Sparse Generalized Kernel Modeling for Nonlinear Systems

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

***Abstract*— A generalized kernel modeling approach is proposed for identification of discrete-time nonlinear systems. Each kernel regressor in the generalized kernel model has an individually fitted diagonal covariance matrix which is determined by maximizing the correlation between the regressor and training data. A state-of-the-art construction algorithm based on orthogonal least squares regression with leave-one-out test statistic and local regularization is applied to select a parsimonious generalized kernel model from the full regression matrix. The effectiveness of the proposed nonlinear modeling approach is demonstrated by the experimental results involving one simulated system and two real data sets.**

I. INTRODUCTION

The class of the orthogonal least squares (OLS) algorithms [1]–[6] provides an effective construction method that is capable of producing parsimonious linear-in-the-weights nonlinear models with excellent generalization performance. Alternatively, the state-of-the-art sparse kernel modeling techniques, such as the support vector machine and relevant vector machine [7]–[12], have been gaining popularity in data modeling applications. These existing sparse regression modeling techniques typically employ a single common kernel variance for all the regressors. The value of this common kernel variance has a crucial influence on the sparsity level and generalization capability of the resulting model, and it has to be determined via cross validation. For example, in [3] a genetic algorithm is applied to determine the appropriate common kernel variance through optimizing the model generalization performance.

We propose a generalized kernel model, in which each kernel regressor has an individually tuned diagonal covariance matrix. This generalized kernel model has the potential of enhancing modeling capability and producing sparser models. The difficult issue is how to determine these kernel covariance matrices. Since the correlation between a kernel regressor and the training data defines the "similarity" between the regressor and training data, we can "shape" the regressor by adjusting the associated kernel covariance matrix in order to maximize the absolute value of this correlation function. We employ the repeated weighted boosting search (RWBS) algorithm [13] to perform kernel covariance fitting. This algorithm is a guided random search method having its root from boosting optimization [14]-[17]. The determination of kernel covariance matrices provides the pool of regressors, from which a sparse subset model can be selected using a standard kernel model construction approach.

We adopt the OLS algorithm with the leave-one-out (LOO) test score and local regularization [6], referred to as the LROLS with LOO for short, to select a sparse generalized kernel model. This construction algorithm selects significant regressors by directly maximizing model generalization capability, without resorting to use a separate validation data set. The algorithm is computationally efficient and can produce very parsimonious models due to local regularization that enforces sparse solutions. Moreover, the model building process is automatic without the need for the user to specify some additional termination criterion. The effectiveness of the propose generalized kernel modeling approach is illustrated by three nonlinear system identification examples.

II. GENERALIZED KERNEL MODELING

Consider a general discrete stochastic nonlinear system represented by [18]:

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

where u_k and y_k are the system input and output variables, respectively, n_u and n_y are positive integers representing the known lags in u_k and y_k, respectively, the observation noise e_k is uncorrelated with zero mean, $\mathbf{x}_k = [y_{k-1} \cdots y_{k-n_y}\ u_{k-1} \cdots u_{k-n_u}]^T$ denotes the system input vector with a known dimension $n = n_y + n_u$, $f_s(\bullet)$ is the unknown system mapping, and $\boldsymbol{\theta}$ is an unknown parameter vector associated with an appropriate model structure. The system model (1) is to be identified from an N-sample system observational data set $D_N = \{\mathbf{x}_k, y_k\}_{k=1}^N$.

We will model the unknown dynamical process (1) by using a generalized kernel regression model of the form:

$$y_k = \hat{y}_k + \epsilon_k = \sum_{i=1}^{N} \theta_i g_i(\mathbf{x}_k) + \epsilon_k = \mathbf{g}^T(k)\boldsymbol{\theta} + \epsilon_k \tag{2}$$

where $\hat{y}_k$ denotes the model output given the input $\mathbf{x}_k$, ϵ_k is the modeling error, $\boldsymbol{\theta} = [\theta_1\ \theta_2 \cdots \theta_N]^T$ is the model weight vector, and $\mathbf{g}(k) = [g_1(\mathbf{x}_k)\ g_2(\mathbf{x}_k) \cdots g_N(\mathbf{x}_k)]^T$ is the regressor vector at the point $\mathbf{x}_k$. The model kernel regressors are given by

$$g_i(\mathbf{x}) = \varphi\left(\sqrt{(\mathbf{x} - \mathbf{x}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \mathbf{x}_i)}\right) \tag{3}$$

with $\varphi(\bullet)$ being a chosen kernel function. The kernel centers are placed directly on the training inputs $\mathbf{x}_i$, but each kernel regressor has a kernel covariance matrix taking the form of

