Sparse Multi-Output Radial Basis Function Network Construction Using Combined Locally Regularized Orthogonal Least Square and D-Optimality Experimental Design

S. Chen[†], X. Hong[‡] and C.J. Harris[†]

[†] Department of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, U.K.

[‡] Department of Cybernetics
University of Reading, Reading RG6 6AY, U.K.

Abstract

A new construction algorithm for multi-output radial basis function (RBF) network modelling is introduce by combining a locally regularized orthogonal least squares (LROLS) model selection with a D-optimality experimental design. The proposed algorithm aims to achieve maximized model robustness and sparsity via two effective and complementary approaches. The LROLS method alone is capable of producing a very parsimonious RBF network model with excellent generalization performance. The D-optimality design criterion further enhances the model efficiency and robustness. A further advantage of the combined approach is that the user only needs to specify a weighting for the D-optimality cost in the combined RBF model selecting criterion and the entire model construction procedure becomes automatic. The value of this weighting does not influence the model selection procedure critically and it can be chosen with ease from a wide range of values.

Keywords: Sparse modelling, orthogonal least squares, regularization, Bayesian learning, optimal experimental design, D-optimality, radial basis function network, multi-output regression.

1 Introduction

The radial basis function (RBF) network has widely been studied [1]–[7]. For single-output nonlinear data modelling or regression, the orthogonal least squares (OLS) algorithm [4],[8] provides an effective means to construct parsimonious RBF networks with good generalization performance. The parsimonious principle alone however is not entirely immune to over-fitting. If data are highly noisy, small models constructed may still fit into noise. A useful technique for overcoming over-fitting is regularization [9]–[12]. From the Bayesian learning viewpoint, regularization is equivalent to adopting a hyperparameter approach [13],[14], and a recent work [15],[16] has combined the OLS algorithm with an individually regularized approach to derive an efficient single-output locally regularized OLS (LROLS) algorithm. Optimal experimental designs [17] have been used to construct smooth model response surfaces based on the setting of the experimental variables under well controlled experimental conditions. In optimal design, model adequacy is evaluated by design criteria that are statistical measures of goodness of experimental designs by virtue of design efficiency and experimental effort. For regression models, quantitatively, model adequacy is measured as function of the eigenvalues of the design matrix. The D-optimality design criterion [17] is most effective in optimizing the parameter efficiency and model robustness via the maximization of the determinant of the design matrix. The traditional nonlinear model structure determination based on optimal experimental designs is however inherent inefficient and computationally prohibitive. Recently, effective model construction algorithms has been proposed for single-output nonlinear modelling based on the computationally efficient OLS and LROLS algorithms, respectively, coupled with the D-optimality experimental design [18],[19].

For the construction of multi-output RBF networks, one approach is to fit multiple single-output models as, for example, in the work [20], and an alternative is to construct a single multi-output RBF network model as, for example, in the work [21]. The latter approach has an advantage: a selected RBF term must be significant in explaining all the outputs, and this can result in overall a smaller number of regressors than the former approach to achieve the same modelling accuracy. Recent work [22] has combine the local regularization approach with the multi-output OLS regression. This paper proposes to combine the multi-output LROLS algorithm [22] with the D-optimality experimental design. Computational efficiency of the resulting algorithm is ensured by the orthogonal forward selection procedure. The local regularization enforces model sparsity and avoids over-fitting while the D-optimality design optimizes model efficiency and parameter robustness. The coupling effects of these two approaches in the combined algorithm further enhance each other. The end result is an efficient yet simple algorithm for constructing sparse multi-output RBF models that generalize well, especially under highly noisy learning conditions. Moreover, the model construction process becomes fully automatic, and there is only one user specified quantity which has no critical influence on the model selection procedure.

2 The multi-output radial basis function network

Consider the general discrete-time nonlinear system represented by the nonlinear model [23]:

$$\mathbf{y}(k) = \mathbf{f}(\mathbf{y}(k-1), \cdots, \mathbf{y}(k-n_y), \mathbf{u}(k-1), \cdots, \mathbf{u}(k-n_u)) + \mathbf{e}(k) = \mathbf{f}(\mathbf{x}(k)) + \mathbf{e}(k) \tag{1}$$

where

$$\mathbf{u}(k) = [u_1(k) \cdots u_{n_i}(k)]^T \in \mathcal{R}^{n_i} \tag{2}$$

and

$$\mathbf{y}(k) = [y_1(k) \cdots y_{n_o}(k)]^T \in \mathcal{R}^{n_o} \tag{3}$$

are the system input and output vector variables with dimensions n_i and n_o, respectively, n_u and n_y are positive integers representing the lags in $\mathbf{u}(k)$ and $\mathbf{y}(k)$, respectively;

$$\mathbf{e}(k) = [e_1(k) \cdots e_{n_o}(k)]^T \in \mathcal{R}^{n_o} \tag{4}$$

is the system white noise vector with covariance $\mathrm{Cov}[\mathbf{e}(k)] = \sigma_e^2 \mathbf{I}_{n_o}$ and $\mathbf{I}_{n_o}$ being the $n_o \times n_o$ identity matrix;

$$\mathbf{x}(k) = [\mathbf{y}^T(k-1) \cdots \mathbf{y}^T(k-n_y)\ \mathbf{u}^T(k-1) \cdots \mathbf{u}^T(k-n_u)]^T \tag{5}$$

denotes the system "input" vector; and $\mathbf{f}(\bullet)$ is the unknown n_o-dimensional system mapping.

The system model (1) is to be identified from an N-sample observation data set $\{\mathbf{x}(k), \mathbf{y}(k)\}_{k=1}^N$ using some suitable functional which can approximate $\mathbf{f}(\bullet)$ with arbitrary accuracy. One class of such functionals is the RBF network model of the form:

$$y_i(k) = \hat{y}_i(k) + e_i(k) = \sum_{j=1}^{M} \theta_{j,i} \phi_j(\mathbf{x}(k)) + e_i(k)\,, \ \ 1 \le k \le N\,, \tag{6}$$

for $1 \le i \le n_o$, where $e_i(k)$ is the error between $y_i(k)$ and the i-th model output $\hat{y}_i(k)$, $\theta_{j,i}$ are the RBF weights, the RBF kernels or regressors

$$\phi_j(\mathbf{x}(k)) = \phi(\|\mathbf{x}(k) - \mathbf{c}_j\|; \rho_j), \tag{7}$$

$\mathbf{c}_j$ are the RBF centers and ρ_j the positive width parameters. Typically, each training data $\mathbf{x}(k)$ is considered as a candidate RBF center, and the total number of candidate regressors in this case is $M = N$. Typical choices of nonlinearity $\phi(\bullet)$ are

$$\left\{ \begin{array}{ll} \phi(v) = v^2 \log(v), & \text{thin-plate-spline}, \\ \phi(v;\rho) = \exp\left(-\frac{v^2}{2\rho^2}\right), & \text{Gaussian}, \\ \phi(v;\rho) = (v^2+\rho^2)^{\frac{1}{2}}\,, & \text{multi-quadric}, \\ \phi(v;\rho) = \frac{1}{\sqrt{v^2+\rho^2}}, & \text{inverse multi-quadric}. \end{array} \right. \tag{8}$$

