NARX-Based Nonlinear System Identification Using Orthogonal Least Squares Basis Hunting

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

***Abstract*—An orthogonal least squares technique for basis hunting (OLS-BH) is proposed to construct sparse radial basis function (RBF) models for NARX-type nonlinear systems. Unlike most of the existing RBF or kernel modelling methods, which places the RBF or kernel centers at the training input data points and use a fixed common variance for all the regressors, the proposed OLS-BH technique tunes the RBF center and diagonal covariance matrix of individual regressor by minimizing the training mean square error. An efficient optimization method is adopted for this basis hunting to select regressors in an orthogonal forward selection procedure. Experimental results obtained using this OLS-BH technique demonstrate that it offers a state-of-the-art method for constructing parsimonious RBF models with excellent generalization performance.**

***Index Terms*—Basis hunting (BH), neural networks, nonlinear system identification, orthogonal least squares (OLS), sparse kernel regression.**

I. Introduction

A BASIC principle in nonlinear system modelling is the parsimonious principle of ensuring the smallest possible model that explains the data [1]. Popular forward selection using the orthogonal least squares (OLS) algorithm [2]–[11] provides an effective means of constructing parsimonious linear-in-the-weights nonlinear models that generalize well. Alternatively, the support vector machine (SVM) and other sparse kernel modelling techniques [12]–[22] have been widely adopted in data modelling applications. These sparse regression modelling techniques in effect choose the basis or kernel centers from the training input data points and use a fixed common variance for all the regressor units. It is well-known that the value of this common variance has a critical influence on the model generalization capability and the level of model sparsity. Since these model construction algorithms do not provide this basis variance, it has to be treated as a hyperparameter and learned via costly cross validation. For example, in [6] a genetic algorithm (GA) is applied to determine the appropriate common basis variance through optimizing the model generalization performance using a separate validation data set.

A recent work [23] has developed a construction algorithm for nonlinear system identification based on a general radial basis function (RBF) model. The method as usual considers all the training input points as candidate RBF centers but the algorithm individually fits a diagonal covariance matrix to each RBF regressor by maximizing the correlation function of each candidate regressor over the training data set. The locally regularized OLS algorithm based on the leave-one-out mean square error [11] is then applied to select a sparse representation from the resulting candidate regressor set. The experimental results reported in [23] show that this approach yields sparser models with excellent generalization capability, in comparison with the standard approach of adopting a single common RBF variance. Moreover, the RBF covariance matrices are optimized using the training data set, and there is no need to involve an additional validation data set for this optimization. A drawback of this approach is an increase in computational complexity, particularly when the number of the data points is large, since each data point needs to be fitted with a diagonal RBF covariance matrix.

We propose a novel method for regression modelling using the general RBF model. The proposed algorithm tunes the RBF center and diagonal covariance matrix of each regressor by minimizing the training mean square error (MSE) in an orthogonal forward selection procedure. This basis hunting process is performed using a global optimization algorithm called the repeated weighted boosting search (RWBS) [24]. Because the RBF centers are not restricted to the training input data and each regressor has an individually optimized diagonal covariance matrix, this orthogonal least squares basis hunting (OLS-BH) method is capable of producing very sparse models that generalize well. Our modelling experimental results demonstrate that this OLS-BH algorithm can produce much more parsimonious models with equally good generalization capability, in comparison with the existing state-of-the-art sparse RBF and kernel modelling techniques. Because the number of the selected RBF regressors is typically very small and optimization is only performed for this small set of RBF units, the proposed OLS-BH algorithm requires far less computation, compared with the algorithm developed recently in [23].

II. General RBF Modelling for Nonlinear System

For notational simplicity, we consider the class of discrete stochastic nonlinear systems that can be represented by the following NARX structure:

$$\begin{aligned} y_k &= f_s(y_{k-1}, \ldots, y_{k-n_y}, u_{k-1}, \ldots, u_{k-n_u}; \mathbf{w}) + e_k \\ &= f_s(\mathbf{x}_k; \mathbf{w}) + e_k \end{aligned} \tag{1}$$

where u_k and y_k are the system input and output variables, respectively, n_u and n_y are the known lags in u_k and y_k, respectively, the observation noise e_k is uncorrelated with zero mean, $f_s(\bullet)$ is the unknown system mapping, $\mathbf{x}_k = [y_{k-1} \cdots y_{k-n_y} u_{k-1} \cdots u_{k-n_u}]^T$ denotes the system input vector with a known dimension $n = n_y + n_u$, and $\mathbf{w}$ is

Manuscript received February 17, 2006; revised July 28, 2006. Manuscript received in final form January 3, 2007. Recommended by Associate Editor G. Yen. The work of S. Chen was supported by the United Kingdom Royal Academy of Engineering.

S. Chen and C. J. Harris are with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ U.K. (e-mail: sqc@ecs.soton.ac.uk; cjh@ecs.soton.ac.uk).