S. Chen and C.J. Harris are with School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. E-mails: {sqc,chj}@ecs.soton.ac.uk

X. Hong is with Department of Cybernetics, University of Reading, Reading, RG6 6AY, U.K. E-mail: x.hong@reading.ac.uk

X.X. Wang is with Neural Computing Research Group, Aston University, Birmingham B4 7ET, U.K. E-mail: x.wang@aston.ac.uk

$\boldsymbol{\Sigma}_i = \text{diag}\{\sigma_{i,1}^2, \cdots, \sigma_{i,n}^2\}$. Over the training set D_N, the model (2) can be written in the matrix form as

$$\mathbf{y} = \mathbf{G}\boldsymbol{\theta} + \boldsymbol{\epsilon} \tag{4}$$

by defining $\mathbf{y} = [y_1 \ y_2 \cdots y_N]^T$, $\boldsymbol{\epsilon} = [\epsilon_1 \ \epsilon_2 \cdots \epsilon_N]^T$ and $\mathbf{G} = [\mathbf{g}_1 \ \mathbf{g}_2 \cdots \mathbf{g}_N]$ with $\mathbf{g}_i = [g_i(\mathbf{x}_1) \ g_i(\mathbf{x}_2) \cdots g_i(\mathbf{x}_N)]^T$, $1 \le i \le N$. Note that $\mathbf{g}_k$ is the kth column of the regression matrix $\mathbf{G}$, while $\mathbf{g}(k)$ is the kth row of $\mathbf{G}$.

Let an orthogonal decomposition of $\mathbf{G}$ be $\mathbf{G} = \boldsymbol{\Phi}\mathbf{A}$, where

$$\mathbf{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,N} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{N-1,N} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{5}$$

and $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \ \boldsymbol{\phi}_2 \cdots \boldsymbol{\phi}_N]$ satisfying $\boldsymbol{\phi}_i^T \boldsymbol{\phi}_j = 0$, if $i \ne j$. The regression model (4) can alternatively be expressed as

$$\mathbf{y} = \boldsymbol{\Phi}\mathbf{w} + \boldsymbol{\epsilon} \tag{6}$$

where the weight vector $\mathbf{w} = [w_1 \ w_2 \cdots w_N]^T$, defined in the new space $\boldsymbol{\Phi}$, satisfy the triangular system $\mathbf{A}\boldsymbol{\theta} = \mathbf{w}$. The space spanned by the original model bases $\mathbf{g}_i$, $1 \le i \le N$, is identical to the space spanned by the orthogonal bases $\boldsymbol{\phi}_i$, $1 \le i \le N$, and $\hat{y}_k$ can equivalently be expressed by

$$\hat{y}_k = \boldsymbol{\phi}^T(k)\mathbf{w} \tag{7}$$

where $\boldsymbol{\phi}(k) = [\phi_{k,1} \ \phi_{k,2} \cdots \phi_{k,N}]^T$ is the kth row of $\boldsymbol{\Phi}$.

III. Sparse model construction algorithm

The objective of sparse modeling is to construct a subset model consisting of N_s ($\ll N$) significant regressors from the full model defined in (2), which can adequately model the underlying system (1).

A. Determination of the full regression matrix

To specify the pool of regressors or the full regression matrix $\mathbf{G}$, we need to determine all the associated diagonal covariance matrices $\boldsymbol{\Sigma}_i$, $1 \le i \le N$. The correlation between a regressor $\mathbf{g}_i$ and the training data, defined by

$$C(\boldsymbol{\Sigma}_i) = \frac{\mathbf{y}^T\mathbf{g}_i}{\sqrt{\mathbf{y}^T\mathbf{y}}\sqrt{\mathbf{g}_i^T\mathbf{g}_i}} \tag{8}$$

represents the "similarity" between $\mathbf{g}_i$ and $\mathbf{y}$. We should choose $\boldsymbol{\Sigma}_i$ so that $|C(\boldsymbol{\Sigma}_i)|$ is maximized. It can easily be shown that this is a good strategy to specify the pool of regressors. Let us first define the least squares cost or mean square error (MSE) associated with an m-term model as

$$S_m = \frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2 \tag{9}$$

Obviously $S_0 = \mathbf{y}^T\mathbf{y}/N = \|\mathbf{y}\|^2/N$. Assuming that $\mathbf{g}_i$ is selected to form a one-term model, the associated reduction in the MSE value can be shown to be $\Delta S = S_0 - S_1 = \left(\mathbf{y}^T\mathbf{g}_i\right)^2/\mathbf{g}_i^T\mathbf{g}_i$, which can be rewritten as

$$\Delta S = \left(\mathbf{y}^T\mathbf{y}\right)\frac{\left(\mathbf{y}^T\mathbf{g}_i\right)^2}{\left(\mathbf{y}^T\mathbf{y}\right)\left(\mathbf{g}_i^T\mathbf{g}_i\right)} = \|\mathbf{y}\|^2\left|C(\boldsymbol{\Sigma}_i)\right|^2 \tag{10}$$

Since $\|\mathbf{y}\|^2$ is a constant, maximizing $|C(\boldsymbol{\Sigma}_i)|$ leads to a maximum reduction in the MSE value.