The multi-output RBF network model (6) can be written in a more concise form as

$$\mathbf{y}_i = \mathbf{\Phi} \boldsymbol{\theta}_i + \mathbf{e}_i\,, \ \ 1 \le i \le n_o \tag{9}$$

by defining

$$\mathbf{y}_i = \begin{bmatrix} y_i(1) \\ y_i(2) \\ \vdots \\ y_i(N) \end{bmatrix}, \ \ \mathbf{e}_i = \begin{bmatrix} e_i(1) \\ e_i(2) \\ \vdots \\ e_i(N) \end{bmatrix}, \ \ \boldsymbol{\theta}_i = \begin{bmatrix} \theta_{1,i} \\ \theta_{2,i} \\ \vdots \\ \theta_{M,i} \end{bmatrix}, \tag{10}$$

for $1 \le i \le n_o$, and

$$\mathbf{\Phi} = [\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2 \cdots \boldsymbol{\phi}_M] \tag{11}$$

with

$$\boldsymbol{\phi}_j = [\phi_j(\mathbf{x}(1))\ \phi_j(\mathbf{x}(2)) \cdots \phi_j(\mathbf{x}(N))]^T\ ,\ \ 1 \le j \le M\,. \tag{12}$$

Further define

$$\mathbf{Y} = [\mathbf{y}_1\ \mathbf{y}_2 \cdots \mathbf{y}_{n_o}]\ ,\ \ \boldsymbol{\Theta} = [\boldsymbol{\theta}_1\ \boldsymbol{\theta}_2 \cdots \boldsymbol{\theta}_{n_o}]\ ,\ \ \mathbf{E} = [\mathbf{e}_1\ \mathbf{e}_2 \cdots \mathbf{e}_{n_o}]\,. \tag{13}$$

The RBF network model (6) is given in the matrix form as

$$\mathbf{Y} = \boldsymbol{\Phi}\boldsymbol{\Theta} + \mathbf{E}\,. \tag{14}$$

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

$$\boldsymbol{\Phi} = \mathbf{W}\mathbf{A} \tag{15}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,M} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{M-1,M} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{16}$$

and

$$\mathbf{W} = [\mathbf{w}_1\ \mathbf{w}_2 \cdots \mathbf{w}_M] \tag{17}$$

which satisfies $\mathbf{w}_j^T\mathbf{w}_l = 0$, if $j \neq l$. The RBF model (14) can alternatively be expressed as

$$\mathbf{Y} = \mathbf{W}\mathbf{G} + \mathbf{E} \tag{18}$$

where the orthogonal weight matrix

$$\mathbf{G} = [\mathbf{g}_1\ \mathbf{g}_2 \cdots \mathbf{g}_{n_o}] \tag{19}$$

with

$$\mathbf{g}_i = [g_{1,i}\ g_{2,i} \cdots g_{M,i}]^T\ ,\ \ 1 \le i \le n_o\,, \tag{20}$$

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

$$\mathbf{A}\boldsymbol{\Theta} = \mathbf{G}\,. \tag{21}$$

Knowing $\mathbf{A}$ and $\mathbf{G}$, $\boldsymbol{\Theta}$ can readily be solved from (21).

3 The multi-output LROLS algorithm with D-optimality design

Before discussing this combined multi-output model construction algorithm, its two components, the LROLS algorithm and the D-optimality experimental design, are briefly discussed.

3.1 The LROLS algorithm

The multi-output LROLS algorithm is based on the following regularized error criterion [22]:

$$J_R(\mathbf{G},\boldsymbol{\lambda}) = \text{trace}\left(\mathbf{E}^T\mathbf{E} + \mathbf{G}^T\boldsymbol{\Lambda}\mathbf{G}\right) = \sum_{i=1}^{n_o}\left(\mathbf{e}_i^T\mathbf{e}_i + \mathbf{g}_i^T\boldsymbol{\Lambda}\mathbf{g}_i\right) = \sum_{i=1}^{n_o}\mathbf{e}_i^T\mathbf{e}_i + \sum_{j=1}^{M}\left(\sum_{i=1}^{n_o} g_{j,i}^2\right)\lambda_j \, , \quad (22)$$

where $\boldsymbol{\lambda} = [\lambda_1 \ \lambda_2 \cdots \lambda_M]^T$ is the regularization parameter vector, and the diagonal matrix $\boldsymbol{\Lambda} = \text{diag}\{\lambda_1, \ \lambda_2, \cdots, \lambda_M\}$. The original multi-output OLS algorithm [21] can be viewed as a special case with $\lambda_j = 0$, $\forall j$. After some simplification, the criterion (22) can be expressed as [22]:

$$\text{trace}\left(\mathbf{E}^T\mathbf{E} + \mathbf{G}^T\boldsymbol{\Lambda}\mathbf{G}\right) = \text{trace}\left(\mathbf{Y}^T\mathbf{Y} - \mathbf{G}^T(\mathbf{W}^T\mathbf{W} + \boldsymbol{\Lambda})\mathbf{G}\right) \quad (23)$$

or

$$\text{trace}\left(\mathbf{E}^T\mathbf{E} + \mathbf{G}^T\boldsymbol{\Lambda}\mathbf{G}\right) = \sum_{i=1}^{n_o}\mathbf{y}_i^T\mathbf{y}_i - \sum_{j=1}^{M}\left(\sum_{i=1}^{n_o} g_{j,i}^2\right)(\mathbf{w}_j^T\mathbf{w}_j + \lambda_j)\, . \quad (24)$$

Normalizing (23) by trace$(\mathbf{Y}^T\mathbf{Y})$ yields

$$\frac{\text{trace}\left(\mathbf{E}^T\mathbf{E} + \mathbf{G}^T\boldsymbol{\Lambda}\mathbf{G}\right)}{\text{trace}(\mathbf{Y}^T\mathbf{Y})} = 1 - \sum_{j=1}^{M}\frac{\left(\sum_{i=1}^{n_o} g_{j,i}^2\right)(\mathbf{w}_j^T\mathbf{w}_j + \lambda_j)}{\text{trace}(\mathbf{Y}^T\mathbf{Y})}\, . \quad (25)$$

Define the regularized error reduction ratio due to the regressor $\mathbf{w}_l$ as

$$[\text{rerr}]_l = \frac{\left(\sum_{i=1}^{n_o} g_{l,i}^2\right)(\mathbf{w}_l^T\mathbf{w}_l + \lambda_l)}{\text{trace}(\mathbf{Y}^T\mathbf{Y})}\, . \quad (26)$$

Based on this ratio, significant regressors can be selected in a forward-regression procedure [22]. At the l-th stage, a regressor is chosen as the l-th term of the subset model if it produces the largest $[\text{rerr}]_l$ among the remaining $M - l + 1$ candidates, and the selection is terminated at the M_s-th stage when

$$1 - \sum_{l=1}^{M_s}[\text{rerr}]_l < \xi \quad (27)$$

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 can be found in [22]. Notice that, in the selection procedure, if $\mathbf{w}_l^T\mathbf{w}_l$ 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 [13] can readily be extended to the multi-output case and thus used to "optimize" the regularization parameters. This leads to the updating formulas for the regularization parameters (see [22]):

$$\lambda_j^{\text{new}} = \frac{\gamma_j^{\text{old}}}{N - \gamma^{\text{old}}} \cdot \frac{\sum_{i=1}^{n_o}\mathbf{e}_i^T\mathbf{e}_i}{\sum_{i=1}^{n_o} g_{j,i}^2}\, , \quad 1 \le j \le M\, , \quad (28)$$

where

$$\gamma_j = \frac{\mathbf{w}_j^T\mathbf{w}_j}{\lambda_j + \mathbf{w}_j^T\mathbf{w}_j} \quad (29)$$

and

$$\gamma = \sum_{j=1}^{M} \gamma_j \, . \tag{30}$$

Usually a few iterations (typically 10 to 30) are sufficient to find an optimal $\boldsymbol{\lambda}$.

It is worth emphasizing that, for this multi-output LROLS algorithm, the choice of ξ is less critical than the original OLS algorithm. This is because multiple regularizers enforce sparsity. If, for example, ξ is chosen too small, those unnecessarily selected terms will have a very large λ_l associated with each of them, effectively forcing their weights to zero [15],[16]. Nevertheless, an appropriate value for ξ is desired. Alternatively, the Akaike information criterion (AIC) [24],[25] can be adopted to terminate the subset model selection process. The AIC can be viewed as a model structure regularization by conditioning the model size using a penalty term to penalize large sized models. However, the use of AIC or other information based criteria in forward regression only affects the stopping point of the model selection, but does not penalizes the regressor that may cause poor model performance (e.g. too large variance of parameter estimate or ill-posedness of the regression matrix), if it is selected. Or simply the penalty term in AIC does not determine which regressor should be selected. Optimal experimental design criteria offer better solutions as they are directly linked to model efficiency and parameter robustness.

