complex models for predicting future information. Being generally unreliable, such prediction models consume time and resources, since a large amount of data must be gathered.

REFERENCES

[1] S. X. Bai, Y. K. Tsai, M. Hafsi, and K. Deng, "Production scheduling in a price competition," *Comput. Math. Applicat.*, vol. 33, no. 5, pp. 5–19, 1997.

[2] V. V. Filippov, "Note on continuity of dependence of solutions of differential inclusion $y' \in F(t, y)$ on the right hand side," *Moscow Univ. Math. Bull.*, vol. 50, no. 3, pp. 13–18, 1995.

[3] R. F. Hartl, S. P. Sethi, and R. G. Vickson, "A survey of the maximum principles for optimal control problems with state constraints," *SIAM Rev.*, vol. 37, no. 2, pp. 181–218, 1995.

[4] 2003A. Herbon, E. Khmelnitsky, and O. Maimon, "Effective information horizon length in measuring offline performance of stochastic dynamic systems," *Eur. J. Oper. Res.*, to be published.

[5] O. Maimon, E. Khmelnitsky, and K. Kogan, *Optimal Flow Control in Manufacturing Systems: Production Planning and Scheduling*. Norwell, MA: Kluwer, 1998.

[6] A. Mehrez, M. S. Hung, and B. H. Ahn, "An industrial ocean-cargo shipping problem," *Dec. Sci.*, vol. 26, no. 3, pp. 395–423, 1995.

[7] R. Neck, "Stochastic control theory and operational research," *Eur. J. Oper. Res.*, vol. 17, pp. 283–301, 1984.

Sparse Kernel Regression Modeling Using Combined Locally Regularized Orthogonal Least Squares and D-Optimality Experimental Design

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

***Abstract*—The note proposes an efficient nonlinear identification algorithm 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 model with excellent generalization performance. The D-optimality design criterion further enhances the model efficiency and robustness. An added advantage is that the user only needs to specify a weighting for the D-optimality cost in the combined 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.**

***Index Terms*—Bayesian learning, D-optimality, optimal experimental design, orthogonal least squares, regularization, sparse modeling.**

I. INTRODUCTION

A basic principle in practical nonlinear data modeling is the parsimonious principle that ensures the smallest possible model that explains the data. A large class of nonlinear models and neural networks can be classified as a kernel regression model [1]–[3]. For this class of nonlinear models, the orthogonal least squares (OLS) algorithm [4], [5] is an efficient learning procedure for constructing sparse regression models. If data are highly noisy, however, the parsimonious principle alone may not be entirely immune to over fitting, and small models constructed may still fit into noise. A useful technique for overcoming over fitting is regularization [6]–[8]. From the powerful Bayesian learning viewpoint, a regularization parameter is equivalent to the ratio of the related hyperparameter to the noise parameter and an effective Bayesian learning method is an evidence procedure which iteratively optimizes model parameters and associated hyperparameters [9]. Adopting this Bayesian learning method to regression models, the locally regularized orthogonal least squares (LROLS) algorithm [10]–[12] has recently been proposed, which introduces an individual regularizer for each weight. This LROLS algorithm provides an efficient procedure for constructing sparse models from noisy data that generalize well.

Optimal experimental designs [13] 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 kernel regression models, quantitatively, model adequacy is measured as function of the eigenvalues of the design matrix, as it is known that the eigenvalues of the design matrix are linked to the covariance matrix of the least squares parameter estimate. There are a variety of optimal design criteria based on different aspects of experimental design [13]. The D-optimality criterion 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, incurring the curse of dimensionality. In [14] and [15], this computational difficulty is overcome and an efficient model construction algorithm has been proposed based on the OLS algorithm coupled with the D-optimality experimental design.

This note shows that further advantages can be gained by combining the LROLS algorithm with the D-optimality experimental design. Computational efficiency of the resulting algorithm as usual is ensured by the orthogonal forward selection procedure. The local regularization enforces model sparsity and avoids over-fitting in model parameters while the D-optimality design also optimizes model efficiency and parameter robustness. The coupling effects of these two approaches in the combined algorithm further enhance each other. 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. Some illustrative examples are included to demonstrate the efficiency of this approach.

