Construction of Radial Basis Function Networks with Diversified Topologies

X. Hong, S. Chen and C. J. Harris

Abstract In this review we bring together some of our recent work from the angle of the diversified RBF topologies, including three different topologies; (i) the RBF network with tunable nodes; (ii) the Box-Cox output transformation based RBF network (Box-Cox RBF); and (iii) the RBF network with boundary value constraints (BVC-RBF). We show that the modified topologies have some advantages over the conventional RBF topology for specific problems. For each modified topology, the model construction algorithms have been developed. These proposed RBF topologies are respectively aimed at enhancing the modelling capabilities of; (i)flexible basis function shaping for improved model generalisation with the minimal model;(ii) effectively handling some dynamical processes in which the model residuals exhibit heteroscedasticity; and (iii) achieving automatic constraints satisfaction so as to incorporate deterministic *prior* knowledge with ease. It is shown that it is advantageous that the linear learning algorithms, e.g. the orthogonal forward selection (OFS) algorithm based leave-one-out (LOO) criteria, are still applicable as part of the proposed algorithms.

1 Introduction

The identification of nonlinear systems using only observed finite data sets has become a mature research area over the last two decades [1]. A large class of nonlinear models and neural networks can be classified as a linear-in-the-parameters model [2, 3]. These are well structured for adaptive learning, have provable learning and convergence conditions, have the capability of parallel processing and have

X. Hong
School of Systems Engineering, University of Reading, UK, e-mail: x.hong@reading.ac.uk

S. Chen and C. J. Harris
School of Electronics and Computer Science, University of Southampton,UK, e-mail: sqc@ecs.soton.ac.uk

clear applications in many engineering applications [4, 5, 6]. In particular, the radial basis function (RBF) network is a popular type of linear-in-the-parameters model and has been widely applied in diverse fields of engineering [7, 8, 9, 10]. The ultimate objective of model construction from observed data sets should be to produce a model which captures the true underlying dynamics and predicts accurately the output for unseen data. This translates into the practical principle in nonlinear modelling of finding the smallest model that generalizes well. Sparse models are preferable in engineering applications since a models' computational complexity scales with its model complexity. Furthermore, a sparse model is easier to interpret from the angle of knowledge extraction from observed data sets.

A fundamental concept in the evaluation of model generalization capability is that of cross validation [11] which is often used to derive the information theoretic metrics, e.g. the leave-one-out (LOO) cross validation has been used to derive model selective criteria such as the Akaike information criterion (AIC) [12]. Model selective criteria can be used for predicting a model's performance on unseen data and evaluating a model's quality amongst other competitive models. The forward orthogonal least squares (OLS) algorithm is an efficient nonlinear system identification algorithm [13, 14] which selects regressors in a forward manner by virtue of their contribution to the maximization of the model error reduction ratio (ERR). In order to produce a model with good generalization capabilities, the AIC [12] is usually incorporated into the forward orthogonal least squares (OLS) algorithm to determinate the model construction process. The OLS algorithm has become a popular modelling tool in a wide range of applications [15, 16, 17, 18]. Note that most of model selective criteria are formula of approximating the LOO mean-square error (mse), and due to the approximation, have lost discriminate power in selecting terms if being used in the forward OLS algorithm. The LOO mean-square error (MSE) criterion, which directly measures the model generalization capability, has been introduced into the framework of forward OLS algorithm [19] in which the LOO mean-square error (MSE) criterion is calculated efficiently (as outlined in Section 2). An additional advantage is that the process is fully automatic, so that there is no need for the user to specify a termination criterion of the model construction process.

In this review we bring together some of our recent work from the angle of the diversified RBF topologies, including three different topologies; (i) the RBF network with tunable nodes [20]; (ii) the Box-Cox RBF [21]; and (iii) the BVC-RBF network [22]. The RBF network with tunable nodes is initially described in Section 3. Note that the parameters of the RBF network include its center vectors and variance or the covariance matrices of the basis function as well as the connecting weights from the RBF nodes to the network output. In [19] and many other RBF modelling paradigms [23, 24, 25, 26], the RBF centers are restricted to be selected from the input data sets and a common variance is employed for every RBF node. The common variance should be treated as a hyperparameter and determined via cross-validation, which may be computationally costly. The recent work [20] has introduced a construction algorithm for the tunable RBF network, where each RBF node has a tunable center and an adjustable diagonal covariance matrix. An

OFS procedure is developed to append the RBF units one by one by minimizing the LOO mse. Because the extra flexibility for the basis functions is allowed in the tunable RBF topology and all the parameters are optimized via minimizing the LOO mean-square error (MSE) criterion, the algorithm is computationally efficient and the resultant models have sparser representations with excellent generalization capability, in comparison with the existing sparse kernel modeling methods.

In Section 4, the Box-Cox RBF topology and its fast model construction algorithm [21] is described. It is a common practice to construct the RBF network in order to represent a systems' input/output mapping. For the network training the system output observations are used as the direct target of the model output. Least squares algorithm is often used as the parameter estimator, which is equivalent to the maximum likelihood estimator (MLE) under the assumption that the noise is additive and independent identically distributed (i.i.d) Gaussian with zero mean and constant variance. In practice the variance of process noise may vary with the output, e.g. the variance of noise may increase as the system output increases. For some dynamical processes in which the model residuals exhibit heteroscedasticity, e.g. with nonconstant variance or skewed distribution, or being multiplicative to the model, using conventional RBF models to construct a direct systems' input/output mapping based on the least squares estimator is no longer appropriate. The work [21] has modified RBF topology based on Box-Cox transformation. The fast identification algorithm [21] is developed based on a maximum likelihood estimator (MLE) to find the required Box-Cox transformation. It is shown the OFS-LOO algorithm is readily applicable to construct a sparse Box-Cox RBF model with good generalisation [19, 27, 21].

Finally Section 5 describes the topology of the BVC-RBF network [22]. Note that most of RBF modelling algorithms fit into the statistical learning framework, i.e. the model is determined based on the observational data only. In many modelling tasks, there are more or less prior knowledge available. Although any prior knowledge about the system should help to improve the model generalization, in general incorporating the deterministic prior knowledge into a statistically learning paradigm would make the development of modelling algorithms more difficult if not impossible. The work [22] has introduced the idea of modifying RBF topology in order to enhance its capability of automatic constraints satisfaction. We considered a special type of prior knowledge given by a type of boundary value constraints (BVC), and introduced the BVC-RBF as a new topology of RBF neural network that has the capability of satisfying the BVC automatically. The BVC-RBF network [22] is constructed and parameterized based on the given BVC. It is shown that the BVC-RBF remains as a linear-in-the-parameter structure just as the conventional RBF does. Therefore many of the existing modelling algorithms for a conventional RBF are almost directly applicable to the new BVC-RBF without added algorithmic complexity nor computational cost. Consequently the topology of the BVC-RBF effectively lends itself as a single framework in which both the deterministic prior knowledge and stochastic data are fused with ease.