X. X. Wang is with Institute of Human Genetics, University of Newcastle, Newcastle upon Tyne NE1 3BZ U.K. (e-mail: xunxian.wang@newcastle.ac.uk)

Digital Object Identifier 10.1109/TCST.2007.899728

an unknown parameter vector associated with the appropriate, but yet to be determined, model structure. The NARX model (1) is a special case of the general NARMAX model that takes the form [25], [26]

$$y_k = f_s(y_{k-1}, \ldots, y_{k-n_y}, u_{k-1}, \ldots, u_{k-n_u}, e_{k-1}, \ldots, e_{k-n_e}; \mathbf{w}) + e_k. \quad (2)$$

The technique developed in this contribution can be extended to the generic NARMAX model of (2), see for example [2], [3], and [25].

The system model (1) is to be identified from an N-sample system observational data set $D_N = \{\mathbf{x}_k, y_k\}_{k=1}^N$, using some suitable functional which can approximate $f_s(\bullet)$ with arbitrary accuracy. One class of such functionals is the regression model of the form

$$y_k = \hat{y}_k + e_k = \sum_{i=1}^{M} w_i g_i(\mathbf{x}_k) + e_k \quad (3)$$

where $\hat{y}_k$ denotes the model output given the input $\mathbf{x}_k$, w_i are the model weight parameters, $g_i(\bullet)$ are the model regressors, and M is the number of regressors. The RBF model and the solution of many kernel methods can be represented in the form of (3). We will allow the regressor to be chosen as the following general form:

$$g_i(\mathbf{x}) = \varphi\left(\sqrt{(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}-\boldsymbol{\mu}_i)}\right) \quad (4)$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i = \text{diag}\{\sigma_{i,1}^2, \ldots, \sigma_{i,n}^2\}$ are the ith regressor's basis center and diagonal covariance matrix, respectively, and $\varphi(\bullet)$ is the chosen basis function. Note that, unlike the method given in [23] and other existing kernel modelling techniques, the basis or kernel centers $\boldsymbol{\mu}_i$ are not chosen from the training input points $\mathbf{x}_k$. Rather, the basis centers are also tunable parameters.

The proposed OLS-BH algorithm constructs the regression model (3) by "hunting" the regressors one by one in an orthogonal forward selection procedure. By defining $\mathbf{y} = [y_1 y_2 \cdots y_N]^T$, $\mathbf{e} = [e_1 e_2 \cdots e_N]^T$, $\mathbf{w} = [w_1 w_2 \cdots w_M]^T$ and

$$\mathbf{G} = [\mathbf{g}_1 \mathbf{g}_2 \cdots \mathbf{g}_M] \quad (5)$$

with

$$\mathbf{g}_i = [g_i(\mathbf{x}_1) g_i(\mathbf{x}_2) \cdots g_i(\mathbf{x}_N)]^T, \quad 1 \le i \le M \quad (6)$$

the regression model (3) over the training data set D_N can be written in the matrix form

$$\mathbf{y} = \mathbf{G}\mathbf{w} + \mathbf{e}. \quad (7)$$

Let an orthogonal decomposition of the regression matrix be $\mathbf{G} = \mathbf{P}\mathbf{A}$, where

$$\mathbf{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,M} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{M-1,M} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \quad (8)$$

and

$$\mathbf{P} = [\mathbf{p}_1 \mathbf{p}_2 \cdots \mathbf{p}_M] \quad (9)$$

with the orthogonal columns that satisfy $\mathbf{p}_i^T \mathbf{p}_j = 0$, if $i \ne j$. The regression model (7) can alternatively be expressed as

$$\mathbf{y} = \mathbf{P}\boldsymbol{\theta} + \mathbf{e} \quad (10)$$

where the weight vector $\boldsymbol{\theta} = [\theta_1 \theta_2 \cdots \theta_M]^T$ in the orthogonal model space satisfies the triangular system

$$\mathbf{A}\mathbf{w} = \boldsymbol{\theta}. \quad (11)$$

Knowing $\mathbf{A}$ and $\boldsymbol{\theta}$, $\mathbf{w}$ can readily be solved from (11). For the M-term orthogonal regression model (10), the training MSE

$$J_M = \frac{1}{N}\mathbf{e}^T\mathbf{e} \quad (12)$$

can be expressed as [2]

$$J_M = \frac{1}{N}\mathbf{e}^T\mathbf{e} = \frac{1}{N}\mathbf{y}^T\mathbf{y} - \frac{1}{N}\sum_{i=1}^{M} \mathbf{p}_i^T \mathbf{p}_i \theta_i^2. \quad (13)$$

Now consider using an OLS-BH procedure to "hunt" the regressors one by one. At the l-th stage of this orthogonal forward selection, we will have built up a model consisting of l regressors. The MSE cost for this l-term "subset" model can be expressed recursively as

$$J_l = J_{l-1} - \frac{1}{N}\mathbf{p}_l^T \mathbf{p}_l \theta_l^2 \quad (14)$$

where $J_0 = \mathbf{y}^T\mathbf{y}/N$. At the lth stage of the basis hunting modelling process, the lth regressor is determined by maximizing the error reduction criterion defined as

$$\text{ER}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) = \frac{1}{N}\mathbf{p}_l^T \mathbf{p}_l \theta_l^2. \quad (15)$$

Unlike the original OLS algorithm [2], however, here the maximization is with respect to the basis center $\boldsymbol{\mu}_l$ and the diagonal covariance matrix $\boldsymbol{\Sigma}_l$ of the lth regressor. As usual, θ_l is the associated least squares weight solution. This OLS-BH procedure can be terminated at the Mth stage if

$$J_M < \xi \quad (16)$$

is satisfied, where the small positive scalar ξ is a chosen tolerance. This produces a parsimonious model containing M regressors.