II. KERNEL REGRESSION MODEL

Consider the general discrete-time nonlinear system represented by the nonlinear model [1]

$$\begin{aligned} y(k) &= f(y(k-1), \ldots, y(k-n_y), u(k-1), \ldots, \\ &\qquad u(k-n_u)) + e(k) \\ &= f(\mathbf{x}(k)) + e(k) \end{aligned} \tag{1}$$

where $u(k)$ and $y(k)$ are the system input and output variables, respectively, n_u and n_y are positive integers representing the lags in $u(k)$ and $y(k)$, respectively, $e(k)$ is the system white noise, $\mathbf{x}(k) = [y(k-1), \ldots, y(k-n_y)u(k-1), \ldots, u(k-n_u)]^T$ denotes the system "input" vector, and $f(\bullet)$ is the unknown system mapping. The system

Manuscript received June 20, 2002; revised October 28, 2002 and January 22, 2003. Recommended by Associate Editor E. Bai.

S. Chen and C. J. Harris are with the Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.

X. Hong is with the Department of Cybernetics, University of Reading, Reading RG6 6AY, U.K.

Digital Object Identifier 10.1109/TAC.2003.812790

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

$$y(k) = \hat{y}(k) + e(k) = \sum_{i=1}^{n_M} \theta_i \phi_i(\mathbf{x}(k)) + e(k) \tag{2}$$

where $\hat{y}(k)$ denotes the model output, θ_i are the model weights, $\phi_i(\mathbf{x}(k))$ are the regressors, and n_M is the total number of candidate regressors or model terms.

By letting $\boldsymbol{\phi}_i = [\phi_i(\mathbf{x}(1)), \ldots, \phi_i(\mathbf{x}(N))]^T$, for $1 \le i \le n_M$, and defining

$$\mathbf{y} = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} \quad \boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \cdots \boldsymbol{\phi}_{n_M}] \quad \boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_{n_M} \end{bmatrix}$$
$$\mathbf{e} = \begin{bmatrix} e(1) \\ \vdots \\ e(N) \end{bmatrix} \tag{3}$$

the regression model (2) can be written in the matrix form

$$\mathbf{y} = \boldsymbol{\Phi}\boldsymbol{\theta} + \mathbf{e}. \tag{4}$$

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

$$\boldsymbol{\Phi} = \mathbf{W}\mathbf{A} \tag{5}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,n_M} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n_M-1,n_M} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{6}$$

and

$$\mathbf{W} = [\mathbf{w}_1 \cdots \mathbf{w}_{n_M}] \tag{7}$$

with columns satisfying $\mathbf{w}_i^T \mathbf{w}_j = 0$, if $i \ne j$. The regression model (4) can alternatively be expressed as

$$\mathbf{y} = \mathbf{W}\mathbf{g} + \mathbf{e} \tag{8}$$

where the orthogonal weight vector $\mathbf{g} = [g_1, \ldots, g_{n_M}]^T$ satisfy the triangular system $\mathbf{A}\boldsymbol{\theta} = \mathbf{g}$.

III. Locally Regularized OLS Algorithm With D-Optimality Design

Before describing this combined model construction algorithm, we briefly discuss its two components.