2 Orthogonal forward selection (OFS) algorithm based on leave-one-out (LOO) criteria

Consider the regression problem of approximating the N pairs of training data $D_N = \{\mathbf{x}_k, y_k\}_{k=1}^N$ with a linear-in-the-parameter model defined in

$$y_k = \hat{y}_k + e_k = \sum_{i=1}^{M} w_i g_i(\mathbf{x}_k) + e_k = \mathbf{g}^T(k)\mathbf{w} + e_k \tag{1}$$

where the input $\mathbf{x}_k \in \Re^m$, the desired output $y_k \in \Re$, $\hat{y}_k$ denotes the model output, $e_k = y_k - \hat{y}_k$ is the modelling error, $g_i(\bullet)$ for $1 \le i \le M$ is a known nonlinear basis function mapping, such as RBF, polynomial or B-spline functions, and $\mathbf{g}(k) = [g_1(\mathbf{x}_k)\ g_2(\mathbf{x}_k)\ \cdots\ g_M(\mathbf{x}_k)]^T$, $\mathbf{w} = [w_1\ w_2\ \cdots\ w_M] \in \Re^M$ is the weight vector, M is the number of basis functions. By defining $\mathbf{y} = [y_1\ y_2\ \cdots\ y_N]^T$, $\mathbf{e} = [e_1\ e_2\ \cdots\ e_N]^T$, and $\mathbf{G} = [\mathbf{g}_1\ \mathbf{g}_2\ \cdots\ \mathbf{g}_M]$ with $\mathbf{g}_l = [g_l(\mathbf{x}_1)\ g_l(\mathbf{x}_2)\ \cdots\ g_l(\mathbf{x}_N)]^T$, $1 \le l \le M$. The regression model (1) over the training data set can be written in the matrix form

$$\mathbf{y} = \mathbf{G}\mathbf{w} + \mathbf{e} \tag{2}$$

Here $\mathbf{g}_l$ is the lth column of while $\mathbf{g}^T(k)$ the kth row of $\mathbf{G}$.

Let an orthogonal decomposition of $\mathbf{G}$ be $\mathbf{G} = \mathbf{P}\mathbf{A}$, where $\mathbf{A} = \{\alpha_{ij}\}$ is an $M \times M$ unit upper triangular matrix and $\mathbf{P} = [\mathbf{p}_1\ \mathbf{p}_2\ \cdots\ \mathbf{p}_M]$ is an $N \times M$ matrix with orthogonal columns that satisfy

$$\mathbf{P}^T\mathbf{P} = \text{diag}\{\kappa_1, ..., \kappa_M\} \tag{3}$$

where $\kappa_l = \mathbf{p}_l^T\mathbf{p}_l$ for $1 \le l \le M$. The regression model (2) can be alternatively expressed as

$$\mathbf{y} = \mathbf{P}\theta + \mathbf{e} \tag{4}$$

where $\theta = [\theta_1\ \theta_2\ \cdots\ \theta_M]^T$ satisfies the triangular system $\mathbf{A}\mathbf{w} = \theta$. The model output $\hat{y}_k$ can be equivalently expressed by

$$\hat{y}_k = \mathbf{p}^T(k)\theta \tag{5}$$

where $\mathbf{p}^T(k) = [p_1(\mathbf{x}_k)\ p_2(\mathbf{x}_k)\ \cdots\ p_M(\mathbf{x}_k)]$ is the kth row of $\mathbf{P}$.

Consider the modeling process that has produced the n-unit model. Let us denote the constructed n model columns as $\mathbf{P}_n = [\mathbf{p}_1, ..., \mathbf{p}_n]$, the kth model output of this n unit model identified using the entire training data set as $\hat{y}_k^{(n)} = \sum_{i=1}^n \theta_i p_i(k)$, and the corresponding kth modeling error $e_k^{(n)} = y_k - \hat{y}_k^{(n)}$.

Definition 1: The leave-one-out (LOO) mse: If we "remove" the kth data point from the traing data set and use the remaining $(N-1)$ data points to identify the n-unit model instead, the "test" error of the resulting model can be calculated on the data point removed from training. This LOO modeling error. (This corresponds to the LOO pseudo-modeling error in the context of Box-Cox RBF network (see Section

4)), denoted as $e_k^{(n,-k)}$, is given by [28]

$$e_k^{(n,-k)} = e_k^{(n)} / \eta_k^{(n)} \tag{6}$$

where $\eta_k^{(n)}$ is the LOO error weighting [28]. The LOO mse (This corresponds to the LOO pseudo-mse in the context of Box-Cox RBF network (see Section 4)) for the n-unit model is then defined by

$$J_n = \frac{1}{N} \sum_{k=1}^{N} \left(e_k^{(n,-k)} \right)^2 . \tag{7}$$

which is a measure of the model generalisation capability [28, 11].

For model (5) the computation of the LOO criterion J_n is very efficient, because $e_k^{(n)}$ and $\eta_k^{(n)}$ can be computed recursively using [19, 27]

$$e_k^{(n)} = e_k^{(n-1)} - \theta_n p_n(k) \tag{8}$$

$$\eta_k^{(n)} = \eta_k^{(n-1)} - \frac{p_n^2(k)}{\kappa_n + \nu} \tag{9}$$

where $\nu \geq 0$ is a small regularization parameter.

The orthogonal forward selection (OFS) algorithm based leave-one-out (LOO) criteria was proposed [19, 27], in which the LOO mse J_n was minimized by searching a set of candidate regressors at each forward orthogonal regression stage. It is shown [19] that J_n is concave with respect to the number of model terms, and this means that the model construction process becomes fully automatic without using additional termination criterion. Furthermore note that J_n directly measures the model generalization capability so that there is no need to use a separate validation data set. Other advantages for using LOO mse criteria are that LOO mse J_n has not lost discriminative power in selecting terms as happens with AIC, and that there is no extra tuning parameters in the model selective criterion.

3 RBF network with tunable nodes

A popular approach is to construct the RBF models with the Gaussian basis functions, in which the candidate regressors $g_i(\bullet)$ are formed using the training data set, and a *given* common variance is employed for every RBF node. In order to find a satisfactory value of the common variance, the algorithms in [19, 27] need to be repeated, e.g. via grid search based cross validation. Clearly the true cost of modeling must take into account the cost of determining all the parameters, e.g. optimizing the value the the common variance. This is because most of the complexity for many existing learning algorithms is due to the need to tune parameters that have nonlinear relationship to the system output via cross validation. Therefore a model

with less parameters that are tuned via cross validation could potentially lead to the significant reduction to the true cost of modeling.

Alternatively if the regressors $g_i(\bullet)$ are viewed as the building blocks of the RBF network, then it is intuitive to make these more flexible by relaxing the constraint that each regressor has the same shape, because this allows the model generalization capability to be maximized for a model with the smallest size. The tunable RBF network was recently introduced [20], in which each node of the network has a tunable center and an adjustable diagonal covariance matrix. Clearly the tunable RBF topology has more parameters that are nonlinear to the system output, and nonlinear optimization is necessary, leading to the additional computation costs. Note that it would be computationally prohibitive to tune a large number of extra parameters via cross validation. Significantly the OFS-LOO algorithm, the construction algorithm developed for the tunable RBF network in [20], optimizes all the associated parameters in order to achieve model generalization without cross validation. This potentially leads to considerable saving in terms of the true cost of modeling, despite the fact that more parameters that have nonlinear relationship to the system output are introduced in the tunable RBF topology.

Consider the general RBF regressor of the form [20]

$$g_i(\mathbf{x}) = K\left(\sqrt{(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)}\right) \tag{10}$$

where μ_i is the center vector of the ith RBF unit, the diagonal covariance matrix has the form $\Sigma_i = \text{diag}\{\sigma_{i,1},\cdots,\sigma_{i,m}\}$, and $K(\bullet)$ is the chosen basis or kernel function. The proposed algorithm constructs the RBF units one by one by positioning and shaping the RBF nodes while minimizing the LOO mse J_n. Specifically, at the nth stage of the constructing procedure, the nth RBF unit is determined by minimizing J_n with respect to the node's center vector μ_n and the diagonal covariance matrix Σ_n

$$\min_{\mu_n,\Sigma_n} J_n(\mu_n,\Sigma_n) \tag{11}$$

and the construction procedure is automatically terminated when $J_M \le J_{M+1}$, yielding an M-term RBF network. Intuitively the extra number of tunable parameters in each RBF node can enhance the modeling capability such that the final model size M could be much smaller than that of fixed RBF with each unit having a common variance, leading to another part of saving in computation cost, and this is often confirmed in simulation studies.

