Multi-output regression using a locally regularised orthogonal least-squares algorithm

S. Chen

Abstract: The paper considers data modelling using multi-output regression models. A locally regularised orthogonal least-squares (LROLS) algorithm is proposed for constructing sparse multi-output regression models that generalise well. By associating each regressor in the regression model with an individual regularisation parameter, the ability of the multi-output orthogonal least-squares (OLS) model selection to produce a parsimonious model with a good generalisation performance is greatly enhanced.

1 Introduction

Data-modelling practicioners have traditionally relied on the parsimonious principle to combat over-fitting. Apart from the obvious computational advantage, small models often generalise better. Among various construction algorithms for producing parsimonious models, e.g. [1–6], the orthogonal least-squares (OLS) algorithm [1, 2] has certain advantages. A key feature of this algorithm is its ability to show the contribution of an individual selected model regressor to the modelling accuracy. This ability to provide quantitative information regarding the significance of an individual regressor enables the algorithm to select only the significant regressors and is responsible for producing parsimonious models. In practical modelling problems, full data matrices are usually ill-conditioned and often non-invertible. A simple mechanism is automatically built into the OLS algorithm to avoid any ill-conditioning of learning problems, and the algorithm does not require an inverse of the full data matrix, as many other data modelling algorithms do. The parsimonious principle alone, however, is not entirely immune to over-fitting. If the data are highly noisy, the small models constructed may still fit into the noise. A useful technique for overcoming over-fitting is regularisation [7, 8]. A uniformly regularised OLS (UROLS) algorithm [9] has been proposed for single-output regression, which employs a single uniform regularisation parameter for each weight in the model. From the Bayesian learning viewpoint, a regularisation parameter is equivalent to the ratio of the related hyperparameter to a noise parameter [10].

The Bayesian learning framework is perhaps the most general and powerful learning technique for data modelling. Various Bayesian learning methods can be categorised into three specific classes: the type-II maximum likelihood or evidence procedure, the Markov chain Monte Carlo sampling approach and the variational learning method. For a recent review of these Bayesian learning methods see, for example, [11]. All the Bayesian learning algorithms are conceptually complicated and computationally expensive. The evidence procedure, which iteratively optimises model parameters and associated hyperparameters [10], is relatively simple. Applying the evidence procedure to single-output kernel regression models leads to the relevance vector machine (RVM) method [12]. A key feature of the RVM is the introduction of an individual hyperparameter for each weight in the regression model. During the optimisation process, many of these hyperparameters are driven to large values, so that the corresponding model weights are effectively forced to be zero. A drawback of the RVM method is that the iterative optimisation process involved is inherently ill-conditioned, and numerically robust methods, such as the singular value decomposition or other pseudo-inverse algorithms, often have to be used to solve for the corresponding optimisation problem. Recent work [13] has combined the idea of OLS subset model selection with an individually regularised approach to derive a single-output LROLS algorithm, which does not suffer from this disadvantage.

For multi-output regression, the choice of construction algorithms is far less than for the single-output case. One approach is to fit multiple single-output models as, for example, in [14]. An alternative is to construct a single multi-output regression model as, for example, in [15]. The latter approach has some advantages: a selected regressor must be significant in explaining all the outputs, and this can result in a smaller number of regressors overall than the former approach to achieve the same modelling accuracy. This paper proposes combining the local regularisation approach with the multi-output OLS regression. For an effective updating of regularisation parameters, the single-output evidence procedure of [10] is extended to the multi-output case. In this proposed multi-output LROLS algorithm, regularisation is introduced in the orthogonal weight space, and the Hessian matrix needed for updating the regularisation parameters is diagonal. This offers considerable numerical advantages: the algorithm retains the ability to select significant regressors and local regularisation further enforces sparsity. The end result is therefore a very efficient yet simple algorithm for constructing sparse multi-output regression models that generalise well, especially under highly noisy learning conditions.

IEE Proceedings online no. 20020401

DOI: 10.1049/ip-vis:20020401

Paper first received 22nd October 2001 and in revised form 21st February 2002

The author is with the Department of Electronics and Computer Science, University of Southampton, Highfield, Southampton SO17 1BJ, UK

2 The multi-output regression model

Consider the multi-output regression model of the form

$$y_i(k) = \hat{y}_i(k) + e_i(k) = \sum_{j=1}^{M} \theta_{j,i}\phi_j(k) + e_i(k) \quad 1 \le k \le N \tag{1}$$

for $1 \le i \le n_o$, where $y_i(k)$ is the ith target or desired output, $e_i(k)$ is the error between $y_i(k)$ and the ith model output $\hat{y}_i(k)$, $\theta_{j,i}$ are the model weights, $\phi_i(k)$ are the regressors, M is the total number of candidate regressors, n_o the number of outputs and N the number of training samples. Define

$$\boldsymbol{y}_i = \begin{bmatrix} y_i(1) \\ y_i(2) \\ \vdots \\ y_i(N) \end{bmatrix} \quad \boldsymbol{e}_i = \begin{bmatrix} e_i(1) \\ e_i(2) \\ \vdots \\ e_i(N) \end{bmatrix} \quad \boldsymbol{\theta}_i = \begin{bmatrix} \theta_{1,i} \\ \theta_{2,i} \\ \vdots \\ \theta_{M,i} \end{bmatrix} \tag{2}$$

for $1 \le i \le n_o$ and

$$\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \quad \boldsymbol{\phi}_2 \quad \cdots \quad \boldsymbol{\phi}_M] \tag{3}$$

with

$$\boldsymbol{\phi}_j = [\phi_j(1) \quad \phi_j(2) \quad \cdots \quad \phi_j(N)]^T \quad 1 \le j \le M \tag{4}$$

The multi-output regression model (1) becomes

$$\boldsymbol{y}_i = \boldsymbol{\Phi}\boldsymbol{\theta}_i + \boldsymbol{e}_i \quad 1 \le i \le n_o \tag{5}$$

Further define

$$\boldsymbol{Y} = [\boldsymbol{y}_1 \quad \boldsymbol{y}_2 \quad \cdots \quad \boldsymbol{y}_{n_o}] \quad \boldsymbol{\Theta} = [\boldsymbol{\theta}_1 \quad \boldsymbol{\theta}_2 \quad \cdots \quad \boldsymbol{\theta}_{n_o}]$$
$$\boldsymbol{E} = [\boldsymbol{e}_1 \quad \boldsymbol{e}_2 \quad \cdots \quad \boldsymbol{e}_{n_o}] \tag{6}$$

The regression model (1) can be rewritten in the matrix form as

$$\boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{\Theta} + \boldsymbol{E} \tag{7}$$

Let an orthogonal decomposition of the regression matrix $\boldsymbol{\Phi}$ be

$$\boldsymbol{\Phi} = \boldsymbol{W}\boldsymbol{A} \tag{8}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,M} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{M-1,M} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{9}$$

and

$$\boldsymbol{W} = [\boldsymbol{w}_1 \quad \boldsymbol{w}_2 \quad \cdots \quad \boldsymbol{w}_M] \tag{10}$$

with orthogonal columns that satisfy $\boldsymbol{w}_j^T\boldsymbol{w}_l = 0$, if $j \ne l$. The regression model (7) can alternatively be expressed as

$$\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{G} + \boldsymbol{E} \tag{11}$$

where the orthogonal weight matrix

$$\boldsymbol{G} = [\boldsymbol{g}_1 \quad \boldsymbol{g}_2 \quad \cdots \quad \boldsymbol{g}_{n_o}] \tag{12}$$

with

$$\boldsymbol{g}_i = [g_{1,i} \quad g_{2,i} \quad \cdots \quad g_{M,i}]^T \quad 1 \le i \le n_o \tag{13}$$

and $\boldsymbol{G}$ satisfies the triangular system

$$\boldsymbol{A}\boldsymbol{\Theta} = \boldsymbol{G} \tag{14}$$

Knowing $\boldsymbol{A}$ and $\boldsymbol{G}$, $\boldsymbol{\Theta}$ can readily be solved from (14).