3.2 The D-optimality experimental design

In experimental design, the data covariance matrix $\mathbf{\Phi}^T\mathbf{\Phi}$ is called the design matrix. The least squares (LS) estimate of $\mathbf{\Theta}$ is given by $\hat{\mathbf{\Theta}} = \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{Y}$. Assume that (14) represents the true data generating process and $\mathbf{\Phi}^T\mathbf{\Phi}$ is nonsingular. Then, the estimate $\hat{\mathbf{\Theta}}$ is unbiased and the covariance matrix of the estimate is determined by the design matrix:

$$\left\{ \begin{array}{l} E\left[\hat{\mathbf{\Theta}}\right] = \mathbf{\Theta}, \\ \text{Cov}\left[\hat{\mathbf{\Theta}}\right] \propto \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1} . \end{array} \right. \tag{31}$$

It is well known that the model based on LS estimate tend to be unsatisfactory for an ill conditioned regression matrix (or design matrix). The condition number of the design matrix is given by

$$C = \frac{\max\{\kappa_i, \, 1 \le i \le M\}}{\min\{\kappa_i, \, 1 \le i \le M\}}, \tag{32}$$

with κ_i, $1 \le i \le M$, being the eigenvalues of $\mathbf{\Phi}^T\mathbf{\Phi}$. Too large a condition number will result in unstable LS parameter estimate while a small condition number improves model robustness. The D-optimality design criterion maximizes the determinant of the design matrix for the constructed model. Specifically, let $\mathbf{\Phi}_{M_s}$ be a column subset of $\mathbf{\Phi}$ representing a constructed M_s-term subset model. According to the D-optimality criterion, the selected subset model is the one that maximizes $\det(\mathbf{\Phi}_{M_s}^T\mathbf{\Phi}_{M_s})$. This helps to prevent the selection of an oversized ill-posed model and the problem of high parameter estimate variances. Thus, the D-optimality design is aimed to optimize model efficiency and parameter robustness.

The optimal experimental designs however do not provide means of parameter estimates and have to rely on the LS or regularized LS methods for model parameter estimate. It is straightforward to verify that maximizing $\det(\mathbf{\Phi}_{M_s}^T \mathbf{\Phi}_{M_s})$ is identical to maximizing $\det(\mathbf{W}_{M_s}^T \mathbf{W}_{M_s})$ or, equivalently, minimizing $-\log\det(\mathbf{W}_{M_s}^T \mathbf{W}_{M_s})$ [18]. Note that

$$\det(\mathbf{\Phi}^T\mathbf{\Phi}) = \prod_{i=1}^{M} \kappa_i = \det(\mathbf{A}^T)\det(\mathbf{W}^T\mathbf{W})\det(\mathbf{A}) = \det(\mathbf{W}^T\mathbf{W}) = \prod_{i=1}^{M} \mathbf{w}_i^T\mathbf{w}_i \tag{33}$$

and

$$-\log\Big(\det(\mathbf{W}^T\mathbf{W})\Big) = \sum_{i=1}^{M} -\log(\mathbf{w}_i^T\mathbf{w}_i). \tag{34}$$

By utilizing the additive property of (34) the D-optimality design criterion can be incorporated naturally and efficiently with the orthogonal forward regression procedure.

3.3 The combined LROLS and D-optimality algorithm

The combined LROLS and D-optimality algorithm can be viewed as based on the combined criterion of

$$J_C(\mathbf{G}, \boldsymbol{\lambda}, \beta) = J_R(\mathbf{G}, \boldsymbol{\lambda}) + \beta \sum_{j=1}^{M} -\log(\mathbf{w}_j^T\mathbf{w}_j), \tag{35}$$

where β is a fixed small positive weighting for the D-optimality cost. In this combined algorithm, the updating of the model weights and regularization parameters is exactly as in the LROLS algorithm, but the selection is according to the combined regularized error reduction ratio defined as

$$[\text{crerr}]_l = \frac{\left(\sum_{i=1}^{n_o} g_{l,i}^2\right)(\mathbf{w}_l^T\mathbf{w}_l + \lambda_l) + \beta\log(\mathbf{w}_l^T\mathbf{w}_l)}{\text{trace}(\mathbf{Y}^T\mathbf{Y})} \tag{36}$$

and the selection is terminated with an M_s-term model when

$$[\text{crerr}]_l \le 0 \ \text{ for } \ M_s + 1 \le l \le M. \tag{37}$$

The iterative RBF model selection procedure can now be summarized:

Initialization. Set λ_j, $1 \le j \le M$, to the same small positive value (e.g. 0.001), and choose a fixed β. Set iteration index $I = 1$.

Step 1. Given the current $\boldsymbol{\lambda}$, select a subset model with M_I terms using the forward regression based on $[\text{crerr}]_l$.

Step 2. Update $\boldsymbol{\lambda}$ using (28)–(30) with $M = M_I$. If $\boldsymbol{\lambda}$ remains sufficiently unchanged in two successive iterations or a pre-set maximum iteration number is reached, stop; otherwise set $I = I + 1$ and go to *Step 1*.

The introduction of the D-optimality cost into the algorithm further enhances the efficiency and robustness of the selected subset model and, as a consequence, the combined algorithm can often produce sparser models with equally good generalization properties, compared with the LROLS algorithm. Note that the model selection procedure is simplified and it is no longer necessary to specify the tolerance ξ, as the algorithm automatically terminates when condition (37) is reached. Unlike the combined OLS and D-optimality algorithm [18], the value of weighting β does not critically influence the performance of this combined LROLS and D-optimality algorithm and β can be chosen with ease from a large range of values. This will be demonstrated in the following modelling examples. It should also be emphasized that the computational complexity of this algorithm is not significantly more than that of the OLS algorithm. This is simply because after the 1st iteration, which has a complexity of the OLS algorithm, the model set contains only M_1 $(\ll M)$ terms, and the complexity of the subsequent iteration decreases dramatically. Typically, after a few iterations, the model set will converge to a constant size of very small M_s. A few more iterations will ensure the convergence of $\boldsymbol{\lambda}$. Thus, this combined LROLS and D-optimality design algorithm offers an efficient procedure to construct sparse multi-output RBF models with excellent generalization performance without the need to apply costly cross-validation.

4 Nonlinear system modelling examples

Three examples were used to illustrate the effectiveness of the multi-output LROLS algorithm with the D-optimality design and to compare it with the combined OLS algorithm and D-optimality design. The RBF network model used in the simulation employed the thin-plate-spline nonlinearity.