A. The Locally Regularized OLS Algorithm

The LROLS algorithm [10]–[12] adopts the following regularized error criterion:

$$J_R(\mathbf{g}, \boldsymbol{\lambda}) = \mathbf{e}^T\mathbf{e} + \sum_{i=1}^{n_M} \lambda_i g_i^2 = \mathbf{e}^T\mathbf{e} + \mathbf{g}^T \boldsymbol{\Lambda} \mathbf{g} \tag{9}$$

where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_{n_M}]^T$ is the regularization parameter vector, and $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \ldots, \lambda_{n_M}\}$. The criterion (9) has its root in the Bayesian learning framework. In fact, according to the Bayesian learning theory (e.g., [9] and [16]), the optimal $\mathbf{g}$ is obtained by maximizing the posterior probability of $\mathbf{g}$, which is given by

$$p(\mathbf{g}|\mathbf{y}, \mathbf{h}, \varepsilon) = \frac{p(\mathbf{y}|\mathbf{g}, \mathbf{h}, \varepsilon) p(\mathbf{g}|\mathbf{h}, \varepsilon)}{p(\mathbf{y}|\mathbf{h}, \varepsilon)} \tag{10}$$

where $p(\mathbf{g}|\mathbf{h}, \varepsilon)$ is the prior with $\mathbf{h} = [h_1, \ldots, h_{n_M}]^T$ denoting the vector of hyperparameters and ε a noise parameter [the inverse of the variance of $e(k)$], $p(\mathbf{y}|\mathbf{g}, \mathbf{h}, \varepsilon)$ is the likelihood, and $p(\mathbf{y}|\mathbf{h}, \varepsilon)$ is the evidence that does not depend on $\mathbf{g}$ explicitly. Under the assumption that $e(k)$ is white and has a Gaussian distribution, the likelihood is expressed as

$$p(\mathbf{y}|\mathbf{g}, \mathbf{h}, \varepsilon) = \left(\frac{\varepsilon}{2\pi}\right)^{N/2} \exp\left(-\frac{\varepsilon}{2} \mathbf{e}^T \mathbf{e}\right). \tag{11}$$

If the Gaussian prior is chosen

$$p(\mathbf{g}|\mathbf{h}, \varepsilon) = \prod_{i=1}^{n_M} \frac{\sqrt{h_i}}{\sqrt{2\pi}} \exp\left(-\frac{h_i g_i^2}{2}\right) \tag{12}$$

maximizing $\log(p(\mathbf{g}|\mathbf{y}, \mathbf{h}, \varepsilon))$ with respect to $\mathbf{g}$ is equivalent to minimizing the following Bayesian cost function:

$$J_B(\mathbf{g}, \mathbf{h}, \varepsilon) = \varepsilon \mathbf{e}^T \mathbf{e} + \mathbf{g}^T \mathbf{H} \mathbf{g} \tag{13}$$

where $\mathbf{H} = \mathrm{diag}\{h_1, \ldots, h_{n_M}\}$. It is easily seen that the criterion (9) is equivalent to the criterion (13) with the relationship

$$\lambda_i = \frac{h_i}{\varepsilon}, \qquad 1 \le i \le n_M. \tag{14}$$

It can readily be shown [12] that with $\mathbf{g}$ set to its optimal values, i.e., $dJ_R/d\mathbf{g} = 0$, the criterion (9) can be expressed as (also see Appendix A)

$$\mathbf{e}^T\mathbf{e} + \mathbf{g}^T \boldsymbol{\Lambda} \mathbf{g} = \mathbf{y}^T \mathbf{y} - \sum_{i=1}^{n_M} \left(\mathbf{w}_i^T \mathbf{w}_i + \lambda_i\right) g_i^2. \tag{15}$$

Normalizing (15) by $\mathbf{y}^T\mathbf{y}$ yields

$$\left(\mathbf{e}^T\mathbf{e} + \mathbf{g}^T \boldsymbol{\Lambda} \mathbf{g}\right) / \mathbf{y}^T \mathbf{y} = 1 - \sum_{i=1}^{n_M} \left(\mathbf{w}_i^T \mathbf{w}_i + \lambda_i\right) g_i^2 / \mathbf{y}^T \mathbf{y}. \tag{16}$$

As in the case of the OLS algorithm [4], the regularized error reduction ratio due to $\mathbf{w}_i$ is defined by

$$[\mathrm{rerr}]_i = \left(\mathbf{w}_i^T \mathbf{w}_i + \lambda_i\right) g_i^2 / \mathbf{y}^T \mathbf{y}. \tag{17}$$

Based on this ratio, significant regressors can be selected in a forward regression procedure, and the selection process is terminated at the n_sth stage when

$$1 - \sum_{l=1}^{n_s} [\mathrm{rerr}]_l < \xi \tag{18}$$

is satisfied, where $0 < \xi < 1$ is a chosen tolerance. This produces a sparse model containing $n_s (\ll n_M)$ significant regressors.

