
White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/81022/

Monograph:
Billings, S.A. and Hong, X. (1997) Dual-Orthogonal Radial Basis Function Networks for Nonlinear Time Series Prediction. Research Report. ACSE Research Report 672 . Department of Automatic Control and Systems Engineering

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



Dual-orthogonal Radial Basis Function Networks For Nonlinear Time Series Prediction

S. A. Billings and X.Hong
Department of Automatic Control and Systems Engineering,
University of Sheffield, Mappin Street, Sheffield S1 3JD

Abstract — A new structure of Radial Basis Function (RBF) neural network called the Dual-orthogonal RBF Network (DRBF) is introduced for nonlinear time series prediction. The hidden nodes of a conventional RBF network compare the *Euclidean* distance between the network input vector and the centers, and the node responses are radially symmetrical. But in time series prediction where the system input vectors are lagged system outputs, which are usually highly correlated, the *Euclidean* distance measure may not be appropriate. The DRBF network modifies the distance metric by introducing a classification function which is based on the estimation data set. Training the DRBF networks consists of two stages. Learning the classification related basis functions and the important input nodes, followed by selecting the regressors and learning the weights of the hidden nodes. In both cases, a forward Orthogonal Least Squares (OLS) selection procedure is applied, initially to select the important input nodes and then to select the important centers. Simulation results of single step and multi-step ahead predictions over a test data set are included to demonstrate the effectiveness of the new approach.

Keywords — time series, OLS, radial basis function, classification function

Research Report No. 672

March 1997

200391383

1 Introduction

The Radial Basis Function (RBF) model was traditionally used for strict interpolation in multi-dimensional space (Powell, 1985). More recently, RBF neural networks have been employed in non-linear systems identification and time series prediction (S. Chen *et al*, 1991, M. Casdagli, 1989, E. S. Chng *et al*, 1996). These networks approximate an unknown function by locally constructing receptive fields around a set of centers. The centers are assumed to sample the data set and to reflect the distribution of the data, but the set of candidate centers can be very large. In practice, a network with a finite basis selected from the data set is usually adopted. With a predetermined number of centers, the centers are either randomly selected from the data (Broomhead and Lowe, 1988), or determined using a k-means clustering or a related technique (Moody and Darken, 1989). Usually the network weights are learnt at a later stage using a least squares based method. Alternatively, the learning problem can be reformulated as a subset model selection problem and a forward orthogonal least squares procedure can be used to identify appropriate radial basis function centers from the network training data and to adjust the network weights (S. Chen *et al*, 1989).

The present study focuses on the problem of time series prediction using RBF neural networks. Throughout, the measure of model quality will be interpreted as the expected fit and prediction over future data. A model with good approximation properties and fewer parameters will also be prefered. It is important to realize that the properties of the basis function plays an important role in achieving these aims. If the form of the basis functions is preselected, then the trained RBF will be closely related to the clustering quality of the training data towards the centers. Classical clustering is a process of partitioning and constructing homogeneous data sets without prior knowledge of the data distribution. The data are therefore partitioned into groups according to the similarity between them. The criterion used to compare the similarity is a distance concept, and if the distances are small enough the data are classified as the same kind. An analogy between RBF basis functions and classical clustering is obvious, because the strength of a node response is determined by the distance between the network input vector and the corresponding center. Usually, the measure of distance used in RBF neural networks is the *Euclidean* distance, but this is only strictly appropriate when the components of the data are uncorrelated (Kleinbaum, Kupper and Muller, 1987, W. R. Klecka, 1980). In the application of time series prediction this observation is important because the lagged system outputs, which are usually highly correlated, form the system input vector. However, if the real similarity between the input and the center cannot be sensed, the clustering quality will be affected, and the resulting basis function will not be effective. The problem can also be viewed as how to abstract the information contained in the input data set most efficiently. If each regressor can not be

formed efficiently, then more regressors may be needed than necessary and this can produce a deterioration in the expected fit to future data.

In this paper, a new RBF neural network called a Dual-orthogonal Radial Basis Function (DRBF) time series predictor is proposed. The main contribution is that a new distance metric is adopted which is based on a classification function of the set of input vectors. It is shown that the importance of each input node can be different based on the new distance metric and the training of the network can be configured as a two stage procedure or a dual orthogonal least squares (OLS) procedure (Billings and Chen, 1989, Billings, *et al*, 1988, 1889). The first stage involves learning the new classification related basis functions and selecting the important input nodes, followed by a second stage selecting the important centers or regressors and learning the weights of the hidden nodes, with both stages being based on the forward orthogonal least squares procedure. By discarding redundant input nodes, an appropriate model structure can be determined (Billings, *et al*, 1992). The effectiveness of the new approach is illustrated by simulation results.

2 Dual-orthogonal RBF Neural Network

2.1 Problem formulation

Conventional RBF networks approximate an unknown function by locally constructing receptive fields around a set of centers. The centers are assumed to sample the data set and reflect the distribution of the data. Each center is compared with the network input vector and the corresponding node is activated if the distance between the network input vector and the center is small enough. A distance is a measure for comparing the similarity of data groups. Each node produces a radially symmetrical response if a *Euclidean* distance measure is used.

A RBF neural network can be formulated as

$$y(t) = \sum_{i=1}^{M} p_i(t)\theta_i + \xi(t) \tag{1}$$

where $t = 1, 2, 3, \cdots, N$, and N is the sample size of the estimation set, and $\xi(t)$ are the residuals. The regressors take the form

$$p_i(t) = \Phi(v_i(t), \beta_i) \tag{2}$$

$$v_i(t) = \|\mathbf{x}(t) - \mathbf{c}_i\| \tag{3}$$

$$\mathbf{x}(t) = [y(t-1), \cdots, y(t-n_y)]^T \tag{4}$$

$\| \bullet \|$ denotes the Euclidean norm, β_i are some positive scalars called widths, $\Phi(v_i(t), \beta_i)$ is a function from $\Re^+ \rightarrow \Re$, and $\mathbf{c}_i \in \Re^{n_y}, 1 \leq i \leq M$ are the RBF centers. The distance between the input vector $\mathbf{x}(t)$ and the centers $\mathbf{c}_i$ is denoted by $v_i(t)$. The thin-plate-spline function

$$\Phi(v) = v^2 \log v \qquad (v \geq 0) \tag{5}$$

will be used in the present study, but other choices of RBF can easily be adopted. One disadvantage of the above formulation for time series applications is that the *Euclidean* distance measure is not always appropriate for measuring the closeness between the input vector $\mathbf{x}(t)$ and the center. One reason is that the input vector $\mathbf{x}(t)$ is itself highly autocorrelated. Another reason arises due to the node response which is radially symmetrical whereas the data may be distributed differently in each dimension. Chakravarthy and Ghosh proposed that an elliptic basis function can be used which has a different width in each dimension (1996), and they suggested a gradient descent method for learning the parameters. Unfortunately this procedure destroys the linear-in-the-parameters structure and negates the quick learning advantage of RBF neural networks.

Another disadvantage of the conventional RBF neural network time series predictor lies in the selection of input nodes when the minimum lag n_y is a large number. If all the lagged outputs up to n_y are selected as input nodes, the network will be unnecessarily complex and the performance might deteriorate due to irrelevant inputs and an oversized structure (Billings, *et al*, 1992). An oversized network tends to fit to the noise in the training data set and does not generalise well. Thus, in the case when n_y can be large, it is necessary to include some preprocessing procedure for input node selection. One solution to this problem is to use a mutual information criterion (Zheng and Billings, 1996).