In [20], a simple yet efficient global search algorithm called the repeating weighted boosting search (RWBS) algorithm [29] was proposed to solve the task of the nonconvex optimization problem (11). The procedure is summarized here. Let $\mathbf{u}$ be the vector that contains μ_n and Σ_n. Giving the following initial conditions:

$$\left.\begin{array}{l} e_k^{(0)} = y_k \text{ and } \eta_k^{(0)} = 1, \;\; 1 \le k \le N \\ J_0 = \frac{1}{N}\mathbf{y}^T\mathbf{y} = \frac{1}{N}\sum_{k=1}^{N} y_k^2 \end{array}\right\} \tag{12}$$

Specify the RWBS algorithmic parameters, namely, the population size P_s, the number of generations in the repeated search N_G, and the number of weighted boosting search iterations M_I.

Outer loop: generations For ($l = 1; l \leq N_G; l = l + 1$) {

Generation Initialization: Initialize the population by setting $\mathbf{u}_1^{[l]} = \mathbf{u}_{best}^{[l-1]}$ and randomly generating the rest of the population members $\mathbf{u}_i^{[l]}, 2 \leq i \leq P_S$, where $\mathbf{u}_{best}^{[l-1]}$ denotes the solution found in the previous generation. If $l = 1$, $\mathbf{u}_1^{[l]}$ is also randomly chosen.

Weighted boosting search initialization: Assign the initial distribution weighting factors $\delta_i(0) = 1/P_s$, $1 \leq i \leq P_s$, for the population. Then

1) For $1 \leq i \leq P_S$, generate $\mathbf{g}_n^{i)}$ from $\mathbf{u}_i^{[l]}$, the candidates for the nth model column, and orthogonalize them

$$\alpha_{j,n}^{i)} = \mathbf{p}_j^T \mathbf{g}_n^{i)} / \mathbf{p}_j^T \mathbf{p}_j \qquad 1 \leq j < n \tag{13}$$

$$\mathbf{p}_n^{i)} = \mathbf{g}_n^{i)} - \sum_{j=1}^{n-1} \alpha_{j,n}^{i)} \mathbf{p}_j \tag{14}$$

$$\theta_n^{i)} = (\mathbf{p}_n^{i)})^T \mathbf{y} / \left((\mathbf{p}_n^{i)})^T \mathbf{p}_n^{i)} + \nu \right) \tag{15}$$

2) For $1 \leq i \leq P_s$, calculate the LOO cost for each $\mathbf{u}_i^{[l]}$

$$e_k^{(n)}(i) = e_k^{(n-1)} - p_n^{i)}(k)\theta_n^{i)}, \qquad 1 \leq k < N \tag{16}$$

$$\eta_k^{(n)}(i) = \eta_k^{(n-1)} - \left(p_n^{i)}(k) \right)^2 / \left((\mathbf{p}_n^{i)})^T \mathbf{p}_n^{i)} + \nu \right), \qquad 1 \leq k < N \tag{17}$$

$$J_n^{i)} = \frac{1}{N} \sum_{k=1}^{N} \left(\frac{e_k^{(n)}(i)}{\eta_k^{(n)}(i)} \right)^2 \tag{18}$$

where $p_n^{i)}(k)$ is the kth element of $\mathbf{p}_n^{i)}$.

Inner loop: weighted boosting search For $(t = 1; t \leq M_I; t = t + 1)$ {

Step 1: Boosting

1. Find

$$i_{best} = \arg \min_{1 \leq i \leq P_s} J_n^{i)} \tag{19}$$

$$i_{worst} = \arg \max_{1 \leq i \leq P_s} J_n^{i)} \tag{20}$$

Denote $\mathbf{u}_{best}^{[l]} = \mathbf{u}_{i_{best}}^{[l]}$ and $\mathbf{u}_{worst}^{[l]} = \mathbf{u}_{i_{worst}}^{[l]}$,

2. Normalize the cost function values

$$\vec{J_n^{i)}} = \frac{J_n^{i)}}{\sum_{j=1}^{P_s} J_n^{j)}}, \quad 1 \le i \le P_s. \tag{21}$$

3. Compute a weighting factor β_t according to

$$\xi_t = \sum_{i=1}^{P_s} \delta_i(t-1)\vec{J_n^{i)}}, \quad \beta_t = \frac{\xi_t}{1-\xi_t}. \tag{22}$$

4. Update the distribution weightings for $1 \le i \le P_s$

$$\delta_i(t) = \begin{cases} \delta_i(t-1)\beta_t^{\vec{J_n^{i)}}} & \text{for } \beta \le 1 \\ \delta_i(t-1)\beta_t^{1-\vec{J_n^{i)}}} & \text{for } \beta > 1 \end{cases} \tag{23}$$

and normalize them

$$\delta_i(t) = \frac{\delta_i(t)}{\sum_{j=1}^{P_s} \delta_j(t)}, \quad 1 \le i \le P_s. \tag{24}$$

Step 2: Parameter Updating

1. Construct the $(Ps+1)$th point using

$$\mathbf{u}_{P_s+1} = \sum_{i=1}^{P_s} \delta_i(t)\mathbf{u}_i^{[l]} \tag{25}$$

2. Construct the $(Ps+2)$th point using

$$\mathbf{u}_{P_s+2} = \mathbf{u}_{best}^{[l)]} + \left(\mathbf{u}_{best}^{[l]} - \mathbf{u}_{P_s+1}\right) \tag{26}$$

3. Calculate $\mathbf{g}_n^{P_s+1)}$ and $\mathbf{g}_n^{P_s+2)}$ from $\mathbf{u}_{P_s+1}$ and $\mathbf{u}_{P_s+2}$, orthogonalize these two candidate model columns (as in (13)-(15), and compute the corresponding LOO cost function values $J_n^{i)}$, $i = P_s+1, P_s+2$ (as in (16)-(18)). Then find

$$i_* = \arg \min_{i=P_s+1,P_s+2} J_n^{i)}. \tag{27}$$

$(\mathbf{u}_{i_*}, J_n^{i_*)})$, which replace $(\mathbf{u}_{worst}^{[l]}, J_n^{i_{worst})})$ in the population.

} **End of inner loop** This solution found in the lth generation is $\mathbf{u} = \mathbf{u}_{best}^{[l]}$.

} **End of outer loop** This yields the solution $\mathbf{u} = \mathbf{u}_{best}^{(N_G)}$, i.e., μ_n, Σ_n of the nth RBF node, the nth model column $\mathbf{g}_n$, the orthogonalization coefficients $\alpha_{j,n}$, $1 \le j < n$, the corresponding orthogonal model column $\mathbf{p}_n$, and the weight θ_n, as well as the

n-term modelling errors $e_k^{(n)}$ and the associated LOO modelling error weightings $\eta_k^{(n)}$ for $1 \leq k \leq N$.

Note that the algorithmic parameters P_s, N_G and M_I are found empirically, and some general rules are discussed in [29].

Table 1 Comparative results for Boston housing data set [20]; The results were averaged over 100 realisations and quoted as the mean $\pm$ standard deviation

algorithm	RBF type	model size	training MSE	test MSE
ε-SVM	fixed	243.2 $\pm$ 5.3	6.7986 $\pm$ 0.4444	23.1750 $\pm$ 9.0459
LROLS-LOO	fixed	58.6 $\pm$ 11.3	12.9690 $\pm$ 2.6628	17.4157 $\pm$ 4.6670
OFS-LOO	**tunable**	**34.6 $\pm$ 8.4**	**10.0997$\pm$ 3.4047**	**14.0745 $\pm$ 3.6178**

Example 1 [20]: Boston Housing Data

This benchmark data set is available at the University of California, Irvine (UCI) repository [30]. The data set comprises 506 data points with 14 variables. The task of predicting the median house value was performed from the remaining 13 attributes. 456 data points were randomly selected from the data set for training and the remaining 50 data points were used as a test data set. The experiment was repeated and the average results over 100 repetitions were given [20]. Three construction algorithms, the ε-SVM [24], the LROLS-LOO [27] and the OFS-LOO [20] were compared, and the Gaussian basis function was used to form the basis function. Table 1 summarize the results for three algorithms over the 100 realizations. The experiments parameters setting can be found [20]. Discussions on the computational complexity comparison can be found [20], in which it is argued that the OFS-LOO algorithm is highly competitive in terms of the real cost of modeling.