The hyperparameters specify the prior distributions of $\mathbf{g}$. Since initially we do not know the optimal value of $\mathbf{g}$, λ_i should be initialized to the same small value, and this corresponds to choose a same flat distribution for each prior of g_i in (12). The beauty of Bayesian learning is "let data speak"—it learns not only the model parameters $\mathbf{g}$, but also the related hyperparameters $\mathbf{h}$. This can be done for example by iteratively optimizing $\mathbf{g}$ and $\mathbf{h}$ using an evidence procedure [9], [16]. Applying this evidence procedure leads to the updating formulas for the regularization parameters (see Appendix B)

$$\lambda_i^{\text{new}} = \frac{\gamma_i^{\text{old}}}{N - \gamma^{\text{old}}} \frac{\mathbf{e}^T\mathbf{e}}{g_i^2}, \qquad 1 \le i \le n_M \tag{19}$$

where

$$\gamma_i = \frac{\mathbf{w}_i^T\mathbf{w}_i}{\lambda_i + \mathbf{w}_i^T\mathbf{w}_i} \quad \gamma = \sum_{i=1}^{n_M} \gamma_i. \tag{20}$$

Usually, a few iterations (typically 10 to 20) are sufficient to find an optimal $\boldsymbol{\lambda}$.

It is worth emphasizing that, for the 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 added terms will have a very large λ_l associated with each of them, effectively forcing their weights to zero [10]–[12]. Nevertheless, an appropriate value for ξ is desired. Alternatively, the Akaike information criterion (AIC) [17], [18] 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.

B. D-Optimality Experimental Design

In experimental design, the matrix $\mathbf{\Phi}^T\mathbf{\Phi}$ is called the design matrix. The least-square estimate of $\boldsymbol{\theta}$ is given by $\hat{\boldsymbol{\theta}} = \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{y}$. Assume that (4) represents the true data generating process and $\mathbf{\Phi}^T\mathbf{\Phi}$ is nonsingular. Then, the estimate $\hat{\boldsymbol{\theta}}$ is unbiased and the covariance matrix of the estimate is determined by the design matrix

$$\begin{cases} E\left[\hat{\boldsymbol{\theta}}\right] = \boldsymbol{\theta} \\ \text{Cov}\left[\hat{\boldsymbol{\theta}}\right] = \frac{1}{\varepsilon}\left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1} \end{cases}. \tag{21}$$

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

$$C = \frac{\max\{\kappa_i, 1 \le i \le n_M\}}{\min\{\kappa_i, 1 \le i \le n_M\}} \tag{22}$$

with $\kappa_i, 1 \le i \le n_M$, being the eigenvalues of $\mathbf{\Phi}^T\mathbf{\Phi}$. Too large a condition number will result in unstable least square 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 $\mathbf{\Phi}_{n_s}$ be a column subset of $\mathbf{\Phi}$ representing a constructed n_s-term subset model. According to the D-optimality criterion, the selected subset model is the one that maximizes $\det(\mathbf{\Phi}_{n_s}^T\mathbf{\Phi}_{n_s})$. 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.

