Orthogonal least-squares algorithm for training multioutput radial basis function networks

S. Chen
P.M. Grant
C.F.N. Cowan

Indexing terms: Signal processing, Algorithms, Radial basis function networks

Abstract: A constructive learning algorithm for multioutput radial basis function networks is presented. Unlike most network learning algorithms, which require a fixed network structure, this algorithm automatically determines an adequate radial basis function network structure during learning. By formulating the learning problem as a subset model selection, an orthogonal least-squares procedure is used to identify appropriate radial basis function centres from the network training data, and to estimate the network weights simultaneously in a very efficient manner. This algorithm has a desired property, that the selection of radial basis function centres or network hidden nodes is directly linked to the reduction in the trace of the error covariance matrix. Nonlinear system modelling and the reconstruction of pulse amplitude modulation signals are used as two examples to demonstrate the effectiveness of this learning algorithm.

1 Introduction

The radial basis function (RBF) network has been applied to many diverse fields in signal processing in the past few years. The RBF method was originally employed as a numerical interpolation technique in multidimensional space [1], and was later adopted as a one-hidden-layer feedforward network [2]. An excellent review on this topic is given in Reference 3. Considerable attention has been focused on how to derive linear learning methods by exploiting the structural characteristics of the RBF network. Each hidden node in an RBF network has a radially symmetric response around a node parameter vector called a centre, and the network output layer is a set of linear combiners with weights. A common learning strategy is to randomly select some network input vectors as the RBF centres, thus effectively fixing the network hidden layer. The weights in the output layer can then be learnt using the least-squares (LS) method [2].

Paper 9134F (C4, E5), first received 16th January and in revised form 20th August 1992

S. Chen and P.M. Grant are with the Department of Electrical Engineering, University of Edinburgh, King's Buildings, Edinburgh EH9 3JL, United Kingdom

C.F.N. Cowan is with the Department of Electronic and Electrical Engineering, Loughborough University of Technology, Loughborough LE11 3TU, United Kingdom

Arbitrarily choosing some data points as centres, however, may not always satisfy the requirement that centres should suitably sample the network input domain. Furthermore, such an approach may require an unnecessarily large RBF network to achieve a given level of performance and, as a result, causes numerical ill-conditioning. These shortcomings can be overcome by formulating the learning problem as one of selecting subset models while preserving the advantages of linear learning. For RBF networks with a scalar output, an intelligent learning algorithm has been derived based on the orthogonal LS (OLS) method, which constructs RBF networks in a rational way [4, 5]. The algorithm chooses appropriate RBF centres one by one from training data points until a satisfactory network is obtained. Each selected centre maximises the increment to the explained variance of the desired output, and so learning does not suffer numerical ill-conditioning problems. An alternative linear learning procedure is the hybrid clustering and LS algorithm [6, 7]. The main attraction of this algorithm is that it can naturally be implemented in a recursive form. However, the hybrid learning algorithm requires that the number of hidden nodes must first be given.

In contrast to most learning algorithms, which can only work if a fixed network structure has first been specified, the OLS algorithm is a structural identification algorithm, and it constructs an adequate network structure in an intelligent way during learning. The present study continues this theme and extends this OLS learning algorithm to multioutput RBF networks. The basic idea is to use the trace of the desired output covariance matrix as the selection criterion for choosing RBF centres, instead of the variance in the single-output case. A brief summary of the RBF network architecture and a discussion on the approximation capability of RBF networks are first given. The derivation of the OLS algorithm is then presented. Two applications are used to illustrate the OLS algorithm: the first case considers modelling multiinput–multioutput (MIMO) nonlinear systems based on an RBF network. In the second, the reconstruction of pulse amplitude modulation (PAM) signals is viewed as a multiclass classification problem, and an RBF network is constructed to approximate the optimal Bayesian solution.

This work was supported by the UK Science and Engineering Research Council under award GR/G/53095. The authors gratefully acknowledge contributions from Prof. S.A. Billings on the topics reported in this study.

2 Network architecture and approximation ability

The RBF network has a topology of the one-hidden-layer neural network. Denote the network input and output dimensions as n_I and n_o, respectively, and let n_h be the number of hidden nodes. The outputs of hidden nodes are specified by

$$\phi_j = \phi(\|\boldsymbol{x} - \boldsymbol{c}_j\|; \sigma_j) \quad 1 \leqslant j \leqslant n_h \tag{1}$$

where $\boldsymbol{x} \in R^{n_I}$ is a network input vector, $\boldsymbol{c}_j \in R^{n_I}$ are the RBF centres, σ_j are the real positive scalars known as the widths, $\|\cdot\|$ denotes the Euclidean norm, and $\phi(\cdot\,; \sigma)$ is a nonlinear function from R^+ to R, and is referred to as the nonlinearity of hidden nodes. Each output node is a linear combiner defined by

$$f_{r_i}(\boldsymbol{x}) = \sum_{j=1}^{n_h} \phi_j \theta_{ji} \quad 1 \leqslant i \leqslant n_0 \tag{2}$$

where θ_{ji} are the weights. The nonlinearity $\phi(\cdot\,; \sigma)$ has a radially symmetric shape. Although there is a variety of choices for this node nonlinearity, these choices belong to either the class one: $\phi(r; \sigma) \to 0$ as $r \to \infty$, or the class two: $\phi(r; \sigma) \to \infty$ as $r \to \infty$. Two typical choices of $\phi()$ are the Gaussian function

$$\phi(r; \sigma) = \exp(-r^2/\sigma^2) \tag{3}$$

and the thin-plate-spline function

$$\phi(r; 1) = r^2 \log(r) \tag{4}$$

The overall input–output mapping of the network is $f_r : R^{n_I} \to R^{n_o}$.