4 Box-Cox output transformation based RBF network (Box-Cox RBF)

In this section we review a modified RBF topology [21], in which a conventional RBF network was introduced to represent the Box-Cox transformed system output, rather than the actual system output. One of the motivations of [21] is to provide a computationally efficient approach to construct a sparse Box-Cox RBF network for some systems with the heteroscedasticity. Provided that there is a suitable Box-Cox transformation, the pseudo model errors that are the model residuals between the transformed system output and model output can be stabilized so that it follows a normal assumption [32, 33, 34]. Provided that the optimal parameter λ used in Box-Cox transformation, the number and location of candidate RBF centers are known, various orthogonal forward regression (OFR) algorithms [35, 13, 36, 37] are readily

applicable to model structure selection and parameter estimation for the proposed Box-Cox transformed based RBF network.

Consider the problem of approximating the N pairs of training data $\{\mathbf{x}_k, y_k\}_{k=1}^N$, where y_k is positive system output. If the original system output is not negative, then $y_k + c \rightarrow y_k > 0$ is used where c is a chosen positive number just large enough to enable y_k to be positive. The Box-Cox transformation is a transformation to the system output given by

$$h(y,\lambda) = \begin{cases} (y^\lambda - 1)/(\lambda \tilde{y}^{\lambda-1}) & \text{if } \lambda \neq 0 \\ \tilde{y}\log(y) & \text{if } \lambda = 0 \end{cases} \tag{28}$$

where $\tilde{y} = \sqrt[N]{\prod_{k=1}^N y_k}$, the geometric mean of the output observations.

The Box-Cox transformation based RBF networks (Box-Cox RBF) [21] is illustrated in Figure 1. For a given λ, the Box-Cox RBF network with a single output can be formulated as

$$h(y_k,\lambda) = \hat{h}_k + e_k = \sum_{i=1}^M w_i g_i(\mathbf{x}_k) + e_k = \mathbf{g}^T(k)\mathbf{w} + e_k. \tag{29}$$

Here $e_k = h(y_k,\lambda) - \hat{h}_k$ is referred as the pseudo error. (In order to reduce the number of notations, e_k is still used here in spite of the difference between (1) and (29). This allows that the algorithm in Section 2 to be shared for the different topologies.) The regressors $g_i(\mathbf{x}_k)$ are formed using some known RBF functions (see Section 2). Note that

$$\lim_{\lambda\rightarrow 0} h(y,\lambda) = \lim_{\lambda\rightarrow 0}[(y^\lambda - 1)/(\lambda\tilde{y}^{\lambda-1})] = \tilde{y}\log(y) \tag{30}$$

and the inverse of Box-Cox transformation upon $\hat{h}_k$ for given $\lambda \neq 0$ is

$$\hat{y}_k = h^{-1}(\hat{h}_k) = \sqrt[\lambda]{1 + \lambda\tilde{y}^{\lambda-1}\hat{h}_k}. \tag{31}$$

If $\lambda = 0$, then $\hat{y}_k = \exp[\hat{h}_k/\tilde{y}]$.

Supposing all the training data were used as centres to construct the candidate regressors $g_i(\mathbf{x}_k)$, (29) can be rewritten in a vector form as

$$\mathbf{e} = \mathbf{h}(\lambda) - \mathbf{G}\mathbf{w} \tag{32}$$

in which $\mathbf{h}(\lambda) = [h(y_1,\lambda),...,h(y_N,\lambda)]^T \in \Re^N$ is transformed system outputs' vector. $\mathbf{e} = [e_1,...,e_N]^T \in \Re^N$ is the pseudo-error vector.

The parameter estimation for the Box-Cox RBF network is to adapt model parameters based on the fundamentals of feedback learning and weight adjustment found in conventional parametric optimization so that the model produces a good approximation to the true system, e.g. to minimize pseudo errors as shown Figure 1. Compared to the conventional RBF neural networks, there is an additional task of determining the required Box-Cox transformation, i.e. finding the optimal λ. The method introduced in [21] is based on the underlying assumption that there exists a

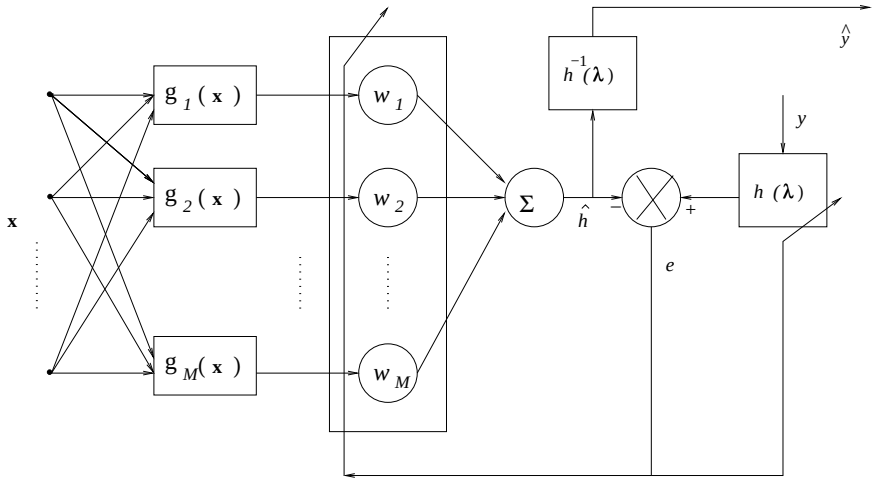

Fig. 1 The Topology of the Box-Cox RBF network.

suitable Box-Cox RBF network such that the resultant model residuals, or pseudo errors e_k, become Gaussian with zero mean and constant variance σ^2 [32, 33]. This leads to a fast algorithm for determining λ based on MLE, as described below.

Because the parameter estimators for linear-in-the-parameters models rely on the well-conditioning of the model, yet using the full data set to form RBF regressors usually results in ill-conditioning. Initially we consider the singular value decomposition (SVD) of matrix $\mathbf{G}$ with orthonormal matrix $\mathbf{Q}_N \in \Re^{N\times N}$, such that

$$\mathbf{Q}_N^T \mathbf{G} \mathbf{Q}_N = \Sigma_N = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_L, 0, ..., 0) \in \Re^{N\times N} \tag{33}$$

where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_L$ are L nonnegative singular values of $\mathbf{G}$. Denote $\Sigma = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_L) \in \Re^{L\times L}$, and the submatrix of the first L columns of $\mathbf{Q}_N$ as $\mathbf{Q} = [\mathbf{q}_1, ..., \mathbf{q}_L] \in \Re^{N\times L}$, and $\mathbf{q}_k = [q_k(\mathbf{x}_1), ..., q_k(\mathbf{x}_N)]^T$. (32) becomes

$$\mathbf{e}(\vartheta_\lambda) = \mathbf{h}(\lambda) - \mathbf{Q}\Sigma\mathbf{Q}^T\mathbf{w} = \mathbf{h}(\lambda) - \mathbf{Q}\vartheta \tag{34}$$

in which $\vartheta = [\vartheta_1, ..., \vartheta_L]^T \in \Re^L$, ϑ_λ is defined as $\vartheta_\lambda = [\vartheta^T, \lambda]^T$. Denote $\mathbf{e}(\vartheta_\lambda)$ $= [e_1(\vartheta_\lambda), \cdots, e_N(\vartheta_\lambda)]^T$.

