. The indirect-modeling approach recovers the transmitted symbols by filtering the channel observations, usually using an adaptive linear filter [4], without estimating a channel model explicitly. This is by far the widest used equalizer structure and it is considered in the present study.

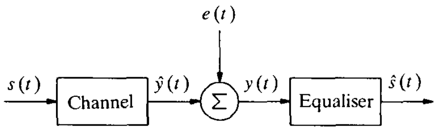


Fig. 1 Discrete-time model of data transmission system.

The structure of the symbol-decision and indirect-modeling equalizer is shown in Fig. 2. The operation of the equalizer at each sample $t$ is based on $m$ most recent channel observations and a decision is made regarding the transmitted symbol at sample $t-\tau$, where the integers $m$ and $\tau$ are known as the equalizer order and delay, respectively. How the $m$ channel observations are processed determines the performance and the complexity of the equalizer. The indirect modeling approach is sometimes referred to as the inverse modeling because, traditionally, the equalization problem is viewed as an inverse filtering in which the equalizer forms an approximation to the inverse of the distorting channel [4]. From this view point, the filter within the architecture of Fig. 2 becomes linear and, the resulting equalizer is called a linear transversal equalizer (LTE). This view point however has certain shortcomings. Firstly, it completely ignores the fact that $s(t)$ is binary. It ought to exploit this information to the benefit of performance, as is the case in the MLSE. The inverse modeling also implies that increasing the equalizer order $m$ should lead to a more accurate approximation and hopefully better equalization performance. This is however not true due to the noise enhancement. Previous research [5], [6] has demonstrated that the LTE does not achieve the full performance potential of the given symbol-decision structure in Fig. 2. Better performance can be obtained if some more complex filtering method is employed.

Manuscript received April 29, 1991; revised May 5, 1992. This work was supported by the U.K. Science and Engineering Research Council under Award GR/E10357.

The authors are with the Department of Electrical Engineering, University of Edinburgh, King's Buildings, Edinburgh EH9 3JL, Scotland.

IEEE Log Number 9202044.

1045–9227/93$03.00 © 1993 IEEE

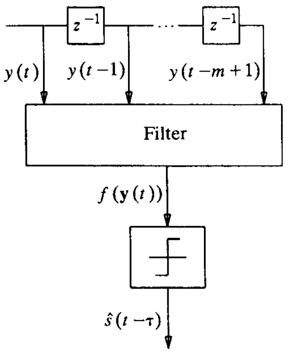


Fig. 2 Architecture of symbol-decision equalizer without estimating channel model.

The use of the multilayer perception [5] and the polynomial filter [6] as equalizers can achieve significant performance improvement over the LTE. This is because these two nonlinear equalizers are able to approximate the optimal symbol-decision equalizer solution implicitly. The multilayer perceptron however, has problems of slow convergence and unpredictable solutions during the training while the polynomial equalizer suffers a drawback of exponentially increasing filter dimension. Furthermore, it is not known how to specify parameters of these two nonlinear equalizers even when the channel transfer function, the symbol and the noise statistics are given. Their parameters can only be set by experiment in an ad hoc manner. This is in contrast to the LTE whose parameters are completely specified by the Wiener filter solution when the channel statistics are provided.

The present study proposes a novel strategy for an adaptive equalizer design based on the radial basis function (RBF) network [7]–[9]. The optimal solution for the symbol-decision architecture of Fig. 2 is first derived using the Bayes decision theory [10], [11]. It is then shown that this optimal Bayesian equalizer solution has an identical structure to the RBF network. For a known channel, therefore, all the parameters of the RBF network are specified explicitly, giving rise to precisely the Bayesian solution. Moreover, the training of an RBF network to realize the optimal equalizer can be carried out very efficiently by exploiting the underlying data structure. This involves a supervised clustering to position the RBF centers at the desired channel output states. Because noisy channel observations form Gaussian clusters and the means of these data clusters are the desired channel states, rapid convergence is guaranteed. The supervised clustering algorithm is very simple and robust, and it represents a radical departure from traditional training methods which are mostly based on minimizing the mean square error between the desired filter output and the actual filter output. Often communications channels are time-varying and, during data transmission, the desired channel output states will be changing. The RBF network can track these changes using a decision-directed clustering algorithm. Similar to the multilayer perceptron and the polynomial equalizers, the adaptive RBF equalizer is capable of compensating nonlinear distortion. Compared with the two previous nonlinear equalizers, however, the adaptive RBF equalizer has significant performance and implementation advantages because its structure is explicitly equivalent to the underlying optimal Bayesian solution. Computer simulation results are used to demonstrate the optimal performance of RBF equalizers.

## II. Optimal Symbol-Decision Equalizer

The general symbol-decision equalizer depicted in Fig. 2 is characterized by the equalizer order $m$ and delay $\tau$. For the general channel of $n_h + 1$ taps given in (1), there are $n_s = 2^{n_h+m}$ combinations of the channel input sequence

$$\boldsymbol{s}(t) = [s(t)\cdots s(t-m+1-n_h)]^T. \tag{2}$$

This gives rise to $n_s$ points or values of the noise-free channel output vector

$$\hat{\boldsymbol{y}}(t) = [\hat{y}(t)\cdots\hat{y}(t-m+1)]^T. \tag{3}$$

These points will be referred to as the desired channel states, and they can be partitioned into two classes according to the value of $s(t-\tau)$

$$\left.\begin{aligned} Y^+_{m,\tau} &= \{\hat{\boldsymbol{y}}(t)|s(t-\tau)=1\},\\ Y^-_{m,\tau} &= \{\hat{\boldsymbol{y}}(t)|s(t-\tau)=-1\}.\end{aligned}\right\} \tag{4}$$

The two sets $Y^+_{m,\tau}$ and $Y^-_{m,\tau}$ contain the information of the channel transfer function, the symbol statistics and the equalizer constraints. Each desired state $\boldsymbol{y}_i^+ \in Y^+_{m,\tau}$ or $\boldsymbol{y}_i^+ \in Y^-_{m,\tau}$ has *a priori* probability of appearance $p_i$. Under the previous assumptions on symbol statistics, all the desired states have a same probability of appearance $p = 1/n_s$. The numbers of the states in $Y^+_{m,\tau}$ and $Y^-_{m,\tau}$ are denoted as $n_s^+$ and $n_s^-$, respectively.