The Dual-orthogonal Radial Basis Function (DRBF) neural network is proposed in the present study to overcome most of the above limitations. A new distance measure is used in order to achieve improved clustering, and a dual orthogonal least squares estimator is used first to determine the significant lags and then to select the most appropriate regressors which make up the RBF network. Simulation results are included to demonstrate the application of the new algorithm.

2.2 New distance metric

In the DRBF neural network time series predictor a new distance metric is adopted, which is based on a classification function of the set of input vectors. The basic idea arises from discriminant analysis. In discriminant analysis a classification function which is a linear combination of variables in the group is usually derived. A classification function value is then used to determine if any new vector falls within the group (Kleinbaum, Kupper and Muller, 1987, W. R. Klecka, 1980). The classification function realises a mapping from a multidimensional vector to a scalar, and carrys information about the distribution of the data set and the dimensionality of the discriminant space. The classification function value of a vector indicates the similarity between the vector and the distribution of the data set. This suggests that a similar approach could be applied as a new distance metric in RBF networks. Based on this idea, a classification function is constructed according to the set of input vectors in the estimation set. The effect of constructing the classification function is to find a classification hyperplane in the multi-dimensional input data space, where the mean squared distance from the data to the hyperplane is minimised. The classification function of the input vector $\mathbf{x}(t)$ takes the form

$$d[\mathbf{x}(t)] = \mathbf{x}(t)^T \mathbf{a} \tag{6}$$

where $\mathbf{a} = [a_1, a_2, \cdots, a_{n_y}]^T \in \Re^{n_y}$ is the weight vector of the classification function. The classification hyperplane is given by the classification function

$$d = \mathbf{x}^T \mathbf{a} \tag{7}$$

where d is a constant, and $\mathbf{x} \in \Re^{n_y}$ represents any vector on the hyperplane. This means that all data falling on the hyperplane have the same classification function value d. The geometrical interpretation of the classification function is illustrated in Fig.1, where $\mathbf{x}(t_0)$ and $\mathbf{x}$ are two vectors on the hyperplane, and $\mathbf{n}$ is the unit normal vector.

The hyperplane given by Eq.(7) can also be expressed as

$$[\mathbf{x} - \mathbf{x}(t_0)]^T \mathbf{n} = 0 \tag{8}$$

and it is clear that

$$\mathbf{n} = \frac{\mathbf{a}}{\|\mathbf{a}\|} \tag{9}$$

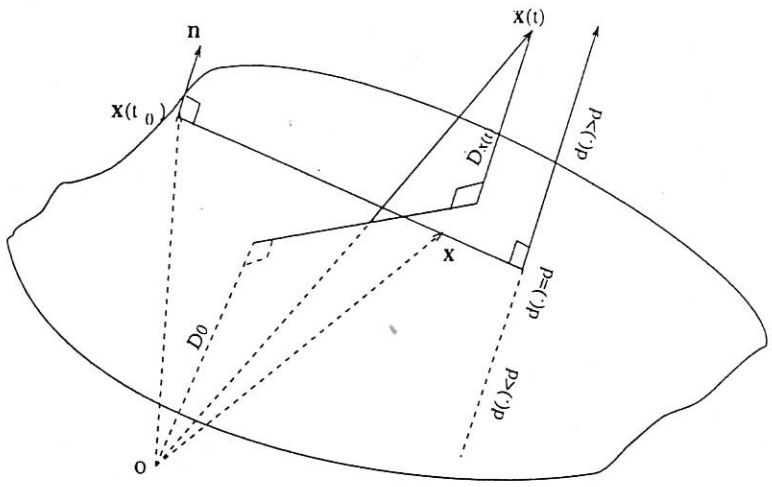


Figure 1: The geometry of the classification function of the input data set

Substituting Eq.(9) into Eq.(7) and comparing with Eq.(8), yields

$$\mathbf{x}(t_0)^T\mathbf{n} = \frac{d}{\|\mathbf{a}\|} \tag{10}$$

The distance from the origin to the hyperplane, from Fig.1, is given by

$$D_0 = \mathbf{x}(t_0)^T\mathbf{n} \tag{11}$$

Consider the input data vector $\mathbf{x}(t)$ and denote the distance from the hyperplane as $D_{\mathbf{x}(t)}$. From Fig.1, it is seen that

$$\mathbf{x}(t)^T\mathbf{n} = D_0 + D_{\mathbf{x}(t)} \tag{12}$$

Substituting Eq's.(9), (10) and (11) into Eq.(12), yields

$$\begin{aligned} D_{\mathbf{x}(t)} &= \frac{\mathbf{x}(t)^T\mathbf{a} - d}{\|\mathbf{a}\|} \\ &= \frac{d[\mathbf{x}(t)] - d}{\|\mathbf{a}\|} \end{aligned} \tag{13}$$

Suppose that two autocorrelated vectors $\mathbf{x}(t_1)$ and $\mathbf{x}(t_2)$ are formed from the same data set which is used to compute the classification function. The classification function carrys information about the distribution of the data set. How an input vector conforms to the data distribution is reflected in the deviation of the corresponding classification function value from the constant d, that is, the distance of the vector to the classification hyperplane. Therefore, the distance

between the two vectors $\mathbf{x}(t_1)$ and $\mathbf{x}(t_2)$ can be defined as the difference of their respective distances to the classification hyperplane $\|D_{\mathbf{x}(t_1)} - D_{\mathbf{x}(t_2)}\|$. Obviously the classification function can be defined to enable the norm of the weight vector to be a unit vector, that is, $\|\mathbf{a}\| = 1$. In that case, using Eq.(13), yields

$$\|D_{\mathbf{x}(t_1)} - D_{\mathbf{x}(t_2)}\| = \|d[\mathbf{x}(t_1)] - d[\mathbf{x}(t_2)]\| \tag{14}$$

The distance between the two vectors equals the absolute difference of their corresponding classification function values.

2.3 Dual-orthogonal RBF (DRBF) neural network time series predictor

The Dual-orthorgonal RBF network consists of a two stage sequential learning process based on the forward orthogonal least squares algorithm, learning the new classification related basis functions and selecting the important input nodes, followed by selecting the important centers or regressors and learning the weights of the hidden nodes. The topology of the algorithm is illustrated in Fig.2.

The first stage

The learning of the classification function can be realised using least squares, where the mean squared distance from all the input vectors in the estimation data set to the classification hyperplane

$$\frac{1}{N - n_y} \sum_{t=n_y+1}^{N} D^2_{\mathbf{x}(t)}$$

is minimised. In the present study the forward orthogonal least squares algorithm will be applied to learn the new classification related basis functions and to select the appropriate lags. It will be shown that the importance of each input node can be determined based on a metric called the error reduction ratio(**err**) (S. Chen, *et al*, 1989) , which provides a measure of the energy distribution due to each input node towards the classification function d.