An appropriate value for ξ is problem dependent and must be learned empirically. Alternatively, the Akaike information criterion (AIC) [27], [28] can be adopted to terminate the OLS-BH procedure. Specifically, for the l-term model, the AIC is defined as

$$\text{AIC}_l = N \log(J_l) + l\chi \quad (17)$$

where χ is the critical value of the chi-squared distribution with one degree of freedom and for a given level of significance. An appropriate value for χ can be shown to be $\chi = 2.0$ [27]. If the AIC reaches the minimum at $l = M$, then the OLS-BH procedure is terminated, yielding an M-term model. The termination of the OLS-BH process can also be decided using cross validation [29]–[31]. Instead of using the pure MSE criterion (12), other criteria can also be adopted for the OLS-BH procedure, and these include regularization, optimal experimental design, and leave-one-out criterion [10], [11].

III. Orthogonal Least Squares Basis Hunting

The task of lth stage of the OLS-BH regression is to determine the lth regressor by minimizing the training MSE cost function $J_l(\mathbf{u})$ over $\mathbf{u} \in U$, where the vector $\mathbf{u}$ contains the regressor's basis center $\boldsymbol{\mu}_l$ and diagonal covariance matrix $\boldsymbol{\Sigma}_l$. This task may be carried out with a gradient-based optimization method. A gradient-based method, however, depends on the initial condition and may become trapped at the local minima. Alternatively, the standard global optimization methods, such as the GA [32], [33] and adaptive simulated annealing (ASA) [34], [35], can be used. We opt to perform this optimization task using the RWBS algorithm [24]. The RWBS algorithm is a simple yet efficient global search algorithm that adopts some ideas from boosting [36]–[39]. In a comparative study investigated in [24], the RWBS algorithm was shown to achieve a similar global convergence speed as the GA and ASA for several global optimization applications. The RWBS algorithm has additional advantages of requiring minimum programming effort and having fewer algorithmic parameters that require to tune. The procedure of using the RWBS algorithm to determine the basis parameters, $\boldsymbol{\mu}_l$ and $\boldsymbol{\Sigma}_l$, at the lth modelling stage of the OLS-BH regression is summarized as follows.

Give the RWBS algorithmic parameters: the population size P_S, the number of generations in the repeated search N_G, and the accuracy for terminating the weighted boosting search ξ_B.

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

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

Weighted Boosting Search Initialization: Assign the initial distribution weightings $\delta_i(0) = 1/P_S$, $1 \leq i \leq P_S$, for the population. Then

1) For $1 \leq i \leq P_S$, generate $\mathbf{g}_l^{(i)}$ from $\mathbf{u}_i^{(m)}$, the candidates for the lth model column, and orthogonalize them

$$\alpha_{j,l}^{(i)} = \frac{\mathbf{p}_j^T \mathbf{g}_l^{(i)}}{\mathbf{p}_j^T \mathbf{p}_j}, \quad 1 \leq j < l, \tag{18}$$

$$\mathbf{p}_l^{(i)} = \mathbf{g}_l^{(i)} - \sum_{j=1}^{l-1} \alpha_{j,l}^{(i)} \mathbf{p}_j. \tag{19}$$

2) For $1 \leq i \leq P_S$, calculate the cost function value of each $\mathbf{u}_i^{(m)}$

$$\theta_l^{(i)} = \frac{\left(\mathbf{p}_l^{(i)}\right)^T \mathbf{y}}{\left(\mathbf{p}_l^{(i)}\right)^T \mathbf{p}_l^{(i)}}, \tag{20}$$

$$J_l^{(i)} = J_{l-1} - \frac{1}{N}\left(\mathbf{p}_l^{(i)}\right)^T \mathbf{p}_l^{(i)} \left(\theta_l^{(i)}\right)^2. \tag{21}$$

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

Step 1: Boosting

1) Find

$$i_{\text{best}} = \arg \min_{1 \leq i \leq P_S} J_l^{(i)} \text{ and } i_{\text{worst}} = \arg \max_{1 \leq i \leq P_S} J_l^{(i)}.$$

Denote $\mathbf{u}_{\text{best}}^{(m)} = \mathbf{u}_{i_{\text{best}}}^{(m)}$ and $\mathbf{u}_{\text{worst}}^{(m)} = \mathbf{u}_{i_{\text{worst}}}^{(m)}$.