Because of the additive white Gaussian noise, the noisy observation vector

$$\boldsymbol{y}(t) = [y(t)\cdots y(t-m+1)]^T \tag{5}$$

is a random process having conditional Gaussian density functions centered at each of the desired channel states. It is apparent that channel observations form clusters and the means of these data clusters are the desired states. Determining the value of the transmitted symbol $s(t-\tau)$ based on the observation vector (5) is a decision problem. The Bayes decision theory [10], [11] provides the optimal solution to the general decision problem and, therefore, can be employed to derive the optimal solution for the general equalizer of Fig. 2. It is straightforward to verify that this optimal Bayesian equalizer solution is defined as [6]

$$\hat{s}(t-\tau) = \operatorname{sgn}(f_B(\boldsymbol{y}(t))) = \begin{cases} 1, & f_B(\boldsymbol{y}(t)) \geq 0\\ -1, & f_B(\boldsymbol{y}(t)) < 0\end{cases} \tag{6}$$

with the optimal Bayesian filter or decision function given by

$$\begin{aligned} f_B(\boldsymbol{y}(t)) = &\sum_{i=1}^{n_s^+} p_i\left(2\pi\sigma_e^2\right)^{-m/2}\exp\left(-\left\|\boldsymbol{y}(t)-\boldsymbol{y}_i^+\right\|^2/2\sigma_e^2\right)\\ &-\sum_{j=1}^{n_s^-} p_j\left(2\pi\sigma_e^2\right)^{-m/2}\exp\left(-\left\|\boldsymbol{y}(t)-\boldsymbol{y}_j^-\right\|^2/2\sigma_e^2\right)\end{aligned} \tag{7}$$

where the first sum is over $\boldsymbol{y}_i^+ \in Y_{m,\tau}^+$, the second sum is over $\boldsymbol{y}_j^- \in Y_{m,\tau}^-$, and $\sigma_e^2$ is the noise variance.

The optimal equalizer solution clearly depends on the noise distribution as well as the desired channel states. Multiplying $f_B(\boldsymbol{y}(t))$ by a positive constant does not change the optimally. What is critical is the optimal decision boundary defined by

$$\{\boldsymbol{y} | f_B(\boldsymbol{y}) = 0\}. \tag{8}$$

It partitions the observation space into two decision regions corresponding to the two decisions $\hat{s}(t-\tau) = \pm 1$. Because the Bayesian decision function (7) is nonlinear, the optimal boundary (8) is a hypersurface in the observation space. The decision boundary of any linear equalizer is a hyperplane in the observation space. Therefore a performance gap always exists between the LTE and the optimal equalizer. For equiprobable symbols, the coefficients in the optimal filter (7) become redundant and can be removed, giving rise to the following simpler form of the optimal decision function.

$$\begin{aligned} f_B(\boldsymbol{y}(t)) = & \sum_{i=1}^{n_s^+} \exp\left(-\|\boldsymbol{y}(t) - \boldsymbol{y}_i^+\|^2/2\sigma_e^2\right) \\ & - \sum_{j=1}^{n_s^-} \exp\left(-\|\boldsymbol{y}(t) - \boldsymbol{y}_j^-\|^2/2\sigma_e^2\right). \end{aligned} \tag{9}$$

An example is given to illustrate these results.

For the purpose of graphical display, the equalizer order is chosen as $m = 2$. Let the channel transfer function be

$$H(z) = 0.5 + 1.0z^{-1}. \tag{10}$$

All the combinations of $\boldsymbol{s}(t)$ and the desired channel states are listed in Table I. The states of $Y_{2,1}^+$ and $Y_{2,1}^-$ are also plotted in Fig. 3 using the "square" □ and "cross" ×, respectively. When $\hat{\boldsymbol{y}}(t)$ is at a particular state, the observation vector $\boldsymbol{y}(t)$ is a stochastic process having a Gaussian density function with a mean equal to the given state and a variance equal to that of the noise. For noise variance $\sigma_e^2 = 0.125$, 1000 samples of $\boldsymbol{y}(t)$ are plotted in Fig. 3 using dots. It is seen that observations form clusters around the desired channel states. The optimal decision boundary computed using (9) is a curve, which is a two-dimensional hypersurface. When a $\boldsymbol{y}(t)$ is observed at the right-hand region of he boundary, the decision $\hat{s}(t-1) = 1$ is made. If the observation vector $\boldsymbol{y}(t)$ appears in the left-hand region of the boundary, $\hat{s}(t-1) = -1$ is made. This way of making decisions is optimal because it produces the minimum average error probability or bit error rate.

TABLE I
INPUT AND DESIRED CHANNEL STATES. $H(z) = 0.5 + 1.0z^{-1}$, $m = 2$ AND $\tau = 1$

| No. | $s(t)$ | $s(t-1)$ | $s(t-2)$ | $\hat{y}(t)$ | $\hat{y}(t-1)$ |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1.5 | 1.5 |
| 2 | 1 | 1 | −1 | 1.5 | −0.5 |
| 3 | −1 | 1 | 1 | 0.5 | 1.5 |
| 4 | −1 | 1 | −1 | 0.5 | −0.5 |
| 5 | 1 | −1 | 1 | −0.5 | 0.5 |
| 6 | 1 | −1 | −1 | −0.5 | −1.5 |
| 7 | −1 | −1 | 1 | −1.5 | 0.5 |
| 8 | −1 | −1 | −1 | −1.5 | −1.5 |

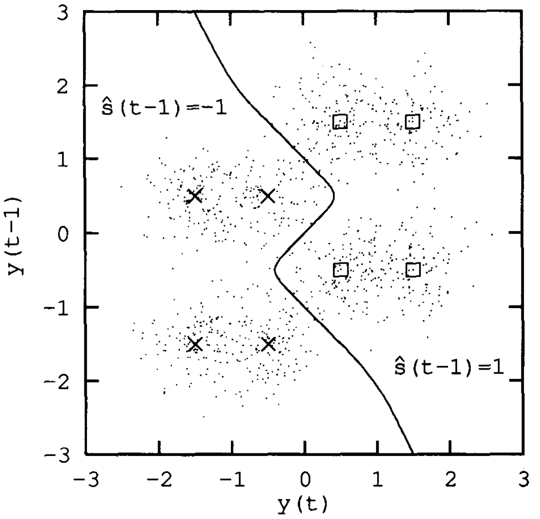


Fig. 3. Desired states, data clusters, and optimal decision boundary, $H(z) = 0.5 + 1.0z^{-1}$, SNR = 10 dB and 1000 samples of $\boldsymbol{y}(t)$.