Using Eq.(13), yields

$$\frac{1}{N - n_y} \sum_{t=n_y+1}^{N} D^2_{\mathbf{x}(t)} = \frac{1}{N - n_y} \sum_{t=n_y+1}^{N} \frac{[\mathbf{x}(t)^T\mathbf{a} - d]^2}{\|a\|^2} \tag{15}$$

The weight vector $\mathbf{a}$ and classification function value d can be determined from Eq.(15) as

follows. Consider fitting a linear model

$$\mathbf{d} = \mathbf{X}\mathbf{a} + \mathbf{e} \tag{16}$$

which can be written in matrix form as

$$\begin{vmatrix} d \\ d \\ \vdots \\ d \\ d \end{vmatrix} = \begin{vmatrix} y(n_y) & y(n_y-1) & \cdots & y(1) \\ y(n_y+1) & y(n_y) & \cdots & y(2) \\ \vdots & \vdots & \cdots & \vdots \\ y(N-2) & y(N-3) & \cdots & y(N-n_y-1) \\ y(N-1) & y(N-2) & \cdots & y(N-n_y) \end{vmatrix} \begin{vmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n_y-1} \\ a_{n_y} \end{vmatrix} + \begin{vmatrix} e(n_y+1) \\ e(n_y+2) \\ \vdots \\ e(N-1) \\ e(N) \end{vmatrix} \tag{17}$$

where $\mathbf{d}^T = [d, \cdots, d] \in \Re^{N-n_y}$ denotes the classification function value vector, and the value of d is at first arbitrarily assigned as a nonnegative constant. The matrix $\mathbf{X} \in \Re^{(N-n_y)\times n_y}$ denotes the regression matrix, and $\mathbf{e}^T = [e(n_y+1), \cdots, e(N)]$ is the corresponding residual vector.

The regression matrix $\mathbf{X}$ can also be written both in terms of the $(N-n_y)$ row vectors, i.e. the input vectors as

$$\mathbf{X} = [\mathbf{x}(n_y+1)\ \ \mathbf{x}(n_y+2)\cdots\mathbf{x}(N)]^T \tag{18}$$

and in terms of the n_y column vectors

$$\mathbf{X} = [\mathbf{x}_1\ \ \mathbf{x}_2\cdots\mathbf{x}_{n_y}] \tag{19}$$

where

$$\begin{aligned} \mathbf{x}_j &= [y(n_y+1-j), y(n_y+2-j), \cdots, y(N-j)]^T \\ &= [x_j(n_y+1), x_j(n_y+2), \cdots, x_j(N)]^T, \quad j = 1, \cdots, n_y \end{aligned}$$

Perform an orthogonal decomposition of $\mathbf{X}$

$$\mathbf{X} = \mathbf{Z}\mathbf{R} \tag{20}$$

where $\mathbf{R}$ is an $n_y \times n_y$ unit upper triangular matrix and $\mathbf{Z} \in \Re^{(N-n_y)\times n_y}$ is an orthogonal matrix. This can also be written both in terms of the $(N-n_y)$ row vectors or input vectors as

$$\mathbf{Z} = [\mathbf{z}(n_y+1)\ \ \mathbf{z}(n_y+2)\cdots\mathbf{z}(N)]^T \tag{21}$$

where

$$\mathbf{z}(t) = [z_1(t), z_2(t), \cdots, z_{n_y}(t)], \quad t = n_y+1, \cdots, N$$

and in terms of the n_y column vectors

$$\mathbf{Z} = [\mathbf{z}_1 \ \mathbf{z}_2 \cdots \mathbf{z}_{n_y}] \tag{22}$$

where $\mathbf{z}_j = [z_j(n_y+1), z_j(n_y+2), \cdots, z_j(N)]^T, \quad j = 1, \cdots, n_y.$ which satisfies

$$\mathbf{Z}^T\mathbf{Z} = \mathbf{H} = diag\{h_1, h_2, \cdots, h_{n_y}\} \tag{23}$$

with

$$h_j = < \mathbf{z}_j, \mathbf{z}_j >, \quad j = 1, 2, \cdots, n_y \tag{24}$$

$< \bullet, \bullet >$ denotes the inner product, and $\mathbf{z}_j, \quad j = 1, \cdots, n_y.$ is a set of orthogonal bases which span the same space as that of $\mathbf{x}_j, \quad j = 1, \cdots, n_y.$

Substituting Eq.(18) and Eq.(21) into Eq.(20), yields

$$\mathbf{x}(t) = \mathbf{z}(t)\mathbf{R}, \quad t = n_y + 1, \cdots, N \tag{25}$$

Rearranging Eq.(16) using Eq.(20) yields

$$\mathbf{d} = \mathbf{X}\mathbf{a} + \mathbf{e} = (\mathbf{X}\mathbf{R}^{-1})(\mathbf{R}\mathbf{a}) + \mathbf{e} = \mathbf{Z}\mathbf{q} + \mathbf{e} \tag{26}$$

where

$$\mathbf{R}\mathbf{a} = \mathbf{q} \tag{27}$$

$$\mathbf{q} = [q_1, q_2, \cdots, q_{n_y}]^T \tag{28}$$

Pre-multiplying both sides of Eq.(26) by $\mathbf{Z}^T$, and taking the expected value, yields

$$E(\mathbf{Z}^T\mathbf{d}) = E(\mathbf{Z}^T\mathbf{Z}\mathbf{q}) + E(\mathbf{Z}^T\mathbf{e}) \tag{29}$$

Assuming that $e(t)$ is uncorrelated with the past output $\mathbf{x}(t)$ and, from Eq.(25), is in turn uncorrelated with the past $\mathbf{z}(t)$, then $E(\mathbf{Z}^T\mathbf{e}) = 0$ holds. It may be further shown from Eq.'s(23), (24) and (29) that

$$q_j = \frac{< \mathbf{z}_j, \mathbf{d} >}{< \mathbf{z}_j, \mathbf{z}_j >}, \quad j = 1, 2, \cdots, n_y \tag{30}$$

The weight vector $\mathbf{a}$ can then be obtained from Eq.(27) through backsubstitution. Set

$$\mathbf{a} = \frac{\mathbf{a}}{\|\mathbf{a}\|}$$

$$\mathbf{q} = \frac{\mathbf{q}}{\|\mathbf{a}\|}$$

$$d = \frac{d}{\|\mathbf{a}\|}$$

to enable the norm of the weight vector of the classification function to be a unit vector, i.e. $\|\mathbf{a}\| = 1$ (see section 2.2)

From Eq.(6), Eq.(25) and Eq.(27), the classification function value of the input vector $\mathbf{x}(t)$ is

$$d[\mathbf{x}(t)] = \mathbf{x}(t)^T\mathbf{a} = \mathbf{z}(t)^T\mathbf{q} \tag{31}$$

By defining

$$\mathbf{b}_i^T = \mathbf{c}_i^T\mathbf{R}^{-1} = [b_{i1}, \cdots, b_{in_y}] \tag{32}$$

and applying Eq.(6), Eq.(27) and Eq.(32), the classification function value of the center vector $\mathbf{c}_i$ is

$$d(\mathbf{c}_i) = \mathbf{c}_i^T\mathbf{a} = \mathbf{b}_i^T\mathbf{q} \tag{33}$$

From Eq.(14), the distance $v_i(t)$ between the $\mathbf{x}(t)$ and $\mathbf{c}_i$ is

$$v_i(t) = \|d[\mathbf{x}(t)] - d(\mathbf{c}_i)\| = \sqrt{\sum_{j=1}^{n_y} q_j^2(z_j(t) - b_{ij})^2} \tag{34}$$

By virtue of coefficients $q_j \neq 1$ and viewing $\mathbf{b}_i$'s as the new centers of the hyper-ellipsoid, an inspection of Eq.(2) together with Eq.(34) shows that the basis function is in fact a hyper-elliptic basis function with the orthogonal basis $\mathbf{z}_j$'s as axes. This is illustrated in Fig. 2.