We apply the RWBS algorithm [13] to perform the associated optimization tasks for fitting kernel covariance matrices. The RWBS algorithm is a simple yet efficient global search algorithm that adopts some ideas from boosting [14]-[17]. The RWBS optimizer is given in Appendix. Once the full regression matrix $\mathbf{G}$ has been designed, the LROLS with LOO [6] can be used to select a subset model.

B. Efficient subset model selection

The weight vector $\mathbf{w}$ is obtained as the regularized least squares solution obtained by minimizing the cost

$$J_R(\mathbf{w}, \boldsymbol{\lambda}) = \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} + \sum_{i=1}^{N}\lambda_i w_i^2 \tag{11}$$

where $\boldsymbol{\lambda} = [\lambda_1 \ \lambda_2 \cdots \lambda_N]^T$ is the regularization parameter vector, which is optimized based on the evidence procedure with the iterative updating formulas [5],[6]

$$\lambda_i^{\text{new}} = \frac{\gamma_i^{\text{old}}}{N - \gamma^{\text{old}}}\frac{\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}}{w_i^2}, \quad 1 \le i \le N \tag{12}$$

where

$$\gamma_i = \frac{\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i}{\lambda_i + \boldsymbol{\phi}_i^T\boldsymbol{\phi}_i} \quad \text{and} \quad \gamma = \sum_{i=1}^{N}\gamma_i \tag{13}$$

Usually a few iterations (typically less than 10) are sufficient to find a local optimal $\boldsymbol{\lambda}$. The Bayesian interpretation of the criterion $J_R(\mathbf{w}, \boldsymbol{\lambda})$ together with the full derivation of the updating formulas (12) and (13) can be found in [5].

A forward selection procedure is used to construct a sparse model by incrementally minimizing the LOO test score. Assume that an m-term model is selected from the full model (6). The LOO test error [19]-[22], denoted as $\epsilon_k^{(m,-k)}$, for the selected m-term model can be shown to be [4],[6]

$$\epsilon_k^{(m,-k)} = \epsilon_k^{(m)}/\eta_k^{(m)} \tag{14}$$

where $\epsilon_k^{(m)}$ is the m-term modeling error and $\eta_k^{(m)}$ is the associated LOO error weighting given by

$$\eta_k^{(m)} = 1 - \sum_{i=1}^{m}\frac{\phi_{k,i}^2}{\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i + \lambda_i} \tag{15}$$

The mean square LOO error for the model with a size m is defined by

$$J_m = E\left[\left(\epsilon_k^{(m,-k)}\right)^2\right] = \frac{1}{N}\sum_{k=1}^{N}\frac{\left(\epsilon_k^{(m)}\right)^2}{\left(\eta_k^{(m)}\right)^2} \tag{16}$$

This LOO statistic is a measure of the model generalization performance and it can be computed efficiently because $\epsilon_k^{(m)}$ and $\eta_k^{(m)}$ can be calculated recursively according to

$$\epsilon_k^{(m)} = y_k - \sum_{i=1}^{m}\phi_{k,i}w_i = \epsilon_k^{(m-1)} - \phi_{k,m}w_m \tag{17}$$