The RBF network has a very general approximation ability [8, 9]. Under very mild assumptions on the nonlinearity $\phi()$, any continuous function $f: D_f \subset R^{n_I} \to R^{n_o}$ can be uniformly approximated to within an arbitrary accuracy by an RBF network f_r on D_f provided that there are a sufficient number of hidden nodes, where D_f is a compact subset of R^{n_I}. A sufficient condition on $\phi()$ to guarantee the universal approximation is $\phi()$ being continuous and bounded [9]. This is obviously a very mild assumption, and the class one nonlinearity such as eqn. 3 satisfies this requirement. The class two nonlinearity such as eqn. 4 does not satisfy this condition. According to Powell [10], however, RBF networks based on the class two nonlinearity also have excellent approximation ability. In fact, it is easier to achieve a good approximation if $\phi(r; \sigma) \to \infty$ as $r \to \infty$ than if $\phi(r; \sigma) \to 0$ as $r \to \infty$. Based on these theoretical results, it can be concluded that the choice of $\phi()$ is not crucial for network performance. Although each hidden node may have a different width parameter σ_j, a same width is sufficient for universal approximation [9]. All the widths in the network can therefore be fixed to a value σ, and this can result in a simpler training strategy. Some choices of the nonlinearity, such as eqn. 4, do not require to specify a width.

3 Learning based on orthogonal least-squares method

The task of network learning is to choose appropriate centres $\boldsymbol{c}_j$ and to determine the corresponding weights θ_{ji}, based on a given set of network training inputs and desired outputs $\{\boldsymbol{x}(t), \boldsymbol{d}(t)\}_{t=1}^N$, where $\boldsymbol{x}(t) = [x_1(t) \cdots x_{n_I}(t)]^T$ and $\boldsymbol{d}(t) = [d_1(t) \cdots d_{n_o}(t)]^T$. To avoid nonlinear learning, the RBF centres are to be selected from training data, and this is equivalent to a problem of subset model selection. The full model is defined by considering all the training data $\{\boldsymbol{x}(k)\}_{k=1}^N$ as candidates for centres.

Assume that a nonlinearity $\phi()$ is chosen and a fixed width σ is given. A candidate centre $\boldsymbol{c}_j = \boldsymbol{x}(k)$ gives rise to a candidate hidden node ϕ_j in the full RBF network of N hidden nodes. The desired outputs can be expressed as

$$d_i(t) = \sum_{j=1}^{N} \phi_j(t)\theta_{ji} + e_i(t) \quad 1 \leqslant i \leqslant n_o \tag{5}$$

where $e_i(t)$ are the errors between the desired outputs and the network outputs. The model in eqn. 5 is a linear regression model. $\phi_j(t)$ are known as the regressors, which are some fixed functions of the input vector $\boldsymbol{x}(t)$. By defining

$$\boldsymbol{d}_i = [d_i(1) \cdots d_i(N)]^T \quad 1 \leqslant i \leqslant n_o \tag{6}$$

$$\boldsymbol{e}_i = [e_i(1) \cdots e_i(N)]^T \quad 1 \leqslant i \leqslant n_0 \tag{7}$$

$$\boldsymbol{\Phi}_j = [\phi_j(1) \cdots \phi_j(N)]^T \quad 1 \leqslant j \leqslant N \tag{8}$$

then for $1 \leqslant t \leqslant N$, eqn. 5 can be collectively written as

$$[\boldsymbol{d}_1 \cdots \boldsymbol{d}_{n_o}] = [\Phi_1 \cdots \Phi_N] \begin{bmatrix} \theta_{11} & \cdots & \theta_{1n_o} \\ \vdots & & \vdots \\ \theta_{N1} & \cdots & \theta_{Nn_o} \end{bmatrix} + [\boldsymbol{e}_1 \cdots \boldsymbol{e}_{n_o}] \tag{9}$$

or, more concisely, in the matrix form

$$\boldsymbol{D} = \Phi\Theta + \boldsymbol{E} \tag{10}$$

The parameter matrix Θ can readily be solved using the LS principle.