2) Normalize the cost function values

$$\bar{J}_l^{(i)} = \frac{J_l^{(i)}}{\sum_{j=1}^{P_S} J_l^{(j)}}, 1 \leq i \leq P_S.$$

3) Compute a weighting factor β_t according to

$$\eta_t = \sum_{i=1}^{P_S} \delta_i(t-1) \bar{J}_l^{(i)}, \beta_t = \frac{\eta_t}{1 - \eta_t}.$$

4) Update the distribution weightings for $1 \leq i \leq P_S$

$$\delta_i(t) = \begin{cases} \delta_i(t-1)\beta_t^{\bar{J}_l^{(i)}}, & \text{for } \beta_t \leq 1 \\ \delta_i(t-1)\beta_t^{1-\bar{J}_l^{(i)}}, & \text{for } \beta_t > 1 \end{cases}$$

and normalize them

$$\delta_i(t) = \frac{\delta_i(t)}{\sum_{j=1}^{P_S} \delta_j(t)}, \quad 1 \leq i \leq P_S.$$

Step 2: Parameter updating

1) Construct the $(P_S + 1)$th point using the formula

$$\mathbf{u}_{P_S+1} = \sum_{i=1}^{P_S} \delta_i(t) \mathbf{u}_i^{(m)}.$$

2) Construct the $(P_S + 2)$th point using the formula

$$\mathbf{u}_{P_S+2} = \mathbf{u}_{\text{best}}^{(m)} + \left(\mathbf{u}_{\text{best}}^{(m)} - \mathbf{u}_{P_S+1}\right).$$

3) Calculate $\mathbf{g}_l^{(P_S+1)}$ and $\mathbf{g}_l^{(P_S+2)}$ from $\mathbf{u}_{P_S+1}$ and $\mathbf{u}_{P_S+2}$, orthogonalize these two candidate model columns [as in (18) and (19)], and compute their corresponding cost function values $J_l^{(i)}$, $i = P_S + 1, P_S + 2$ [as in (20) and (21)]. Then find

$$i_* = \arg \min_{i=P_S+1, P_S+2} J_l^{(i)}.$$

4) The pair $(\mathbf{u}_{i_*}, J_l^{(i_*)})$ then replaces $(\mathbf{u}_{\text{worst}}^{(m)}, J_l^{(i_{\text{worst}})})$ in the population.

If $\|\mathbf{u}_{P_S+1} - \mathbf{u}_{P_S+2}\| < \xi_B$, exit **inner loop**.

$\}$ **End of inner loop**

The solution found in the mth generation is $\mathbf{u} = \mathbf{u}_{\text{best}}^{(m)}$.

$\}$ **End of outer loop**

This yields the solution $\mathbf{u} = \mathbf{u}_{\text{best}}^{(N_G)}$, i.e., $\boldsymbol{\mu}_l$ and $\boldsymbol{\Sigma}_l$ of the lth regressor, the lth model column $\mathbf{g}_l$, the orthogonalization coefficients $\alpha_{j,l}$, $1 \leq j < l$, as well as the corresponding orthogonal model column $\mathbf{p}_l$, the weight θ_l and the MSE of the l-term model J_l.

The motivation and analysis of the RWBS algorithm as a global optimizer are detailed in [24]. Appropriate values for the algorithmic parameters P_S, N_G, and ξ_B depend on the dimension of $\mathbf{u}$ and how hard the objective function to be optimized. Generally, these algorithmic parameters have to be found empirically, just as in any global optimization algorithm. In the inner loop optimization, there is no need for every member of the population to converge to a (local) minimum, and it is sufficient to locate where the minimum lies. Thus, ξ_B can be set to a relatively large value. This makes the search efficient, achieving convergence with a small number of the cost function evaluations. Instead of choosing ξ_B, we may simply set a maximum number of iterations N_I for the inner loop. The values of P_S and N_G should be set to be sufficiently large so that the parameter space will be sampled sufficiently.

Finally, we make a computational complexity comparison between this proposed algorithm and our previous algorithm of [23]. The proposed algorithm performs M tasks of $2n$-dimensional nonlinear optimization, while the previous algorithm of [23] performs N tasks of n-dimensional nonlinear optimization, where M is the number of RBF units constructed by the proposed algorithm, N is the number of training data samples, and n is the dimension of the model input space. Since M is much smaller than N, the saving in computational requirements by the proposed algorithm is self-evident.

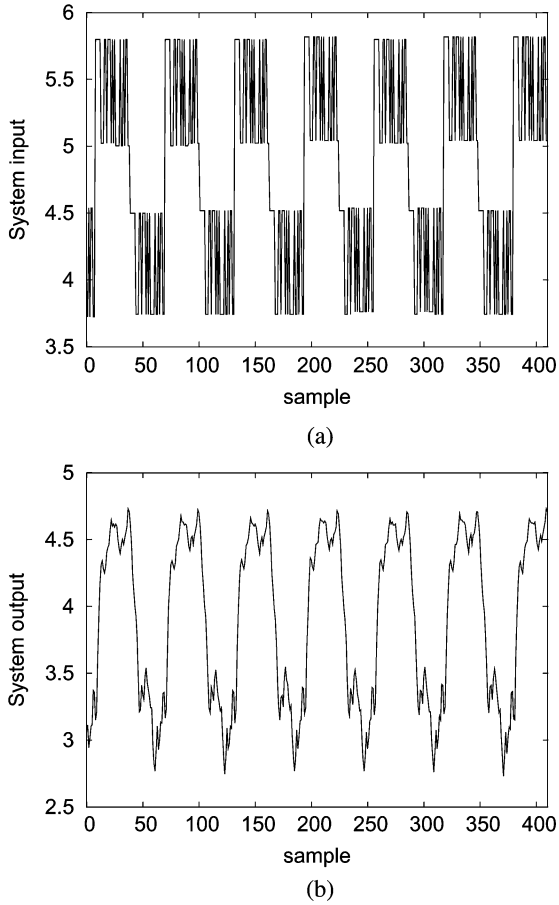

Fig. 1. Engine data set: (a) the input u_k and (b) the output y_k.