From Eq.(26), and taking into account the orthogonality of $\mathbf{z}_j$, $j = 1, \cdots, n_y$, the sum of squares of the classification value is

$$< \mathbf{d}, \mathbf{d} >= \sum_{j=1}^{n_y} q_j^2 < \mathbf{z}_j, \mathbf{z}_j > + < \mathbf{e}, \mathbf{e} > \tag{35}$$

Define the fraction of the increment towards the classification function value d by the basis $\mathbf{z}_j$ as the error reduction ratio (**err**) (Billings and Chen, 1989, Billings, *et al*, 1988, 1889)

$$\mathbf{err}_j = \frac{q_j^2 < \mathbf{z}_j, \mathbf{z}_j >}{< \mathbf{d}, \mathbf{d} >} \tag{36}$$

The value $\mathbf{err}_j$ is representative of the energy distribution of the j'th orthogonal basis towards the classification function value, or the projection along the orthogonal basis by the classification function value. The distribution of the overall energy of the classification function value can be viewed as hyper-elliptical due to the different projection along each basis. A reduction in

dimension may be possible by reducing some of the unimportant bases. In practice, an upper limit n_y for the input lag is made but the number of significant input nodes may be much smaller than n_y, $n_w \leq n_y$. By adopting an efficient forward selection procedure (Billings and Chen, 1989, Billings, *et al*, 1988, 1889) for the construction of the classification function, that is, by stepwise selecting a significant input node with the largest **err** subject to its orthogonality with the previous selected nodes, a procedure for input node selection is achieved. Input node selection for RBF neural networks was also studied by Zheng and Billings (1996), where a mutual information criterion was proposed. The advantage of a mutual information measure is that both linear and nonlinear correlations are taken into account. However, the classification function based approach has an advantage of being less computationally intensive. Both approaches attempt to use the prior information learnt from the data so as to avoid building an oversized network.

If n_w prominent input nodes are selected, then

$$\mathbf{q} = [q_1, q_2, \cdots, q_{n_w}]^T$$

and $\mathbf{R}$ is reduced to an $n_w \times n_w$ unit upper triangular matrix. The dimension of the new centers $\mathbf{b}_i$'s is reduced to n_w. The distance between $\mathbf{x}(t)$ and $\mathbf{c}_i$

$$v_i(t) = \|d[\mathbf{x}(t)] - d(\mathbf{c}_i)\| = \sqrt{\sum_{j=1}^{n_w} q_j^2 (z_j(t) - b_{ij})^2} \tag{37}$$

will be used together with Eq.(2) in the formulation of the DRBF network.

The second stage

The second stage in the DRBF network involves selecting M important regressors $p_i(t)$, $i = 1, \cdots, M$ in Eq.(1) from the data set and learning the weights of the hidden nodes. This will also be based on the forward regression OLS procedure (Billings and Chen, 1989, Billings, *et al*, 1988, 1889). In a previous study (S. Chen *et al*, 1989), RBF centers were selected either from all the estimation data set or from a subset of the set. Similarly, the new centers $\mathbf{b}_i$, $i = 1, \cdots, M$ in DRBF network can be selected from the estimation data or a subset of data after performing the transformation defined by Eq.(32). The forward OLS estimator can learn the weights θ_i, $i = 1, \cdots, M$ and select the most relevant regressors. The regressors can be selected using the error reduction ratio $[ERR]_i$, which is defined as the increment towards the overall output variance $E[y^2(t)]$ due to each regressor $p_i(t)$ divided by the overall output variance (see Appendix A). Note that to avoid confusion with the first stage selection procedure we have denoted the error

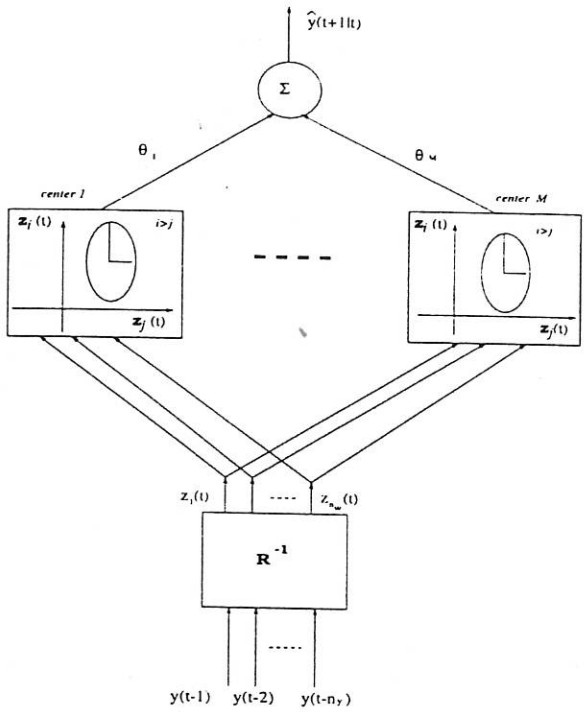


Figure 2: The topology of the DRBF neural network time series predictor

reduction ratio in the second stage as $[ERR]$.

The second stage procedure can be terminated using either a prediction risk (Barron, A. R., 1984, Liu, 1995), Akaike's information criterion (AIC) (S. Chen *et al*, 1989), or when a desired error tolerance is achieved. All these criteria are designed to produce a small one-step ahead generalization error and a model with good approximation ability and fewer parameters is preferable. The prediction risk is the expected prediction error over a test data set, and is given by

$$\sigma_{val}^2 = \sigma_\xi^2 \times (1.0 + \frac{2n_{eff}}{N}) \tag{38}$$

where the σ_ξ^2 is the variance of the prediction errors in the estimation data set, the n_{eff} denotes the effective number of parameters in the model, and N is the number of data points in the estimation set. If a desired error tolerance ρ is used to select the model, the regressors will be selected until

$$1 - \sum_{i=1}^{n_{eff}} [ERR]_i \geq \rho \tag{39}$$

fails to hold, providing a simple and effective model selection procedure. In the present study,

Akaike's information criterion

$$\mathrm{AIC} = N \log_e(\sigma_\xi^2) + 4n_{eff} \tag{40}$$

will be used. Minimizing this function with respect to the n_{eff} gives rise to the optimal number of centers M.

A model with the least one-step ahead prediction errors may not result in the least multi-step ahead prediction errors. The criterion for a good multi-step time series predictor should take the multi-step ahead prediction performance into account. Because the multi-step ahead prediction error arises from many sources, and the complexity accumulates as the number of iteration steps increases, the generalization errors of multi-step ahead predictions are difficult to analyse. Often the only sensible approach is to use cross-validation on a test data set.

Remarks:

i). It has been shown that the training of the DRBF network can be achieved using a dual-orthogonal least squares procedure. The learning of the classification function related distance metrics and input node selection is configured as the first stage, while the learning of the hidden nodes and corresponding weights is configured as the second stage. The DRBF neural network auguments the first stage of the conventional RBF training, and in this stage a classification function based new distance metric is used in order to achieve an improved basis function approximation property. The idea was motivated from the consideration that the approximation property of the basis function is important to the model quality. That is to say, the resulting model could have a good approximation yet fewer parameters due to the fact that the basis functions have been more appropriately chosen.

ii). Although the DRBF involves a more complex training procedure, the resulting network will generally have a more concise structure because the redundant input nodes will automatically be discarded at the first stage. This is an advantage especially when n_y is high (Billings, *et al*, 1992).

iii). The DRBF neural network can be viewed as an elliptical basis function network involving extra parameters in the basis functions. In DRBF the complexity to learn the parameters in each basis function through nonlinear optimisation is avoided because the learning process is composed of two sequential stages thus preserving the advantage of the RBF linear-in-the-parameters structure and learning characteristics.