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

$$\begin{aligned} y_1(k) &= 0.1\sin(\pi y_2(k-1)) + \left(0.8 - 0.5\exp\left(-y_1^2(k-1)\right)\right) y_1(k-1) \\ &\quad - \left(0.3 + 0.9\exp\left(-y_1^2(k-1)\right)\right) y_1(k-2) + \epsilon_1(k)\,, \\ y_2(k) &= 0.6 y_2(k-1) + 0.2 y_2(k-1) y_2(k-2) + 1.2\tanh(y_1(k-2)) + \epsilon_2(k)\,, \end{aligned} \tag{38}$$

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 $0.04\mathbf{I}_2$. The first 500 data samples were used for training and the other 500 samples for validating the obtained model. The underlying dynamics of the simulated time series was governed by

$$\begin{aligned} y_{d1}(k) &= 0.1\sin(\pi y_{d2}(k-1)) + \left(0.8 - 0.5\exp\left(-y_{d1}^2(k-1)\right)\right) y_{d1}(k-1) \\ &\quad - \left(0.3 + 0.9\exp\left(-y_{d1}^2(k-1)\right)\right) y_{d1}(k-2)\,, \\ y_{d2}(k) &= 0.6 y_{d2}(k-1) + 0.2 y_{d2}(k-1) y_{d2}(k-2) + 1.2\tanh(y_{d1}(k-2))\,. \end{aligned} \tag{39}$$

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. 1. A two-output RBF network was used to model this time series, with the input vector to the RBF network given by

$$\mathbf{x}(k) = [y_1(k-1)\ y_1(k-2)\ y_2(k-1)\ y_2(k-2)]^T. \tag{40}$$

As each training input was used as a candidate RBF center, the number of candidate regressors in the RBF model (6) was $M = 500$.

For the multi-output modelling, the covariance of the modelling error $\mathbf{E}$, $\mathrm{Cov}(\mathbf{E}) = \mathbf{E}^T\mathbf{E}$, is a $n_o \times n_o$ matrix. Typical scalar measures of modelling accuracy include trace$(\mathrm{Cov}(\mathbf{E}))$ and $\det(\mathrm{Cov}(\mathbf{E}))$. Since $\det(\mathrm{Cov}(\mathbf{E}))$ is well-known to be a better measure of modelling accuracy, we will adopt the following scalar measure

$$s_m = \log(\det(\mathrm{Cov}(\mathbf{E}))) \tag{41}$$

in our modelling comparison. Table 1 compares the values of s_m over the training and testing sets for the RBF models constructed by the combined LROLS and D-optimality algorithm with those of the combined OLS and D-optimality algorithm, given a wide range of β values. Note that for this example the true system noise $\boldsymbol{\epsilon}(k)$ had a $s_m = -6.43775$. It can be seen clearly that using the D-optimality alone without regularization the constructed models can still fit into the noise unless the weighting β is set to some appropriate value. Combining regularization with D-optimality design, the results obtained are consistent over a wide range of β values and, effectively, the value of β has no serious influence on the model construction process. The generalization capability of an identified model can best be tested by examining the iterative model output. If the iterative model output can closely realize the behaviour shown in Fig. 1, the identified model truly captures the underlying dynamics of the system and does not simply fits the noise containing in the training data. Given the same initial conditions, the 49-term RBF model identified by the combined LROLS and D-optimality algorithm with $\beta = 1.0$ were used to iteratively generate the network outputs $\hat{y}_{di}(k)$, $i = 1, 2$, with the input

$$\mathbf{x}_d(k) = [\hat{y}_{d1}(k-1)\ \hat{y}_{d1}(k-2)\ \hat{y}_{d2}(k-1)\ \hat{y}_{d2}(k-2)]^T. \tag{42}$$

The iterative model outputs so generated are plotted in Fig. 2. It can be seen that the constructed RBF model appeared to capture the underlying dynamics of the system well.

Example 2. This was a simulated single-input two-output nonlinear system. The data were generated using the model

$$\begin{aligned} y_1(k) &= 0.5y_1(k-1) + u(k-1) + 0.4\tanh(u(k-2)) + \\ & \quad 0.1\sin(\pi y_1(k-2))y_2(k-1) + \epsilon_1(k) \, , \\ y_2(k) &= 0.3y_2(k-1) + 0.1y_2(k-2)y_1(k-1) + \\ & \quad 0.4\exp\left(-u^2(k-1)\right) y_1(k-2) + \epsilon_2(k) \, , \end{aligned} \tag{43}$$

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 $0.04\mathbf{I}_2$. The data set contained 1000 samples, with the first 500 data points used for training and the last 500 data samples for model validation. A two-output RBF network with the input

$$\mathbf{x}(k) = [y_1(k-1)\ y_1(k-2)\ y_2(k-1)\ y_2(k-2)\ u(k-1)\ u(k-2)]^T \tag{44}$$

was employed to fit the noisy training data. The goodness of a fitted model was also evaluated by computing the iterative model outputs with the input

$$\mathbf{x}_d(k) = [\hat{y}_{d1}(k-1)\ \hat{y}_{d1}(k-2)\ \hat{y}_{d2}(k-1)\ \hat{y}_{d2}(k-2)\ u(k-1)\ u(k-2)]^T . \tag{45}$$

For this example, the true system noise again had $s_m = -6.43775$. The modelling accuracies over both the training and testing sets are compared in Table 2 for the two algorithms, the combined LROLS and D-optimality and the combined OLS and D-optimality, with a range of β values. Again it is seen that, for the combined LROLS and D-optimality algorithm, the model construction process is insensitive to the value of β. The modelling accuracies in terms of $\log(\det(\text{Cov}(\mathbf{E}_d)))$ for the two algorithms are compared in Table 3, where $\text{Cov}(\mathbf{E}_d)$ denotes the covariance of the iterative model error. The one-step predictions $\hat{\mathbf{y}}(k)$ of the 35-term RBF model produced by the combined LROLS and D-optimality algorithm with $\beta = 10.0$ are illustrated in Fig. 3, and the iterative model outputs$\hat{\mathbf{y}}_d(k)$ generated by the same RBF model are shown in Fig. 4.

Example 3. This example was a two-input two-output data set collected from a turbo-alternator (Appendix A11.3 in [26]). The data set contained 100 samples. The system inputs were the in-phase current deviation $u_1(k)$ and the out-of-phase current deviation $u_2(k)$, and the system outputs were the voltage deviation $y_1(k)$ and the frequency deviation $y_2(k)$. The two-output RBF network with the input vector

$$\begin{aligned} \mathbf{x}(k) &= [y_1(k-1)\ y_1(k-2)\ y_1(k-3)\ y_2(k-1)\ y_2(k-2)\ y_2(k-3) \\ & \quad u_1(k-1)\ u_1(k-2)\ u_2(k-1)\ u_2(k-2)]^T \end{aligned} \tag{46}$$

was used to fit this data set. As the data set was too short to be divided into a training set and a testing set, the model validation in this case could only be performed by evaluating the iterative model outputs $\hat{y}_{di}(k)$, $i = 1, 2$, with the input

$$\begin{aligned} \mathbf{x}_d(k) \;=\; & [\hat{y}_{d1}(k-1)\; \hat{y}_{d1}(k-2)\; \hat{y}_{d1}(k-3)\; \hat{y}_{d2}(k-1)\; \hat{y}_{d2}(k-2)\; \hat{y}_{d2}(k-3) \\ & u_1(k-1)\; u_1(k-2)\; u_2(k-1)\; u_2(k-2)]^T \end{aligned} \qquad (47)$$

over the training set of 100 samples. Table 4 compares the training accuracies of the two algorithms, the combined LROLS and D-optimality and the combined OLS and D-optimality, given three values of β. Although there were no statistics over a testing data set to confirm the generalization capability of a resulting model, it can be seen from Table 4 that the combined LROLS and D-optimality algorithm performed more consistently with different β values. Note that with $\beta = 0.001$, the two algorithms had similar training accuracies, suggesting that the corresponding models should have similarly good generalization capability. Figs. 5 and 6 depicted the model one-step predictions and the iterative model outputs, respectively, over the training data for the 34-term RBF model constructed by the combined LROLS and D-optimality algorithm with $\beta = 0.001$.

5 Conclusions

A locally regularized OLS algorithm with the D-optimality design has been proposed for constructing sparse multi-output RBF network models. The efficiency of the subset model selection procedure is ensured as usual with the orthogonal forward regression. By combining the two effective and complementary approaches for sparse and robust modelling, namely the local regularization and D-optimality experimental design, the end result is an effective construction algorithm that is capable of producing sparse multi-output RBF network models with excellent generalization performance. It has been shown that the performance of the algorithm is insensitive to the D-optimality cost weighting, and the model construction process is fully automated. The complexity of this combined model construction procedure is only slightly more than that of the efficient OLS algorithm.