IV. Experimental Results

Two real data sets were used to investigate the proposed OLS-BH regression construction method. The basis function (4) was chosen to be Gaussian. The RWBS algorithmic parameters P_S, N_I, and N_G were chosen empirically, and it was found that the values of P_S, N_I, and N_G did not critically influence the modelling results. The OLS-BH procedure was terminated automatically when the AIC criterion (17) reached its minimum at $l = M$.

Example 1: This example constructed a model representing the relationship between the fuel rack position (input u_k) and the engine speed (output y_k) for a Leyland TL11 turbocharged, direct injection diesel engine operated at low engine speed. Detailed system description and experimental setup can be found in [40]. The input/output (I/O) data set, depicted in Fig. 1, contained 410 samples. The first 210 data points were used in training and the last 200 points in model validation. The previous study [10], [11] has shown that this data set can be modelled adequately as $y_k = f_s(\mathbf{x}_k) + e_k$ with $\mathbf{x}_k = [y_{k-1}u_{k-1}u_{k-2}]^T$. With $P_S = 40$, $N_I = 600$ and $N_G = 7$, the OLS-BH algorithm automatically produced 11 Gaussian RBF regressors, and the resultant model is listed in Table I. The MSE values of this 11-term Gaussian RBF model over the training and testing sets were 0.000496 and 0.000503, respectively. Fig. 2(a) depicts the model prediction $\hat{y}_k$ superimposed on the system output y_k and Fig. 2(b) shows the model prediction error $\hat{e}_k = y_k - \hat{y}_k$, for this 11-term Gaussian RBF model. To achieve a similar modelling accuracy, the algorithm presented in [23] required 15 Gaussian RBF regressors. Furthermore, computational complexity of the OLS-BH procedure was much less than the algorithm of [23], since the former only required 11 optimization stages corresponding to the 11 selected regressors while the latter involved 210 optimization fittings required for the 210 candidate regressors.

Example 2: This example constructed a model for the gas furnace data set [41, Series J]. The data set contained 296 pairs of I/O points, where the input u_k was the coded input gas feed rate and the output y_k represented CO_2 concentration from the gas furnace. The I/O data set is depicted in Fig. 3. The model input vector was defined by $\mathbf{x}_k = [y_{k-1}y_{k-2}y_{k-3}u_{k-1}u_{k-2}u_{k-3}]^T$. The odd samples of $\{y_k, \mathbf{x}_k\}$ were used for training while the even samples were left out for testing the constructed model. The RWBS algorithmic parameters were set to $P_S = 40$, $N_I = 600$, and $N_G = 21$. The search space for this example was much larger than that of the example one, and therefore, we chose a much larger value for the number of generations. The OLS-BH algorithm automatically constructed a model with six Gaussian RBF regressors, and the MSE values of this constructed model over the training and testing data sets were 0.0513758 and 0.0760216, respectively. Table II lists this constructed six-term model, while Fig. 4 depicts the corresponding model prediction and prediction error. We also

TABLE I
MODEL PRODUCED BY THE OLS-BH PROCEDURE FOR THE ENGINE DATA SET

l	centre vector $\boldsymbol{\mu}_l$			covariance matrix $\boldsymbol{\Sigma}_l$			weight w_l
1	4.93040e+0	5.23146e+0	5.62203e+0	9.30716e+0	1.06996e+1	1.02338e+1	7.17435e+0
2	3.43152e+0	4.72652e+0	5.97125e+0	2.22072e+0	3.97365e+0	4.37818e+0	-4.38864e-1
3	4.72032e+0	5.18036e+0	4.25972e+0	1.21502e+1	1.21557e+0	8.78971e+0	2.61235e-1
4	4.51753e+0	5.40863e+0	4.61350e+0	1.86911e+0	3.25467e+0	4.81354e+0	-2.56098e+0
5	3.08483e+0	4.55273e+0	6.07243e+0	1.44145e+1	1.37853e+1	7.29465e+0	-8.74277e-1
6	4.68020e+0	6.33932e+0	4.60687e+0	1.66835e+0	1.26637e+1	5.29689e+0	1.59228e+0
7	4.60424e+0	3.54278e+0	6.12259e+0	2.08854e+0	1.35888e+1	3.25107e+0	-3.88554e-1
8	2.72135e+0	3.56534e+0	3.63845e+0	1.98233e-1	1.25966e+1	2.88064e-1	-2.95654e-1
9	3.18971e+0	5.83400e+0	4.34225e+0	4.68226e-1	1.16443e+1	1.09179e+1	-1.75496e-1
10	3.11184e+0	4.38409e+0	5.53172e+0	4.90252e+0	1.07918e+1	2.29096e+0	-5.17402e-1
11	2.12043e+0	3.45263e+0	5.74940e+0	7.71450e-1	1.57594e+1	7.43154e+0	3.40360e-1