3 Simulation results

Simulation results of time series prediction using both conventional RBF and Dual-orthogonal RBF (DRBF) predictors are presented in this section. Initially full models were created by using all the data points in the estimation data set as centers $\mathbf{c}_i$ for the RBF model, and transforming these to form the new centers $\mathbf{b}_i$ for the DRBF model. Subset models were then selected from the full models using the OLS scheme. The performance of single-step and multi-step ahead preditions over a new data set were then evaluated.

Consider the Mackey-Glass equation (M. Casdagli, 1989):

$$\frac{dy(t)}{dt} = -by(t) + \frac{ay(t-\tau)}{1+y^c(t-\tau)} \tag{41}$$

where $a = 0.2$, $b = 0.1$, $c = 10$. This system was simulated with the sampling rate $T_s = 1$. Two examples are presented here with $\tau = 30$ and $\tau = 17$ respectively. In each case, a sequence of 1000 points was generated with the initial values set as $y(t) = 0.5, t = 1, \cdots, \tau$, and then a sequence of Gaussian noise $\xi(t) \sim N(0, 0.05^2)$ was added to the data. The initial data from $t = 1$ to $t = \tau$ was discarded, and the estimation data set consisted of 500 points from $t = \tau + 1$ to $t = \tau + 500$. The test data of $500 - \tau$ points ranges from $t = \tau + 501$ to $t = 1000$. The time series are ploted in Fig.3.

To make a comparison of the predictive performances between RBF and DRBF networks initially, for the RBF and for the second stage in the DRBF networks, the model selection procedures were terminated to fixed model sizes of 20, 40, 60 and 80 centers. Afterwards the AIC values were used as the selection criterion.

For $\tau = 30$, two types of models were estimated. One was a RBF neural network with 30 input nodes $[y_{t-1}, y_{t-2}, \cdots, y_{t-30}]$. The other was a DRBF neural network, where the maximum lag was set to be 30, but the maximum number of input nodes was set to be 5. The 5 input nodes were selected as the best subset from the 30 input nodes $[y_{t-1}, y_{t-2}, \cdots, y_{t-30}]$. At the first stage, the classification function was constructed using the least squares method and a forward regression procedure was used to sequentially choose 5 input nodes, and the coefficents $\mathbf{q}$. The input nodes selected were $[y_{t-30}, y_{t-1}, y_{t-26}, y_{t-5}, y_{t-14}]$, and

$$\mathbf{q} = [0.9365, 0.4886, -0.1428, -0.1059, 0.1305]$$

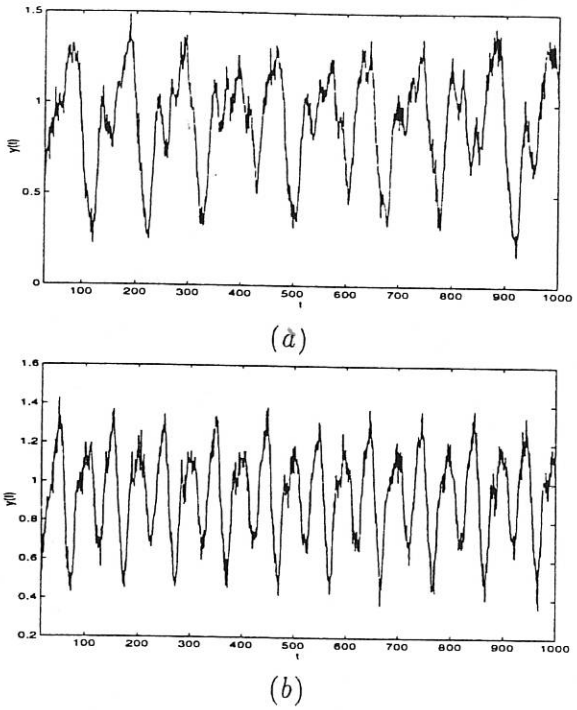


Figure 3: The Mackey-Glass time series Eq.(32). (a) $\tau = 30$ and (b) $\tau = 17$

$$\mathbf{R} = \begin{vmatrix} 1 & 0.8712 & 0.9897 & 0.8853 & 0.9259 \\ & 1 & 0.0908 & 0.9065 & 0.5515 \\ & & 1 & 0.2979 & 1.1065 \\ & & & 1 & 1.0260 \\ & & & & 1 \end{vmatrix}$$

The candidates for new centers $\mathbf{b}_i$ were then formed from the whole data in the estimation data set through the transformation defined by Eq.(32). From Eq.(2) and Eq.(37) each new candidate center forms a new candidate regressor. After all the candidate regressors were formed, the forward OLS procedure was used to select the important regressors and the procedure was terminated according to a desired model size. The model sizes of 20, 40, 60 and 80 centers were used, and one step ahead and multi-step ahead predictions were computed over the test data set from $t = 531$ to $t = 1000$. The results are shown in Fig.4's(a),(b),(c) and (d) respectively. The RBF performs better in the one step ahead prediction task, but deteriorates quickly for multi-step ahead predictions. Compared with RBF networks with the same model size, the DRBF has a much better multi-step ahead prediction performance except in the case when the prediction step is 2 and a model size of 80 centers. This suggests that the DRBF performs well

as a time series predictor.

All the DRBF network results show that the simplest model size of 20 centers produces the best results for all prediction steps. The forward OLS procedure can be terminated using Akaike's information criterion (AIC) which penalizes large size models. Minimising the AIC to select the best model for the DRBF networks achieved $M = 19$, number of centers, which is very close to 20. This suggests that the optimal number of hidden nodes for DRBF network with 5 input nodes is around 20, and also suggests that DRBF networks with a small structure can have excellent predictive properties.

For $\tau = 17$, two types of models were used. A RBF neural network with 17 input nodes $[y_{t-1}, y_{t-2}, \cdots, y_{t-17}]$, and a DRBF neural network where the maximum lag was set to be 17, but the maximum number of input nodes was set to be 5. Only the 5 most significant input nodes were selected from the 17 nodes $[y_{t-1}, y_{t-2}, \cdots, y_{t-17}]$. The classification function was learnt and a forward regression was used to sequentially choose 5 input nodes, and the coefficents $\mathbf{q}$. The resulting input nodes were $[y_{t-3}, y_{t-17}, y_{t-1}, y_{t-13}, y_{t-5}]$, and

$$\mathbf{q} = [0.8568, 0.4408, 0.7423, -0.2924, -0.2267]$$

$$\mathbf{R} = \begin{vmatrix} 1 & 0.9245 & 0.9946 & 0.9521 & 0.9951 \\ & 1 & -0.0512 & 0.7484 & 0.1266 \\ & & 1 & -0.2356 & -0.0501 \\ & & & 1 & 0.3535 \\ & & & & 1 \end{vmatrix}$$

The whole data in the estimation data set formed the candidates for new centers $\mathbf{b}_i$ through the transformation Eq.(32). Each new candidate centre, through Eq.(2) and Eq.(37), forms a new candidate regressor. The forward OLS procedure was used to select the important regressors and the procedure was terminated according to a desired model size. Model sizes of 20, 40, 60 and 80 centers were used, and one step ahead and multi-step ahead predictions were computed over the test data set from $t = 518$ to $t = 1000$. The results are shown in Fig.5's(a),(b),(c) and (d) respectively. The RBF performs better in the one step ahead prediction task for model sizes of 40, 60 and 80 centers, but deteriorates quickly when performing multi-step ahead predictions. For all sizes of network structure, the DRBF has a much better multi-step ahead prediction performance except for 2 step ahead predictions for models sizes of 40, 60, 80 centers, and for 5 step ahead predictions for model size of 80 centers.