Consider the MLE for ϑ_λ under the assumption that the pseudo errors, e_k, is Gaussian with zero mean and constant variance σ^2 [32, 33]. Specifically, suppose that there exists a suitable Box-Cox transformation given by (28) such that the transformed output observations $h(y, \lambda)$ satisfy the normal assumption with the probability density function [32, 33] in relation to the original observations y_k, $k = 1, \cdots, N$ proportional to the following function

$$\frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{e_k^2(\vartheta_\lambda)}{2\sigma^2}\right\} \mathscr{J}(k, \lambda) \tag{35}$$

where

$$e_k(\vartheta_\lambda) = h(y_k,\lambda) - \sum_{i=1}^{L} q_i(\mathbf{x}_k)\vartheta_i \tag{36}$$

and $\mathscr{J}(k,\lambda)$ is the Jacobian of the Box-Cox transformation given by [32, 33]

$$\mathscr{J}(k,\lambda) = \frac{\partial h(y,\lambda)}{\partial y}|_{y=y_k} = \left[\frac{y_k}{\tilde{y}}\right]^{\lambda-1}. \tag{37}$$

Define a loglikelihood function as follows [32, 33]

$$L(\theta_\lambda) = -N\log(\sigma) - \sum_{k=1}^{N}\frac{e_k^2(\vartheta_\lambda)}{2\sigma^2} \tag{38}$$

in which (37) is applied. Hence MLE of ϑ_λ can be solved by nonlinear least squares algorithm such as Gauss-Newton algorithm to minimize the mean squares pseudo errors $\sum_{k=1}^{N} e_k^2(\vartheta_\lambda)$.

Consider the minimization of $\sum_{k=1}^{N} e_k^2(\vartheta_\lambda)$ with respect to ϑ_λ by using Gauss-Newton algorithm [38]. Denote an iteration step variable l by a superscript (l). With an initial $\vartheta_\lambda^{(0)}$, the iteration formula is given by

$$\vartheta_\lambda^{(l)} = \vartheta_\lambda^{(l-1)} + \alpha\{[\underline{\mathbf{Q}}^{(l)}]^T\underline{\mathbf{Q}}^{(l)}\}^{-1}[\underline{\mathbf{Q}}^{(l)}]^T\mathbf{e}(\vartheta_\lambda^{(l-1)}) \tag{39}$$

where $\alpha > 0$ is a small positive step size. $\underline{\mathbf{Q}}$ (the superscript (l) is removed here for notational simplicity) is the Jacobian matrix of $e_k(\vartheta_\lambda)$ with respect to ϑ_λ, given by

$$\underline{\mathbf{Q}} = \begin{bmatrix} \frac{\partial}{\partial\vartheta_1}e_1(\vartheta_\lambda) & \frac{\partial}{\partial\vartheta_2}e_1(\vartheta_\lambda) & \cdots & \frac{\partial}{\partial\vartheta_L}e_1(\vartheta_\lambda) & \frac{\partial}{\partial\lambda}e_1(\vartheta_\lambda) \\ \frac{\partial}{\partial\theta_1}e_2(\vartheta_\lambda) & \frac{\partial}{\partial\vartheta_2}e_2(\theta_\lambda) & \cdots & \frac{\partial}{\partial\vartheta_L}e_2(\vartheta) & \frac{\partial}{\partial\lambda}e_2(\vartheta_\lambda) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial}{\partial\vartheta_1}e(\vartheta_\lambda,N) & \frac{\partial}{\partial\vartheta_2}e_N(\vartheta_\lambda) & \cdots & \frac{\partial}{\partial\vartheta_L}e_N(\vartheta_\lambda) & \frac{\partial}{\partial\lambda}e_N(\vartheta_\lambda) \end{bmatrix} \in \Re^{N\times(L+1)} \tag{40}$$

or equivalently

$$\underline{\mathbf{Q}} = [-\mathbf{Q}, \nabla_\lambda h(k,\lambda)] \tag{41}$$

where

$$\nabla_\lambda h(k,\lambda) = [\frac{\partial}{\partial\lambda}h(y_1,\lambda),\ \frac{\partial}{\partial\lambda}h(y_2,\lambda)...,\ \frac{\partial}{\partial\lambda}h(y_N,\lambda)]^T \in \Re^N, \tag{42}$$

in which,

$$\frac{\partial}{\partial\lambda}h(y_k,\lambda) = \frac{\lambda y_k^\lambda\log[y_k] - (y_k^\lambda - 1)(1+\lambda\log\tilde{y})}{\lambda^2\tilde{y}^{\lambda-1}} \tag{43}$$

as derived from (28). Hence, due to the fact that $\mathbf{Q}$ is orthonormal,

$$\underline{\mathbf{Q}}^T\underline{\mathbf{Q}} = \begin{bmatrix} \mathbf{I} & \mathbf{b}(\lambda) \\ \mathbf{b}^T(\lambda) & d(\lambda) \end{bmatrix} \tag{44}$$

in which $\mathbf{I}$ is an unit matrix.

$$\begin{aligned}\mathbf{b}(\lambda) &= -\mathbf{Q}^T\nabla_\lambda h(t,\lambda) = -[\mathbf{q}_1^T\nabla_\lambda h(t,\lambda),\ ...,\ \mathbf{q}_L^T\nabla_\lambda h(t,\lambda))]^T \\ d(\lambda) &= \{\nabla_\lambda h(\lambda)\}^T\nabla_\lambda h(\lambda)\end{aligned} \tag{45}$$

At the lth iteration step with previous parameter estimator as $\vartheta_\lambda^{(l-1)} = [\vartheta^{(l-1)}, \lambda^{(l-1)}]^T$. Denote $\underline{\mathbf{K}}^{(l)} = \left\{[\underline{\mathbf{Q}}^{(l)}]^T\underline{\mathbf{Q}}^{(l)}\right\}^{-1}$. Apply the inverse of matrix block decomposition lemma to (44), in which $\mathbf{b}(\lambda)$, $d(\lambda)$, $\underline{\mathbf{Q}}$ are replaced by $\mathbf{b}(\lambda^{(l-1)})$, $d(\lambda^{(l-1)})$ and $\underline{\mathbf{Q}}^{(l)}$, to yield,

$$\underline{\mathbf{K}}^{(l)} = \frac{1}{h(\lambda^{(l-1)})}\begin{bmatrix}\mathbf{I}+\mathbf{b}(\lambda^{(l-1)})\mathbf{b}^T(\lambda^{(l-1)}) & -\mathbf{b}(\lambda^{(l-1)}) \\ -\mathbf{b}^T(\lambda^{(l-1)}) & 1\end{bmatrix} \tag{46}$$

where

$$h(\lambda^{(l-1)}) = d(\lambda^{(l-1)}) - \mathbf{b}^T(\lambda^{(l-1)})\mathbf{b}(\lambda^{(l-1)}). \tag{47}$$

The proposed algorithm is fast and stable, as the update of $\underline{\mathbf{K}}^{(l)}$ over iteration step l is simplified with no need of matrix inversion. Following deriving the MLE for λ by using the above fast Gauss-Newton algorithm, the Box-Cox transformation is readily applied to form the transformed output.

For system modelling and control, it is desirable that the model is represented as (29) with a minimal number of M basis functions. Provided that the optimal parameter λ used in Box-Cox transformation, the number and location of candidate RBF centers are known, various orthogonal forward regression(OFR) algorithms [35, 13, 36, 37] are readily applicable for model structure selection and parameter estimation for the Box-Cox RBF network, simply by using the transformed system output as target of the RBF networks output. This is based on the assumption that the MLE estimator of λ as derived above can be treated as true parameter of λ. For the complete algorithm to construct a sparse Box-Cox RBF model with good generalisation, see [21], which simply extends the algorithm [19, 27] (see also Section 2) to Box-Cox RBF model.