## III. THE RADIAL BASIS FUNCTION NETWORK

Consider the RBF network [7]–[9], which is a two-layered processing structure depicted in Fig. 4. The hidden layer consists of an array of computing units. Each unit contained a parameter vector called a center, and the unit calculates a squared distance between the center and the network input vector. The squared distance is then divided by a parameter called a width and the result is passed through a nonlinear function. The second layer is essentially a linear combiner with a set of connection weights. The overall response of the RBF network is a mapping $f_r$

$$f_r(\boldsymbol{y}) = \sum_{i=1}^{n} w_i \phi\left(\|\boldsymbol{y} - \boldsymbol{c}_i\|^2/\rho_i\right) \tag{11}$$

where $n$ is the number of computing units, $\boldsymbol{c}_i$ are the RBF centers, $\rho_i$ are the widths of the units and $w_i$ are the weights.

Comparing the network response (11) with the optimal equalizer filter (7), it is obvious that they have the same structure. The RBF network is therefore an ideal processing means to implement the optimal Bayesian equalizer. Given channel statistics, it is known exactly how to specify all the parameters of the RBF network. The number of the hidden units $n$ is equal to the number of the desired channel states and the RBF centers are placed at these desired states. The nonlinear function $\phi$ is obviously chosen as the following exponential function

$$\phi(y) = \exp(-y) \tag{12}$$

and all the widths have a same value $\rho$, which is twice as large as the noise variance. Each hidden unit then implements a component conditional density function in (7) and, when the

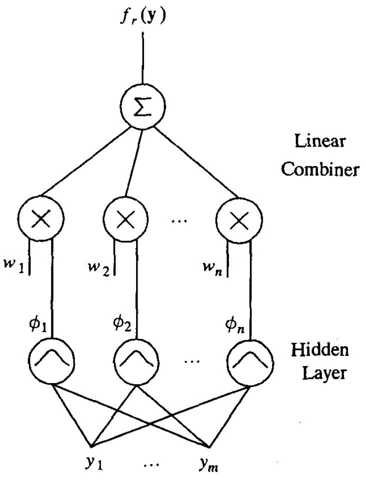


Fig. 4. Schematic of radial basis function network.

weights are further set to the corresponding coefficients in (7), the RBF network realizes precisely the Bayesian equalizer. For the case of equiprobable symbols, the network can be simplified considerably by fixing half of the weights to 1 and the other half to –1. In this case the second layer becomes a summer.

Whether a RBF network can realize optimal equalizer solution depends crucially on whether the centers can be positioned correctly at the desired channel states. The width $\rho$ is a less influential parameter. It is not too difficult to see that $\rho$ need not be accurately set to $2\sigma_e^2$. As far as the decision boundary is concerned, the influence of $\rho$ on the hidden units cancel out each other to a certain extent. This is important because in practice only an estimate of the noise variance is available. Ideally $n$ should be equal to $n_s$, and this requires a correct estimation of the channel order $n_h$. If the estimate $\hat{n}_h$ is larger than the true value $n_h$, more centers than really required will be employed. Apart from introducing unnecessary computations, this will not affect the performance of the RBF network. For example, if $\hat{n}_h = n_h + 1$, the number of centers is twice as many as the desired channel states, and a pair of two centers will converge to each desired state. This still results in the optimal equalization solution. If however $\hat{n}_h$ is smaller than $n_h$, less centers than the desired states will be used and some performance loss can be expected. These discussions will be further illustrated later using simulation.

## IV. Supervised Learning

In reality the channel transfer function is unavailable. An efficient learning strategy is necessary in order for a RBF network to learn the optimal equalizer solution. In the case of equiprobable symbols, the weights of the network can be fixed and learning involves finding the desired channel states so that the centers of the network can be positioned at these states. Denote the $n_s$ combinations of $\boldsymbol{s}(t)$ as $\boldsymbol{s}_i$; $1 \le i \le n_s$. During the training period, transmitted symbols are known to the equalizer. At each sample $t$, it can be inferred from $\boldsymbol{s}(t)$ which member of the desired states occurs. Furthermore, as mentioned previously, noisy observations form Gaussian clusters centered at the desired states. This suggests that a supervised $\kappa$-means clustering procedure can effectively filter out the noise so that the RBF centers converge to the desired states. The computational procedure of this clustering algorithm is summarized as follows:

TABLE II
Complexity of Supervised Clustering (13) for $m$-Inputs and $n$-Units RBF Network

| |
|---|
| $m$ multiplications |
| $m$ divisions |
| $m + 1$ additions |

TABLE III
Complexity of Supervised LMS (15) for $m$-Inputs and $n$-Units RBF Network

| |
|---|
| $n \times m + 2n + 1$ multiplications |
| $n$ divisions |
| $2n \times m + n$ additions |
| $n$ evaluations of $\exp(-y)$ |

```
if ($\boldsymbol{s}(t) == \boldsymbol{s}_i$){
    $\boldsymbol{c}_i(t) = \text{counter}_i * \boldsymbol{c}_i(t-1) + \boldsymbol{y}(t)$;
    $\text{counter}_i = \text{counter}_i + 1$;
    $\boldsymbol{c}_i(t) = \boldsymbol{c}_i(t)/\text{counter}_i$;
}                                                        (13)
```

Because of the underlying data structure, a rapid convergence of this supervised clustering procedure is guaranteed. The computational requirement of this supervised clustering algorithm is given in Table II. For nonstationary channels, the following adaptive version of (13) is preferred:

```
if ($\boldsymbol{s}(t) == \boldsymbol{s}_i$){
    $\boldsymbol{c}_i(t) = \boldsymbol{c}_i(t-1) + g_c * (\boldsymbol{y}(t) - \boldsymbol{c}_i(t-1))$;
}                                                        (14)
```

where $g_c$ is the learning rate for centers. This version of the supervised clustering algorithm is simpler than (13). It should be emphasized that the algorithm (13) or (14) is all that is required to train a RBF network in the case of equiprobable symbols.