3 Multi-output locally regularised OLS algorithm

With local regularisation, each orthogonal regressor $\boldsymbol{w}_j$ has an associated regularisation parameter λ_j. Denote the regularisation parameter vector as $\boldsymbol{\lambda} = [\lambda_1 \ \lambda_2 \cdots \lambda_M]^T$ and a diagonal matrix $\boldsymbol{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_M\}$. The multi-output LROLS algorithm is based on the following regularised error criterion:

$$J_R(\boldsymbol{G}, \boldsymbol{\lambda}) = \text{trace}(\boldsymbol{E}^T\boldsymbol{E} + \boldsymbol{G}^T\boldsymbol{\Lambda}\boldsymbol{G}) = \sum_{i=1}^{n_o}(\boldsymbol{e}_i^T\boldsymbol{e}_i + \boldsymbol{g}_i^T\boldsymbol{\Lambda}\boldsymbol{g}_i) = \sum_{i=1}^{n_o}\boldsymbol{e}_i^T\boldsymbol{e}_i + \sum_{j=1}^{M}\left(\sum_{i=1}^{n_o} g_{j,i}^2\right)\lambda_j \tag{15}$$

The original multi-output OLS algorithm [15] can be viewed as a special case of this LROLS algorithm with $\lambda_j = 0$, $\forall j$. It is also possible to derive a multi-output UROLS algorithm by setting $\lambda_j = \lambda$, $\forall j$, just as in the single-output case [9].

After some simplification (see the Appendix, Section 7.1), the criterion (15) can be expressed as

$$\text{trace}(\boldsymbol{E}^T\boldsymbol{E} + \boldsymbol{G}^T\boldsymbol{\Lambda}\boldsymbol{G}) = \text{trace}(\boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{G}^T(\boldsymbol{W}^T\boldsymbol{W} + \boldsymbol{\Lambda})\boldsymbol{G}) \tag{16}$$

or

$$\text{trace}(\boldsymbol{E}^T\boldsymbol{E} + \boldsymbol{G}^T\boldsymbol{\Lambda}\boldsymbol{G}) = \sum_{i=1}^{n_o}\boldsymbol{y}_i^T\boldsymbol{y}_i - \sum_{j=1}^{M}\left(\sum_{i=1}^{n_o} g_{j,i}^2\right)(\boldsymbol{w}_j^T\boldsymbol{w}_j + \lambda_j) \tag{17}$$

Normalising (16) by $\text{trace}(\boldsymbol{Y}^T\boldsymbol{Y})$ yields

$$\frac{\text{trace}(\boldsymbol{E}^T\boldsymbol{E} + \boldsymbol{G}^T\boldsymbol{\Lambda}\boldsymbol{G})}{\text{trace}(\boldsymbol{Y}^T\boldsymbol{Y})} = 1 - \sum_{j=1}^{M}\frac{\left(\sum_{i=1}^{n_o} g_{j,i}^2\right)(\boldsymbol{w}_j^T\boldsymbol{w}_j + \lambda_j)}{\text{trace}(\boldsymbol{Y}^T\boldsymbol{Y})} \tag{18}$$

Define the regularised error reduction ratio due to the regressor $\boldsymbol{w}_l$ as

$$[err]_l = \frac{\left(\sum_{i=1}^{n_o} g_{l,i}^2\right)(\boldsymbol{w}_l^T\boldsymbol{w}_l + \lambda_l)}{\text{trace}(\boldsymbol{Y}^T\boldsymbol{Y})} \tag{19}$$

Based on this ratio, significant regressors can be selected in a forward-regression procedure, exactly as in the case of the multi-output OLS algorithm [15]. The selection is terminated at the M_sth stage when

$$1 - \sum_{l=1}^{M_s}[err]_l < \xi \tag{20}$$

is satisfied, where $0 < \xi < 1$ is a chosen tolerance. This produces a sparse model containing M_s ($\ll M$) significant regressors. The detailed algorithm-selection procedure is given in the Appendix (Section 7.2). Notice that, in the selection procedure, if $\boldsymbol{w}_l^T\boldsymbol{w}_l$ is too small (near zero), this term will not be selected. Thus, any ill-conditioning or singular situations can automatically be avoided.

The Bayesian evidence procedure [10] can readily be extended to the multi-output case and thus used to 'optimise' the regularisation parameters. From the

Bayesian viewpoint, the following error criterion is equivalent to the criterion (15):

$$J_B(\boldsymbol{G}, \boldsymbol{h}, \beta) = \frac{1}{2}\text{trace}(\beta \boldsymbol{E}^T\boldsymbol{E} + \boldsymbol{G}^T\boldsymbol{H}\boldsymbol{G}) = \frac{\beta}{2}\sum_{i=1}^{n_o}\boldsymbol{e}_i^T\boldsymbol{e}_i + \frac{1}{2}\sum_{j=1}^{M}\left(\sum_{i=1}^{n_o} g_{j,i}^2\right)h_j \quad (21)$$

where β is a noise parameter, $\boldsymbol{h} = [h_1\ h_2 \cdots h_M]^T$ is the hyperparameter vector and $\boldsymbol{H} = \text{diag}\{h_1, h_2, \ldots, h_M\}$. The relationship between a regularisation parameter and its corresponding hyperparameter is obviously given by

$$\lambda_i = \frac{h_i}{\beta} \quad (22)$$

Following MacKay [10], it can be shown that the log evidence for $\boldsymbol{h}$ and β is (see the Appendix, Section 7.3)

$$\log(\text{evidence}) = -\frac{1}{2}\text{trace}(\beta \boldsymbol{E}^T\boldsymbol{E} + \boldsymbol{G}^T\boldsymbol{H}\boldsymbol{G}) - \frac{1}{2}\log(\det(\boldsymbol{B})) + \frac{n_o}{2}\sum_{j=1}^{M}\log(h_j) + \frac{n_o N}{2}\log(\beta) + c \quad (23)$$

where c is a constant that does not depend on $\boldsymbol{h}$ and β, and the $(n_o M) \times (n_o M)$ diagonal matrix

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_0 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{B}_0 \end{bmatrix} \quad (24)$$

with the $M \times M$ diagonal matrix $\boldsymbol{B}_0$ given by

$$\boldsymbol{B}_0 = \boldsymbol{H} + \beta \boldsymbol{W}^T\boldsymbol{W} = \text{diag}\{h_1 + \beta \boldsymbol{w}_1^T\boldsymbol{w}_1, h_2 + \beta \boldsymbol{w}_2^T\boldsymbol{w}_2, \ldots, h_M + \beta \boldsymbol{w}_M^T\boldsymbol{w}_M\} \quad (25)$$

Setting the derivatives of log(evidence) with respect to $\boldsymbol{h}$ and β to zeros yields the updating formulas for $\boldsymbol{h}$ and β, respectively, as given in Section 7.3. Substituting these updating formulas into (22) results in the updating formulas for the regularisation parameters:

$$\lambda_j^{new} = \frac{\gamma_j^{old}}{N - \gamma^{old}} \cdot \frac{\sum_{i=1}^{n_o}\boldsymbol{e}_i^T\boldsymbol{e}_i}{\sum_{i=1}^{n_o} g_{j,i}^2} \quad 1 \le j \le M \quad (26)$$

where

$$\gamma_j = \frac{\boldsymbol{w}_j^T\boldsymbol{w}_j}{\lambda_j + \boldsymbol{w}_j^T\boldsymbol{w}_j} \quad (27)$$

and

$$\gamma = \sum_{j=1}^{M}\gamma_j \quad (28)$$

The iterative regression-model selection procedure can now be summarised:

Initialisation: Set λ_j, $1 \le j \le M$ to the same small positive value, e.g. 0.001.

Step 1. Given the current $\boldsymbol{\lambda}$, use the procedure described in Section 7.2 to select a subset model with M_s terms.

Step 2. Update $\boldsymbol{\lambda}$ using (26)–(28) with $M = M_s$. If $\boldsymbol{\lambda}$ remains sufficiently unchanged in two successive iterations or a pre-set maximum iteration number is reached, stop; otherwise go to step 1.

At the beginning of the iterative loop, the value of ξ for terminating the subset model selection can deliberately be chosen to be smaller than really needed, so that step 1 produces a M_s-term model which is larger than is really needed. This ensures that no significant terms are lost when $\boldsymbol{\lambda}$ is far from its optimal value. When $\boldsymbol{\lambda}$ has converged (typically after 10 to 30 iterations), an appropriate value of ξ should then be used to produce a parsimonious final model.