From a geometric viewpoint, the regressors Φ_j form a set of basis vectors. These bases, however, are generally correlated. An orthogonal transformation can be performed to transfer from the set of Φ_j into a set of orthogonal basis vectors. This can be achieved by decomposing Φ into

$$\Phi = \boldsymbol{W}A \tag{11}$$

where

$$A = \begin{bmatrix} 1 & \alpha_{12} & \cdots & \alpha_{1N} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 1 & \alpha_{N-1N} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{12}$$

and

$$\boldsymbol{W} = [\boldsymbol{w}_1 \cdots \boldsymbol{w}_N] \tag{13}$$

with orthogonal columns that satisfy

$$\boldsymbol{w}_i^T \boldsymbol{w}_j = 0 \quad \text{if } i \neq j \tag{14}$$

The space spanned by the set of $\boldsymbol{w}_j$ is the same space spanned by the set of Φ_j, and eqn. 10 can be rewritten as

$$\boldsymbol{D} = \boldsymbol{W}G + \boldsymbol{E} \tag{15}$$

The OLS solution

$$G = \begin{bmatrix} g_{11} & \cdots & g_{1n_o} \\ \vdots & & \vdots \\ g_{N1} & \cdots & g_{Nn_o} \end{bmatrix} \tag{16}$$

and the ordinary LS solution Θ satisfy the triangular system

$$A\Theta = G \tag{17}$$

The classic and modified Gram–Schmidt methods [11] can be used to derive A and G, and thus to solve for Θ from eqn. 17. Alternatively, the Householder transformation method [12] can be used to obtain a similar orthogonal decomposition.

The number N of all the candidate regressors is generally very large, but an adequate network may only require $n_h (\ll N)$ significant regressors. These significant regressors or hidden nodes can be selected using the OLS algorithm, similar to the case of selecting subset models for the general linear regression model [13, 14]. A criterion for determining the significance of candidates, however, must first be chosen. In the single-output case, the contribution of a candidate to the variance of the desired output is used to define how significant this candidate is [13]. For the multioutput case, the trace of the desired output covariance matrix is a natural choice. Because the error matrix $\boldsymbol{E}$ is orthogonal to $\boldsymbol{W}$, after some simple calculation it can be shown that the trace of the covariance of $d(t)$ is

$$\text{trace}\,(\boldsymbol{D}^T\boldsymbol{D}/N) = \sum_{j=1}^{N}\left(\sum_{i=1}^{n_o} g_{ji}^2\right)\boldsymbol{w}_j^T\boldsymbol{w}_j/N + \text{trace}\,(\boldsymbol{E}^T\boldsymbol{E}/N) \quad (18)$$

The error reduction ratio due to $\boldsymbol{w}_k$ can be defined as

$$[err]_k = \left(\sum_{i=1}^{n_o} g_{ki}^2\right)\boldsymbol{w}_k^T\boldsymbol{w}_k/\text{trace}\,(\boldsymbol{D}^T\boldsymbol{D}) \quad 1 \leqslant k \leqslant N \quad (19)$$

Based on this ratio, significant regressors can be selected in a forward regression procedure. At the kth step of the selection procedure, a candidate regressor is selected as the kth regressor of the subset network if it produces the largest value of $[err]_k$ from among the rest of the $N - k + 1$ candidates. The selection is terminated when

$$1 - \sum_{k=1}^{n_h}[err]_k < \rho \quad (20)$$

where $0 < \rho < 1$ is a chosen tolerance. This gives rise to a subset network containing n_h significant hidden nodes. The selection procedure is very similar to that for single-output models [5, 13].

If the desired output vector has a zero-mean vector, the first term in the right-hand side of eqn. 18 is the part of the trace of the desired output covariance matrix that can be explained by the regressors, and the second term is the unexplained trace of the desired output covariance. Thus

$$\left(\sum_{i=1}^{n_o} g_{ki}^2\right)\boldsymbol{w}_k^T\boldsymbol{w}_k/N \quad (21)$$

is the increment to the explained trace due to $\boldsymbol{w}_k$, and each selected centre maximises the increment to the explained trace of the desired output covariance. The selection of centres is therefore directly linked to the reduction in the error covariance trace. Another advantage of this algorithm is that numerical ill-conditioning can easily be avoided. It can be shown that $\boldsymbol{w}_k^T\boldsymbol{w}_k = 0$ implies that Φ_k is a linear combination of Φ_1 to Φ_{k-1}. If $\boldsymbol{w}_k^T\boldsymbol{w}_k$ is less than a small positive threshold, the candidate regressor Φ_k will not be selected, and this ensures a well conditioned LS solution. It is worth pointing out that the algorithm does not attempt to find an 'optimal' solution for subset network selection. In theory, the optimal subset network could be constructed by testing all the possible subset networks, which, however, is impossible to do even for a modest N.

The tolerance ρ is important in balancing the accuracy and the complexity of the final network. The ideal value for ρ can be learnt by interacting with the selection procedure [13, 14]. The terminating criterion in eqn. 20 emphasises the network performance. Because a more accurate performance is often achieved at the expense of using a larger network, a trade-off between performance and complexity is often desired. This can be achieved using an alternative terminating criterion based on the Akaike information criterion [15]:

$$AIC(\chi) = N \log\,(\det\,(N^{-1}\boldsymbol{E}^T\boldsymbol{E})) + n_h\chi \quad (22)$$

where $\det(\cdot)$ is the determinant operator, and χ is the critical value of the chi-squared distribution with one degree of freedom and for a given level of significance. The criterion of eqn. 22 can be combined with the orthogonal selection procedure. The significant regressors are selected by the OLS selection procedure, based on their significances as indicated by their error reduction ratios, and the selection is terminated when $AIC(\chi)$ reaches its minimum.