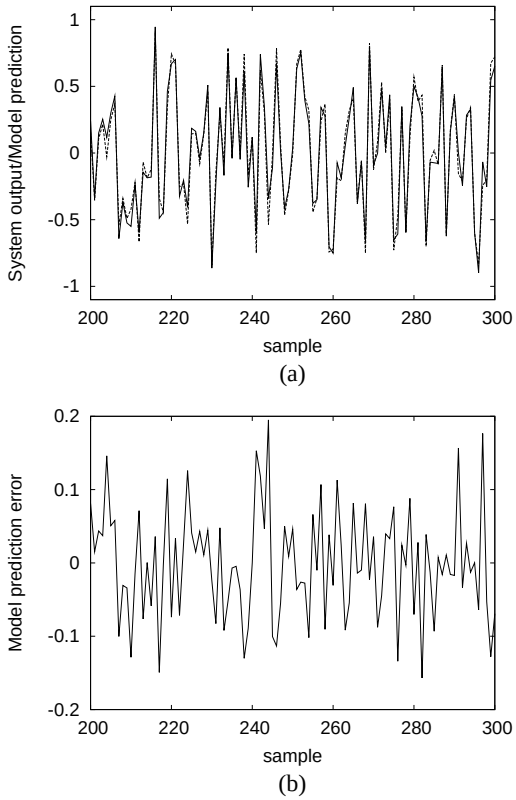

Fig. 1. Performance of the 15-term generalized Gaussian kernel model for the simulated system: (a) model prediction (dashed) superimposed on noisy system output (solid), and (b) model prediction error.

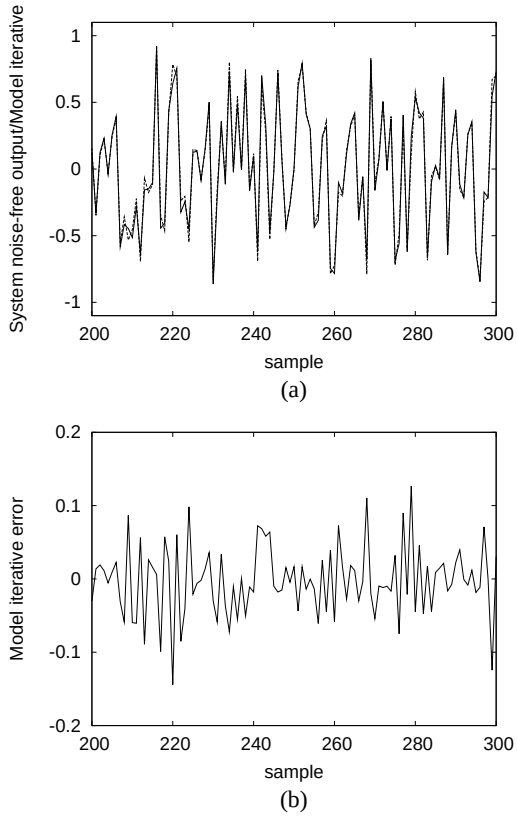

Fig. 2. Performance of the 15-term generalized Gaussian kernel model for the simulated system: (a) iterative model output (dashed) superimposed on noise-free system output (solid), and (b) iterative model error.

$$\eta_k^{(m)} = \eta_k^{(m-1)} - \frac{\phi_{k,m}^2}{\boldsymbol{\phi}_m^T \boldsymbol{\phi}_m + \lambda_m} \tag{18}$$

respectively. The idea of delete-one cross validation and the associated LOO test error are explained in [4],[6].

The subset model selection procedure can be carried as follows: at the mth stage of the selection procedure, a model term is selected among the remaining m to N candidates if the resulting m-term model produces the smallest LOO test score J_m. It has been shown in [4] that the LOO statistic J_m is convex with respect to the model size m. That is, there exists an "optimal" model size N_s such that for $m \leq N_s$ J_m decreases as m increases while for $m \geq N_s + 1$ J_m increases as m increases. Thus the selection procedure is automatically terminated with an N_s-term model when $J_{N_s+1} > J_{N_s}$, without the need for the user to specify a separate termination criterion. The details of the iterative procedure for constructing a sparse model based on the LROLS with LOO can be found in [6].

IV. Modeling examples

Three examples were used to demonstrate the effectiveness of the proposed model construction algorithm.

Example 1. This was the system considered in [23]. The underlying dynamic system was governed by

$$z_k = \frac{z_{k-1} z_{k-2} z_{k-3} u_{k-2} (z_{k-3} - 1) + u_{k-1}}{1 + z_{k-2}^2 + z_{k-3}^2} \tag{19}$$

where the system input u_k was a random signal uniformly distributed in the interval $[-1,\ 1]$. The noisy system output was given by $y_k = z_k + e_k$, where the noise e_k was Gaussian distributed with zero mean and standard deviation 0.05. Four hundred noisy samples were generated. The first 200 data points were used for training, and the other 200 samples were used for model validation. The generalized Gaussian kernel model with the input vector

$$\mathbf{x}_k = [y_{k-1}\ y_{k-2}\ y_{k-3}\ u_{k-1}\ u_{k-2}]^T \tag{20}$$

was used to construct a model from the noisy training set.