The ideal value of ξ can usually be learnt by interacting with the selection procedure [1, 16], or a cross-validation using a separate testing data set can be used to learn an appropriate value for ξ. Alternatively, the selection can be terminated when the Akaike information criterion

$$AIC(\chi) = N\log(\det(N^{-1}\boldsymbol{E}^T\boldsymbol{E})) + M_s\chi \quad (29)$$

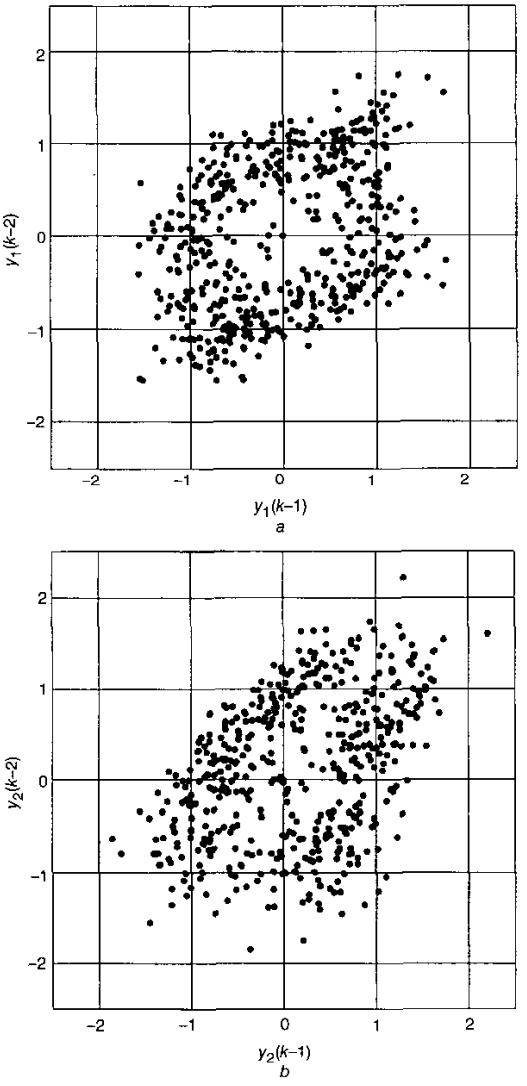

Fig. 1 *Two-dimensional representations of the noisy time series observations*

Initial conditions were $y_1(0) = y_1(-1) = y_2(0) = y_2(-1) = 0.0$, and the covariance of the noise was $0.04I_2$

a Phase plot of noisy time series $y_1(k)$

b Phase plot of noisy time series $y_2(k)$

Table 1: Comparison of the OLS and LROLS algorithms for the simulated two-output nonlinear time series modelling example

Algorithm	Training set Cov(*E*)	Testing set Cov(*E*)
OLS	3.404865×10^{-2} 4.109623×10^{-4} 4.109623×10^{-4} 3.359714×10^{-2}	5.330108×10^{-2} 3.248144×10^{-3} 3.248144×10^{-3} 4.879024×10^{-2}
	$\log(\det(\mathrm{Cov}(E))) = -6.77349$	$\log(\det(\mathrm{Cov}(E))) = -5.95610$
LROLS	3.550233×10^{-2} 5.746453×10^{-5} 5.746453×10^{-5} 3.481578×10^{-2}	5.070231×10^{-2} 2.858339×10^{-3} 2.858339×10^{-3} 4.560991×10^{-2}
	$\log(\det(\mathrm{Cov}(E))) = -6.69587$	$\log(\det(\mathrm{Cov}(E))) = -6.07293$

Cov(*E*) = one-step prediction error covariance

reaches its minimum [17], where χ is the critical value of the chi-squared distribution with one degree of freedom for a given level of significance. It should be pointed out, however, that the choice of ξ is less critical than the original OLS algorithm. In the original OLS selection procedure, when data is very noisy, it is possible that the normalised error measure $1 - \sum [err]_l$ continuously decreases as more terms are added. This may lead to over-fitting unless the value of ξ is chosen carefully. As is demonstrated in the single-output case [13], multiple regularisers enforce sparsity, and $1 - \sum [rerr]_l$ will not continuously decrease as more terms are added. This is because those unnecessarily added terms will have a very large λ_l associated with them, effectively forcing their weights to be zero. This also helps to determine how many regressors to include in the final model.

4 Nonlinear system modelling examples

Three examples are used to illustrate the multi-output LROLS algorithm and to compare it with the original OLS algorithm. The regression model employed is the multi-output radial basis function (RBF) network of the form:

$$\hat{y}_i(k) = \sum_{j=1}^{M} \theta_{j,i}\phi_j(k) = \sum_{j=1}^{M} \theta_{j,i}\phi(\|\boldsymbol{x}(k) - \boldsymbol{c}_j\|) \quad 1 \le i \le n_o \tag{30}$$

with the thin-plate-spline function

$$\phi(r) = r^2 \log(r) \tag{31}$$

where the input vector to the RBF network is

$$\boldsymbol{x}(k) = [x_1(k) \quad x_2(k) \quad \cdots \quad x_{n_I}(k)]^T \tag{32}$$

and $\boldsymbol{c}_j$, of dimension n_I, are the RBF centres.

Example 1: This was a simulated two-output time-series process. The data set contained 1000 noisy observations generated using the model

$$\begin{aligned} y_1(k) &= 0.1\sin(\pi y_2(k-1)) \\ &\quad + (0.8 - 0.5\exp(-y_1^2(k-1)))y_1(k-1) \\ &\quad - (0.3 + 0.9\exp(-y_1^2(k-1)))y_1(k-2) + \epsilon_1(k) \\ y_2(k) &= 0.6y_2(k-1) + 0.2y_2(k-1)y_2(k-2) \\ &\quad + 1.2\tanh(y_1(k-2)) + \epsilon_2(k) \end{aligned} \tag{33}$$

given the initial conditions $y_1(0) = y_1(-1) = y_2(0) = y_2(-1) = 0$, where the zero-mean Gaussian noise $\boldsymbol{\epsilon}(k) = [\epsilon_1(k) \ \epsilon_2(k)]^T$ had a covariance $0.04\boldsymbol{I}_2$ with $\boldsymbol{I}_2$ being the 2×2 identity matrix. The first 500 data samples were used for training and the other 500 samples for validating the obtained model. The noisy training data set is depicted in Fig. 1. A two-output RBF network was used to model this

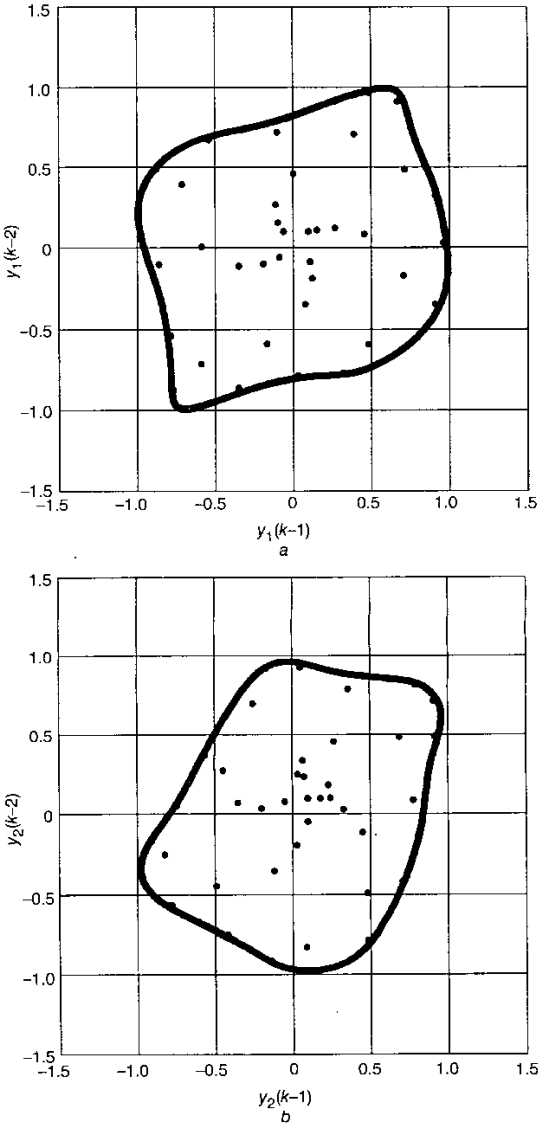

Fig. 2 *Two-dimensional representations of the noise-free time series observations*

Initial conditions were $y_{d1}(0) = y_{d1}(-1) = y_{d2}(0) = y_{d2}(-1) = 0.1$
a Phase plot of noise-free time series $y_{d1}(k)$
b Phase plot of noise-free time series $y_{d2}(k)$

time series, with the input vector to the RBF network given by

$$x(k) = [y_1(k-1) \quad y_1(k-2) \quad y_2(k-1) \quad y_2(k-2)]^T \tag{34}$$

As each training input was used as a candidate RBF centre, the number of candidate regressors M in (30) was 500.