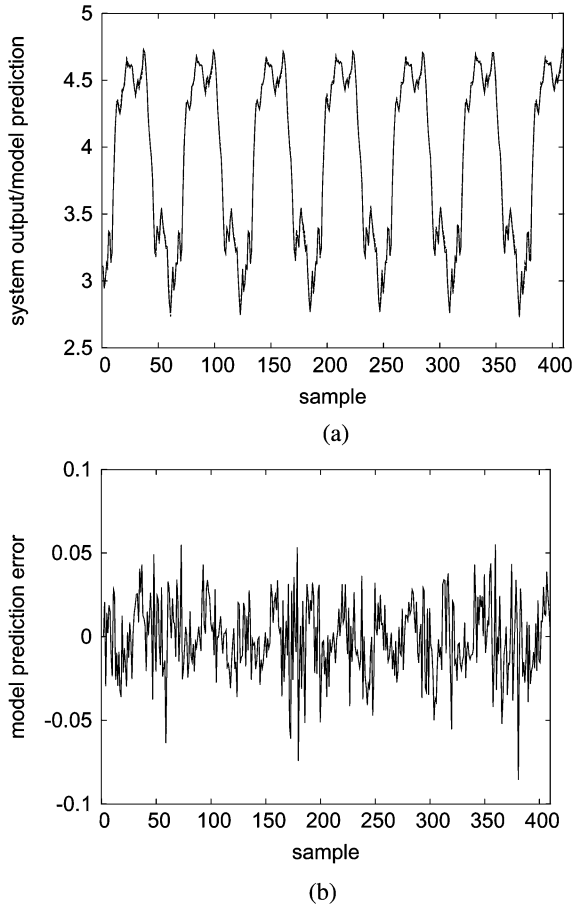

Fig. 2. OLS-BH modelling for the engine data set: (a) the model prediction $\hat{y}_k$ (dashed line) of the constructed 11-term model superimposed on the system output y_k (solid line) and (b) the corresponding model prediction error $\hat{e}_k = y_k - \hat{y}_k$.

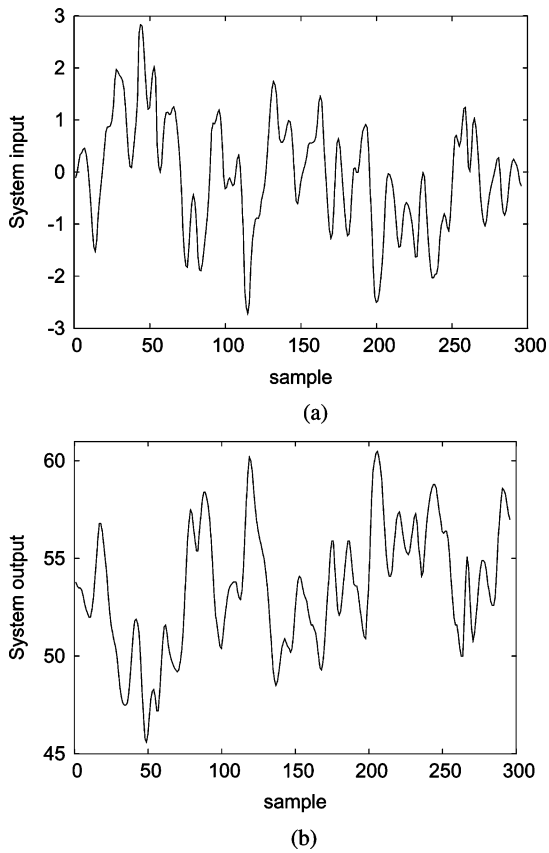

Fig. 3. Gas furnace data set: (a) the input u_k and (b) the output y_k.

applied the algorithm developed in [23] to this data set, and it needed 18 Gaussian RBF regressors to achieve a similar modelling accuracy as the six-term model produced by the OLS-BH method. Moreover, the computational complexity of the OLS-BH algorithm was a fraction of the complexity required by the algorithm of [23].

V. CONCLUSION

A novel construction algorithm has been proposed for parsimonious nonlinear system identification based on the general RBF model. Unlike most of the sparse RBF or kernel regression modelling methods, which restrict basis or kernel centres to the training input data points and use a single basis variance for all the regressors, the proposed OLS-BH algorithm has the ability to tune the center vector and diagonal covariance matrix of individual regressor by minimizing the training mean square error. An efficient yet simple global optimization search algorithm called the RWBS has been employed to "hunt" model bases one by one in an OLS regression procedure. The model construction procedure is automatically terminated using the AIC criterion. The proposed OLS-BH technique provides enhanced modelling capability with very sparse representations. Using the state-of-the-art sparse regression modelling algorithm recently developed in [23] as a benchmark, the modelling experiments involved two real-data sets have been conducted and it has been