The $N = 200$ candidates' kernel covariance matrices were first determined by the RWBS algorithm, and the LROLS with LOO then selected a 15-term generalized Gaussian kernel model. The MSE values of this model over the training and testing sets were 0.003244 and 0.005195, respectively. The model prediction $\hat{y}_k$ and prediction error $\epsilon_k = y_k - \hat{y}_k$ over the first 100 data points of the test set are depicted in Fig. 1. The constructed 15-term model was also used to iteratively generate the model output according to $\hat{y}_{d,k} = f_m(\hat{\mathbf{x}}_{d,k})$ with the input $\hat{\mathbf{x}}_{d,k} = [\hat{y}_{d,k-1}\ \hat{y}_{d,k-2}\ \hat{y}_{d,k-3}\ u_{k-1}\ u_{k-2}]^T$, where $f_m(\bullet)$ denotes the model mapping. The iterative model output $\hat{y}_{d,k}$ and iterative error, defined by $\hat{\epsilon}_{d,k} = z_k - \hat{y}_{d,k}$, are shown in Fig. 2 over the first 100 data points of the test set.

For this example, the previous experiments had found out that it was difficult to select a sparse Gaussian kernel regression model using a common kernel variance [6]. Var-

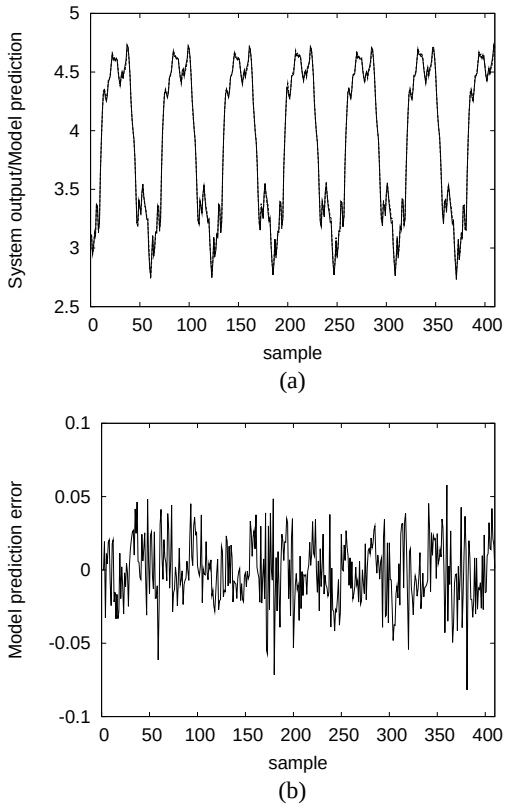

Fig. 3. Performance of the 15-term generalized Gaussian kernel model for the engine data set: (a) model prediction (dashed) superimposed on system output (solid), and (b) model prediction error.

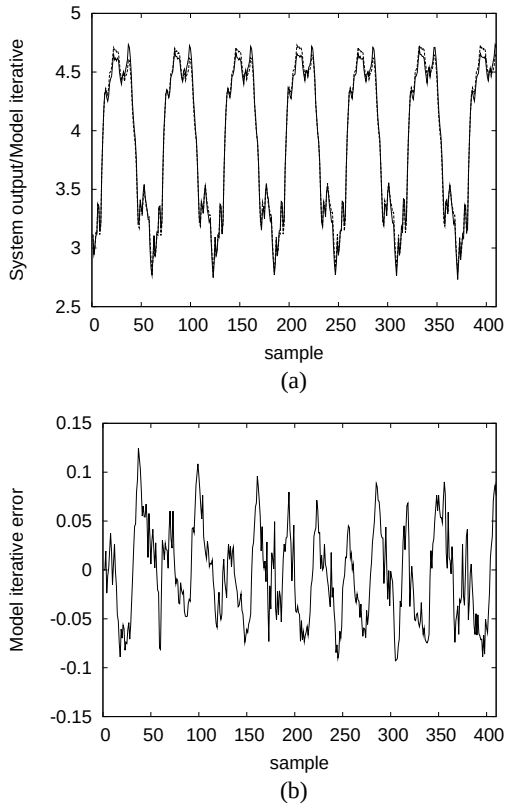

Fig. 4. Performance of the 15-term generalized Gaussian kernel model for the engine data set: (a) iterative model output (dashed) superimposed on system output (solid), and (b) iterative model error.

ious existing kernel regression techniques were used in [6] to fit a thin-plate-spline regression model for this system, and the best result obtained had a 31-term thin-plate-spline regression model with the MSE values of 0.003192 and 0.005892 over the training and validation sets, respectively.