Example 2: [21] Non-stationary time series data: Beveridge wheat price indices from 1500 to 1869[39]. The comparison study comprises two different topologies, the conventional RBF network and the Box-Cox RBF. For both topologies, all the data ($N = 370$) were used as training data set, and the input vector was set as $\mathbf{x}_k = [y_{k-1}, y_{k-2}\ , y_{k-3}, y_{k-4}, y_{k-5}]^T$. The thin-plate-spline basis function $g_i(\mathbf{x}_k) = \|\mathbf{x}_k - \mathbf{c}_i\|^2 \log\|\mathbf{x}_k - \mathbf{c}_i\|$ was used as basis function with all data sets initially used as candidate centres $\mathbf{c}_i$'s. The experimental results is given in Figure 2 and the further details can be found in [21].

5 The RBF Network with Boundary Value Constraints (BVC-RBF)

In this section we describe a newly introduced RBF topology [22] which aims to handle effectively a special type of prior knowledge given by a type of boundary value constraints (BVC). In many modelling tasks, there are more or less some prior knowledge available. Note that most of the RBF modelling algorithms are conditioned on that the model is determined based on the observational data only, so that these fit into the statistical learning framework. However, despite the fact that the availability of prior knowledge about the system could help to improve the model generalization, incorporating the deterministic prior knowledge into a statistically learning paradigm would generally make the development of modelling algorithms more difficult if not impossible.

The new topology of RBF network [22] is referred as the BVC-RBF and as shown in Figure 3. The BVC-RBF is constructed and parameterized based on the given BVC and has the capability of satisfying the BVC automatically. Because the BVC-RBF remains as a linear-in-the-parameter structure just as the conventional RBF does, it is advantageous that many of the existing modelling algorithms for a conventional RBF are almost directly applicable without added algorithmic complexity nor computational cost. Consequently the BVC-RBF effectively lends itself as a single framework in which both the deterministic prior knowledge and stochastic data are fused with ease.

Consider the identification of a semi-unknown system. Given a training data set D_N consisting of N input/output data pairs $\{\mathbf{x}_k, y_k\}_{k=1}^N$, the goal is to find the underlying system dynamics

$$y_k = f(\mathbf{x}_k) + \varepsilon_k \tag{48}$$

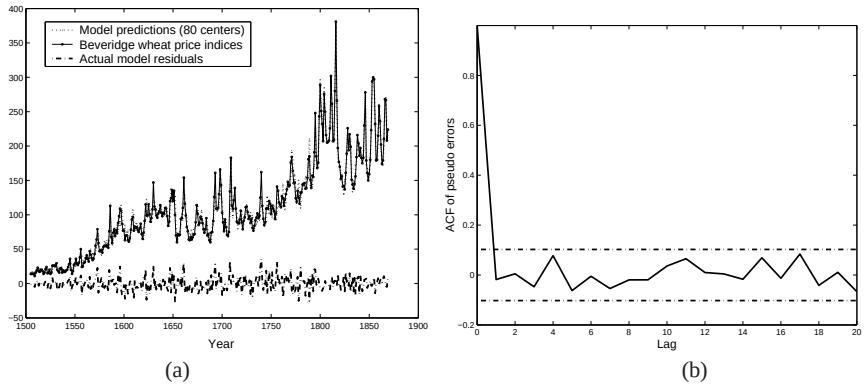

Fig. 2 (a)Modelling results of the Box-Cox RBF networks (80 centres) for Example 2; and (b) Autocorrelation function coefficients based on pseudo errors of Box-Cox RBF network (80 centre model) for Example 2, where the dotted line is calculated as $\pm\frac{1.96}{\sqrt{N}}$.

The underlying function $f : \Re^m \rightarrow \Re$ is unknown. ε_k is the noise, which is often assumed to be independent and identically distributed (i.i.d.) with constant variance. In addition, it is required that the model *strictly* satisfies a set of $\mathscr{L}$ boundary value constraints (BVC) given by

$$f(\mathbf{x}'_j) = d_j, \quad j = 1, ..., \mathscr{L} \tag{49}$$

where $\mathbf{x}'_j \in \Re^m$ and $d_j \in \Re$ are known. Note that the information from the given BVC is fundamentally different from that of the observational data set D_N and should be treated differently. The BVC is a deterministic condition but D_N is subject to observation noise and possesses stochastic characteristics. The BVC may represent the fact that at some critical regions, there is a complete knowledge about the system.

If the underlying function $f(.)$ is represented by a conventional RBF network (formulated as (1)), then resultant RBF network using the conventionl modelling procedure, e.g. Section 2, cannot meet the BVC given by (49). Clearly the prior knowledge about the system from BVC should help to improve the model generalization, but equally this makes the modelling process more difficult, since with constraints we are facing a constrained optimization problem. A simple yet effective treatment was introduced to ease the problem [22], as summarized below.

The design goal in [22] is to find a new topology of RBF such that the BVC is automatically satisfied, and as a consequence the system identification can be carried out without added algorithmic complexity nor computational cost compared to any modelling algorithm for a conventional RBF. The BVC-RBF is parameterized and dependent upon the given BVC as shown below. Consider the following BVC-RBF model representation

$$\hat{y}_k = \sum_{i=1}^{M} g_i(\mathbf{x}_k) w_i + \bar{h}(\mathbf{x}_k) \tag{50}$$

where the proposed RBF function for BVC-RBF model [22] is given by

$$g_i(\mathbf{x}_k) = s(\mathbf{x}_k) \exp(-\frac{\|\mathbf{x}_k - \mathbf{c}_i\|^2}{\tau_1^2}) \tag{51}$$

where $s(\mathbf{x}_k) = \sqrt[\mathscr{L}]{\prod_{j=1}^{\mathscr{L}} \|\mathbf{x}_k - \mathbf{x}'_j\|}$ is the geometric mean of the data sample $\mathbf{x}_k$ to the set of boundary values $\mathbf{x}'_j$, $j = 1, ..., \mathscr{L}$. $\mathbf{c}_i \in \Re^m$ is the RBF centers, τ_1 is a positive scalar.

$$\bar{h}(\mathbf{x}_k) = \sum_{j=1}^{\mathscr{L}} \alpha_j \exp(-\frac{\|\mathbf{x}_k - \mathbf{x}'_j\|^2}{\tau_2^2}) \tag{52}$$

τ_2 is also a positive scalar. α_j is a set of parameters that is obtained by solving a set of linear equations $g(\mathbf{x}_j) = d_j$, $j = 1, ..., \mathscr{L}$. That is