TABLE II
MODEL PRODUCED BY THE OLS-BH PROCEDURE FOR THE GAS FURNACE DATA SET

l	centre vector $\boldsymbol{\mu}_l$ covariance matrix $\boldsymbol{\Sigma}_l$						weight w_l
1	5.83088e+1	5.62198e+1	5.64866e+1	7.47858e-1	-2.69253e+0	-1.58904e+0	8.68847e+1
	3.35482e+2	2.81978e+3	7.83954e+3	7.98400e+3	9.43025e+3	5.03522e+3	
2	6.05300e+1	4.63109e+1	4.51138e+1	-1.45846e+0	2.52120e+0	2.03603e+0	8.50866e+1
	3.80413e+2	3.89503e+3	8.55855e+3	1.97749e+3	9.88717e+3	6.16581e+3	
3	5.59832e+1	5.61848e+1	5.10818e+1	-6.79874e-1	-1.10821e-1	-8.01872e-1	-8.98341e+1
	2.07585e+2	2.62504e+3	2.63505e+3	7.46915e+3	6.51597e+3	4.35037e+3	
4	5.49372e+1	5.11708e+1	4.99839e+1	1.16323e+0	-6.44229e-1	-2.28130e+0	2.98737e+1
	4.98259e+3	1.38044e+3	7.64429e+3	6.09445e+3	3.16484e+3	4.38496e+1	
5	5.97103e+1	6.09311e+1	4.82014e+1	1.15097e+0	8.92954e-1	-1.70440e+0	-5.61899e+1
	4.65400e+3	6.94119e+2	4.77786e+3	1.87229e+3	6.70391e+3	8.46690e+1	
6	5.41696e+1	5.23214e+1	5.40583e+1	1.60523e+0	2.63507e+0	1.21269e+0	3.86157e+0
	9.01061e+3	6.17733e+2	9.51149e+1	4.69803e+3	2.79922e+3	2.97989e+3	

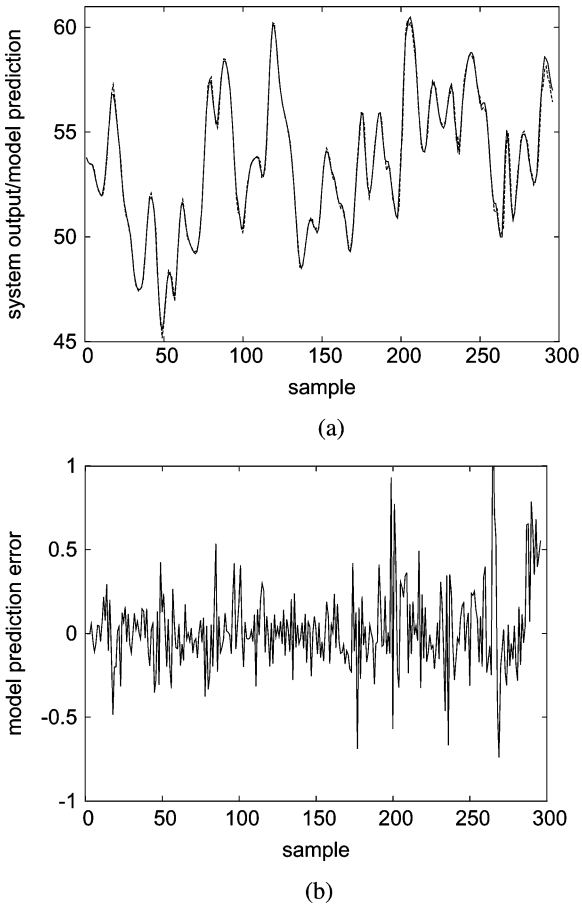

Fig. 4. OLS-BH modelling for the gas furnace data set: (a) the model prediction $\hat{y}_k$ (dashed line) of the constructed six-term model superimposed on the system output y_k (solid line) and (b) the corresponding model prediction error $\hat{e}_k = y_k - \hat{y}_k$.

shown that the proposed OLS-BH construction method is capable of producing much sparser model representations with the same excellent generalization performance, at a fraction of the complexity required by the previous algorithm [23].

REFERENCES

[1] S. A. Billings and S. Chen, "The determination of multivariable nonlinear models for dynamic systems," in *Control and Dynamic Systems, Neural Network Systems Techniques and Applications*, C. T. Leondes, Ed. San Diego, CA: Academic Press, 1998, vol. 7, pp. 231–278.

[2] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[3] S. Chen, S. A. Billings, C. F. N. Cowan, and P. M. Grant, "Practical identification of NARMAX models using radial basis functions," *Int. J. Control*, vol. 52, pp. 1327–1350, 1990.

[4] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Netw.*, vol. 2, no. 2, pp. 302–309, Mar. 1991.

[5] S. Chen, E. S. Chng, and K. Alkadhimi, "Regularized orthogonal least squares algorithm for constructing radial basis function networks," *Int. J. Control*, vol. 64, no. 5, pp. 829–837, 1996.

[6] 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. Neural Netw.*, vol. 10, no. 5, pp. 1239–1243, Sep. 1999.