In the previous study [15], the OLS algorithm identified a RBF network of 50 centres for this times series, where the noise covariance was $0.01I_2$. In the current example, the noise level was much higher. Both the OLS and LROLS algorithm were used to construct RBF networks of 50 centres. The covariances of the resulting network prediction errors between the noisy observations $y_i(k)$ and the one-step network predictions $\hat{y}_i(k)$, $i = 1, 2$, over both the training and testing sets are listed in Table 1. It can be seen that the generalisation performance of the LROLS algorithm is better than that of the OLS algorithm. The underlying dynamics of the simulated time series was governed by

$$\begin{aligned} y_{d1}(k) &= 0.1 \sin(\pi y_{d2}(k-1)) \\ &\quad + (0.8 - 0.5\exp(-y_{d1}^2(k-1)))y_{d1}(k-1) \\ &\quad - (0.3 + 0.9\exp(-y_{d1}^2(k-1)))y_{d1}(k-2) \\ y_{d2}(k) &= 0.6y_{d2}(k-1) + 0.2y_{d2}(k-1)y_{d2}(k-2) \\ &\quad + 1.2\tanh(y_{d1}(k-2)) \end{aligned} \tag{35}$$

Given the initial conditions $y_{d1}(0) = y_{d1}(-1) = y_{d2}(0) = y_{d2}(-1) = 0.1$, the response of this noise-free time series is depicted in Fig. 2. The generalisation capability of an identified model can best be tested by examining the iterative model output. If the iterative model output can

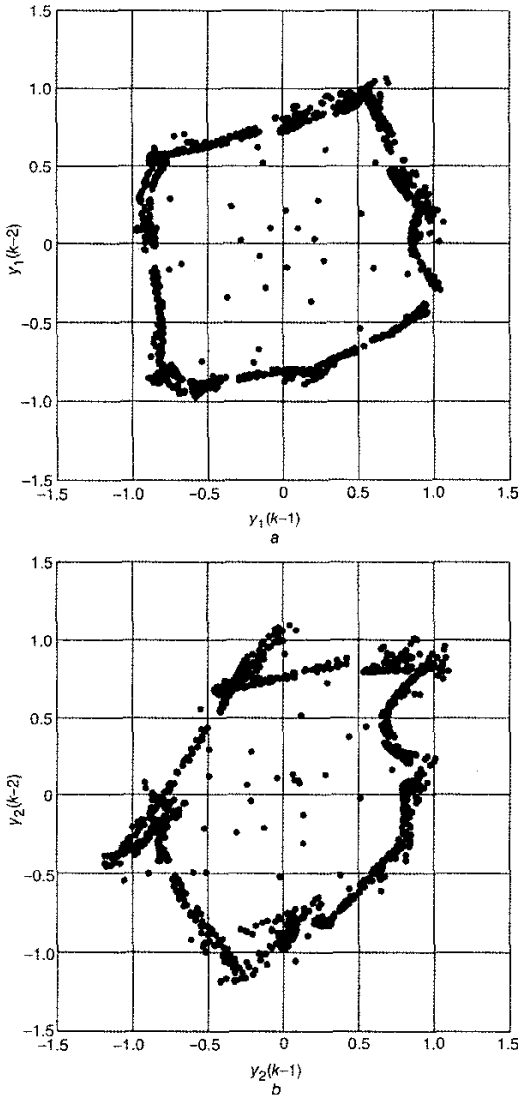

Fig. 3 *Two-dimensional representations of the iterative model outputs*

Initial conditions were $\hat{y}_{d1}(0) = \hat{y}_{d1}(-1) = \hat{y}_{d2}(0) = \hat{y}_{d2}(-1) = 0.1$, and the model was constructed by the OLS algorithm using very noisy data
a Phase plot of the iterative model output $\hat{y}_{d1}(k)$
b Phase plot of the iterative model output $\hat{y}_{d2}(k)$

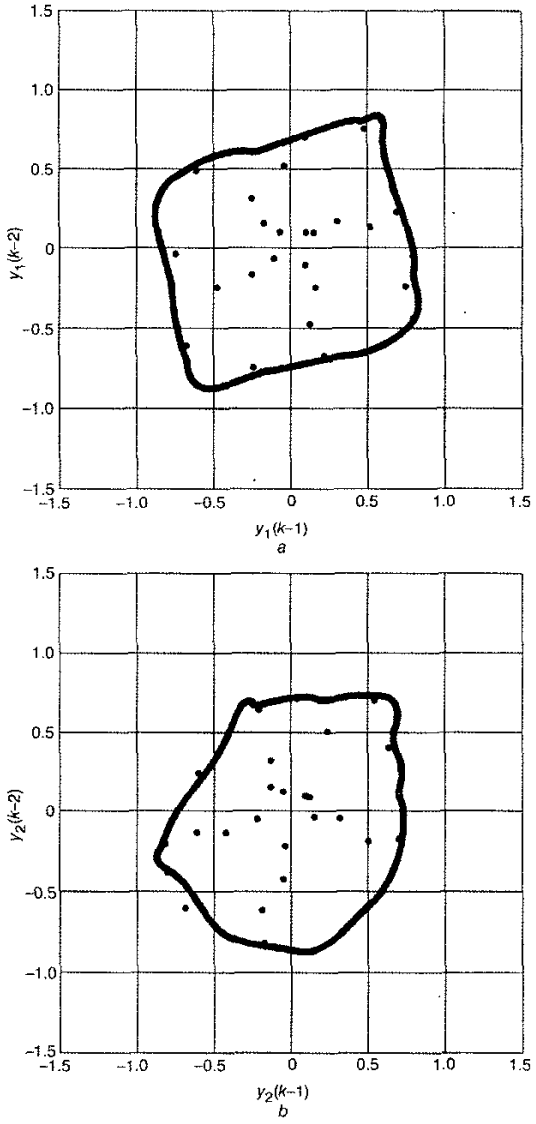

Fig. 4 *Two-dimensional representations of the iterative model outputs*

Initial conditions were $\hat{y}_{d1}(0) = \hat{y}_{d1}(-1) = \hat{y}_{d2}(0) = \hat{y}_{d2}(-1) = 0.1$, and the model was constructed by the LROLS algorithm using very noisy data
a Phase plot of the iterative model output $\hat{y}_{d1}(k)$
b Phase plot of the iterative model output $\hat{y}_{d2}(k)$

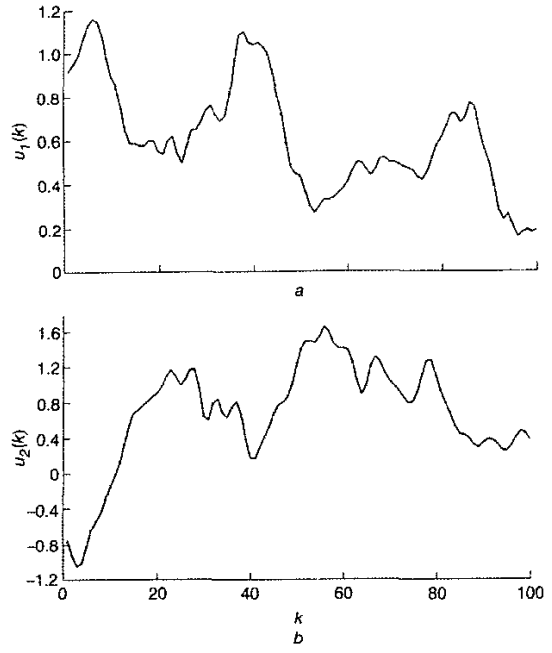

Fig. 5 *System input data for the turbo-alternator example*
a In-phase current deviation $u_1(k)$
b Out-of-phase current deviation $u_2(k)$

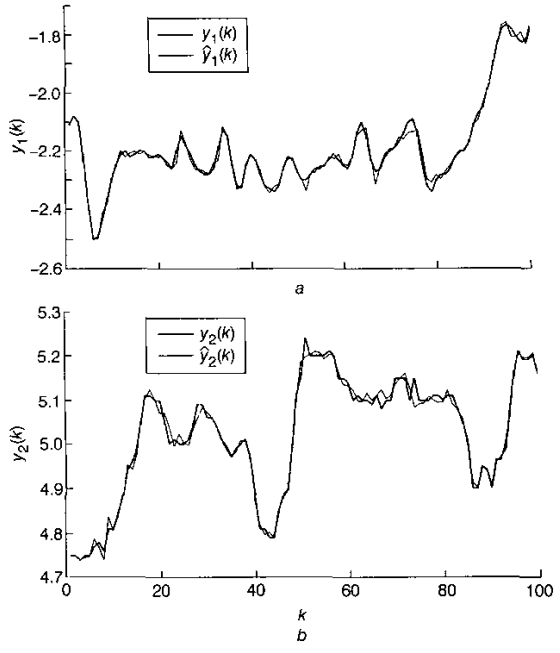

Fig. 6 *One-step predictions superimposed on turbo-alternator outputs*
The 45-term model was constructed by the LROLS algorithm
a Voltage deviation/one-step prediction $y_1(k)/\hat{y}_1(k)$
b Frequency deviation/one-step prediction $y_2(k)/\hat{y}_2(k)$

closely realise the behaviour shown in Fig. 2, the identified model truly captures the underlying dynamics of the system and does not simply fit the noise contained in the training data. Given the same initial conditions, the two RBF models identified by the OLS and LROLS algorithms were used to iteratively generate the network outputs $\hat{y}_{di}(k)$, $i = 1, 2$, with the input

$$\boldsymbol{x}_d(k) = [\hat{y}_{d1}(k-1) \quad \hat{y}_{d1}(k-2) \quad \hat{y}_{d2}(k-1) \quad \hat{y}_{d2}(k-2)]^T \tag{36}$$

The iterative model outputs so generated are plotted in Figs. 3 and 4, respectively. It can be seen that the model constructed by the LROLS algorithm captured the underlying dynamics of the system better than the OLS algorithm did.