If the assumption of equiprobable symbols is violated, it is advisable to adjust the weights of the network so that the network can learn the general equalizer solution (7). The adaptation of the weights is achieved using the following supervised least mean square (LMS) algorithm:

$$\left.\begin{array}{ll} \phi_i(t) = \exp\left(-\|\boldsymbol{y}(t) - \boldsymbol{c}_i(t)\|^2/\rho\right), & 1 \le i \le n, \\ \epsilon(t) = s(t-\tau) - \sum_{i=1}^{n} w_i(t-1)\phi_i(t), & \\ w_i(t) = w_i(t-1) + g_w\epsilon(t)\phi_i(t), & 1 \le i \le n, \end{array}\right\} \tag{15}$$

where $g_w$ is the learning rate for weights. The computational complexity of this LMS algorithm is listed in Table III. The properties of the LMS algorithm are well understood [12].

Knowledge of the noise variance $\sigma_e^2$ and the channel order $n_h$ are required to specify the network structure and simple techniques can be employed to provide their estimates. If the centers are positioned correctly at the desired channel states and the $i$th desired state appears at sample $t$, it is easy to verify that the noise variance is given by

$$\sigma_e^2 = \boldsymbol{E}[\|\boldsymbol{y}(t) - \boldsymbol{c}_i\|^2/m] \tag{16}$$

where $\boldsymbol{E}[.]$ is the expectation operator. This suggests a simple estimator for the noise variance. If $\boldsymbol{s}(t) = \boldsymbol{s}_i$ at $t$, the noise variance estimate is adjusted according to

$$\hat{\sigma}_e^2(t) = ((t-1)\hat{\sigma}_e^2(t-1) + \|\boldsymbol{y}(t) - \boldsymbol{c}_i(t)\|^2/m)/t. \tag{17}$$

Under the assumptions introduced in Section I, the autocorrelations of $y(t)$ satisfy

$$\gamma_q = \boldsymbol{E}[y(t)y(t-q)] = \begin{cases} \sum_{j=0}^{n_h} h_j^2 + \sigma_e^2, & q = 0 \\ \sum_{j=q}^{n_h} h_j h_{j-q}, & 0 < q \leq n_h \\ 0, & q > n_h. \end{cases} \tag{18}$$

Note that $\gamma_{n_h} \neq 0$ if $h_0 \neq 0$. The channel order $n_h$ can therefore be inferred from the autocorrelations of $y(t)$. This involves the calculation of the normalized sample autocorrelations of $y(t)$

$$\hat{\gamma}_q = \sum_{t=1}^{N-q} (y(t) - \overline{y})(y(t+q) - \overline{y}) \Big/ \left(\sum_{t=1}^{N} (y(t) - \overline{y})^2\right) \qquad q \geq 0 \tag{19}$$

where $N$ is the number of samples, and $\overline{y}$ is the sample mean of $y(t)$

$$\overline{y} = N^{-1} \sum_{t=1}^{N} y(t). \tag{20}$$

A $\hat{\gamma}_q$ is regarded as significant if it is outside the 95% confidence bands $\pm 1.96 N^{-1/2}$. The last significant sample autocorrelation provides an estimate $\hat{n}_h$ for the channel order.

## V. Decision-Directed Learning

For time-varying channels, the channel states also become time-varying. A RBF network must have tracking ability to cope with this situation. During data transmission, supervised learning no longer applies and adaptation has to rely on unsupervised or decision-directed learning. The unsupervised $\kappa$-means clustering procedure is often employed as a part of the general learning algorithm to adjust RBF centers [13], [14]. This involves computing the squared distance between the centers and the network input vector, selecting a minimum squared distance and moving the corresponding center closer to the input vector. This unsupervised clustering scheme can also be used in the current application for the RBF network to track time-varying channel states. The computational procedure of this unsupervised clustering is as follows:

$$\left.\begin{array}{l} d_i(t) = \|\boldsymbol{y}(t) - \boldsymbol{c}_i(t-1)\|^2, \qquad 1 \leq i \leq n \\ k = \arg[\min\{d_i(t), \qquad 1 \leq i \leq n\}] \\ \boldsymbol{c}_k(t) = \boldsymbol{c}_k(t-1) + g_c(\boldsymbol{y}(t) - \boldsymbol{c}_k(t-1)) \\ \boldsymbol{c}_i(t) = \boldsymbol{c}_i(t-1), \qquad 1 \leq i \leq n, \text{ and } i \neq k \end{array}\right\} \tag{21}$$

If it is also necessary to adapt the weights during data transmission, a decision-directed LMS algorithm can be incorporated into the adaptive algorithm. The decision-directed LMS algorithm is identical to the LMS algorithm (15) except that the error signal $\epsilon(t)$ is derived as the difference between the estimated symbol $\hat{s}(t-\tau)$ and the network response.

For the equalization application, the unsupervised clustering algorithm (21) is unnecessarily complex and a decision-directed version of the supervised clustering algorithm (14) provides a simpler alternative. The basic idea is to infer the state membership using estimated symbols. Because of the decision delay $\tau$, at sample $t$, the algorithm actually determines the state membership at $t - \tau$ and the computational procedure of this decision-directed clustering algorithm is as follows:

```
if ($\hat{\boldsymbol{s}}(t-\tau) == \boldsymbol{s}_i$){
    $\boldsymbol{c}_i(t) = \boldsymbol{c}_i(t-1) + g_c * (\boldsymbol{y}(t-\tau) - \boldsymbol{c}_i(t-1))$;
}                                                        (22)
```

where

$$\hat{\boldsymbol{s}}(t-\tau) = [\hat{s}(t-\tau) \cdots \hat{s}(t-\tau-m+1-n_h)]^T. \tag{23}$$

After the initial training and in normal operation, the equalizer decisions are correct with high probability. This ensures that inferring state memberships from estimated symbols are correct often enough to allow a rapid convergence of the centers to the channel states. This algorithm is all required to track variations in the channel characteristics during data transmission in the case of equiprobable symbols.