$$\alpha = \bar{\mathbf{H}}^{-1}\mathbf{d} \tag{53}$$

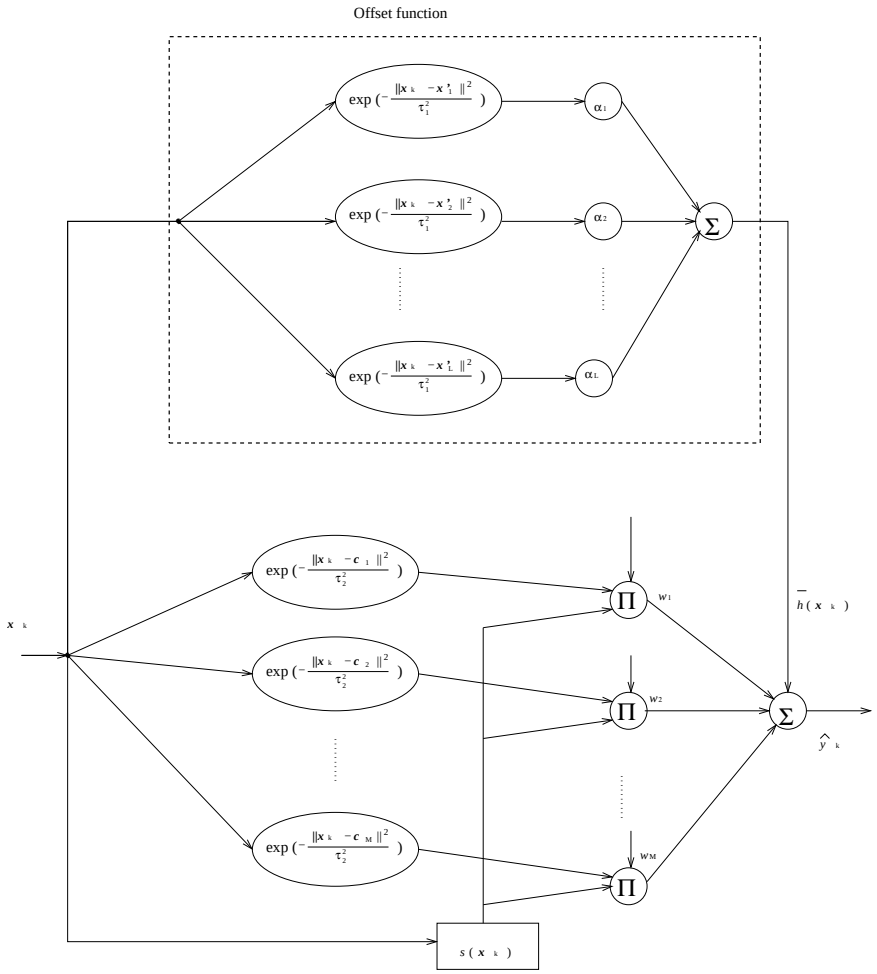

Fig. 3 A graphical illustration of the BVC-RBF network.

where $\alpha = [\alpha_1, ..., \alpha_{\mathscr{L}}]^T$, $\mathbf{d} = [d_1, ..., d_L]^T$ and $\bar{\mathbf{H}}$ is given by

$$\bar{\mathbf{H}} = \begin{pmatrix} 1 & e^{-\frac{\|\mathbf{x}'_1-\mathbf{x}'_2\|^2}{\tau_2^2}} & \dots & e^{-\frac{\|\mathbf{x}'_1-\mathbf{x}'_{\mathscr{L}}\|^2}{\tau_2^2}} \\ e^{-\frac{\|\mathbf{x}'_2-\mathbf{x}'_1\|^2}{\tau_2^2}} & 1 & \dots & e^{-\frac{\|\mathbf{x}'_2-\mathbf{x}'_{\mathscr{L}}\|^2}{\tau_2^2}} \\ \dots & \dots & \dots & \dots \\ e^{-\frac{\|\mathbf{x}'_{cal\,L}-\mathbf{x}'_1\|^2}{\tau_2^2}} & e^{-\frac{\|\mathbf{x}'_{\mathscr{L}}-\mathbf{x}'_2\|^2}{\tau_2^2}} & \dots & 1 \end{pmatrix} \tag{54}$$

In the case of the ill-conditioning, the regularization technique is applied to the above solution. It is easy to verify that with the proposed topology of BVC-RBF neural networks, the BVC is automatically satisfied [22]. In general, $g_i(\mathbf{x}_k)$ and $\bar{h}(\mathbf{x}_k)$ act as building blocks of the BVC-RBF networks in (50), with a novel feature compared to most of the existent neural networks architecture. That is, by resorting to the given boundary conditions, its topology is designed for the boundary constraints satisfaction, or more generally, for incorporating given prior knowledge. Note that the boundary condition satisfaction via the network topology is an inherent, but often overlooked, feature for any model representation. For example, the autoregressive with exogenous output (ARX) model automatically satisfies the boundary condition of $f(\mathbf{0}) = 0$, and for the conventional RBF with the Gaussian basis functions, $f(\infty) = 0$. The aim of [22] is to introduce and exploit the boundary condition satisfaction via the network topology in a controlled manner, so that the modelling performance may be enhanced by incorporating the a prior knowledge via boundary conditions satisfaction.

Substituting (50) into (48) and defining an auxiliary output variable $z_k = y_k - \bar{h}(\mathbf{x}_k)$, we have

$$z_k = \sum_{i=1}^{M} g_i(\mathbf{x}_k) w_i + e_k \tag{55}$$

conforming to (1), except that the auxiliary output variable z_k is used as the target of the first term in (50) (the adjustable part of BVC-RBF). Aiming for improved model robustness, the D-optimality in experimental design [40] has been incorporated in the D-optimality based model selective criterion [41] to select M regressors in a forward regression manner. For completeness the combined D-optimality based orthogonal least squares algorithm [41] is used in the following example [22].

Example 3[22]: The Matlab logo was generated by the first eigenfunction of the L-shaped membrane. A 31×31 meshed data set $f(x_1, x_2)$ is generated by using Matlab command *membrane.m*, which is defined over a unit square input region $x_1 \in [0,1]$ and $x_2 \in [0,1]$. The data set $y(x_1, x_2) = f(x_1, x_2) + \varepsilon(x_1, x_2)$ is then generated by adding a noise term $\varepsilon(x_1, x_2) \sim N(0, 0.01^2)$. We use all the data points within the boundary as the training data set D_N consisting of the set of $\{x_1, x_2, y(x_1, x_2)\}$ coordinates ($N = 721$). For comparison, the combined D-optimality based orthogonal least squares algorithm was applied [41] to identify a sparse conventional RBF model. The modeling results are shown in Figure 4 and Table 2. It is shown that the BVC-RBF can achieve significant improvement over the RBF in terms of the modeling performance to the true function. In particular we note that the BVC can be satisfied with the proposed BVC-RBF model, but not by the conventional RBF. The detail of the parameters setting for the experiment can be found in [22].

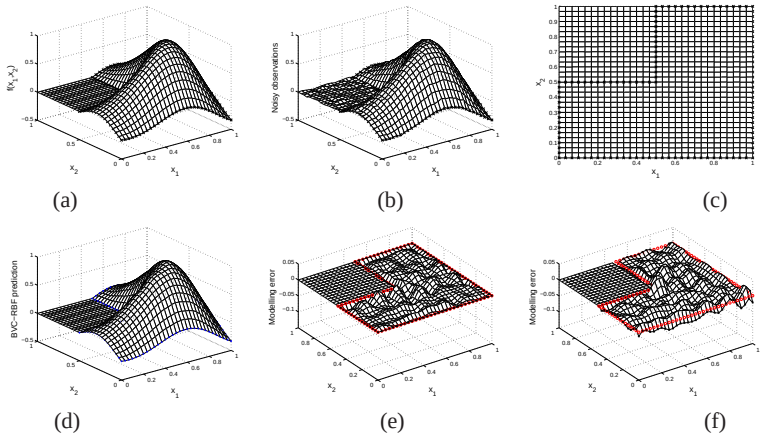

Fig. 4 Example 3; (a) the true function $f(x_1, x_2)$; (b) noisy data $y(x_1, x_2)$; (c) the boundary points and (d) the prediction of the resultant BVC-RBF model; (e) the modelling error between the true function and the model prediction $(\hat{y}(x_1, x_2) - f(x_1, x_2))$ for the BVC-RBF model; and (f) The modelling error for the RBF model. IEEE©2008 IEEE

Table 2 A comparison between the conventional RBF and the BVC-RBF network for Example 3.