Example 2: This example was a two-input two-output data set collected from a turbo-alternator [18]. The data set contained 100 samples. The system inputs, the in-phase current deviation $u_1(k)$ and the out-of-phase current deviation $u_2(k)$, are plotted in Fig. 5; whereas the system outputs, the voltage deviation $y_1(k)$ and the frequency deviation $y_2(k)$ are shown in Fig. 6. The two-output RBF network with the input vector

$$\begin{aligned}\boldsymbol{x}(k) = [&y_1(k-1) \quad y_1(k-2) \quad y_1(k-3) \quad y_2(k-1) \\ &y_2(k-2) \quad y_2(k-3) \quad u_1(k-1) \\ &u_1(k-2) \quad u_2(k-1) \quad u_2(k-2)]^T\end{aligned} \tag{37}$$

was used to fit this data set. In the previous study [15], the OLS algorithm constructed a 45-centre RBF model for this example. As this data set contained very low noise, it was expected that the LROLS algorithm should produce a similar model. The modelling accuracies of the two 45-centre RBF networks constructed by the OLS and LROLS algorithms, respectively, are compared in Table 2. The model validation in this case was performed by evaluating the iterative model outputs $\hat{y}_{di}(k)$, $i = 1, 2$, with the input

$$\begin{aligned}\boldsymbol{x}_d(k) = [&\hat{y}_{d1}(k-1) \quad \hat{y}_{d1}(k-2) \quad \hat{y}_{d1}(k-3) \quad \hat{y}_{d2}(k-1) \\ &\hat{y}_{d2}(k-2) \quad \hat{y}_{d2}(k-3) \quad u_1(k-1) \\ &u_1(k-2) \quad u_2(k-1) \quad u_2(k-2)]^T\end{aligned} \tag{38}$$

Table 2: Comparison of the OLS and LROLS algorithms for the turbo-alternator modelling example

Algorithm	Training set Cov($\boldsymbol{E}$)	Training set Cov($\boldsymbol{E}_d$)
OLS	2.698050×10^{-4} -1.011401×10^{-5} -1.011401×10^{-5} 2.565515×10^{-4}	4.980833×10^{-4} -2.739734×10^{-4} -2.739734×10^{-4} 9.454893×10^{-4}
	$\log(\det(\mathrm{Cov}(\boldsymbol{E}))) = -16.4875$	$\log(\det(\mathrm{Cov}(\boldsymbol{E}_d))) = -14.7422$
LROLS	2.703650×10^{-4} -1.521883×10^{-5} -1.521883×10^{-5} 2.307177×10^{-4}	4.885013×10^{-4} -2.652641×10^{-4} -2.652641×10^{-4} 9.176416×10^{-4}
	$\log(\det(\mathrm{Cov}(\boldsymbol{E}))) = -16.5938$	$\log(\det(\mathrm{Cov}(\boldsymbol{E}_d))) = -14.7886$

Cov($\boldsymbol{E}$) = one-step prediction error covariance, Cov($\boldsymbol{E}_d$) = iterative model error covariance

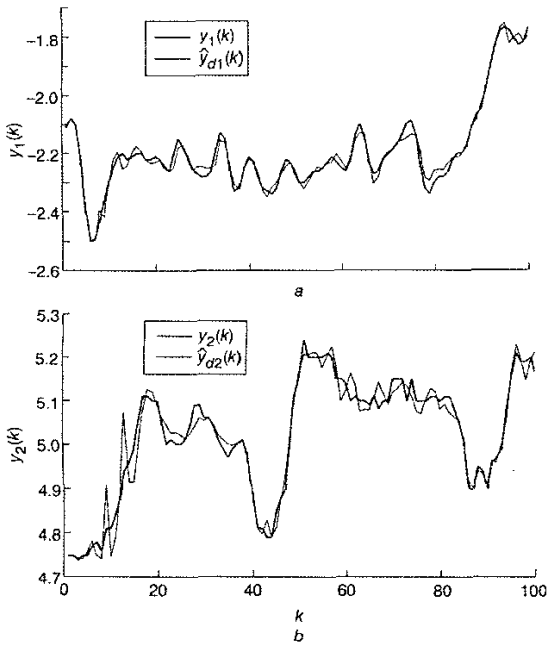

Fig. 7 *Iterative model outputs superimposed on turbo-alternator outputs*

The 45-term model was constructed by the LROLS algorithm
a Voltage deviation/iterative output $y_1(k)/\hat{y}_{d1}(k)$
b Frequency deviation/iterative output $y_2(k)/\hat{y}_{d2}(k)$

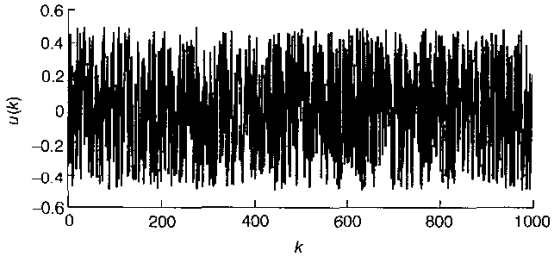

Fig. 8 *System input of the simulated single-input two-output nonlinear system example*

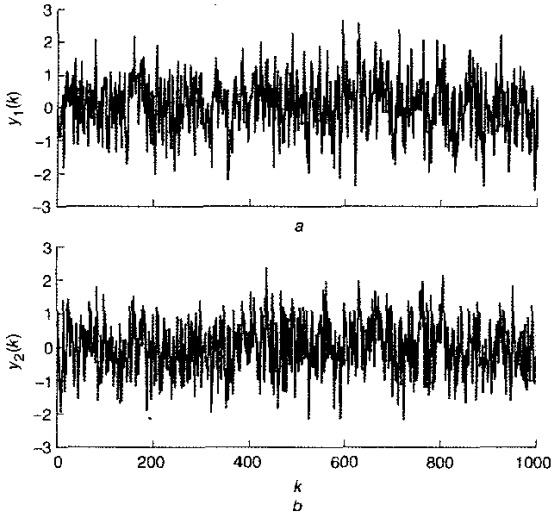

Fig. 9 *Two system outputs of the simulated single-input two-output nonlinear system example*

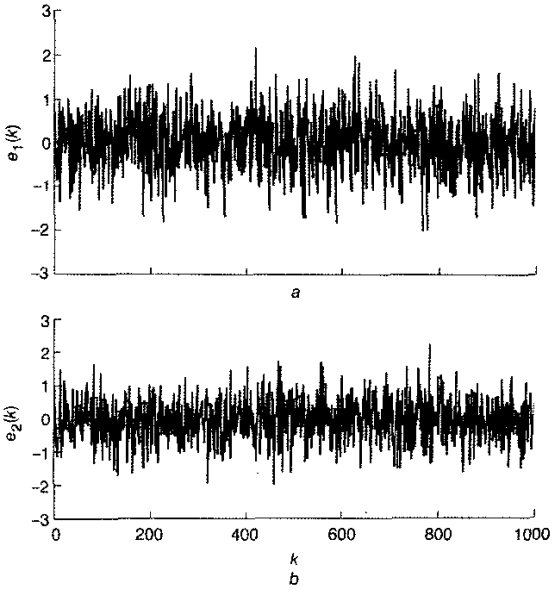

Fig. 10 *Two system noises of the simulated single-input two-output nonlinear system example*

The results given in Table 2 show that the model constructed by the LROLS algorithm is marginally better than the model constructed by the OLS algorithm. The one-step model predictions and the iterative model outputs are

Table 3: OLS selection procedure for the simulated single-input two-output nonlinear system modelling example

Stage l	Accuracy $1-\sum [err]_l$
1	0.9266648402
2	0.7812246356
3	0.6857712253
4	0.6712676332
5	0.6617631403
⋮	⋮
53	0.5513383266
54	0.5490380984
55	0.5468145452
56	0.5444882713
57	0.5420129044
58	0.5396682820
59	0.5370542849
60	0.5336903155
⋮	⋮
69	0.5124715883
70	0.5099673726
71	0.5077152658
72	0.5054002839
73	0.5028909486
74	0.5000084333
75	0.4975149006
76	0.4948750220
⋮	⋮