References

[1] M.J.D. Powell, “Radial basis functions for multivariable interpretation: a review,” in J.C. Mason and M.G. Cox, Eds., *Algorithms for Approximation*. Oxford: Oxford University Press, 1987, pp.143–167.

[2] D.S. Broomhead and D. Lowe, “Multivariable functional interpolation and adaptive networks,” *Complex Systems*, Vol.2, pp.321–355, 1988.

[3] J.E. Moody and C.J. Darken, “Fast learning in networks of locally tuned processing units,” *Neural Computation*, Vol.1, No.2, pp.281–294, 1989.

[4] 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.

[5] S. Chen, S.A. Billings and P.M. Grant, “Recursive hybrid algorithm for nonlinear systems identification using radial basis function networks,” *Int. J. Control*, Vol.55, No.5, pp.1051–1070, 1992.

[6] M. Brown and C.J. Harris, *Neurofuzzy Adaptive Modeling and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[7] S. Chen, “Nonlinear time series modelling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning,” *Electronics Letters*, Vol.31, No.2, pp.117–118, 1995.

[8] 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.

[9] A.E. Hoerl and R.W. Kennard, “Ridge regression: biased estimation for non-orthogonal problems,” *Technometrics*, Vol.12, pp.55–67, 1970.

[10] C.M. Bishop, “Improving the generalisation properties of radial basis function neural networks,” *Neural Computation*, Vol.3, No.4, pp.579–588, 1991.

[11] S. Chen, E.S. Chng and K. Alkadhimi, “Regularised orthogonal least squares algorithm for constructing radial basis function networks,” *Int. J. Control*, Vol.64, No.5, pp.829–837 1996.

[12] 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.

[13] D.J.C. MacKay, “Bayesian interpolation,” *Neural Computation*, Vol.4, No.3, pp.415–447, 1992.

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

[15] S. Chen, “Kernel-based data modelling using orthogonal least squares selection with local regularisation,” in *Proc. 7th Annual Chinese Automation and Computer Science Conf. in U.K.* (Nottingham, U.K.), Sept.22, 2001, pp.27–30.

[16] S. Chen, “Locally regularised orthogonal least squares algorithm for the construction of sparse kernel regression models,” in *Proc. 6th Int. Conf. Signal Processing* (Beijing, China), Aug.26-30, 2002, Vol.2, pp.1229–1232

[17] A.C. Atkinson and A.N. Donev, *Optimum Experimental Designs*. Oxford: Clarendon Press, 1992.

[18] X. Hong and C.J. Harris, “Nonlinear model structure design and construction using orthogonal least squares and D-optimality design,” *IEEE Trans. Neural Networks*, Vol.13, No.5, pp.1245–1250, 2002.

[19] S. Chen, X. Hong and C.J. Harris, “Sparse kernel regression modelling using combined locally regularized orthogonal least squares and D-optimality experimental design,” submitted to *IEEE Trans. Automatic Control*, 2002.

[20] S.A. Billings, S. Chen and M.J. Korenberg, “Identification of MIMO non-linear systems using a forward-regression orthogonal estimator,” *Int. J. Control*, Vol.49, pp.2157–2189, 1989.

[21] S. Chen, P.M. Grant and C.F.N. Cowan, “Orthogonal least squares algorithm for training multi-output radial basis function networks,” *IEE Proc. Part F*, Vol.139, No.6, pp.378–384, 1992.

[22] S. Chen, “Multi-output regression using a locally regularised orthogonal least square algorithm,” *IEE Proceedings – Vision, Image and Signal Processing*, Vol.149, No.4, pp.185–195, 2002.

[23] 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.

[24] H. Akaike, “A new look at the statistical model identification,” *IEEE Trans. Automatic Control*, Vol.AC-19, pp.716–723, 1974.

[25] 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.

[26] G.M. Jenkins and D.G. Watts, *Spectral Analysis and Its Applications*. San Francisco: Holden-Day, 1986.

Table 1: Comparison of modelling accuracy for the simulated two-output nonlinear time series modelling example. Cov$(\mathbf{E})$: one-step prediction error covariance.

D-optimality	training set $\log(\det(\mathrm{Cov}(\mathbf{E})))$		testing set $\log(\det(\mathrm{Cov}(\mathbf{E})))$		number of terms	
weighting β	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt
0.001	-6.78104	-18.1385	-6.07734	-5.32683	102	470
0.01	-6.68156	-10.1001	-6.08521	-5.39079	62	302
0.1	-6.55440	-6.87149	-6.09854	-5.95289	50	72
1.0	-6.43524	-6.51637	-6.03528	-6.04794	49	49
10.0	-6.38538	-6.43935	-6.12874	-6.10428	44	44

Table 2: Comparison of modelling accuracy for the simulated single-input two-output nonlinear system example. Cov$(\mathbf{E})$: one-step prediction error covariance.

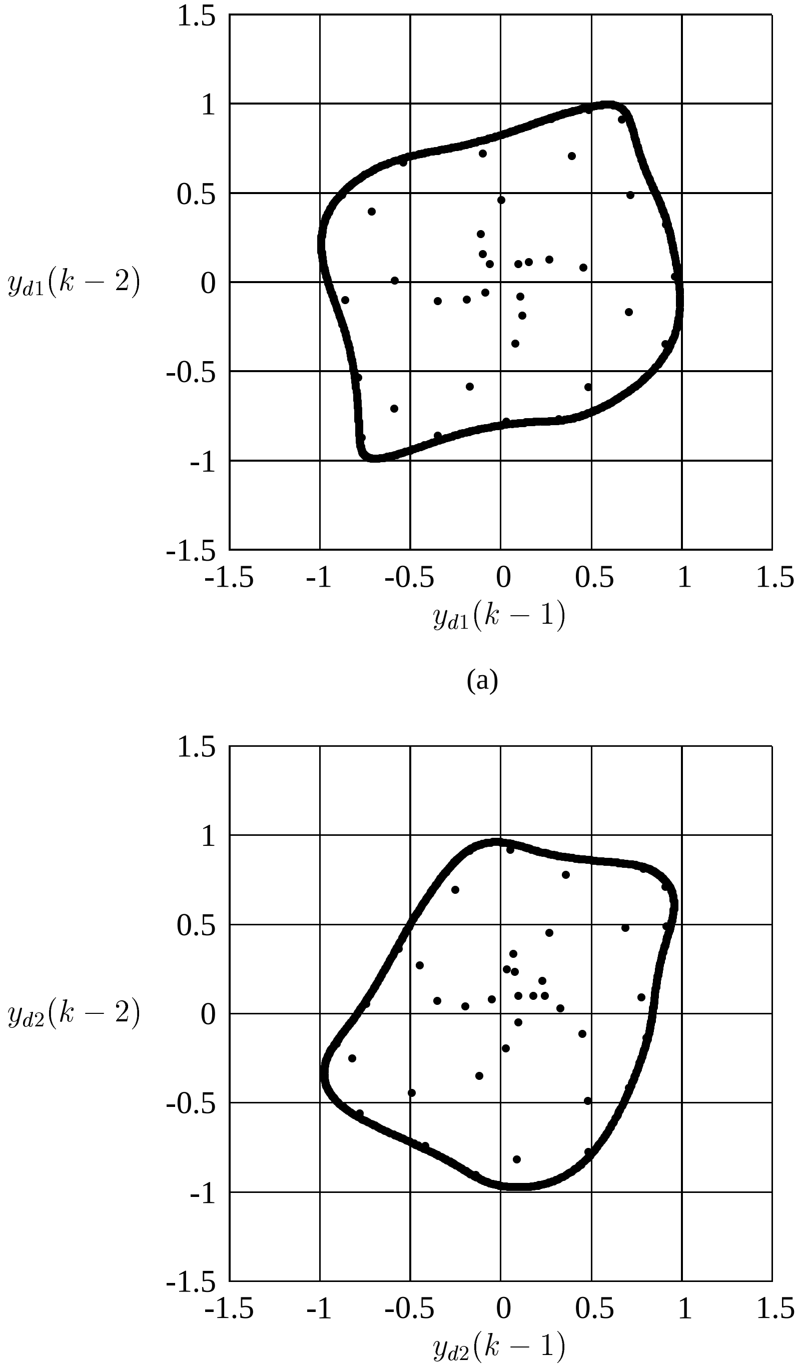
D-optimality	training set $\log(\det(\mathrm{Cov}(\mathbf{E})))$		testing set $\log(\det(\mathrm{Cov}(\mathbf{E})))$		number of terms	
weighting β	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt
0.01	-6.59701	-10.8873	-6.10548	-5.41334	44	320
0.1	-6.56962	-6.84887	-6.07789	-5.95589	38	61
1.0	-6.49324	-6.56252	-6.13198	-6.08903	35	36
10.0	-6.50340	-6.55698	-6.11586	-6.06297	35	35