Example 2. 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 a low engine speed. Detailed system description and experimental setup can be found in [24]. The data set contained 410 samples. The first 210 data points were used in training and the last 200 points in model validation. The previous results [5],[6] had shown that when fitting a Gaussian kernel model with a single common variance, $\sigma^2 = 1.69$ was the optimal value for this kernel variance, and the model input was given by $\mathbf{x}_k = [y_{k-1}\ u_{k-1}\ u_{k-2}]^T$. Various kernel modeling techniques were employed in [6] to fit this data set, and the best Gaussian kernel model constructed consisted of 22 terms. The MSE values of this model over the training and validation sets were 0.000453 and 0.000490, respectively.

The proposed modeling approach was applied to construct a generalized Gaussian kernel model, yielding a 15-term subset model. The MSE values of this model were 0.000482 over the training set and 0.000496 over the validation set, respectively. The model prediction $\hat{y}_k$ and prediction error $\epsilon_k = y_k - \hat{y}_k$ generated by this model are illustrated in Fig. 3. This obtained 15-term model was used to iteratively generate the model output $\hat{y}_{d,k} = f_m(\hat{\mathbf{x}}_{d,k})$ with the input vector $\hat{\mathbf{x}}_{d,k} = [\hat{y}_{d,k-1}\ u_{k-1}\ u_{k-2}]^T$. The iterative model output $\hat{y}_{d,k}$ and the iterative model error $\epsilon_{d,k} = y_k - \hat{y}_{d,k}$, are depicted in Fig. 4.

Example 3. The gas furnace data set (Series J in [25]) contained 296 pairs of input-output points, where the input u_k was the coded input gas feed rate and the output y_k was the CO_2 concentration. All the 296 data points were used in training, with the model input vector 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 previous experiments had found out that the existing state-of-the-art kernel regression techniques failed to fit a Gaussian kernel regression model using a common kernel variance [6]. Various existing kernel modeling methods were used in [6] to fit a thin-plate-spline regression model, and the best thin-plate-spline model obtained contained 28 terms with the training MSE 0.053306.

By adopting the proposed generalized kernel model approach, a 21-term generalized Gaussian kernel model was identified with the training MSE 0.053452. The model prediction and prediction error generated by this 21-term generalized Gaussian kernel model are shown in Fig. 5. The obtained model was also used to iteratively produce the model output $\hat{y}_{d,k} = f_m(\hat{\mathbf{x}}_{d,k})$ given the input $\hat{\mathbf{x}}_{d,k} = [\hat{y}_{d,k-1}\ \hat{y}_{d,k-2}\ \hat{y}_{d,k-3}\ u_{k-1}\ u_{k-2}\ u_{k-3}]^T$. The iterative model output $\hat{y}_{d,k}$ and the associated iterative modeling error $\epsilon_{d,k} = y_k - \hat{y}_{d,k}$ are illustrated in Fig. 6.

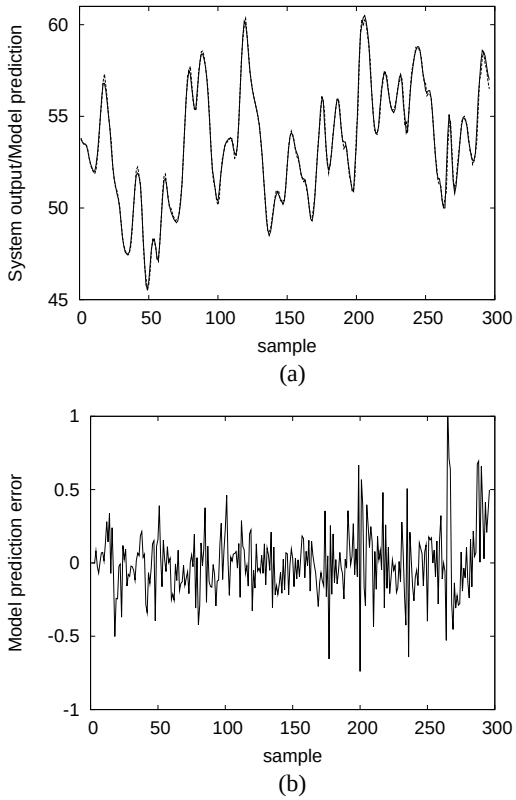

Fig. 5. Performance of the 21-term generalized Gaussian kernel model for the gas furnace data set: (a) model prediction (dashed) superimposed on system output (solid), and (b) model prediction error.