## VI. A Comparison with Other Nonlinear Equalizers

The approach reported in this study represents a radically new thinking to adaptive equalizer design. Traditional adaptive algorithms for equalizers are based on the criterion of minimizing the mean square error between the desired filter output and the actual filter output, that is, these learning algorithms adjust the filter parameters to achieve a minimum of the criterion

$$\boldsymbol{E}[(s(t-\tau) - f(\boldsymbol{y}(t)))^2]. \tag{24}$$

The ultimate performance criterion for an equalizer is however bit error rate and there exists no clear relationship between the mean square error criterion (24) and the bit error rate criterion. The optimal Bayesian equalizer (7) or (9) will not necessarily produce a good mean square error performance and yet it provides the minimum average bit error rate achievable under the general structure of Fig. 2. The two nonlinear equalizers proposed in [5] and [6] rely on the mean square error criterion

(24) for training. Because the multilayer perception is highly nonlinear in the parameters, its error surface (24) is very complicated. Therefore, training times for multilayer perceptron equalizers are typically very long and the adaptive algorithm may become trapped at bad local minima. An alternative nonlinear equalizer approximates the Bayesian solution (7) using a polynomial filter [6]. The polynomial equalizer suffers a drawback of exponentially increasing filter dimension. A consequence of this is that the autocorrelation matrix of the input vector to the adaptive algorithm often becomes very ill-conditioned and has a large eigenvalue spreading ratio. Therefore, it is necessary to pass the output of the polynomial filter through a sigmoid function in order to alleviate this numerical difficulty. The training of the RBF network as an equalizer is based on the method that exploits the underlying data generating mechanism. The channel outputs form clusters in the observation space. The clustering algorithm positions the network centers at the means of data clusters efficiently. This approach is obviously guaranteed to converge rapidly and it is directly linked to the bit error rate performance since the trained RBF network will realize explicitly the Bayesian solution.

It is interesting to compare the two different strategies employed by the RBF network and those equalizers requiring a channel estimator. The MLSE identifies the channel model directly while the RBF equalizer identifies the end-results of the channel. This has an important implication for channels involving nonlinear distortion. When nonlinear distortion is taken into account, the general channel model can be defined as

$$y(t) = f_h(s(t), \cdots, s(t-n_h); \Theta) + e(t) \tag{25}$$

where $f_h$ is some nonlinear function and $\Theta$ is a channel parameter vector. The identification of this nonlinear channel model requires to specify the nonlinear structure as well as to estimate the channel parameters, which is a very difficult task particularly in the context of real-time adaptation. The present approach avoids all these difficulties and it can straightforwardly be applied to the equalization involving nonlinear channels. Because the clustering algorithm always converges to the set of the desired channel states, the RBF network can achieve the full Bayesian performance regardless whether the channel is linear or nonlinear. Difficulties in online identification of a nonlinear channel model, on the other hand, should not be underestimated. Even when the nonlinear form $f_h$ is given, it is not always possible to identify all the parameters in the model unless the system input signal is persistently exciting [15]. For a linear channel model, persistent excitation means that $s(t)$ should contain sufficient frequency components. Since $s(t)$ is generally white, it is an ideal input signal for identifying the linear channel model (1). For a nonlinear channel model, however, persistent excitation requires an additional condition that $s(t)$ should cover a sufficient range of amplitudes. The binary nature of $s(t)$ therefore represents a worst scenario and, as a consequence, parameters in some nonlinear channel models may not be identifiable. This can be demonstrated by considering the following nonlinear channel model

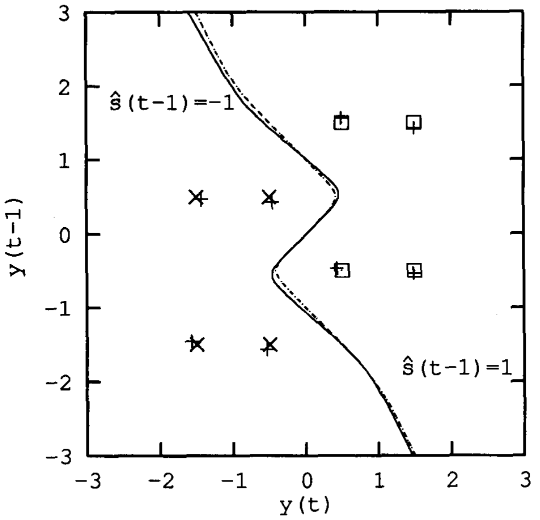


Fig. 5. Comparison of decision boundaries. Dotted: optimal, solid: RBF network, +: centers, $H(z) = 0.5 + 1.0z^{-1}$, SNR = 10 dB, $m = 2$, $\tau = 1$, $\rho = 2\sigma_e^2$, and $\hat{n}_h = 1$.

$$\begin{aligned} y(t) = &\sum_{i_1=0}^{2} h_{i_1} s(t-i_1) \\ &+ \sum_{i_1=0}^{2}\sum_{i_2=i_1}^{2} h_{i_1 i_2} s(t-i_1)s(t-i_2) \\ &+ \sum_{i_1=0}^{2}\sum_{i_2=i_1}^{2}\sum_{i_3=i_2}^{2} h_{i_1 i_2 i_3} s(t-i_1)s(t-i_2)s(t-i_3) \\ &+ e(t). \end{aligned} \tag{26}$$

The input vector to the adaptive channel estimator consists of 19 monomials $s(t-i_1)$, $s(t-i_1)s(t-i_2)$ and $s(t-i_1) \cdot s(t-i_2)s(t-i_3)$. The rank of the 19×19 autocorrelation matrix of the estimator input vector, however, is only 8. It is therefore impossible to identify all the 19 parameters in (26).

## VII. Simulation Study

In all the results, $s(t)$ was an equiprobable random number taking values from $\{\pm 1\}$. Therefore the weights of the RBF network were fixed and learning only involved supervised or decision-directed clustering to position the centers at the desired channel states.

For the same system shown in Fig. 3, 160 samples of training data were used to train the RBF network using (13). Initially correct estimates of the channel order and the noise variance were assumed, giving rise to $\rho = 2\sigma_e^2$ and $\hat{n}_h = 1$. The trained RBF network produced the decision boundary shown in Fig. 5. It is seen that convergence of the RBF center to the desired channel states was achieved and the supervised clustering algorithm performed well even though the noise level was very high. For better signal to noise ratio (SNR) conditions, training samples can be reduced.

The influence of the width $\rho$ was next investigated. The width $\rho$ was set to $4\sigma_e^2$ and $\sigma_e^2$, respectively. The first case represented an estimate of the noise variance that was twice as large as the true value and, in the second case, the estimate was only half of the true noise variance. These two RBF networks were trained and they produced the decision boundaries which were practically the same as that obtained using the true noise variance shown in Fig. 5. This confirms that the performance of RBF equalizers is relatively insensitive to the width $\rho$. Obviously the width $\rho$ has no effect on the supervised clustering learning for centers. The simple variance estimator (17) was also incorporated into the training procedure to estimate the noise variance. After 160 samples of training it produced $\hat{\sigma}_e^2 = 0.105$ compared with the true value $\sigma_e^2 = 0.125$. This estimate was apparently accurate enough for the RBF network to realize the optimal decision boundary.