Table 3: Comparison of modelling accuracy for the simulated single-input two-output nonlinear system example. Cov$(\mathbf{E}_d)$: model iterative error covariance over the entire 1000-sample data set.

D-optimality	$\log(\det(\mathrm{Cov}(\mathbf{E}_d)))$		number of terms	
weighting β	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt
0.01	-5.65089	-5.37460	44	320
0.1	-5.66776	-5.65160	38	61
1.0	-5.65614	-5.71936	35	36
10.0	-5.72100	-5.70334	35	35

Table 4: Comparison of modelling accuracy for the turbo-alternator modelling example. Cov$(\mathbf{E})$: one-step prediction error covariance, and Cov$(\mathbf{E}_d)$: model iterative error covariance

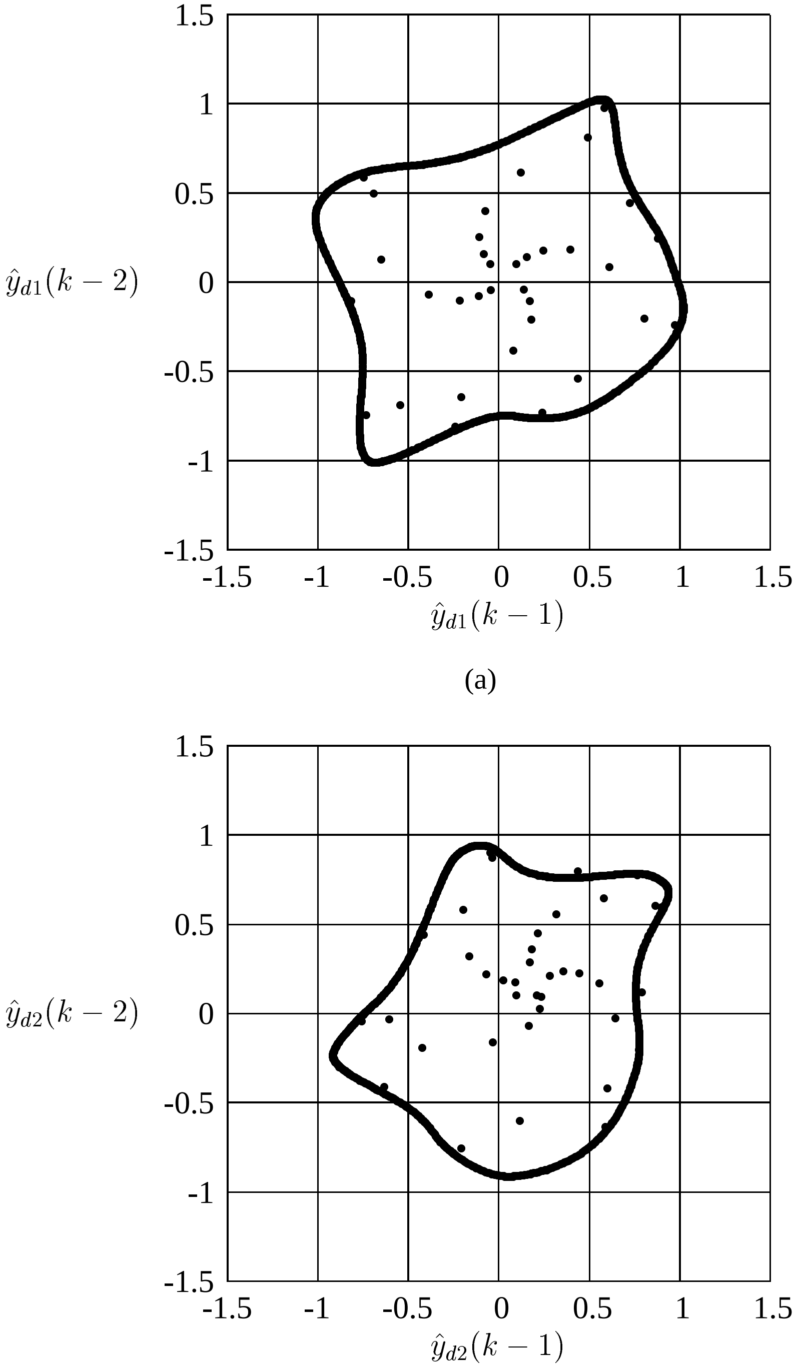
D-optimality	training set $\log(\det(\mathrm{Cov}(\mathbf{E})))$		training set $\log(\det(\mathrm{Cov}(\mathbf{E}_d)))$		number of terms	
weighting β	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt
0.00001	-18.4925	-28.2112	-13.3163	-27.6729	64	96
0.0001	-16.5032	-20.8628	-13.7963	-18.0451	49	78
0.001	-15.2006	-15.7269	-13.4131	-13.4300	34	40



(a)

(b)

Figure 1: Two dimensional representations of the noise-free time series observations: (a) phase plot of noise-free time series $y_{d1}(k)$, and (b) phase plot of noise-free time series $y_{d2}(k)$. Initial conditions were $y_{d1}(0) = y_{d1}(-1) = y_{d2}(0) = y_{d2}(-1) = 0.1$.



(a)

(b)

Figure 2: Two dimensional representations of the iterative model outputs: (a) phase plot of iterative model output $\hat{y}_{d1}(k)$, and (b) phase plot of iterative model output $\hat{y}_{d2}(k)$. Initial conditions were $\hat{y}_{d1}(0) = \hat{y}_{d1}(-1) = \hat{y}_{d2}(0) = \hat{y}_{d2}(-1) = 0.1$. The 49-term RBF model was constructed by the combined LROLS and D-optimality algorithm with $\beta = 1.0$ from very noisy data.