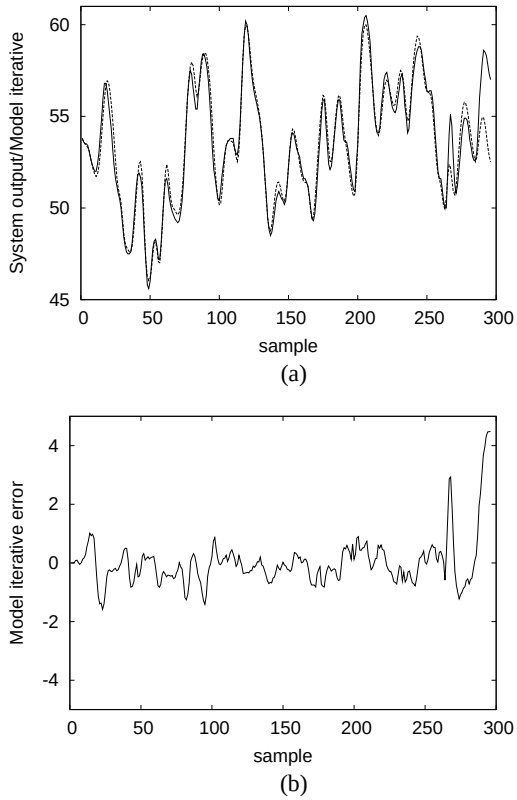

Fig. 6. Performance of the 21-term generalized Gaussian kernel model for the gas furnace data set: (a) iterative model output (dashed) superimposed on system output (solid), and (b) iterative model error.

V. Conclusions

Nonlinear system identification has been considered using a generalized kernel model. Each regressor in the generalized kernel model has an individually fitted diagonal covariance matrix, which is determined by maximizing a correlation criterion using a guided random search algorithm called the RWBS. The OLS algorithm based on the leave-one-out test statistic and local regularization then automatically selects a sparse model from the resulting pool of candidate regressors. The effectiveness of the proposed modeling approach has been demonstrated by the experimental results involving one simulated system and two real data sets.

Acknowledgement

S. Chen wish to thank the support of the United Kingdom Royal Academy of Engineering.

References

[1] 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.

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

[3] S. Chen, Y. Wu and B.L. Luk, "Combined genetic algorithm optimisation and regularised orthogonal least squares learning for radial basis function networks," *IEEE Trans. Neural Networks*, Vol.10, No.5, pp.1239–1243, 1999.

[4] 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 and Applications*, Vol.150, No.3, pp.245–254, 2003.

[5] S. Chen, X. Hong and C.J. Harris, "Sparse kernel regression modeling using combined locally regularized orthogonal least squares and D-optimality experimental design," *IEEE Trans. Automatic Control*, Vol.48, No.6, pp.1029–1036, 2003.

[6] S. Chen, X. Hong, C.J. Harris and P.M. Sharkey, "Sparse modeling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Trans. Systems, Man and Cybernetics, Part B*, Vol.34, No.2, pp.898–911, 2004.

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

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

[9] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, UK: Cambridge University Press, 2000.

[10] M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Machine Learning Research*, Vol.1, pp.211–244, 2001.

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

[12] W. Chu, S.S. Keerthi and C.J. Ong, "Bayesian support vector regression using a unified loss function," *IEEE Trans. Neural Networks*, Vol.15, No.1, pp.29–44, 2004.

[13] S. Chen, X.X. Wang and C.J. Harris, "Experiments with repeating weighted boosting search for optimization in signal processing applications," submitted to *IEEE Trans. Systems, Man and Cybernetics, Part B*, Vol.35, No.4, pp.682-693, 2005.

[14] R.E. Schapire, "The strength of weak learnability," *Machine Learning*, Vol.5, No.2, pp.197–227, 1990.

[15] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Computer and System Sciences*, Vol.55, No.1, pp.119–139, 1997.

[16] G. Ridgeway, D. Madigan and T. Richardson, "Boosting methodology for regression problems," in: D. Heckerman and J. Whittaker, eds., *Proc. Artificial Intelligence and Statistics*, 1999, pp.152–161.

[17] R. Meir and G. Rätsch, "An introduction to boosting and leveraging," in: S. Mendelson and A. Smola, eds., *Advanced Lectures in Machine Learning*. Springer Verlag, 2003, pp.119–184.
[18] 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.
[19] R.H. Myers, *Classical and Modern Regression with Applications*. 2nd Edition, Boston: PWS-KENT, 1990.
[20] L.K. Hansen and J. Larsen, "Linear unlearning for cross-validation," *Advances in Computational Mathematics*, Vol.5, pp.269–280, 1996.
[21] G. Monari and G. Dreyfus, "Withdrawing an example from the training set: an analytic estimation of its effect on a non-linear parameterised model," *Neurocomputing*, Vol.35, pp.195–201, 2000.
[22] G. Monari and G. Dreyfus, "Local overfitting control via leverages," *Neural Computation*, Vol.14, pp.1481–1506, 2002.
[23] K.S. Narendra and K. Parthasarathy, "Identification and control of dynamic systems using neural networks," *IEEE Trans. Neural Networks*, Vol.1, No.1, pp.4-27, 1990.
[24] S.A. Billings, S. Chen and R.J. Backhouse, "The identification of linear and non-linear models of a turbocharged automotive diesel engine," *Mechanical Systems and Signal Processing*, Vol.3, No.2, pp.123–142, 1989.
[25] G.E.P. Box and G.M. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden Day Inc., 1976.