It is straightforward to verify that maximizing $\det(\mathbf{\Phi}_{n_s}^T\mathbf{\Phi}_{n_s})$ is identical to maximizing $\det(\mathbf{W}_{n_s}^T\mathbf{W}_{n_s})$ or, equivalently, minimizing $-\log\det(\mathbf{W}_{n_s}^T\mathbf{W}_{n_s})$ [14], [15]. Note that

$$\det(\mathbf{\Phi}^T\mathbf{\Phi}) = \prod_{i=1}^{n_M} \kappa_i \tag{23}$$

$$\begin{aligned}\det(\mathbf{\Phi}^T\mathbf{\Phi}) &= \det(\mathbf{A}^T)\det(\mathbf{W}^T\mathbf{W})\det(\mathbf{A}) \\ &= \det(\mathbf{W}^T\mathbf{W}) = \prod_{i=1}^{n_M} \mathbf{w}_i^T\mathbf{w}_i \end{aligned} \tag{24}$$

and

$$-\log\left(\det(\mathbf{W}^T\mathbf{W})\right) = \sum_{i=1}^{n_M} -\log(\mathbf{w}_i^T\mathbf{w}_i). \tag{25}$$

The combined algorithm of OLS and D-optimality design given in [14] and [15] is based on the cost function

$$J_C(\mathbf{g}, \beta) = \mathbf{e}^T\mathbf{e} + \beta\sum_{i=1}^{n_M} -\log(\mathbf{w}_i^T\mathbf{w}_i) \tag{26}$$

where β is a fixed small positive weighting for the D-optimality cost. The model selection is according to the combined error reduction ratio defined as

$$[\text{cerr}]_i = \left(\mathbf{w}_i^T\mathbf{w}_i g_i^2 + \beta\log(\mathbf{w}_i^T\mathbf{w}_i)\right)/\mathbf{y}^T\mathbf{y}. \tag{27}$$

Note that at some stage, say the n_sth stage, the remaining unselected model terms will have $[\text{cerr}]_l \le 0$ for $n_s + 1 \le l \le n_M$, and this terminates the model construction process. Obviously, for this model construction algorithm to produce desired sparse models, the value of β should be set appropriately. It has been suggested in [14] and [15] that an appropriate value for β should be determined using cross validation with two data sets, an estimation set and a validation set. Cross validation in forward model construction is however computationally costly.

C. Combined LROLS and D-Optimality Algorithm

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

$$J_{CR}(\mathbf{g}, \boldsymbol{\lambda}, \beta) = J_R(\mathbf{g}, \boldsymbol{\lambda}) + \beta\sum_{i=1}^{n_M} -\log(\mathbf{w}_i^T\mathbf{w}_i). \tag{28}$$

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

$$[\text{crerr}]_i = \left((\mathbf{w}_i^T\mathbf{w}_i + \lambda_i)g_i^2 + \beta\log(\mathbf{w}_i^T\mathbf{w}_i)\right)/\mathbf{y}^T\mathbf{y} \tag{29}$$

and the selection is terminated with an n_s term model when

$$[\text{crerr}]_l \le 0 \qquad \text{for } n_s + 1 \le l \le n_M. \tag{30}$$

The iterative model selection procedure can now be summarized.

Initialization: Set $\lambda_i, 1 \le i \le n_M$, to the same small positive value (e.g., 0.001), and choose a fixed β. Set iteration $I = 1$.

Step 1) Given the current $\boldsymbol{\lambda}$, use the procedure described in Appendix C to select a subset model with n_I terms.

Step 2) Update $\boldsymbol{\lambda}$ using (19) with $n_M = n_I$. If $\boldsymbol{\lambda}$ remains sufficiently unchanged in two successive iterations or a preset maximum iteration number is reached, stop; otherwise, set $I+ = 1$ and go to Step 1).