All the DRBF network results show that the simplest model size of 20 centers produces the best results for all prediction steps. To select the best model for the DRBF networks, the forward

OLS procedure was then terminated using Akaike's information criterion (AIC). A minimum of AIC was achieved when the number of centers was $M = 23$, which is also close to 20. This suggests that the optimal number of hidden nodes for the DRBF network with 5 input nodes could be around 20, and again comfirms that DRBF networks with a small structure can have excellent predictive properties.

Two effects may contribute to the improvement of the DRBF predictive performance, an improved quality of the classification related basis functions or a more concise structure with less input nodes. To investigate these possibilities for both $\tau = 30$ and $\tau = 17$ three models with 20 centers were compared. For $\tau = 30$, a conventional RBF network with input nodes the same as the DRBF network, that is, $[y_{t-30}, y_{t-1}, y_{t-26}, y_{t-5}, y_{t-14}]$ was trained. Similarly for $\tau = 17$, a conventional RBF network with input nodes the same as the DRBF network, $[y_{t-3}, y_{t-17}, y_{t-1}, y_{t-13}, y_{t-5}]$ was trained. The performance is illustrated in Fig.6's(a)(b). It is seen that the RBF neural network with less input nodes alone performs worse than the DRBF network with the same input nodes for both $\tau = 30$ and for $\tau = 17$. For $\tau = 17$, the conventional RBF network with input nodes $[y_{t-1}, y_{t-2}, \cdots, y_{t-17}]$ diverges. This means that the DRBF networks performs better than conventionl RBF networks with both the same input nodes and the same number of centers. The results suggest the new classification based basis function plays an indispensible role in model input node reduction and input node selection.

4 Conclusions

Conventional RBF networks approximate an unknown function by locally constructing receptive fields around a set of centers. The corresponding node is activated according to the distance between the network input vector and the center, and a radially symmetrical response is produced. However, the *Euclidean* distance measure of distance is strictly precise only when the components of the data are uncorrelated. But in the application of time series prediction the system input vector consists of lagged system outputs which are usually highly correlated. The present study introduced a new structure of RBF neural network called the Dual-orthogonal RBF Network (DRBF) for nonlinear time series prediction. The DRBF modifies the distance metrics by introducing a classification function constructed from the set of input vectors in the estimation data set. It was shown that the training of the network can be implemented as a dual-orthogonal least squares procedure. In two stages, a forward regression selection procedure is applied, initially learning the classification related basis functions and the important input nodes and then selecting the regressors and learning the weights of the hidden nodes. The simulation results of single step and multi-step ahead predictions over a test data set were pre-

sented and these clearly show the effectiveness of the new approach. Although the new DRBF is proposed in the context of time series predictor, it can be applied in a wide range of signal processing applications.

5 Acknowledgements

SAB gratefully acknowledges that part of this work was supported by EPSRC. XH expresses her thanks for the award of an ORS scholarship which made this study possible.

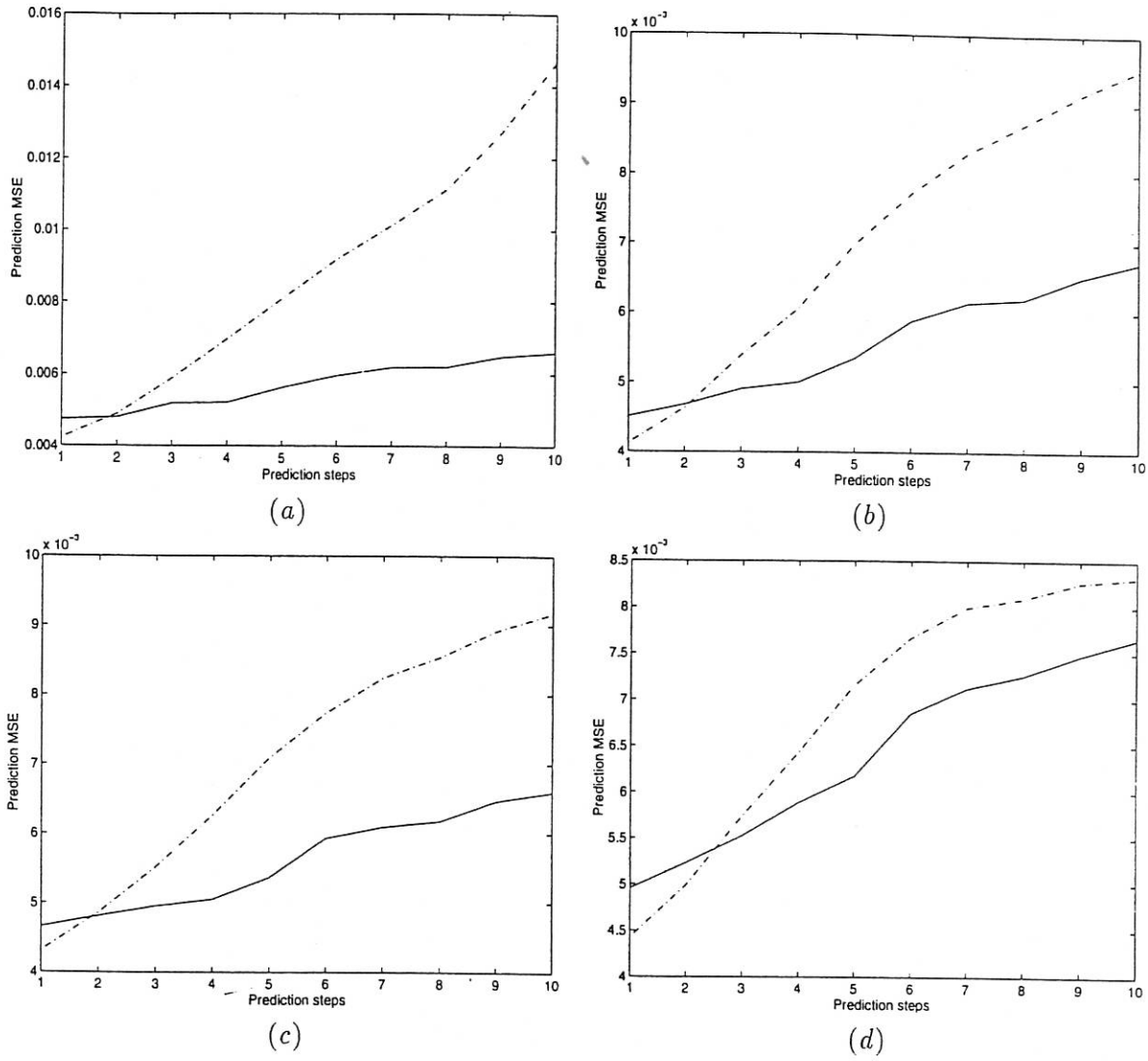


Figure 4: Multi-step ahead prediction performance when $\tau = 30$ (Solid line: DRBF with 5 input nodes $[y_{t-30}, y_{t-1}, y_{t-26}, y_{t-5}, y_{t-14}]$ and dashdot line: RBF with 30 input nodes $[y_{t-1}, y_{t-2}, \cdots, y_{t-30}]$). (a) a model size of 20 centers, (b) a model size of 40 centers, (c) a model size of 60 centers, and (d) a model size of 80 centers.