APPENDIX: FIT KERNEL COVARIANCE MATRICES

The RWBS algorithm for fitting the lth regressor's covariance matrix is summarized. Let $\mathbf{c}$ be the n-dimensional vector containing the diagonal covariance matrix $\mathbf{\Sigma}_l$. Specify the RWBS algorithmic parameters: P_S – population size, N_G – number of generations in the repeated search, and ξ_B – accuracy for terminating the weighted boosting search.

•• **Outer loop: generations** For $n = 1 : N_G$

- *Generation initialization*: Initialize the population by setting $\mathbf{c}_1^{(n)} = \mathbf{c}_{\text{best}}^{(n-1)}$ and randomly generating rest of the population members $\mathbf{c}_i^{(n)}$, $2 \le i \le P_S$, where $\mathbf{c}_{\text{best}}^{(n-1)}$ denotes the solution found in the previous generation. If $n = 1$, $\mathbf{c}_1^{(n)}$ is also randomly chosen
- *Weighted boosting search initialization*: Assign the initial distribution weightings $\delta_i(0) = \frac{1}{P_S}$, $1 \le i \le P_S$, for the population, and calculate the cost function value of each point $\mathbf{c}_i^{(n)}$

$$h_i = 1 - |C(\mathbf{c}_i^{(n)})|,\ 1 \le i \le P_S$$

⋆⋆ **Inner loop: weighted boosting search** Set $t = 0$; For $t = t + 1$

- *Step 1: Boosting*
  1) Find

$$i_{\text{best}} = \arg \min_{1 \le i \le P_S} h_i$$

$$i_{\text{worst}} = \arg \max_{1 \le i \le P_S} h_i$$

  Denote $\mathbf{c}_{\text{best}}^{(n)} = \mathbf{c}_{i_{\text{best}}}^{(n)}$ and $\mathbf{c}_{\text{worst}}^{(n)} = \mathbf{c}_{i_{\text{worst}}}^{(n)}$
  2) Normalize the cost function values

$$\bar{h}_i = \frac{h_i}{\sum_{m=1}^{P_S} h_m},\ 1 \le i \le P_S$$

  3) Compute a weighting factor β_t according to

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

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

$$\delta_i(t) = \begin{cases} \delta_i(t-1)\beta_t^{\bar{h}_i}, & \text{for } \beta_t \le 1 \\ \delta_i(t-1)\beta_t^{1-\bar{h}_i}, & \text{for } \beta_t > 1 \end{cases}$$

  and normalize them

$$\delta_i(t) = \frac{\delta_i(t)}{\sum_{m=1}^{P_S} \delta_m(t)},\ 1 \le i \le P_S$$

- *Step 2: Parameter updating*
  1) Construct the (P_S+1)th point using the formula

$$\mathbf{c}_{P_S+1} = \sum_{i=1}^{P_S} \delta_i(t)\mathbf{c}_i^{(n)}$$

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

$$\mathbf{c}_{P_S+2} = \mathbf{c}_{\text{best}}^{(n)} + \left(\mathbf{c}_{\text{best}}^{(n)} - \mathbf{c}_{P_S+1}\right)$$

  3) Compute the cost function values $h_i = 1 - |C(\mathbf{c}_i)|$, $i = P_S+1, P_S+2$, for these two points and find

$$i_* = \arg \min_{i=P_S+1,P_S+2} h_i$$

  4) The pair $(\mathbf{c}_{i_*}, h_{i_*})$ then replaces $(\mathbf{c}_{\text{worst}}^{(n)}, h_{i_{\text{worst}}})$ in the population
- If $\|\mathbf{c}_{P_S+1} - \mathbf{c}_{P_S+2}\| < \xi_B$, exit **inner loop**

⋆⋆ **End of inner loop**
The solution found in the nth generation is $\mathbf{c}_{\text{best}}^{(n)}$

•• **End of outer loop**
This yields the solution $\mathbf{\Sigma}_l = \mathbf{c}_{\text{best}}^{(N_G)}$, i.e. the diagonal kernel covariance matrix of the lth regressor.