4 Nonlinear system modelling

Consider dynamic systems which are governed by the nonlinear difference equation.

$$y(t) = f[y(t-1), \ldots, y(t-n_y), u(t-1), \ldots, u(t-n_u)] + \varepsilon(t) \quad (23)$$

where $y(t)$ and $\varepsilon(t)$ are the m_y-dimensional system output and noise vectors, respectively; $u(t)$ is the m_u-dimensional system input vector; n_y and n_u are the lags in the output and input, respectively; and the nonlinear function $f: R^{n_I} \rightarrow R^{m_y}$, with the dimension of the input space being

$$n_I = m_y \times n_y + m_u \times n_u \quad (24)$$

Given a set of system outputs $y(t)$ and inputs $u(t)$ (in the case of a time series process) only a set of $y(t)$ is provided, we can introduce

$$\boldsymbol{x}(t) = [y^T(t-1) \cdots y^T(t-n_y)u^T(t-1) \cdots u^T(t-n_u)]^T \quad (25)$$

as the RBF network input vector at sample t, and use $y(t)$ as the corresponding desired output to train an RBF network so that the network $f_r(\,)$ realises or approximates the underlying system dynamics $f(\,)$. The system representation in eqn. 23 is a simplified case of the general nonlinear system known as the NARMAX model [16], and the approach given here is therefore a special version of the general identification scheme reported in Reference 4.

The first example used to test the OLS algorithm was a simulated two-output time series process. One thousand noisy observations were generated using the model

$$\begin{aligned} y_1(t) &= (0.8 - 0.5\exp(-y_1^2(t-1)))y_1(t-1) \\ &\quad - (0.3 + 0.9\exp(-y_1^2(t-1)))y_1(t-2) \\ &\quad + 0.1\sin(y_2(t-1)) + \varepsilon_1(t) \\ y_2(t) &= 0.6y_2(t-1) + 0.2y_2(t-1)y_2(t-2) \\ &\quad + 1.2\tanh(y_1(t-2)) + \varepsilon_2(t) \end{aligned}$$

as training data, where the Gaussian noise $\varepsilon(t) = [\varepsilon_1(t)\varepsilon_2(t)]^T$ had statistics $\boldsymbol{E}[\varepsilon_1(t)] = \boldsymbol{E}[\varepsilon_2(t)] = \boldsymbol{E}[\varepsilon_1(t)\varepsilon_2(t)] = 0.0$ and $\boldsymbol{E}[\varepsilon_1^2(t)] = \boldsymbol{E}[\varepsilon_2^2(t)] = 0.01$. A two-output RBF network was employed to model this nonlinear process, with the network input defined by $\boldsymbol{x}(t) = [y_1(t-1)y_1(t-2)y_2(t-1)y_2(t-2)]^T$. The nonlinearity

$\phi()$ was chosen for eqn. 4. The number of candidates for centres was about 1000, and the OLS algorithm was used to construct an RBF network. During the learning it was found that a suitable value for ρ was 0.0183, and the OLS algorithm identified a subset network of 50 centres. Another 1000 samples of noisy time series were then generated to validate the obtained network. The covariances of the network prediction error between the noisy observation $y(t)$ and the one-step-ahead network prediction $\hat{y}(t) = f_r(\boldsymbol{x}(t))$ for both the training and testing data sets are listed in Table 1. The training data and the selected

Table 1: Covariance of network prediction error for time series example

Training set	$9.66032e-3$	$1.47958e-5$
	$1.47958e-5$	$9.73829e-3$
Testing set	$1.13836e-2$	$3.90550e-5$
	$3.90550e-5$	$1.13188e-2$

RBF centres are plotted in Fig. 1, where it can be seen that the noisy observations have a symmetrical distribution and the selected centres clearly reflect this pattern. In Fig. 2, the one-step-ahead network predictions over the first 100 testing data are superimposed on these testing

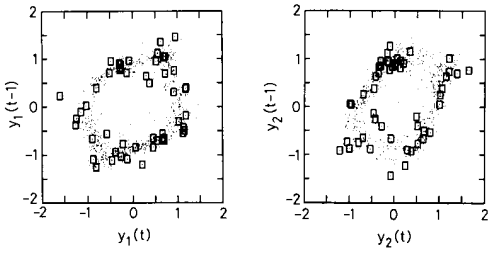

Fig. 1 *Two-dimensional representations of the noisy observations (·) and the RBF centres (□) selected by the OLS algorithm*

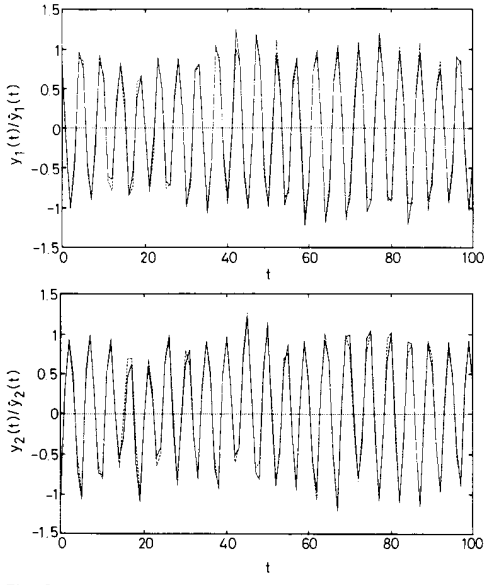

Fig. 2 *One-step-ahead network predictions superimposed on time series testing data*