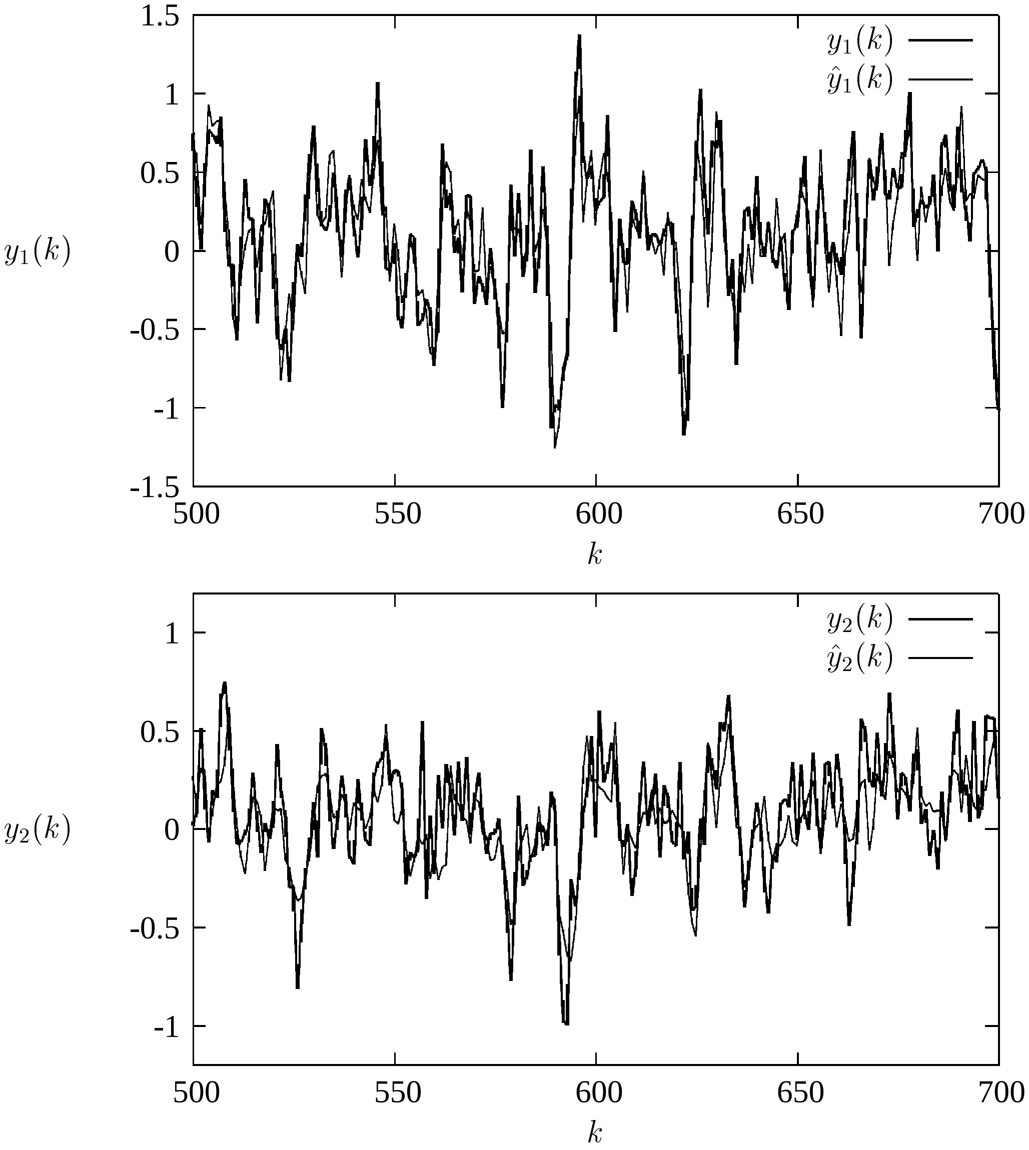


Figure 3: One-step prediction $\hat{\mathbf{y}}(k)$ superimposed on system output $\mathbf{y}(k)$ over the first 200 samples of the test set for the simulated single-input two-output nonlinear system example. The 35-term RBF model was identified by the combined LROLS and D-optimality algorithm with $\beta = 10.0$.

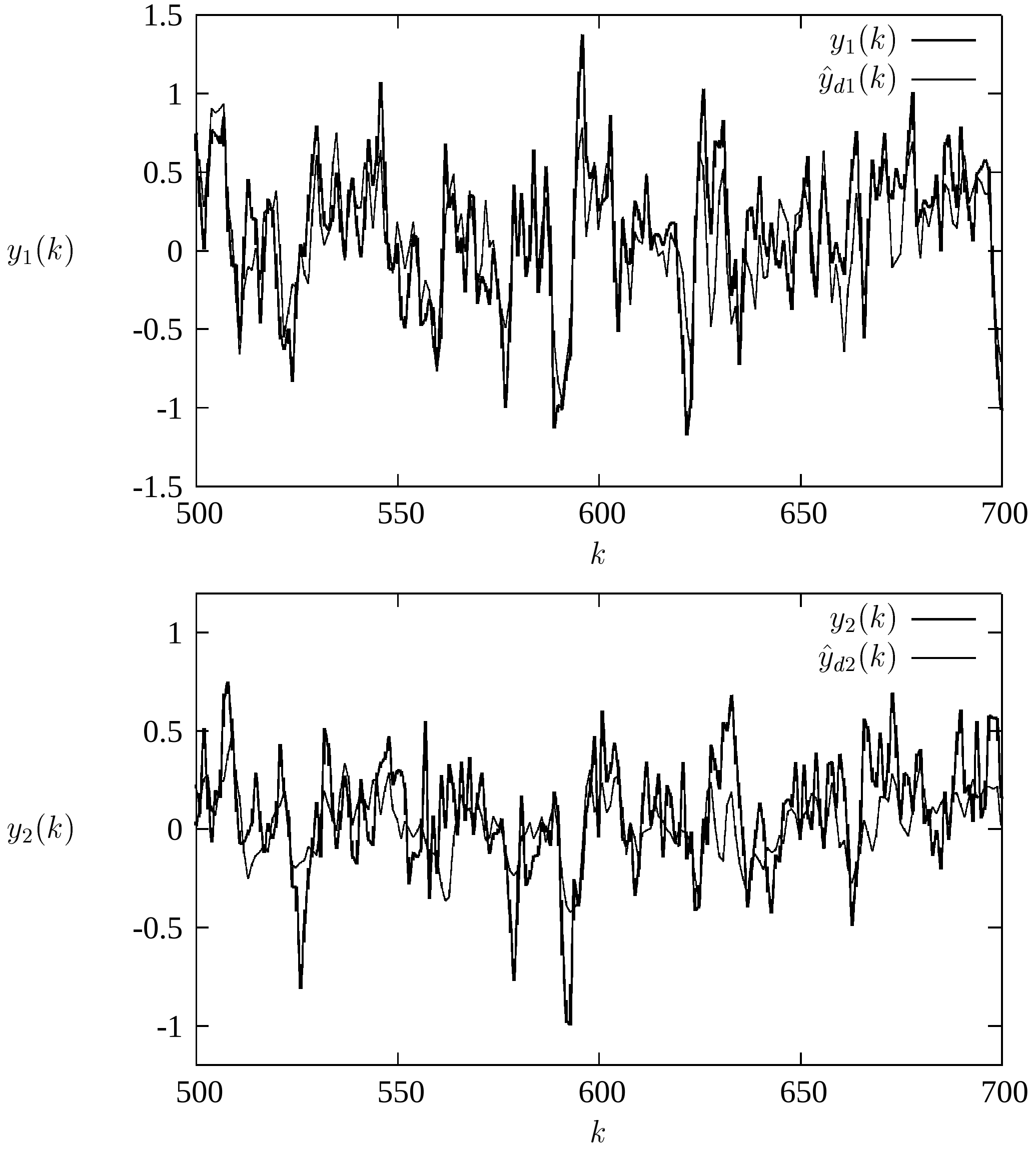


Figure 4: Model iterative output $\hat{\mathbf{y}}_d(k)$ superimposed on system output $\mathbf{y}(k)$ over the first 200 samples of the test set for the simulated single-input two-output nonlinear system example. The 35-term RBF model was identified by the combined LROLS and D-optimality algorithm with $\beta = 10.0$.