The channel (10) used in the previous simulation has an order $n_h = 1$. Assume that it was wrongly estimated as $\hat{n}_h = 2$. This resulted in 16 centers compared with 8 desired channel states. The positions of these 16 centers obtained after 160 samples of supervised clustering learning are plotted in Fig. 6. As expected, two centers converged to a desired state and the optimality of the RBF equalizer was maintained. Another channel

$$H(z) = 1.0 + 0.8z^{-1} + 0.5z^{-2} \tag{27}$$

was used to test the case of $\hat{n}_h < n_h$. Again a poor SNR of 10 dB was chosen and 160 samples were used in training. This was a channel of order $n_h = 2$. Suppose that an incorrect estimate $\hat{n}_h = 1$ was provided. This gave rise to a RBF network with 8 centers compared with 16 desired channel states. Also a wrong estimate of the noise variance was used by setting $\rho = 4\sigma_e^2$. The result obtained is shown in Fig. 7. It is interesting to see that each center converged to the mean of two desired states. Judging from the decision boundaries depicted in Fig. 7, only a small performance loss occurred in this case. The estimator (17) produced $\hat{\sigma}_e^2 = 0.281$ compared with the true noise variance $\hat{\sigma}_e^2 = 0.189$. This estimate is more accurate than the actual one used in the network.

A third channel with a more realistic equalizer order was used to study the performance of the RBF network under a variety of SNR's. The channel transfer function was given by

$$H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}. \tag{28}$$

The equalizer order and delay were chosen as $m = 4$ and $\tau = 1$, respectively. Correct estimates of the channel order and the noise variance were assumed. This gave rise to a network of 64 centers. 640 samples of training data were used in the supervised clustering procedure (13). Bit error rates of the optimal Bayesian equalizer and the trained RBF network are depicted in Fig. 8, where it is seen that the RBF equalizer achieved the optimal performance. The performance of the order-4 Wiener filter is also plotted in Fig. 8 as a comparison. The performance of the Wiener filter is the best that a LTE can hope to realize. For this example, order 4 is also the best choice for the Wiener filter. Higher order Wiener filters cannot improve performance and may even worsen the results due to the noise enhancement, as can be seen from Fig. 9. At the error probability of $10^{-4}$, the RBF equalizer reported a 4.4 dB improvement in SNR over the Wiener filter. Note that for SNR > 15 dB, less than 300 samples are sufficient to train the network. The noise variances were also estimated using (17) and the results are summarized in Table IV. When the width was set to $\rho = 2\hat{\sigma}_e^2$, an identical performance to the case of $\rho = 2\sigma_e^2$ was produced by the RBF network.

The ability of the RBF network to cope with nonlinear distortion was demonstrated using the following non-

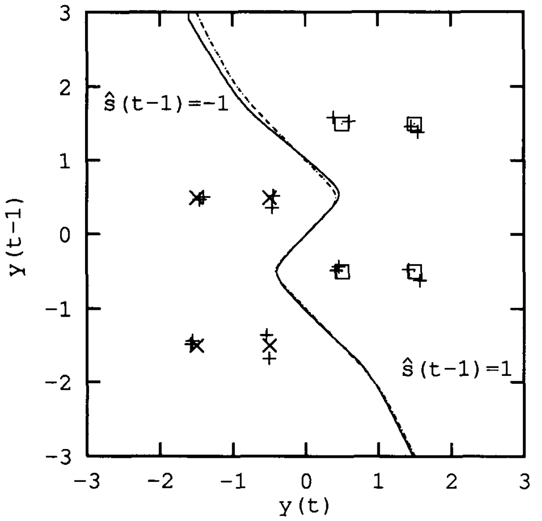


Fig. 6. Comparison of decision boundaries. Dotted: optimal, solid: RBF network, +: centers, $H(z) = 0.5 + 1.0z^{-1}$, SNR = 10 dB, $m = 2$, $\tau = 1$, $\rho = 2\sigma_e^2$, and $\hat{n}_h = 2$.

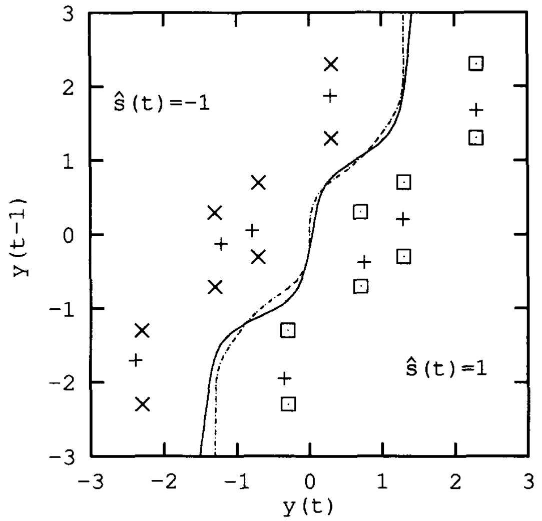


Fig. 7. Comparison of decision boundaries. Dotted: optimal, solid: RBF network, +: centers, $H(z) = 1.0 + 0.8z^{-1} + 0.5z^{-2}$, SNR = 10 dB, $m = 2$, $\tau = 0$, $\rho = 4\sigma_e^2$, and $\hat{n}_h = 1$.

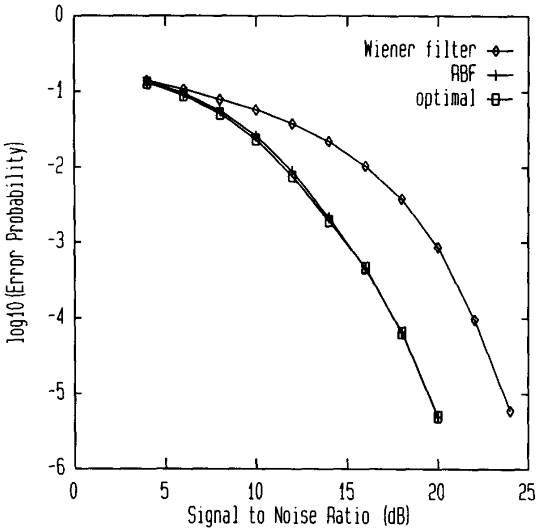


Fig. 8. Comparison of performance. $H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$, $m = 4$ and $\tau = 1$.