——— noisy time series observations
····· network predictions

time series observations. The underlying dynamics of the simulated time series are determined by the autonomous system output $y_d(t) = f(y_d(t-1), y_d(t-2))$, which generates a stable limit cycle as shown in Fig. 3. The identified RBF network was used to iteratively produce the

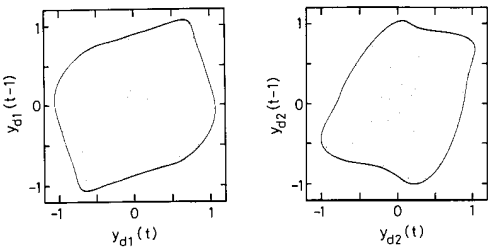

Fig. 3 *Two-dimensional response of the autonomous time series process based on 1000 data samples*

network output $\hat{y}_d(t) = f_r(\boldsymbol{x}_d(t))$, where $\boldsymbol{x}_d(t) = [\hat{y}_{d1}(t-1)\hat{y}_{d1}(t-2)\hat{y}_{d2}(t-1)\hat{y}_{d2}(t-2)]^T$. The iterative network output produces a similar limit cycle, as can be seen in Fig. 4. The output waveform from the iterative network is superimposed on that of the autonomous time

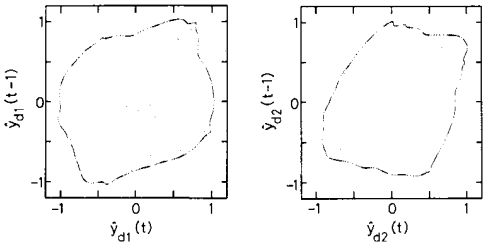

Fig. 4 *Two-dimensional response of the iterative network based on 1000 data samples*

series outputs in Fig. 5, and it is seen that the amplitudes of the two responses agree with each other well. Phase deviations of the two waveforms accumulate only slowly as time elapses. The above testing results confirm that the selected RBF network does capture the underlying dynamics of the system, even though it was identified using noisy observations.

The second example was a two-input–two-output data set collected from a 50 MW turboalternator, operating in parallel with an interconnected system having a capacity of approximately 5000 MW [17]. The data set contains 100 samples. The input $u_1(t)$ was the in-phase current deviation and $u_2(t)$ was the out-of-phase current deviation. The output $y_1(t)$ was the voltage deviation and $y_2(t)$ was the frequency deviation. The system inputs are plotted in Fig. 6, and the system outputs are shown in Fig. 7. A two-output RBF network with the nonlinearity of eqn. 4 was used to identify this system. The network input was defined as

$$\boldsymbol{x}(t) = [y_1(t-1)y_1(t-2)y_1(t-3)y_2(t-1)y_2(t-2) \\ y_2(t-3)u_1(t-1)u_1(t-2)u_2(t-1)u_2(t-2)]^T$$

For the given desired tolerance $\rho = 0.000018$, the OLS algorithm selected a subset network of 45 centres, and the covariance of the network prediction error was

$$\begin{bmatrix} 2.69805e-4 & -1.01140e-5 \\ -1.01140e-5 & 2.56552e-4 \end{bmatrix}$$

The one-step-ahead network predictions $\hat{y}(t) = f_r(\boldsymbol{x}(t))$ and the iterative network outputs $\hat{y}_d(t) = f_r(\boldsymbol{x}_d(t))$, where

$$\boldsymbol{x}_d(t) = [\,\hat{y}_{d1}(t-1)\hat{y}_{d1}(t-2)\hat{y}_{d1}(t-3)\hat{y}_{d2}(t-1)\hat{y}_{d2}(t-2)$$
$$\hat{y}_{d2}(t-3)u_1(t-1)u_1(t-2)u_2(t-1)u_2(t-2)]^T$$

are superimposed on the alternator outputs $y(t)$ in Figs. 7 and 8, respectively. The results in Fig. 8 clearly show that the identified RBF network is an excellent model for the turboalternator, and can be used to investigate the properties of the latter.

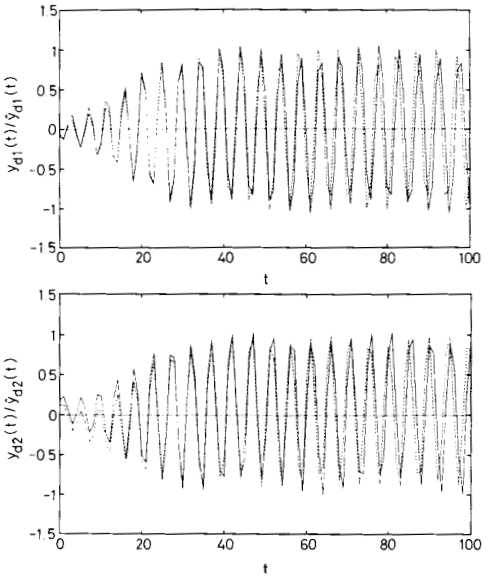

Fig. 5 *Iterative network outputs superimposed on autonomous time series observations*