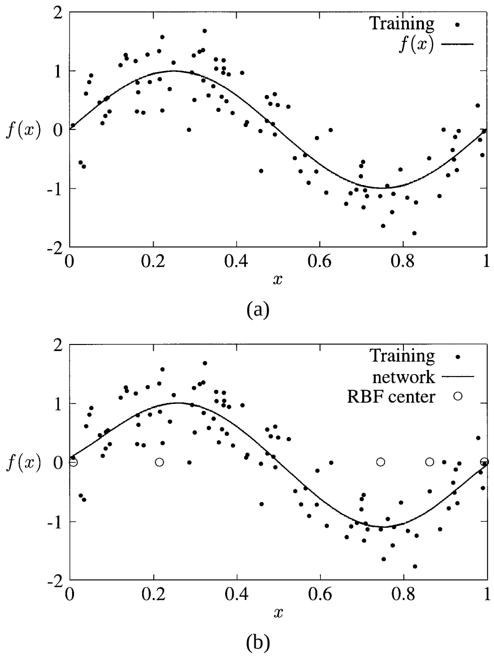
The introduction of the D-optimality cost into the algorithm further enhances the efficiency and robustness of the selected subset model and, as a consequence, the combined algorithm can often produce sparser models with equally good generalization properties, compared with the LROLS algorithm. An additional advantage is that it simplifies the selection procedure. Notice that it is no longer necessary to specify the tolerance ξ and the algorithm automatically terminates when condition (30) is reached. Unlike the combined OLS and D-optimality algorithm [14], [15], the value of weighting β does not critically influence the performance of this combined LROLS and D-optimality algorithm. This is because the LROLS algorithm alone is capable of producing a very sparse model and the selected model terms are most likely to have large values of $\mathbf{w}_i^T\mathbf{w}_i$. Using the OLS algorithm without local regularization, this is not necessarily the case, as model terms with small $\mathbf{w}_i^T\mathbf{w}_i$ can have very large g_i (overfitted) and consequently will be chosen. Note that with regularization, such overfitting will not occur. The D-optimality design also favors the model terms with large $\mathbf{w}_i^T\mathbf{w}_i$ and, therefore, the two component criteria in the combined criterion (28) are not in conflict. Thus, the two methods enhance each other. Consequently, the value of β is less critical in arriving a desired sparse model, compared with the combined OLS and D-optimality algorithm, and the suitable weighting β can be chosen with ease from a large range of values. This will be demonstrated in the following modeling examples. It should also be emphasized that the computational complexity of this algorithm is not significantly more than that of the OLS algorithm or the combined OLS and D-optimality algorithm. This is simply because, after the first iteration, which has a complexity of the OLS algorithm, the model set contains only $n_1(\ll n_M)$ 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 n_s. A few more iterations will ensure the convergence of $\boldsymbol{\lambda}$.

IV. Modeling Examples

Example 1: This example used a radial basis function (RBF) network to model the scalar function

$$f(x) = \sin(2\pi x), \qquad 0 \le x \le 1. \tag{31}$$

For a detailed description of the RBF network see, for example, [5]. The RBF model employed Gaussian kernel function with a variance of 0.04. One hundred training data were generated from $y = f(x) + e$, where the input x was uniformly distributed in $(0, 1)$ and the noise e was Gaussian with zero mean and variance 0.16. The noisy training points y and the underlying function $f(x)$ are plotted in Fig. 1(a). As each training data x was considered as a candidate RBF center, there were $n_M = 100$ regressors in the regression model (2). The training data were extremely noisy. One hundred noise-free data $f(x)$ with equally spaced x were also generated as the testing data set for model validation. In the previous works [10]–[12], it was shown that without regularization the constructed models suffered from a serious over-fitting problem, and the LROLS algorithm was able to overcome this problem and produced a sparse six-term model, with the mean square error (MSE) values over the noisy training set and the noise-free testing set being 0.159 17 and 0.001 81, respectively.



(a)

(b)

Fig. 1. Simple scalar function modeling problem. (a) Noisy training data y (dots) and underlying function $f(x)$ (curve). (b) Model mapping (curve) produced by the LROLS + D-optimality algorithm with $\beta = 10^{-5}$; circles indicate the RBF centers.

Table I compares the MSE values over the training and testing sets for the models constructed by the combined LROLS and D-optimality algorithm with those of the combined OLS and D-optimality algorithm [14], [15], given a wide range of β values. 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. It can also be seen that the combined LROLS and D-optimality algorithm was able to produce a sparser five-term model with equally good generalization properties, compared with the result using the LROLS algorithm alone [10]–[12]. The model map of the five-term model produced by the combined LROLS and D-optimality algorithm with $\beta = 10^{-5}$ is shown in Fig. 1(b).