	Model size M	MSE $\frac{1}{N}\sum(\hat{y}-f)^2$	MSE $\frac{1}{N}\sum(\hat{y}-y)^2$	MSE (boundary) $\frac{1}{\mathscr{D}}\sum_j(\hat{y}(\mathbf{x}'_j)-d_j)^2$
BVC-RBF	68	4.3787×10^{-5}	1.0736×10^{-4}	7.2598×10^{-11}
RBF	91	1.0229×10^{-4}	1.6894×10^{-4}	2.1249×10^{-4}

6 Conclusions

Our recent work on diversified RBF topologies is reviewed. Three different topologies have been introduced aimed at enhancing the modelling capabilities of RBF network by modifying their topologies for specific problems; (i) the RBF network with tunable nodes is introduced with the aim of flexible basis function shaping for achieving the minimal model and improved model generalisation; (ii) the Box-Cox RBF network is aimed at effectively handling some dynamical processes in which the model residuals exhibit heteroscedasticity; and (iii) the BVC-RBF is introduced in order to achieve automatic constraints satisfaction and incorporating deterministic *prior* knowledge with ease. It is advantageous that the model construction algorithms for the diversified RBF topologies are either direct application or extension of linear learning algorithms. In each case, an illustrative example is used to demonstrate the efficacy of the proposed topology, together with the application of the modeling construction algorithm.

References

1. X. Hong, R. J. Mitchell, S. Chen, C. J. Harris, K. Li, and G. W. Irwin, "Model selection approaches for nonlinear system identification: a review," *International Journal of Systems Science*, vol. 39, no. 10, pp. 925–946, 2008.
2. C. J. Harris, X. Hong, and Q. Gan, *Adaptive Modelling, Estimation and Fusion from Data: A Neurofuzzy Approach*, Springer-Verlag, 2002.
3. M. Brown and C. J. Harris, *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall, Hemel Hempstead, 1994.
4. A. E. Ruano, *Intelligent Control Systems using Computational Intelligence Techniques*, IEE Publishing, 2005.
5. R. Murray-Smith and T. A. Johansen, *Multiple Model Approaches to Modelling and Control*, Taylor and Francis, 1997.
6. S. G. Fabri and V. Kadirkamanathan, *Functional Adaptive Control: An Intelligent Systems Approach*, Springer, 2001.
7. J. A. Leonard and M. A. Kramer, "Radial basis function networks for classifying process faults," *IEEE Control Systems Magazine*, vol. 11, no. 3, pp. 31–38, 1991.
8. A. Caiti and T. Parisini, "Mapping ocean sediments by RBF networks," *IEEE J. Ocean Engineering*, vol. 19, no. 4, pp. 577–582, 1994.
9. Y. Li, N. Sundararajan, P. Saratchandran, and Z. Wang, "Robust neuro-H_∞ controller design for aircraft auto-landing," *IEEE Transactions on Aerosp. Electron. Syst.*, vol. 40, no. 1, pp. 158–167, 2004.
10. S. X. Ng, M. S. Yee, and L. Hanzo, "Coded modulation assisted radial basis function aided turbo equalization for dispersive rayleigh-fading channels," *IEEE Trans. on Wireless Communications*, vol. 3, no. 6, pp. 2198–2206, 2004.
11. M. Stone, "Cross validatory choie and assessment of statistical predictions," *Journal of the Royal Statistical Society, Series B*, vol. 36, pp. 117–147, 1974.
12. H. Akaike, "A new look at the statistical model identification," *IEEE Trans. on Automatic Control*, vol. AC-19, pp. 716–723, 1974.
13. S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their applications to non-linear system identification," *International Journal of Control*, vol. 50, pp. 1873–1896, 1989.
14. M. J. Korenberg, "Identifying nonlinear difference equation and functional expansion representations: the fast orthogonal algorithm," *Annals of Biomedical Engineering*, vol. 16, pp. 123–142, 1988.
15. L. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Trans. on Neural Networks*, vol. 5, pp. 807–814, 1992.
16. X. Hong and C. J. Harris, "Neurofuzzy design and model construction of nonlinear dynamical processes from data," *IEE Proc. - Control Theory and Applications*, vol. 148, no. 6, pp. 530–538, 2001.
17. Q. Zhang, "Using wavelets network in nonparametric estimation," *IEEE Trans. on Neural Networks*, vol. 8, no. 2, pp. 1997, 1993.
18. S. A. Billings and H. L. Wei, "The wavelet-narmax representation: A hybrid model structure combining polynomial models with multiresolution wavelet decompositions," *International Journal of Systems Science*, vol. 36, no. 3, pp. 137 – 152, 2005.
19. X. Hong, P. M. Sharkey, and K. Warwick, "Automatic nonlinear predictive model construction using forward regression and the PRESS statistic," *IEE Proc.-Control Theory Appl.*, vol. 150, no. 3, pp. 245–254, 2003.
20. S. Chen, X. Hong, and C. J. Harris, "Construction of tunable radial basis function networks using orthogoanl forward selection," *IEEE Trans. on Trans. on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 39, no. 2, pp. 457–466, 2009.
21. X. Hong, "Modified radial basisfunction neural networks using output transformation," *IET Proc.-Control Theory Appl.*, vol. 1, no. 1, pp. 1–8, 2007.

22. X. Hong and S. Chen, "A new RBF neural network with boundary value constraints," *IEEE Transactions on System, Man, and Cybernetics, Part B*, vol. 39, no. 1, pp. 298–303, 2009.
23. V. Vapnik, *The Nature of Statictical Learning Theory*, New York: Springer-Verlag, 1995.
24. S. R. Gunn, "Support vector machine for classification and regression," in *Technical Report*. ISIS Research Group, Dept of Electronics and Computer Science, University of Southampton, 1998.
25. S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.
26. M.E.Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.
27. S. Chen, X. Hong, C. J. Harris, and P. M. Sharkey, "Sparse modelling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Trans. on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 34, no. 2, pp. 898–911, 2004.
28. R. H. Myers, *Classical and modern regression with applications*, PWS-KENT, Boston, 2nd edn., 1990.
29. S. Chen, X. X. Wang, and C. J. Harris, "Experiments with repeating weighted boosting search for optimization signal processing applications," *IEEE Trans. on Trans. on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 35, no. 4, pp. 682–693, 2005.
30. [online], "Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html," .
31. S. Chen, X. Hong, B. L. Luk, and C. J. Harris, "Nonlienar system identification using particle swarm optimization tuned radial basis function models," *International Journal of Bio-Inspired Computation*, vol. 1, no. 4, pp. 246–258, 2009.
32. G. E. P. Box and D.R. Cox, "An analysis of transformation," *Journal of the Royal Statitical Society. Series B*, vol. 26, no. 2, pp. 211–252, 1964.
33. R. J. Carroll and D. Ruppert, *Transformation and Weighting in Regression*, Chapman and Hall, 1988.
34. A.A.Ding and X. He, "Backpropagation of pseudoerrors: neural networks that are adpative to heterogeneous noise," *IEEE Trans. on Neural Networks*, vol. 14, no. 2, pp. 253–262, 2003.
35. S. Chen, Y. Wu, and B. L. Luk, "Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks," *IEEE Trans. on Neural Networks*, vol. 10, pp. 1239–1243, 1999.
36. X. Hong and C. J. Harris, "Nonlinear model structure design and construction using orthogonal least squares and D-optimality design," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1245–1250, 2002.
37. S. Chen, "Locally regularised orthogonal least squares algorithm for the construction of sparse kernel regression models," in *Proceedings of 6th Int. Cof. Signal Processing*, Beijing, China, 2002, pp. 1229–1232.
38. M. J. D. Powell, "Problems related to unconstrained optimization," in *Numerical Methods for Unconstrained optimization*, W. Murray, Ed., pp. 29–55. London and New York: Academic Press, 1972.
39. K.W.Hipel and A.I. Mcleod, *Time Series Modelling of Water Resourses and Environmental Systems*, Amsterdam: Elsevier, 1994.
40. A. C. Atkinson and A. N. Donev, *Optimum Experimental Designs*, Clarendon Press, Oxford, 1992.
41. X. Hong and C. J Harris, "Experimental design and model construction algorithms for radial basis function networks," *International Journal of Systems Science*, vol. 34, no. 14-15, pp. 733–745, 2003.