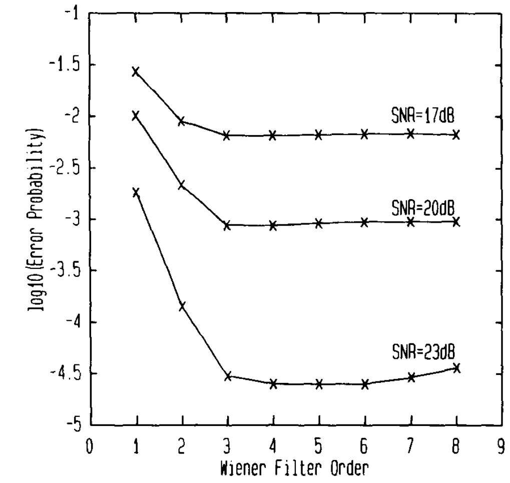


Fig. 9. Performance versus Wiener filter order. $H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$, and $\tau = 1$.

TABLE IV
TRUE NOISE VARIANCES AND ESTIMATED NOISE VARIANCES

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\sigma_e^2$ | 0.3162 | 0.1778 | 0.1000 | 0.0562 | 0.0316 | 0.0178 | 0.0100 |
| $\hat{\sigma}_e^2$ | 0.2319 | 0.1304 | 0.0733 | 0.0412 | 0.0232 | 0.0131 | 0.0073 |

linear channel

$$\left.\begin{aligned} y(t) &= x(t) + 0.2x^2(t) - 0.1x^3(t) + e(t) \\ X(z)/S(z) &= 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}. \end{aligned}\right\} \quad (29)$$

Fig. 10 shows the performance curves of the optimal Bayesian equalizer with $m = 4$ and $\tau = 1$ and the order-4 Wiener filter with $\tau = 1$. The supervised clustering algorithm (13) was used to train a RBF equalizer of 64 centers with a training sequence of 320 samples. The trained RBF equalizer produced a performance curve closely matched to that of the Bayesian equalizer.

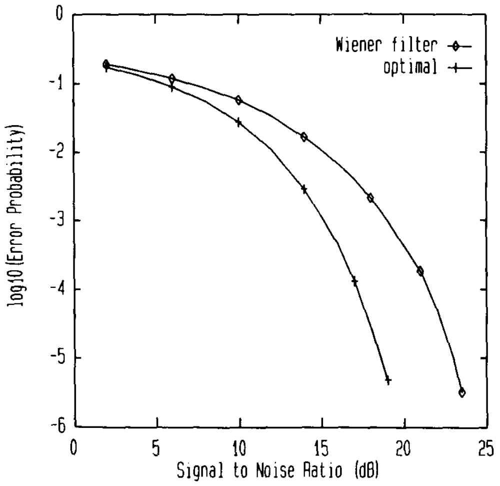


Fig. 10. Comparison of performance. Nonlinear channel (29), $m = 4$ and $\tau = 1$.

For all the above examples, a combined learning of supervised clustering and LMS was also performed to train the centers and the weights simultaneously. The performances obtained were identical to the results reported here. This is not surprising and it confirms that for equiprobable symbols the training of centers is sufficient for the network to realize the optimal equalizer solution.

A further example demonstrated the tracking performance of the RBF network using the decision-directed clustering algorithm (22). The channel was time-varying with a transfer function

$$H(z) = 1.0 + (0.2 + 0.0015t)z^{-1} \quad (30)$$

and the noise variance was chosen as $0.01(1.0 + (0.2 + 0.0015t)^2)$ to provide a constant SNR of 20 dB. Correct estimates of $\sigma_e^2$ and $n_h$ were assumed. The center trajectories obtained using the decision-directed clustering algorithm with a learning rate $g_c = 0.2$ are plotted in Fig. 11. It is clear from Fig. 11 that good tracking was achieved.

Finally, autocorrelations of channel observations were computed using (19) for several channels and the results are given in Fig. 12. For all these four examples, channel orders were correctly revealed from their normalized sample autocorrelations.

## VIII. CONCLUDING REMARKS

A practical application of the RBF network to the equalization of digital communications channels has been reported. A main contribution of this study is the derivation of the structural equivalence between the optimal Bayesian solution of the symbol-decision equalizer and the RBF network. This explains clearly why neural network equalizers outperform the traditional linear equalizer and it highlights the advantages of the RBF equalizer over other nonlinear equalizers. It has been demonstrated that the training of the RBF network to realize the optimal equalizer solution can be achieved efficiently using

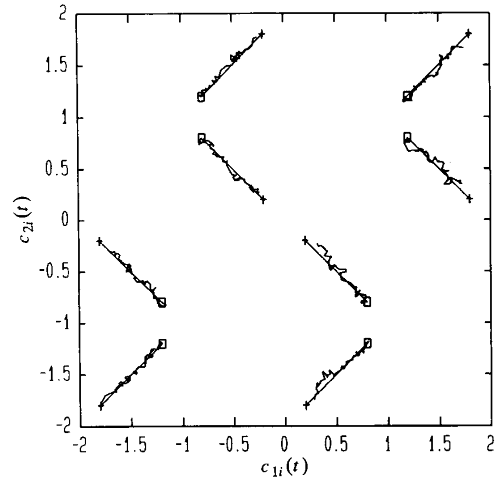
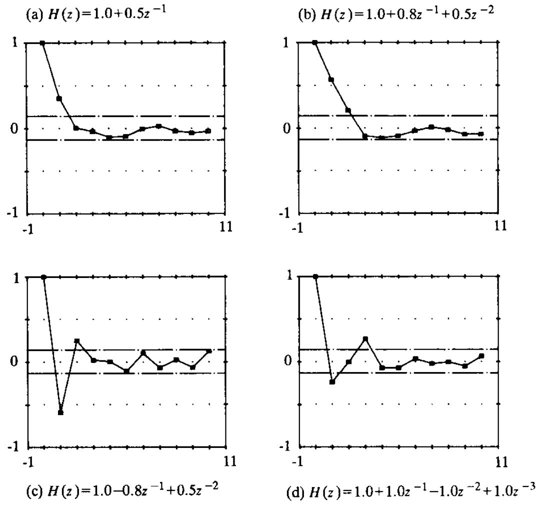


Fig. 11. Tracking performance of decision-directed clustering. $H(z) = 1.0 + (0.2 + 0.0015t)z^{-1}$, SNR = 20 dB, □—× desired state trajectories, $m = 2$ and $\tau = 0$.