Example 2: This was a two-dimensional simulated nonlinear time series given by

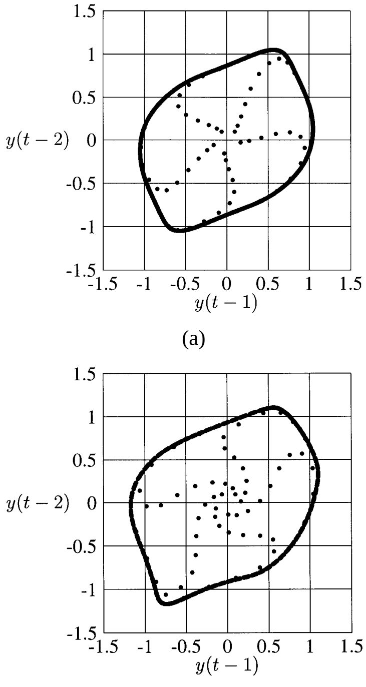
$$\begin{aligned} y(k) = {}& \left(0.8 - 0.5\exp(-y^2(k-1))\right) y(k-1) \\ & - \left(0.3 + 0.9\exp(-y^2(k-1))\right) y(k-2) \\ & + 0.1\sin(\pi y(k-1)) + e(k) \end{aligned} \tag{32}$$

where the noise $e(k)$ was Gaussian with zero mean and variance 0.09. One thousand noisy samples were generated given $y(0) = y(-1) = 0.0$. The first 500 data points were used for training, and the other 500 samples were used for model validation. The underlying noise-free system

$$\begin{aligned} y_d(k) = {}& \left(0.8 - 0.5\exp(-y_d^2(k-1))\right) y_d(k-1) \\ & - \left(0.3 + 0.9\exp(-y_d^2(k-1))\right) y_d(k-2) \\ & + 0.1\sin(\pi y_d(k-1)) \end{aligned} \tag{33}$$

TABLE I
COMPARISON OF MODELING ACCURACY FOR SIMPLE SCALAR FUNCTION MODELING

D-optimality	MSE over noise training data		MSE over noise-free testing data		number of terms	
weighting β	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt
1e-8	0.15766	0.14743	0.00168	0.02138	6	15
1e-7	0.15766	0.14743	0.00168	0.02138	6	15
1e-6	0.15823	0.14743	0.00202	0.02138	6	15
1e-5	0.15705	0.14743	0.00194	0.02138	5	15
1e-4	0.15826	0.14761	0.00246	0.02068	5	15
1e-3	0.15705	0.14933	0.00194	0.01585	5	12
1e-2	0.15705	0.15560	0.00194	0.00423	5	6
1e-1	0.15911	0.15544	0.00223	0.00427	5	6



(a)

(b)

Fig. 2. Two-dimensional time series modeling problem. (a) Phase plot of the noise-free time series ($y_d(0) = y_d(-1) = 0.1$). (b) Phase plot of the iterative RBF model output ($\hat{y}_d(0) = \hat{y}_d(-1) = 0.1$), the model was constructed by the LROLS + D-optimality algorithm with $\beta = 10^{-4}$.

with $y_d(0) = y_d(-1) = 0.1$ was specified by a limit circle, as shown in Fig. 2(a). A Gaussian RBF model of the form

$$\hat{y}(k) = \hat{f}_{\mathrm{RBF}}(\mathbf{x}(k)) \quad \text{with} \quad \mathbf{x}(k) = [y(k-1)y(k-2)]^T \tag{34}$$

was constructed using the noisy training data. The Gaussian kernel function had a variance of 0.81. As each data point $\mathbf{x}(k)$ was considered as a candidate RBF center, there were $n_M = 500$ candidate regressors. The previous study [10]–[12] constructed a sparse 18-term model using the LROLS algorithm, with the MSE values over the training and testing sets being 0.092 64 and 0.096 78, respectively.