——— time series observations
····· network outputs

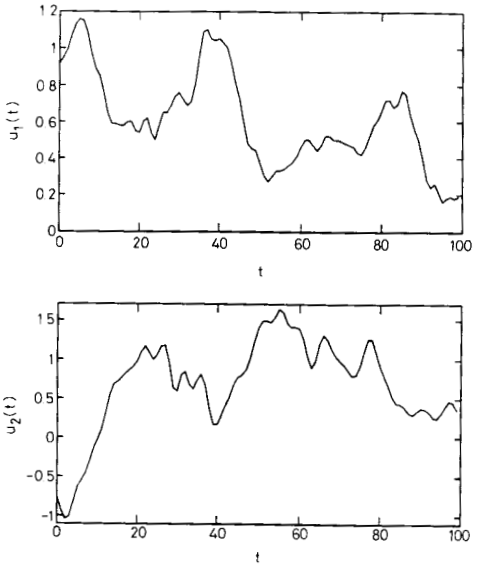

Fig. 6 *System input data set for turbo alternator*

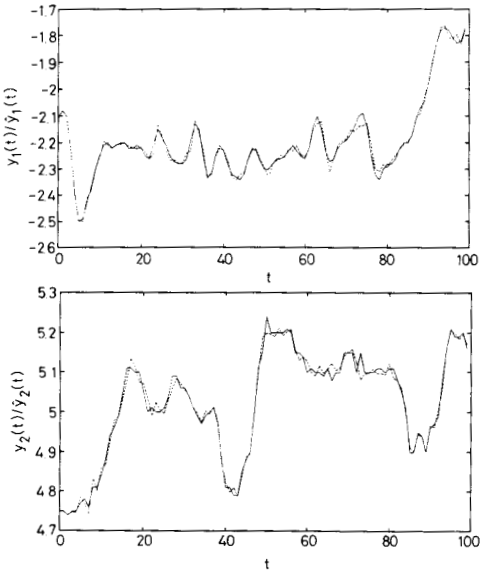

Fig. 7 *One-step-ahead network predictions superimposed on turbo-alternator outputs*

——— system outputs
– – – – network predictions

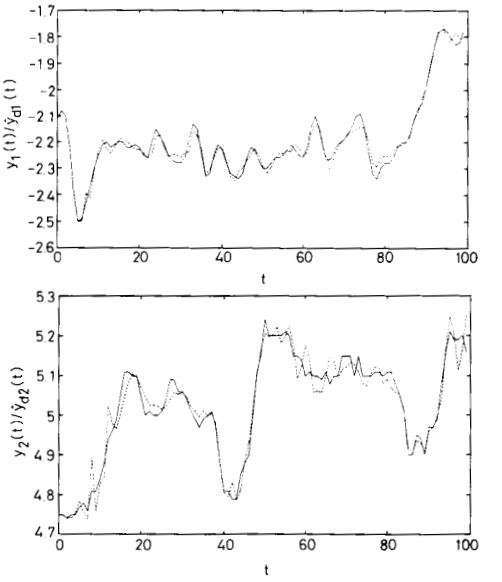

Fig. 8 *Iterative network outputs superimposed on turboalternator outputs*

——— system outputs
– – – – network outputs

5 Reconstruction of PAM signals

An important application of neural networks is pattern classification. Here, the equalisation of communications channels with a multi-ary PAM signalling scheme is viewed as a multiclass classification problem, and an RBF network is constructed to solve it. A general digital communications system is shown in Fig. 9, where the

channel is modelled as a finite impulse response filter with transfer function

$$H(z) = \sum_{i=0}^{n} h_i z^{-i} \tag{26}$$

The channel output is corrupted by an additive white Gaussian noise $e(t)$. The task of the equaliser at sample t

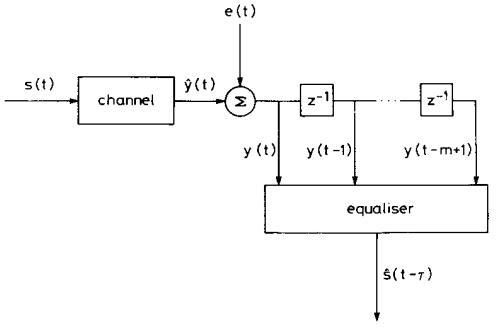

Fig. 9 *Schematic of data transmission system*

is to reconstruct input symbol $s(t-\tau)$, based on the channel observation vector

$$y(t) = [y(t) \cdots y(t-m+1)]^T \tag{27}$$

where the integers m and τ are known as the equaliser order and delay, respectively. In the present study $s(t)$ is assumed to be an M-ary PAM signal

$$s(t) = s^{(i)} \quad 1 \leqslant i \leqslant M \tag{28}$$

where $M = 2^L$ and L is an integer. Fig. 9 is often referred to as the symbol decision structure and the most commonly used equaliser is the linear transversal equaliser [18]. The optimal equaliser solution for the structure of Fig. 9, however, is nonlinear and can be derived based on Bayes decision theory [19].