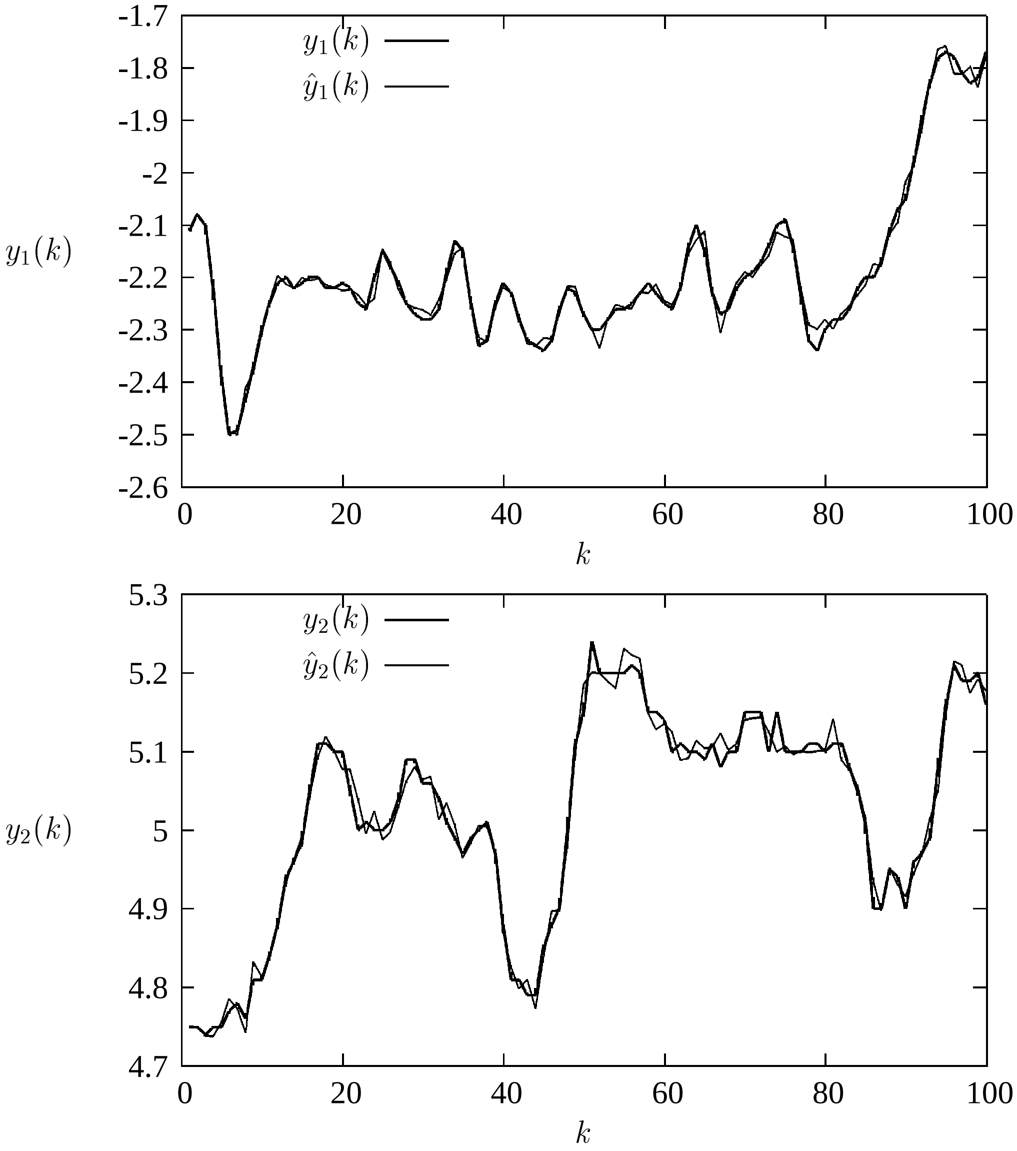


Figure 5: One-step prediction $\hat{\mathbf{y}}(k)$ superimposed on system output $\mathbf{y}(k)$ for the turbo-alternator modelling example. The 34-term RBF model was identified by the combined LROLS and D-optimality algorithm with $\beta = 0.001$.

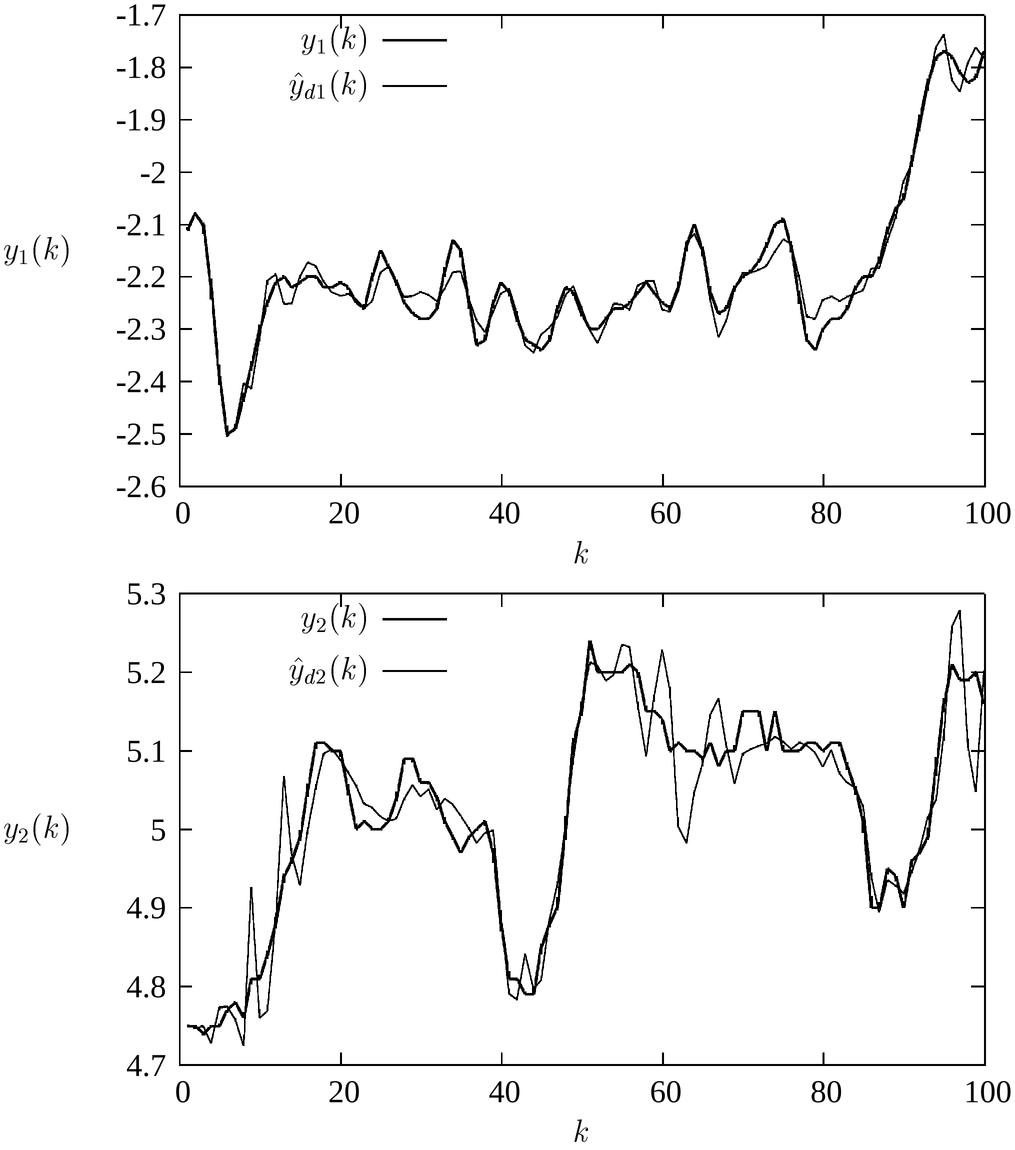


Figure 6: Model iterative output $\hat{\mathbf{y}}_d(k)$ superimposed on system output $\mathbf{y}(k)$ for the turbo-alternator modelling example. The 34-term RBF model was identified by the combined LROLS and D-optimality algorithm with $\beta = 0.001$.