Table 4: LROLS selection procedure for the simulated single-input two-output nonlinear system modelling example after λ has converged

Stage l	Accuracy $1-\sum[err]_l$	Weights $\theta_{l,1}$, $\theta_{l,2}$		Regulariser λ_l
1	0.9290445009	-1.40107×10^{-1}	2.28733×10^{-1}	9.26011×10^{2}
2	0.8102458145	7.41665×10^{-1}	-2.40987×10^{-1}	1.71147×10^{2}
3	0.7230709198	-5.52066×10^{-1}	2.85684×10^{-1}	8.01033
4	0.7048683716	1.15299×10^{-1}	-5.39292×10^{-3}	8.71645×10
5	0.6917827744	-1.10061	-1.79131×10^{-1}	4.34778×10
⋮	⋮	⋮	⋮	⋮
51	0.6314173375	2.40694×10^{-2}	-1.72133×10^{-2}	2.20599×10
52	0.6313490611	-3.83076×10^{-2}	-1.00672×10^{-2}	1.14253×10
53	0.6312695431	-7.54805×10^{-3}	-7.26621×10^{-3}	1.38760×10^{2}
54	0.6312211359	-5.63049×10^{-3}	-2.18998×10^{-2}	2.57337×10
55	0.6311776920	4.03370×10^{-2}	-2.10492×10^{-2}	1.94738×10
56	0.6311154361	2.06592×10^{-2}	-2.15718×10^{-2}	1.71534×10
57	0.6310975544	1.09999×10^{-2}	1.35215×10^{-3}	7.28739×10
58	0.6310970858	3.60116×10^{-5}	-1.53099×10^{-4}	1.44289×10^{4}
59	0.6310970693	6.47015×10^{-6}	9.37922×10^{-6}	1.01024×10^{5}
60	0.6310970658	9.26853×10^{-7}	3.56768×10^{-6}	1.46341×10^{5}
61	0.6310970638	-4.94328×10^{-6}	-1.15381×10^{-6}	1.13584×10^{5}
62	0.6310970617	1.97632×10^{-6}	-1.09804×10^{-6}	3.44474×10^{5}
63	0.6310970617	-1.64595×10^{-10}	4.03026×10^{-10}	2.84426×10^{9}
64	0.6310970617	-2.03159×10^{-12}	7.91477×10^{-12}	1.22579×10^{11}
65	0.6310970617	3.65183×10^{-16}	1.11422×10^{-15}	5.42861×10^{16}
66	0.6310970617	2.61706×10^{-18}	1.62022×10^{-17}	6.09163×10^{16}
67	0.6310970617	-7.07541×10^{-25}	3.91732×10^{-23}	5.30902×10^{22}
68	0.6310970617	1.02075×10^{-27}	-7.85205×10^{-28}	3.36203×10^{27}
69	0.6310970617	1.78144×10^{-30}	1.60239×10^{-28}	1.87551×10^{28}
70	0.6310970617	-2.34416×10^{-32}	-1.12397×10^{-32}	6.75809×10^{31}
71	0.6310970617	9.08491×10^{-47}	-1.11975×10^{-46}	1.06856×10^{46}

superimposed on the turbo-alternator outputs in Figs. 6 and 7, respectively, which were very similar to those shown in [15].

Example 3: This was a simulated single-input two-output nonlinear system. The data were generated using the model

$$\begin{aligned} y_1(k) &= 0.5y_1(k-1) + u(k-1) + 0.4\tanh(u(k-2)) \\ &\quad + 0.1\sin(\pi y_1(k-2))y_2(k-1) + \epsilon_1(k) \\ y_2(k) &= 0.3y_2(k-1) + 0.1y_2(k-2)y_1(k-1) \\ &\quad + 0.4\exp(-u^2(k-1))y_1(k-2) + \epsilon_2(k) \end{aligned} \tag{39}$$

where the system input $u(k)$ was uniformly distributed in $(-0.5, 0.5)$, and the system noises $\boldsymbol{\epsilon}(k) = [\epsilon_1(k)\ \epsilon_2(k)]^T$ were Gaussian with zero means and covariance $0.4\boldsymbol{I}_2$. Figs. 8 and 9 show the system inputs and outputs, respectively. Notice that the system outputs were 'buried' in noise. This can be confirmed by observing the noise realisations used to generate the data, given in Fig. 10. The first 500 data points were used for training, and the two-output RBF network with the input

$$\begin{aligned} \boldsymbol{x}(k) = [&y_1(k-1) \quad y_1(k-2) \quad y_2(k-1) \\ &y_2(k-2) \quad u(k-1) \quad u(k-2)]^T \end{aligned} \tag{40}$$

was employed to fit the training data. The last 500 data samples were used for model validation. The goodness of a fitted model was also evaluated by computing the iterative model outputs with the input

$$\begin{aligned} \boldsymbol{x}_d(k) = [&\hat{y}_{d1}(k-1) \quad \hat{y}_{d1}(k-2) \quad \hat{y}_{d2}(k-1) \\ &\hat{y}_{d2}(k-2) \quad u(k-1) \quad u(k-2)]^T \end{aligned} \tag{41}$$

Because this data set was extremely noisy, the normalised error measure $1-\sum[err]_l$ continuously decreased as more terms were added by the OLS model-selection procedure, as illustrated in Table 3. This would certainly lead to over-fitting. Thus, the value of ξ used to terminate selection was critical in this case for the OLS algorithm. The LROLS selection procedure, after $\boldsymbol{\lambda}$ had converged, is listed in Table 4. Two observations can be made here. First, the modelling accuracy $1-\sum[err]_l$ did not continuously decrease as more terms were added by the LROLS selection procedure. In this particular example, $1-\sum[err]_l$ remained unchanged after the $l=61$ stage. This clearly indicated that the model should contain no more than the first 62 selected terms. Secondly, the regularisation parameters related to the terms from 58 onwards were very large and the corresponding weights were effectively zeros. This clearly indicated that a 57-term model was sufficient. This desired property of enforcing sparsity by local regularisation is very useful in helping to terminate the model-selection procedure at an appropriate stage without using costly cross-validation based on a separate testing data set.

Table 5: Comparison of the OLS and LROLS algorithms for the simulated single-input two-output nonlinear system modelling example

57-term model	OLS		LROLS	
Training set	3.332288×10^{-1}	2.184405×10^{-2}	3.640266×10^{-1}	2.307056×10^{-2}
Cov($\boldsymbol{E}$)	2.184405×10^{-2}	3.224246×10^{-1}	2.307056×10^{-2}	3.718092×10^{-1}
log(det(Cov($\boldsymbol{E}$)))	−2.23526		−2.00385	
Testing set	5.356972×10^{-1}	3.857507×10^{-2}	4.977092×10^{-1}	2.699694×10^{-2}
Cov($\boldsymbol{E}$)	3.857507×10^{-2}	4.888409×10^{-1}	2.699694×10^{-2}	4.380525×10^{-1}
log(det(Cov($\boldsymbol{E}$)))	−1.34560		−1.52650	
Iterative model	6.011033×10^{-1}	7.425149×10^{-2}	5.803848×10^{-1}	7.540281×10^{-2}
Cov($\boldsymbol{E}_d$)	7.425149×10^{-2}	6.193378×10^{-1}	7.540281×10^{-2}	5.806359×10^{-1}
log(det(Cov($\boldsymbol{E}_d$)))	−1.00301		−1.10471	
71-term model	**OLS**		**LROLS**	
Training set	3.100165×10^{-1}	1.578568×10^{-2}	3.640265×10^{-1}	2.307079×10^{-2}
Cov($\boldsymbol{E}$)	1.578568×10^{-2}	3.041483×10^{-1}	2.307079×10^{-2}	3.718081×10^{-1}
log(det(Cov($\boldsymbol{E}$)))	−2.36402		−2.00385	
Testing set	5.595245×10^{-1}	3.987048×10^{-2}	4.977094×10^{-1}	2.699648×10^{-2}
Cov($\boldsymbol{E}$)	3.987048×10^{-2}	4.958190×10^{-1}	2.699648×10^{-2}	4.380519×10^{-1}
log(det(Cov($\boldsymbol{E}$)))	−1.28796		−1.52651	
Iterative model	6.337929×10^{-1}	9.950650×10^{-2}	5.803847×10^{-1}	7.540273×10^{-2}
Cov($\boldsymbol{E}_d$)	9.950650×10^{-2}	6.496305×10^{-1}	7.540273×10^{-2}	5.806359×10^{-1}
log(det(Cov($\boldsymbol{E}_d$)))	−0.91173		−1.10471	

Cov($\boldsymbol{E}$) = one-step prediction error covariance, Cov($\boldsymbol{E}_d$) = iterative model error covariance

The modelling accuracies of the 57-term and 71-term RBF models constructed by the OLS and LROLS algorithms are compared in Table 5. From Table 5, it can be seen that the training-error variances of the models identified by the OLS algorithm were clearly smaller than the system-noise variances, indicating that the models were fitted into the noise, and moreover over-fitting of the 71-term model was more serious than that of the 57-term model. The two models identified by the LROLS did not appear to suffer from over-fitting, and they had the same generalisation accuracy, which was much better than the models constructed by the OLS algorithm without regularisation.