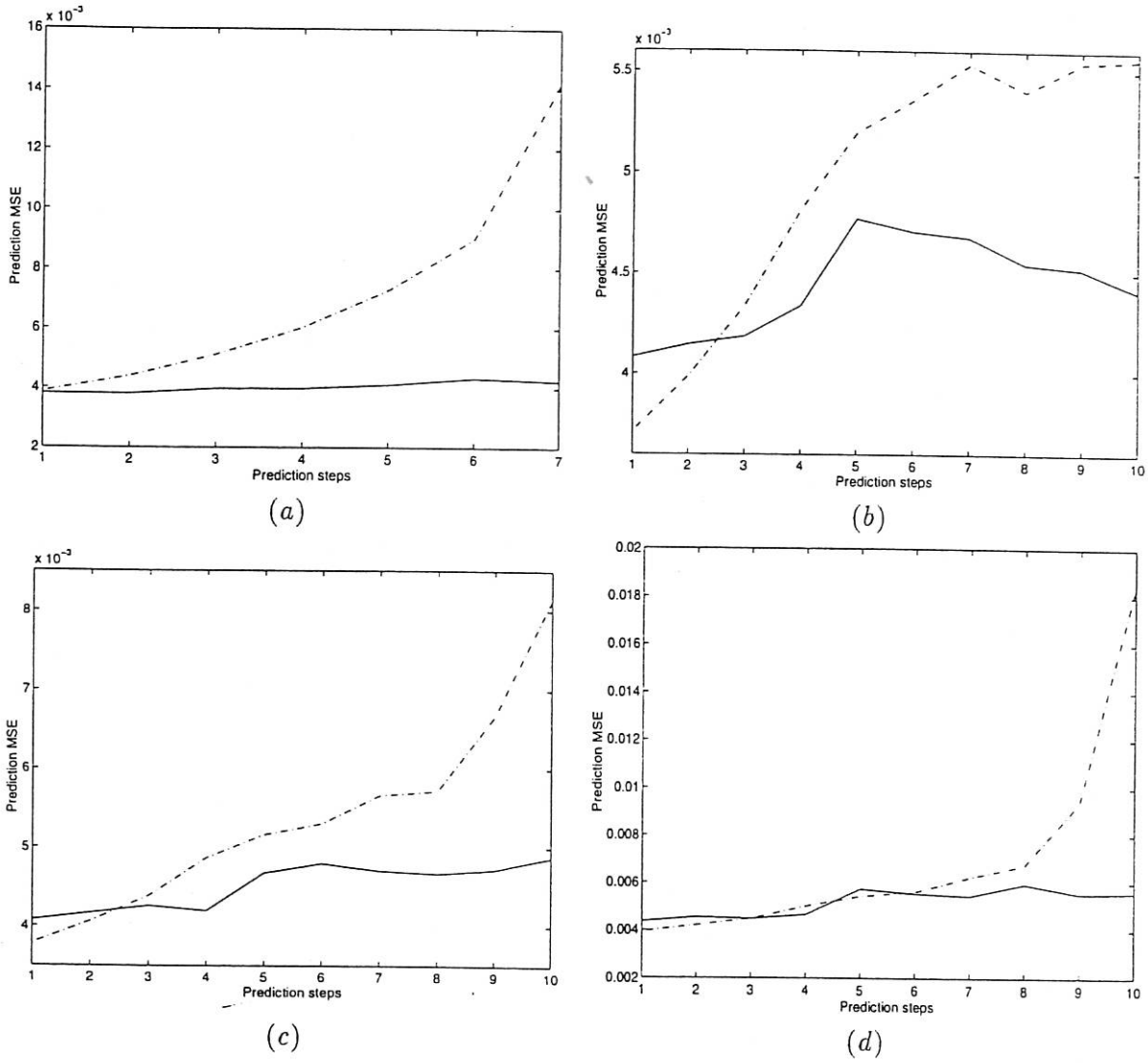
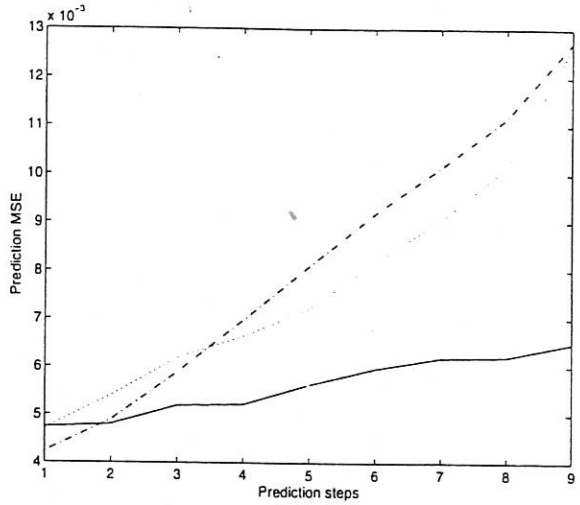
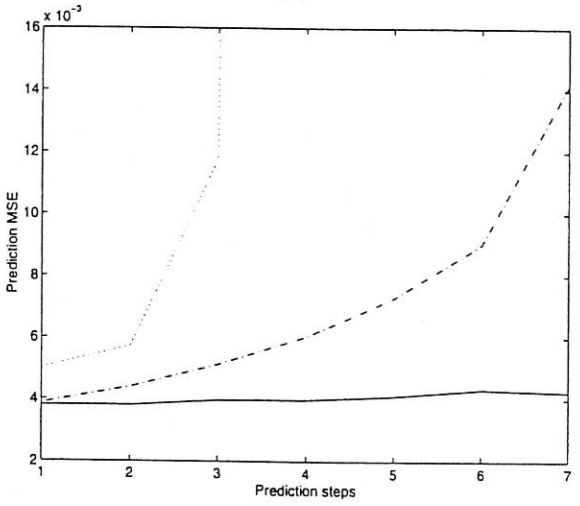


Figure 5: Multi-step ahead prediction performance when $\tau = 17$ (Solid line: DRBF with 5 input nodes $[y_{t-3}, y_{t-17}, y_{t-1}, y_{t-13}, y_{t-5}]$, Dashdot line: RBF with 17 input nodes $[y_{t-1}, y_{t-2}, \cdots, y_{t-17}]$). a) a model size of 20 centers, b) a model size of 40 centers, c) a model size of 60 centers, and d) a model size of 80 centers.