The number of all the possible combinations of the channel input sequence

$$s(t) = [s(t) \cdots s(t-m+1-n)]^T \tag{29}$$

is $n_s = M^{m+n}$, and this gives rise to n_s states of the noise-free channel outcome

$$\hat{y}(t) = [\hat{y}(t) \cdots \hat{y}(t-m+1)]^T \tag{30}$$

The set of these states, denoted as $Y_{m,\tau}$, can be partitioned into M subsets according to the value of $s(t-\tau)$:

$$Y_{m,\tau} = \bigcup_{1 \leqslant i \leqslant M} Y_{m,\tau}^{(i)} \tag{31}$$

where

$$Y_{m,\tau}^{(i)} = \{\hat{y}(t) \mid s(t-\tau) = s^{(i)}\} \quad 1 \leqslant i \leqslant M \tag{32}$$

The task of the equaliser is equivalent to an M-class classification problem. Compute M Bayesian decision variables

$$\eta^{(i)}(t) = \sum \beta_j^{(i)} p_e(y(t) - y_j^{(i)}) \quad 1 \leqslant i \leqslant M \tag{33}$$

where $y_j^{(i)} \in Y_{m,\tau}^{(i)}$, $\beta_j^{(i)}$ is the *a priori* probability of $y_j^{(i)}$, the sum is over the set $Y_{m,\tau}^{(i)}$, and $p_e(\cdot)$ is the probability density function of

$$e(t) = [e(t) \cdots e(t-m+1)]^T \tag{34}$$

Then the minimum error probability decision is

$$\hat{s}(t-\tau) = s^{(i^*)} \quad \text{if } \eta^{(i^*)}(t) = \max\{\eta^{(i)}(t), 1 \leqslant i \leqslant M\} \tag{35}$$

The Bayesian decision procedure effectively partitions the m-dimensional channel observation space into M decision regions. When $y(t)$ is within the ith region, the decision $\hat{s}(t-\tau) = s^{(i)}$ is made.

As the above Bayesian equaliser solution is an M-class classification problem, an L-output RBF network can be trained to approximate this optimal equaliser. Because the noise distribution is generally white Gaussian, the nonlinearity $\phi(\,)$ is obviously chosen as the Gaussian function (eqn. 3) with an ideal width defined by $\sigma^2 = 2\sigma_e^2$, where σ_e^2 is the noise variance. In practice, an estimated σ_e^2 is sufficient for setting the width parameter. The centres of the network should ideally be the channel states $y_j^{(i)}$. These states are, however, unknown, and the OLS algorithm is used to select appropriate centres from the noise data $y(t)$ and to determine the network weights. This approach is best illustrated using a simple example.

Let $s(t)$ be a quarternary PAM signal taking values from the set $\{\pm 1, \pm 3\}$, and let the channel transfer function be $H(z) = 1.0 + 0.5z^{-1}$. Assume that the equaliser has a structure of $m = 2$ and $\tau = 0$. In the absence of noise, channel output vectors are 64 discrete points. Each of these points is shown in Fig. 10, using one of the four

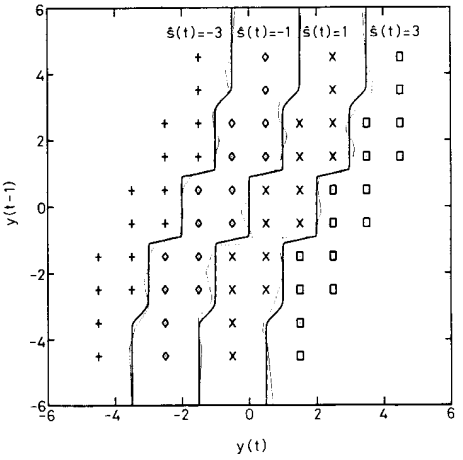

Fig. 10 *Comparison of decision boundaries*
noise variance 0.0625
——— optimal Bayesian
····· RBF network

symbols $\{+, \diamond, \times, \square\}$ which correspond to the input set $\{-3, -1, 1, 3\}$. The Bayesian decision boundaries for a noise variance 0.0625 are plotted in Fig. 10. A two-output RBF network is sufficient for this four-class classification problem. The network inputs are $x(t) = [y(t)y(t-1)]^T$, and the desired outputs are set to $d(t) = [1\ 1]^T$, $[1\ -1]^T$, $[-1\ 1]^T$ and $[-1\ -1]^T$, corresponding to $s(t) = 3$, 1, -1 and -3. 740 points of training data were generated. An RBF network of 74 centres was selected using the OLS learning algorithm, and the decision boundaries of this RBF network are also shown in Fig. 10. This selecting procedure was repeated for a variety of noise variances and the performance of the selected RBF network is compared with that of the optimal Bayesian equaliser in Fig. 11.

Using an RBF network to realise the Bayesian equaliser provides a significant performance improvement over the linear equaliser, at the cost of a considerable increase in computational complexity. An important technique for improving equaliser performance is to use decision feedback. The conventional decision feedback equaliser (DFE) expands the equaliser inputs of the linear transversal equaliser to include past detected symbols [18]. A novel Bayesian DFE has been developed that realises a significant performance gain over the conventional DFE at the cost of only a very small increase in computational load [20, 21]. It is beyond the scope of this paper to study the Bayesian DFE for an M-ary PAM signalling scheme. Readers interested in this can find a detailed discussion on the Bayesian DFE and its realisation using RBF networks in References 20 and 21.