5 Conclusions

A locally regularised OLS algorithm has been developed for constructing sparse multi-output regression models. This multi-output LROLS algorithm combines both the advantages of OLS model selection, which has the ability to select only those significant regressors to explain training data, and local regularisation, which enforces the sparsity of the models. The end result is an efficient construction algorithm that is capable of producing sparse multi-output regression models with excellent generalisation performances. As regularisation is introduced in the orthogonal weight space, the computational requirements of the iterative model selection procedure are simple and straightforward. Any numerical ill-conditioning problems can automatically be avoided. It has also been shown that the decision on when to terminate the model selection procedure is greatly assisted by local regularisation.

6 References

1 CHEN, S., BILLINGS, S.A., and LUO, W.: 'Orthogonal least squares methods and their application to non-linear system identification', *Int. J. Control*, 1989, **50**, (5), pp. 1873–1896

2 CHEN, S., COWAN, C.F.N., and GRANT, P.M.: 'Orthogonal least squares learning algorithm for radial basis function networks', *IEEE Trans. Neural Netw.*, 1991, **2**, (2), pp. 302–309

3 FRIEDMAN, J.H.: 'Multivariate adaptive regression splines', *Ann. Stat.*, 1991, **19**, (1), pp. 1–141

4 KAVLI, T.: 'ASMOD: an algorithm for adaptive spline modeling of observation data', *Int. J. Control*, 1993, **58**, (4), pp. 947–968

5 BROWN, M., and HARRIS, C.J.: 'Neurofuzzy adaptive modeling and control' (Prentice-Hall, Englewood Cliffs, NJ, 1994)

6 CHEN, S.: 'Basis pursuit'. PhD thesis, Department of Statistics, Stanford University, 1995

7 HOERL, A.E., and KENNARD, R.W.: 'Ridge regression: biased estimation for non-orthogonal problems', *Technometrics*, 1970, **12**, pp. 55–67

8 BISHOP, C.M.: 'Improving the generalisation properties of radial basis function neural networks', *Neural Comput.*, 1991, **3**, (4), pp. 579–588

9 CHEN, S., CHNG, E.S., and ALKADHIMI, K.: 'Regularised orthogonal least squares algorithm for constructing radial basis function networks', *Int. J. Control*, 1996, **64**, (5), pp. 829–837

10 MACKAY, D.J.C.: 'Bayesian interpolation', *Neural Comput.*, 1992, **4**, (3), pp. 415–447

11 KANDOLA, J.S.: 'Interpretable modelling with sparse kernels'. PhD thesis, Department of Electronics and Computer Science, University of Southampton, UK, 2001

12 TIPPING, M.E.: 'The relevance vector machine' in SOLLA, S.A., LEEN, T.K., and MÜLLER, K.-R. (Eds.,): 'Advances in neural information processing systems 12' (MIT Press, Cambridge, MA, 2000)

13 CHEN, S.: 'Kernel-based data modelling using orthogonal least squares selection with local regularisation'. Proceedings of the 7th Annual Chinese Automation and Computer Science Conference in UK, Nottingham, UK, 2001 pp. 27–30

14 BILLINGS, S.A., CHEN, S., and KORENBERG, M.J.: 'Identification of MIMO non-linear systems using a forward-regression orthogonal estimator', *Int. J. Control*, 1989, **49**, pp. 2157–2189

15 CHEN, S., GRANT, P.M., and COWAN, C.F.N.: 'Orthogonal least squares algorithm for training multi-output radial basis function networks', *IEE Proc. F, Radar Signal Process*, 1992, **139**, (6), pp. 378–384

16 BILLINGS, S.A., and CHEN, S.: 'Extended model set, global data and threshold model identification of severely non-linear systems', *Int. J. Control*, 1989, **50**, (5), pp. 1897–1923

17 LEONTARITIS, I.J., and BILLINGS, S.A.: 'Model selection and validation methods for non-linear systems', *Int. J. Control*, 1987, **45**, (1), pp. 311–341

18 JENKINS, G.M., and WATTS, D.G.: 'Spectral analysis and its applications' (Holden-Day, San Francisco, 1986), Appendix A11.3

19 CHEN, S., and WIGGER, J.: 'Fast orthogonal least squares algorithm for efficient subset model selection', *IEEE Trans. Signal Process.*, 1995, **43**, (7), pp. 1713–1715

7 Appendixes

7.1 Simplification of criterion (15)

The 'least squares' solution for $\boldsymbol{G}$ is obtained by setting $\partial J_R/\partial \boldsymbol{G} = \boldsymbol{0}$, that is

$$\boldsymbol{W}^T\boldsymbol{Y} = (\boldsymbol{W}^T\boldsymbol{W} + \boldsymbol{\Lambda})\boldsymbol{G} \tag{42}$$

Now

$$\begin{aligned}\boldsymbol{Y}^T\boldsymbol{Y} - 2\boldsymbol{G}^T\boldsymbol{\Lambda}\boldsymbol{G} &= (\boldsymbol{W}\boldsymbol{G} + \boldsymbol{E})^T(\boldsymbol{W}\boldsymbol{G} + \boldsymbol{E}) - 2\boldsymbol{G}^T\boldsymbol{\Lambda}\boldsymbol{G} \\ &= \boldsymbol{G}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{G} + \boldsymbol{E}^T\boldsymbol{E} + \boldsymbol{G}^T\boldsymbol{W}^T\boldsymbol{E} \\ &\quad + \boldsymbol{E}^T\boldsymbol{W}\boldsymbol{G} - 2\boldsymbol{G}^T\boldsymbol{\Lambda}\boldsymbol{G}\end{aligned} \tag{43}$$

Noting (42),

$$\begin{aligned}\boldsymbol{G}^T\boldsymbol{W}^T\boldsymbol{E} - \boldsymbol{G}^T\boldsymbol{\Lambda}\boldsymbol{G} &= \boldsymbol{G}^T\boldsymbol{W}^T(\boldsymbol{Y} - \boldsymbol{W}\boldsymbol{G}) - \boldsymbol{G}^T\boldsymbol{\Lambda}\boldsymbol{G} \\ &= \boldsymbol{G}^T(\boldsymbol{W}^T\boldsymbol{Y} - \boldsymbol{W}^T\boldsymbol{W}\boldsymbol{G} - \boldsymbol{\Lambda}\boldsymbol{G}) \\ &= \boldsymbol{0}\end{aligned} \tag{44}$$

Similarly

$$\boldsymbol{E}^T\boldsymbol{W}\boldsymbol{G} - \boldsymbol{G}^T\boldsymbol{\Lambda}\boldsymbol{G} = \boldsymbol{0} \tag{45}$$

Thus,

$$\boldsymbol{Y}^T\boldsymbol{Y} - 2\boldsymbol{G}^T\boldsymbol{\Lambda}\boldsymbol{G} = \boldsymbol{G}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{G} + \boldsymbol{E}^T\boldsymbol{E} \tag{46}$$

or

$$\boldsymbol{E}^T\boldsymbol{E} + \boldsymbol{G}^T\boldsymbol{\Lambda}\boldsymbol{G} = \boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{G}^T\boldsymbol{\Lambda}\boldsymbol{G} - \boldsymbol{G}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{G} \tag{47}$$

7.2 Algorithm-selection procedure

The modified Gram–Schmidt orthogonalisation procedure calculates the $\boldsymbol{A}$ matrix row by row and orthogonalises $\boldsymbol{\Phi}$ as follows: at the lth stage make the columns $\boldsymbol{\phi}_j$, $l+1 \le j \le M$, orthogonal to the lth column and repeat the operation for $1 \le l \le M-1$. Specifically, denoting $\boldsymbol{\phi}_j^{(0)} = \boldsymbol{\phi}_j$, $1 \le j \le M$, then

$$\left.\begin{aligned}\boldsymbol{w}_l &= \boldsymbol{\phi}_l^{(l-1)} \\ a_{l,j} &= \boldsymbol{w}_l^T\boldsymbol{\phi}_j^{(l-1)}/(\boldsymbol{w}_l^T\boldsymbol{w}_l) \qquad l+1 \le j \le M \\ \boldsymbol{\phi}_j^{(l)} &= \boldsymbol{\phi}_j^{(l-1)} - a_{l,j}\boldsymbol{w}_l \qquad l+1 \le j \le M \\ l &= 1, 2, \ldots, M-1\end{aligned}\right\} \tag{48}$$