[7] S. Chen, "Multi-output regression using a locally regularised orthogonal least square algorithm," *IEE Proc. Vision, Image Signal Process.*, vol. 149, no. 4, pp. 185–195, 2002.

[8] X. Hong and C. J. Harris, "Nonlinear model structure design and construction using orthogonal least squares and D-optimality design," *IEEE Trans. Neural Netw.*, vol. 13, no. 5, pp. 1245–1250, Sep. 2002.

[9] X. Hong, P. M. Sharkey, and K. Warwick, "Automatic nonlinear predictive model construction algorithm using forward regression and the PRESS statistic," *IEE Proc. Control Theory Appl.*, vol. 150, no. 3, pp. 245–254, 2003.

[10] S. Chen, X. Hong, and C. J. Harris, "Sparse kernel regression modelling using combined locally regularized orthogonal least squares and D-optimality experimental design," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 1029–1036, Jun. 2003.

[11] S. Chen, X. Hong, C. J. Harris, and P. M. Sharkey, "Sparse modelling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 34, no. 2, pp. 898–911, Apr. 2004.

[12] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[13] V. Vapnik, S. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," in *Advances in Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. Cambridge, MA: MIT Press, 1997, pp. 281–287.

[14] S. Gunn, "Support vector machines for classification and regression," ISIS Research Group, Dept. Electron. Comput. Sci., Univ. Southampton, Southampton, U.K., 1998.

[15] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learning Res.*, vol. 1, pp. 211–244, 2001.

[16] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.

[17] P. Vincent and Y. Bengio, "Kernel matching pursuit," *Mach. Learning*, vol. 48, no. 1, pp. 165–187, 2002.

[18] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.

[19] K. L. Lee and S. A. Billings, "Time series prediction using support vector machines, the orthogonal and the regularized orthogonal least-squares algorithms," *Int. J. Syst. Sci.*, vol. 33, no. 10, pp. 811–821, 2002.

[20] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: Robustness and sparse approximation," *Neurocomput.*, vol. 48, no. 1–4, pp. 85–105, 2002.

[21] L. Zhang, W. Zhou, and L. Jiao, "Wavelet support vector machine," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 34, no. 1, pp. 34–39, Feb. 2004.

[22] W. Chu, S. S. Keerthi, and C. J. Ong, "Bayesian support vector regression using a unified loss function," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 29–44, Jan. 2004.

[23] S. Chen, X. Hong, C. J. Harris, and X. X. Wang, "Identification of nonlinear systems using generalized kernel models," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 3, pp. 401–411, May 2005.

[24] S. Chen, X. X. Wang, and C. J. Harris, "Experiments with repeating weighted boosting search for optimization signal processing applications," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 35, no. 4, pp. 682–693, Aug. 2005.

[25] S. Chen and S. A. Billings, "Representation of non-linear systems: The NARMAX model," *Int. J. Control*, vol. 49, no. 3, pp. 1013–1032, 1989.

[26] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: A unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[27] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974.

[28] I. J. Leontaritis and S. A. Billings, "Model selection and validation methods for non-linear systems," *Int. J. Control*, vol. 45, no. 1, pp. 311–341, 1987.

[29] M. Stone, "Cross validation choice and assessment of statistical predictions," *J. Royal Stat. Soc. Series B*, vol. 36, pp. 117–147, 1974.

[30] R. H. Myers, *Classical and Modern Regression with Applications*, 2nd ed. Boston: PWS-KENT, 1990.

[31] J. Moody and J. Utans, "Architecture selection strategies for neural networks: Application to corporate bond rating prediction," in *Neural Networks in the Capital Markets*, A.-P. Refenes, Ed. Chichester, U.K.: Wiley, 1995, pp. 277–300.

[32] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[33] K. F. Man, K. S. Tang, and S. Kwong, *Genetic Algorithms: Concepts and Design*. London, U.K.: Springer-Verlag, 1998.

[34] L. Ingber, "Simulated annealing: Practice versus theory," *Math. Comput. Model.*, vol. 18, no. 11, pp. 29–57, 1993.

[35] S. Chen and B. L. Luk, "Adaptive simulated annealing for optimization in signal processing applications," *Signal Process.*, vol. 79, no. 1, pp. 117–128, 1999.

[36] R. E. Schapire, "The strength of weak learnability," *Mach. Learning*, vol. 5, no. 2, pp. 197–227, 1990.

[37] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[38] L. Breiman, "Prediction games and arcing algorithms," *Neural Comput.*, vol. 11, no. 7, pp. 1493–1518, 1999.

[39] R. Meir and G. Rätsch, "An introduction to boosting and leveraging," in *Advanced Lectures in Machine Learning*, S. Mendelson and A. Smola, Eds. New York: Springer-Verlag, 2003, pp. 119–184.

[40] S. A. Billings, S. Chen, and R. J. Backhouse, "The identification of linear and non-linear models of a turbocharged automotive diesel engine," *Mech. Syst. Signal Process.*, vol. 3, no. 2, pp. 123–142, 1989.

[41] G. E. P. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*. San Francisco, CA: Holden Day Inc., 1976.