(a)



(b)

Figure 6: Multi-step ahead prediction performance of a model size of 20 centers, a)when $\tau = 30$ (Solid line: DRBF with 5 input nodes $[y_{t-30}, y_{t-1}, y_{t-26}, y_{t-5}, y_{t-14}]$, Dashdot line: RBF with 30 input nodes $[y_{t-1}, y_{t-2}, \cdots, y_{t-30}]$ and Dotted line: RBF with 5 input nodes $[y_{t-30}, y_{t-1}, y_{t-26}, y_{t-5}, y_{t-14}]$) and b) when $\tau = 17$ (Solid line: DRBF with 5 input nodes $[y_{t-3}, y_{t-17}, y_{t-1}, y_{t-13}, y_{t-5}]$, Dashdot line: RBF with 17 input nodes $[y_{t-1}, y_{t-2}, \cdots, y_{t-17}]$ and Dotted line: RBF with 5 input nodes $[y_{t-3}, y_{t-17}, y_{t-1}, y_{t-13}, y_{t-5}]$)

References

[1] Barron, A. R. (1984). Predicted squared error: A criterion for automatic model selection. In Stanley, J. F. , editor, *Self-organizing Methods in Modelling GMDH Type Algorithms*. pp87-103. Marcel Dekker, Inc. New York.

[2] Billings, S. A. and Chen, S. (1989). Extended model set, global data and threshold model identification of severely non-linear systems. *Int. J. Control*, Vol. 50. No. 5, pp1897-1923.

[3] Billings, S. A., Korenberg, M. J., and Chen, S. (1988). Identification of non-linear output-affine systems using an orthogonal least-squares algorithm. *Int. J. Systems Sci.*, Vol. 19, pp1559-1568.

[4] Billings, S. A., Chen, S. and Korenberg, M. (1989). Identification of MIMO non-linear systems using a forward-regression orthogonal estimator. *Int. J. Control*, Vol. 49, No. 6, pp2157-2189.

[5] Billings, S. A., Jamaluddin, H. B. and Chen, S. (1992). Properties of neural networks with applications to modelling dynamic systems. *Int. J. Control*, Vol. 55, No. 1, pp193-224.

[6] Broomhead, D. S. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, Vol. 2, pp321-355.

[7] Casdagli, M. (1989). Nonlinear prediction of chaotic time series. *Physica D.*, Vol. 35, pp335-356.

[8] Chakravarthy, S. V. and Ghosh, J. (1996). Scale-based clustering using the radial basis function network. *IEEE Transactions on Neural Networks*, Vol. 7, No. 5, pp1250-1261.

[9] Chen, S., Billings, S. A. and Luo, W. (1989). Orthorgonal least squares methods and their applications to non-linear system identification. *Int. J. of Control*, Vol. 50, pp1873-1896.

[10] Chng, E. S., Chen, S. and Mulgrew, B. (1996). Gradient radial basis function networks for nonlinear and nonstationary time series prediction. *IEEE Transactions on Neural Networks*, Vol. 7, No. 1, pp190-194.

[11] Klecka, W. R. (1980). *Discriminant Analysis*. Sage Publications.

[12] D. G. Kleinbaum, L. L. Kupper and K. E. Muller (1987). *Applied Regression Analysis and Other Multivariable Methods*. Second Edition, PWS-KENT Publishing Company, Boston.

[13] Liu, Y. (1995). Unbiased estimate of generalisation error and model selection in neural network. *Neural Networks*, Vol. 8, No. 2, pp311-341.

[14] Moody, J. and Darken, C. (1989). Fast-learning in networks of locally-tuned processing units. *Neural Computation*, NN-4, pp740-747.

[15] Powell, M. J. D. (1985). *IMA Conf. on Algorithms for the Approximation of Functions and Data* . RMCS Shrivenham.

[16] Zheng, G. L. and Billings, S. A. (1996). Radial basis function network configuration using mutual information and the orthogonal least squares algorithm. *Neural Networks*, Vol. 9. No. 9, pp1619-1637.

Appendix A

Eq.(1) may be written in a vector form as

$$\mathbf{y} = \mathbf{P}\Theta + \Xi \tag{42}$$

where $\mathbf{y} = [y(1), \cdots, y(N)]^T$ is the output vector, $\Xi = [\xi(1), \cdots, \xi(N)]^T$ is the residual vector, $\Theta = [\theta_1, \cdots, \theta_M]^T$, and $\mathbf{P}$ is the regression matrix

$$\mathbf{P} = \begin{vmatrix} p_1(1) & p_2(1) & \cdots & p_M(1) \\ p_1(2) & p_2(2) & \cdots & p_M(2) \\ \vdots & \vdots & \cdots & \vdots \\ p_1(N-1) & p_2(N-1) & \cdots & p_M(N-1) \\ p_1(N) & p_2(1) & \cdots & p_M(N) \end{vmatrix} \tag{43}$$

An orthogonal decomposition of $\mathbf{P}$ is given as

$$\mathbf{P} = \mathbf{WA} \tag{44}$$

where $\mathbf{A}$ is an $M \times M$ unit upper triangular matrix and $\mathbf{W}$ is an $N \times M$ matrix with orthogonal columns that satisfy

$$\mathbf{W}^T\mathbf{W} = diag\{\kappa_1, \kappa_2, \cdots, \kappa_M\} \tag{45}$$

with

$$\kappa_i = \mathbf{w}_i^T \mathbf{w}_i, \quad i = 1, \cdots, M \tag{46}$$

Rearranging Eq.(42) yields

$$\mathbf{y} = (\mathbf{P}\mathbf{A}^{-1})(\mathbf{A}\Theta) + \Xi = \mathbf{W}\Gamma + \Xi \tag{47}$$

where $\Gamma = [\gamma_1, \cdots, \gamma_M]^T$ is an auxiliary vector. Because $\xi(t)$ is uncorrelated with the past output signals, it may be shown (Chen, Billings and Luo, 1989) that

$$\gamma_i = \frac{\mathbf{w}_i^T \mathbf{y}}{\mathbf{w}_i^T \mathbf{w}_i}, \quad i = 1, \cdots, M \tag{48}$$

The number of original candidate regressors can be much larger than M, but M significant regressors can be identified using the forward OLS procedure. The principle of the method is shown below. As the orthogonality property $\mathbf{w}_i^T \mathbf{w}_j \neq 0$ for $i \neq j$ holds, Eq.(47) multiplied by itself and then the time average taken, the following equation can be derived

$$\frac{1}{N}\mathbf{y}^T\mathbf{y} = \frac{1}{N}\sum_{i=1}^{M} \gamma_i^2 \mathbf{w}_i^T \mathbf{w}_i + \frac{1}{N}\Xi^T\Xi \tag{49}$$

The output variance $E[y^2(t)] = \frac{1}{N}\mathbf{y}^T\mathbf{y}$ consists of $\frac{1}{N}\sum_{i=1}^{M} \gamma_i^2 \mathbf{w}_i^T \mathbf{w}_i$, the part of output variance expained by the regressors and $\frac{1}{N}\Xi^T\Xi$, the part of unexplained variance. The error reduction ratio $[ERR]_i$, which is defined as the increment towards the overall output variance $E[y^2(t)]$ due to each regressor $p_i(t)$ divided by the overall output variance computed through

$$[ERR]_i = \frac{\gamma_i^2 \mathbf{w}_i^T \mathbf{w}_i}{\mathbf{y}^T\mathbf{y}}, \quad i = 1, \cdots, M \tag{50}$$

The most relevant regressor can be selected forwardly according to the error reduction ratio $[ERR]_i$. The original model coefficient $\Theta = [\theta_1, \cdots, \theta_M]^T$ can be calculated from $\mathbf{A}\Theta = \Gamma$ through backsubstitution.

SHEFFIELD UNIV. LIBRARY ST. GEORGE'S