The last stage of the procedure is simply $\boldsymbol{w}_M = \boldsymbol{\phi}_M^{(M-1)}$. The elements of $\boldsymbol{G}$ are computed by transforming $\boldsymbol{Y}^{(0)} = \boldsymbol{Y}$ in a similar way:

$$\left.\begin{aligned}g_{l,i} &= \boldsymbol{w}_l^T\boldsymbol{y}_i^{(l-1)}/(\boldsymbol{w}_l^T\boldsymbol{w}_l + \lambda_l) \\ \boldsymbol{y}_i^{(l)} &= \boldsymbol{y}_i^{(l-1)} - g_{l,i}\boldsymbol{w}_l\end{aligned}\right\} 1 \le l \le M \quad 1 \le i \le n_o \tag{49}$$

This orthogonalisation scheme can be used to derive a simple and efficient algorithm for selecting subset models in a forward-regression manner. First define

$$\boldsymbol{\Phi}^{(l-1)} = \begin{bmatrix}\boldsymbol{w}_1 & \cdots & \boldsymbol{w}_{l-1} & \boldsymbol{\phi}_l^{(l-1)} & \cdots & \boldsymbol{\phi}_M^{(l-1)}\end{bmatrix} \tag{50}$$

If some of the columns $\boldsymbol{\phi}_l^{(l-1)}, \ldots, \boldsymbol{\phi}_M^{(l-1)}$ in $\boldsymbol{\Phi}^{(l-1)}$ have been interchanged, this will still be referred to as $\boldsymbol{\Phi}^{(l-1)}$ for notational convenience. The lth stage of the selection procedure is given as follows.

Step 1: For $l \le j \le M$ and $1 \le i \le n_o$, compute

$$g_{l,i}^{(j)} = (\boldsymbol{\phi}_j^{(l-1)})^T\boldsymbol{y}_i^{(l-1)}/((\boldsymbol{\phi}_j^{(l-1)})^T\boldsymbol{\phi}_j^{(l-1)} + \lambda_j)$$

$$[rerr]_l^{(j)} = \left(\sum_{i=1}^{n_o}(g_{l,i}^{(j)})^2\right)((\boldsymbol{\phi}_j^{(l-1)})^T\boldsymbol{\phi}_j^{(l-1)} + \lambda_j)/\text{trace}(\boldsymbol{Y}^T\boldsymbol{Y})$$

Step 2: Find

$$[rerr]_l = [rerr]_l^{j_l} = \max\{[rerr]_l^{(j)} \quad l \le j \le M\}$$

Then the j_lth column of $\boldsymbol{\Phi}^{(l-1)}$ is interchanged with the lth column of $\boldsymbol{\Phi}^{(l-1)}$, the first $l-1$ elements of the j_lth column of $\boldsymbol{A}$ are interchanged with those of the lth column of $\boldsymbol{A}$, and the j_lth element of $\boldsymbol{\lambda}$ is interchanged with the lth element of $\boldsymbol{\lambda}$. This effectively selects the j_lth candidate as the lth regressor in the subset model.

Step 3: Perform the orthogonalisation as indicated in (48) to derive the lth row of $\boldsymbol{A}$ and to transform $\boldsymbol{\Phi}^{(l-1)}$ into $\boldsymbol{\Phi}^{(l)}$. Calculate $g_{l,i}$ and update $\boldsymbol{Y}^{(l-1)}$ into $\boldsymbol{Y}^{(l)}$ in the way shown in (49).

The selection is terminated at the M_s stage when the criterion (20) is satisfied and this produces a subset model containing M_s significant regressors. The algorithm described here is in its standard form. A fast implementation can be adopted, as shown in [19] for the single-output case, to reduce complexity.

7.3 Model evidence for $\boldsymbol{h}$ and β

The Bayesian evidence procedure formulated for the single-output case [10] can easily be extended to the current multi-output case. According to MacKay [10] and taking into account that the number of outputs is n_o, the model evidence for $\boldsymbol{h}$ and β can be expressed as

$$P(\boldsymbol{Y}, \boldsymbol{W}|\boldsymbol{h}, \beta) = \frac{Z_{\mathcal{M}}(\boldsymbol{h}, \beta)}{Z_{\boldsymbol{G}}(\boldsymbol{h})Z_{\boldsymbol{Y},\boldsymbol{W}}(\beta)} \tag{51}$$

where

$$Z_{\boldsymbol{G}}(\boldsymbol{h}) = \prod_{j=1}^{M}\left(\frac{\pi}{h_j}\right)^{n_o/2} \tag{52}$$

$$Z_{\boldsymbol{Y},\boldsymbol{W}}(\beta) = \left(\frac{2\pi}{\beta}\right)^{n_oN/2} \tag{53}$$

and

$$Z_{\mathcal{M}}(\boldsymbol{h}, \beta) = e^{-\mathcal{M}_{\text{MAP}}}(2\pi)^{n_oM/2}\det^{-1/2}(\boldsymbol{B}) \tag{54}$$

with $\mathcal{M}_{\text{MAP}}$ being the cost function (21) evaluated at the maximum *a posteriori* probability solution $\boldsymbol{G}$, and $\boldsymbol{B}$ being the $(n_oM) \times (n_oM)$ diagonal matrix defined in (24).

Thus the log evidence can be expressed as

$$\begin{aligned}\log(P(\boldsymbol{Y}, \boldsymbol{W}|\boldsymbol{h}, \beta)) &= -\frac{\beta}{2}\sum_{i=1}^{n_o}\boldsymbol{e}_i^T\boldsymbol{e}_i - \frac{1}{2}\sum_{j=1}^{M}h_j\sum_{i=1}^{n_o}g_{j,i}^2 \\ &\quad - \frac{1}{2}\log(\det(\boldsymbol{B})) + \frac{n_o}{2}\sum_{j=1}^{M}\log(h_j) \\ &\quad + \frac{n_oN}{2}\log(\beta) + c\end{aligned} \tag{55}$$

where

$$c = \frac{n_oM}{2}\log(2\pi) - \frac{n_oM}{2}\log(\pi) - \frac{n_oN}{2}\log(2\pi) \tag{56}$$

Setting

$$\frac{\partial \log(P(\boldsymbol{Y}, \boldsymbol{W}|\boldsymbol{h}, \beta))}{\partial \beta} = 0 \tag{57}$$

yields the re-calculation formula for β

$$\beta\left(\sum_{i=1}^{n_o} \boldsymbol{e}_i^T \boldsymbol{e}_i\right) = n_o N - n_o \sum_{j=1}^{M} \frac{\beta \boldsymbol{w}_j^T \boldsymbol{w}_j}{h_j + \beta \boldsymbol{w}_j^T \boldsymbol{w}_j} \tag{58}$$

Setting

$$\frac{\partial \log(P(\boldsymbol{Y}, \boldsymbol{W}|\boldsymbol{h}, \beta))}{\partial h_j} = 0 \tag{59}$$

provides the re-calculation formulas for h_j, $1 \le j \le M$,

$$h_j\left(\sum_{i=1}^{n_o} g_{j,i}^2\right) = \frac{n_o \beta \boldsymbol{w}_j^T \boldsymbol{w}_j}{h_j + \beta \boldsymbol{w}_j^T \boldsymbol{w}_j} \tag{60}$$

Define

$$\gamma = \sum_{j=1}^{M} \gamma_j \tag{61}$$

with

$$\gamma_j = \frac{\beta \boldsymbol{w}_j^T \boldsymbol{w}_j}{h_j + \beta \boldsymbol{w}_j^T \boldsymbol{w}_j} = \frac{\boldsymbol{w}_j^T \boldsymbol{w}_j}{(h_j/\beta) + \boldsymbol{w}_j^T \boldsymbol{w}_j} \tag{62}$$

Then the re-calculation formulas for β and h_j are, respectively,

$$\beta = \frac{n_o(N - \gamma)}{\sum_{i=1}^{n_o} \boldsymbol{e}_i^T \boldsymbol{e}_i} \tag{63}$$

and

$$h_j = \frac{n_o \gamma_j}{\sum_{i=1}^{n_o} g_{j,i}^2} \tag{64}$$

Noting the relationship $\lambda_j = h_j/\beta$ leads to the re-calculation formulas for λ_j, $1 \le j \le M$,

$$\lambda_j = \frac{\gamma_j}{N - \gamma} \cdot \frac{\sum_{i=1}^{n_o} \boldsymbol{e}_i^T \boldsymbol{e}_i}{\sum_{i=1}^{n_o} g_{j,i}^2} \tag{65}$$