(a) $H(z)$=1.0+0.5$z^{-1}$

(b) $H(z)$=1.0+0.8$z^{-1}$+0.5$z^{-2}$

(c) $H(z)$=1.0−0.8$z^{-1}$+0.5$z^{-2}$

(d) $H(z)$=1.0+1.0$z^{-1}$−1.0$z^{-2}$+1.0$z^{-3}$

Fig. 12. Autocorrelations of channel observations. — · — 95% confidence band, SNR = 10 dB and 200 samples of observations.

the supervised clustering algorithm. For nonstationary channels, a decision-directed version of the clustering algorithm can be employed to provide tracking ability for the RBF network during data transmission.

A useful technique to improve equalization performance is to use the decision feedback [4]. After the submission of this paper, some new results have been obtained by incorporating the decision feedback into the RBF network to improve performance and to minimize processing complexity [16]. For some communications systems, digital symbols, and channels are represented in complex-valued forms. Research has been continuing into complex-valued neural network structures involving complex signals (e.g., [17], [18]). The results presented in this study can readily be extended to the general case of complex-valued symbols and channels by employing the complex-valued RBF network of [18].

## Acknowledgment

The authors acknowledge stimulating discussions with Prof. C. Cowan and Dr. S. McLaughlin on the topics reported in this study.

## References

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

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

[3] B. Mulgrew and C. F. N. Cowan, *Adaptive Filter and Equalisers.* Boston, MA: Kluwer Academic, 1988.

[4] S. U. H. Qureshi, "Adaptive equalization," *Proc. IEEE,* vol. 73, pp. 1349–1387, 1985.

[5] G. J. Gibson, S. Siu, and C. F. N. Cowan, "Application of multilayer perceptrons as adaptive channel equalisers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Glasgow, Scotland, 1989, pp. 1183–1186.

[6] S. Chen, G. J. Gibson, and C. F. N. Cowan, "Adaptive channel equalisation using a polynomial-perceptron structure," *Proc. Inst. Elec. Eng.,* vol. 137, pt. I, no. 5, pp. 257–264, 1990.

[7] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems,* vol. 2, pp. 321–355, 1988.

[8] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE,* vol. 78, pp. 1481–1497, 1990.

[9] 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.

[10] D. F. Specht, "Generation of polynomial discriminant functions for pattern recognition," *IEEE Trans. Electron. Comput.,* vol. EC-16, pp. 308–319, 1967.

[11] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.

[12] B. Widrow and S. D. Stearns, *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1985.

[13] 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.

[14] S. Chen, S. A. Billings, and P. M. Grant, "Recursive hybrid algorithm for nonlinear system identification using radial basis function networks," *Int. J. Control,* vol. 55, no. 5, pp. 1051–1070, 1992.

[15] I. J. Leontaritis and S. A. Billings, "Experimental design and identifiability for nonlinear systems," *Int. J. Systems Sci.,* vol. 18, pp. 189–202, 1987.

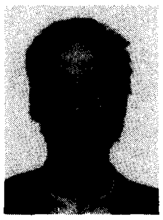
[16] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive Bayesian decision feedback equaliser based on a radial basis function network," in *Proc. ICC '92,* Chicago, IL, 1992, vol. 3, pp.343.3./–343.3.5.

[17] T. L. Clarke, "Generalization of neural networks to the complex plane," in *Proc. Int. Joint Conf. Neural Networks,* San Diego, CA, 1990, vol. II, pp. 435–440.

[18] S. Chen, S. McLaughlin, and B. Mulgrew, "Complex-valued radial basis function network, Part I: network architecture and learning algorithms," *EURASIP Signal Processing J.,* submitted for publication, 1992.

**Sheng Chen** (M'90) was born in Fujian Province, China, in 1957. He received the B.Sc. degree in control engineering from the East China Petroleum Institute in 1982 and the Ph.D. degree in control engineering from the City University at London in 1986.

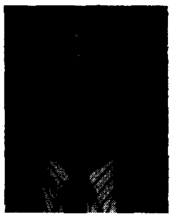
From 1986 to 1989 he was a postdoctoral research associate in the Department of Control Engineering at the University of Sheffield. He joined the Signal Processing Group in the Department of Electrical Engineering at the University of Edinburgh in 1989 as a research fellow. His research interest are in modeling and identification of nonlinear systems, artificial neural networks, and adaptive signal processing for communications.



**Bernard Mulgrew** (M'88) received the Ph.D. degree in 1987 from the University of Edinburgh.

He worked for four years in the Radar Systems Department at Ferranti Defence Systems, Edinburgh, Scotland, after graduation in 1979. From 1983 to 1986 he was a research associate in the Department of Electrical Engineering at the University of Edinburgh, studying the performance and design of adaptive filter algorithms. He is currently a lecturer in the Department of Electrical Engineering, University of Edinburgh, where he gives courses on signals and systems, and signal processing. His research interests are in adaptive signal processing and estimation theory and in their application to radar and communications systems. He is a coauthor of two books on signal processing.

Dr. Mulgrew is an associate member of the IEE and a member of the Audio Engineering Society.



**Peter M. Grant** (M'77–SM'83) was born in St. Andrews, Scotland, in 1944. He received the B.Sc. degree in electronic engineering from the Heriot-Watt University, Edinburgh, Scotland, in 1966, and the Ph.D. degree from the University of Edinburgh in 1975.

From 1966 to 1970 he was with Plessey Company Ltd., before he was appointed to a research fellowship at the University of Edinburgh to study the applications of surface acoustic wave and charge-coupled communication systems. He was subsequently appointed to a lectureship and promoted through to a personal chair in Electronic Signal Processing with responsibility for teaching signal processing and communication systems. He leads the signal processing research group with personal involvement in adaptive filtering and pattern recognition. In 1985–1986 he was appointed as a visiting staff member at the MIT Lincoln Laboratory.

Dr. Grant was the recipient of a James Caird Travelling Scholarship during the academic year 1977–1978 and, as a visiting assistant professor, he researched at the Ginzton Laboratory, Stanford University. He is a fellow of the Institution of Electrical Engineers (London). He serves there as one of the honorary editors of the *IEE Proceedings-Part F, Radar and Signal Processing,* as the senior editor and chairman of the *IEE Proceedings* Editorial Panel, and as a panel member for the *Electronics and Communications Engineering Journal.*