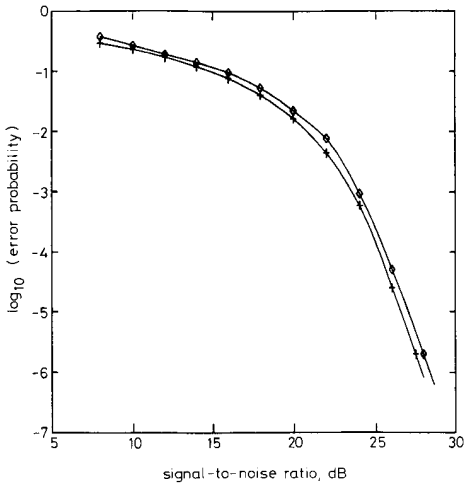

Fig. 11 *Comparison of error rate performance of the optimal Bayesian equaliser with the RBF design*
--◇-- RBF
+—+ optimal

6 Conclusions

An orthogonal least-squares algorithm has been extended for the construction of multioutput radial basis function networks. This learning strategy provides a systematic approach linking the selection of radial basis function centres from the training data set to the reduction of the error covariance trace. Unlike most network learning algorithms, which can only work when a network structure has been specified, this algorithm automatically identifies an adequate network structure during learning. Applications in two different areas of signal processing have been demonstrated.

7 References

1 POWELL, M.J.D.: 'Radial basis functions for multivariable interpolation: a review', *in* MASON, J.C., and COX, M.G. (Eds.): 'Algorithms for approximation' (Oxford, 1987), pp. 143–167
2 BROOMHEAD, D.S., and LOWE, D.: 'Multivariable functional interpolation and adaptive networks', *Complex Syst.*, 1988, **2**, pp. 321–355
3 POGGIO, T., and GIROSI, F.: 'Networks for approximation and learning', *Proc. IEEE*, 1990, **78**, pp. 1481–1497
4 CHEN, S., BILLINGS, S.A., COWAN, C.F.N., and GRANT, P.M.: 'Practical identification of NARMAX models using radial basis functions', *Int. J. Control*, 1990, **52**, pp. 1327–1350
5 CHEN, S., COWAN, C.F.N., and GRANT, P.M.: 'Orthogonal least-squares learning algorithm for radial basis function networks', *IEEE Trans. Neural Networks*, 1991, **2**, pp. 302–309
6 MOODY, J., and DARKEN, C.J.: 'Fast learning in networks of locally tuned processing units', *Neural Comput.*, 1989, **1**, pp. 281–294
7 CHEN, S., BILLINGS, S.A., and GRANT, P.M.: 'Recursive hybrid algorithm for non-linear system identification using radial basis function networks', *Int. J. Control*, 1992, **55**, pp. 1051–1070
8 CYBENKO, G.: 'Approximations by superpositions of a sigmoidal function', *Math. Control, Signals Syst.*, 1989, **2**, pp. 303–314
9 PARK, J., and SANDBERG, I.W.: 'Universal approximation using radial-basis-function networks', *Neural Comput.*, 1991, **3**, pp. 246–257
10 POWELL, M.J.D.: 'Radial basis function approximations to polynomials', *in* Proc. 12th Biennial Numerical Analysis Conf., Dundee, 1987, pp. 223–241
11 BJÖRCK, A.: 'Solving linear least-squares problems by Gram–Schmidt orthogonalization', *Nordisk Tidskr. Informations-Behandling*, 1967, **7**, pp. 1–21
12 GOLUB, G.: 'Numerical methods for solving linear least-squares problems', *Numerische Mathematik*, 1965, **7**, pp. 206–216
13 CHEN, S., BILLINGS, S.A., and LUO, W.: 'Orthogonal least-squares methods and their application to non-linear system identification', *Int. J. Control*, 1989, **50**, pp. 1873–1896
14 BILLINGS, S.A., and CHEN, S.: 'Extended model set, global data and threshold model identification of severely non-linear systems', *Int. J. Control*, 1989, **50**, pp. 1897–1923
15 LEONTARITIS, I.J., and BILLINGS, S.A.: 'Model selection and validation methods for non-linear systems', *Int. J. Control*, 1987, **45**, pp. 311–341
16 CHEN, S., and BILLINGS, S.A.: 'Representation of non-linear systems: the NARMAX model', *Int. J. Control*, 1989, **49**, pp. 1013–1032
17 JENKINS, G.M., and WATTS, D.G.: 'Spectral analysis and its applications' (San Francisco: Holden-Day, 1968)
18 QURESHI, S.U.H.: 'Adaptive equalization', *Proc. IEEE*, 1985, **73**, pp. 1349–1387
19 DUDA, R.O., and HART, P.E.: 'Pattern classification and scene analysis' (New York: John Wiley and Sons, 1973)
20 CHEN, S., MULGREW, B., and McLAUGHLIN, S.: 'Adaptive Bayesian equaliser with decision feedback'. Submitted to *IEEE Trans. Signal Proc.*, 1992
21 CHEN, S., McLAUGHLIN, S., and MULGREW, B.: 'Complex-valued radial basis function network. Part II: application to digital communications channel equalisation'. Submitted to *Signal Proc.*, 1992