The modeling accuracies over both the training and testing sets are compared in Table II 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, when combining with the D-optimality design, the LROLS was able to produce sparser models with equally good generalization performance, compared with the result obtained using the LROLS algorithm alone. It is also clear that for the combined LROLS and D-optimality algorithm the model construction process is insensitive to the value of β. The RBF model produced by the combined LROLS and D-optimality algorithm with $\beta = 10^{-4}$ was used to iteratively generate the time series according to

$$\hat{y}_d(k) = \hat{f}_{\mathrm{RBF}}(\mathbf{x}_d(k)) \quad \text{with} \quad \mathbf{x}_d(k) = [\hat{y}_d(k-1)\hat{y}_d(k-2)]^T \tag{35}$$

given $\mathbf{x}_d(0) = [0.10.1]^T$. The resulting phase plot is shown in Fig. 2(b).

Example 3: This example constructed a model representing the relationship between the fuel rack position (input) and the engine speed (output) for a Leyland TL11 turbocharged, direct injection diesel engine operated at low engine speed. It is known that at low engine speed, the relationship between the input and output is nonlinear [19]. Detailed system description and experimental setup can be found in [19]. The data set contained 410 samples. The first 210 data points were used in modeling and the last 200 points in model validation. An RBF model of the form

$$\hat{y}(k) = \hat{f}_{\mathrm{RBF}}(\mathbf{x}(k))$$

with

$$\mathbf{x}(k) = [y(k-1)u(k-1)u(k-2)]^T \tag{36}$$

was used to model the data. The variance of the RBF kernel function was chosen to be 1.69. As each input vector $\mathbf{x}(k)$ was considered as a candidate RBF center, there were $n_M = 210$ candidate regressors. Previously, a 34-term model was constructed using the LROLS algorithm [10]–[12], and the resulting MSE values over the training and testing sets were 0.000 435 and 0.000 487, respectively.

The MSE values of the models produced by the combined LROLS and D-optimality algorithm and the combined OLS and D-optimality one are compared in Table III, given a range of β values. It can be seen again that the former is insensitive to the weighting value for the D-optimality cost. This real-data identification example really demonstrates the power of combining regularization with the D-optimality design: the ability to produce a much sparser model with similar good generalization performance, compared with relying on regularization alone. The constructed RBF model by the combined LROLS and D-optimality algorithm with $\beta = 10^{-5}$ was used to generate the one-step prediction $\hat{y}(k)$ of the system output according to (36). The iterative model output $\hat{y}_d(k)$ was also produced using

$$\hat{y}_d(k) = \hat{f}_{\mathrm{RBF}}(\mathbf{x}_d(k))$$

with

$$\mathbf{x}_d(k) = [\hat{y}_d(k-1)u(k-1)u(k-2)]^T. \tag{37}$$

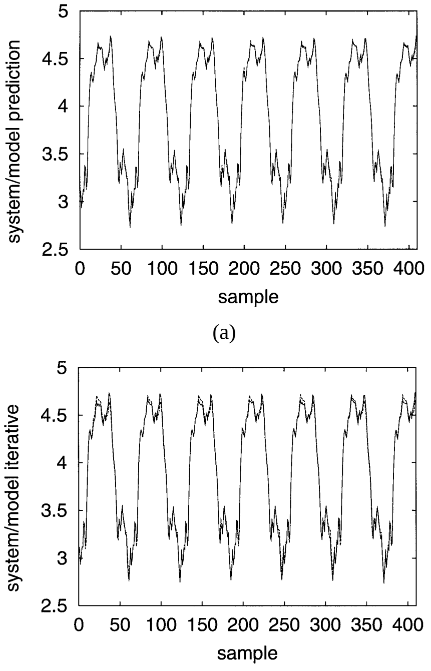
The one-step model prediction and iterative model output for this 22-term model selected by the combined LROLS and D-optimality algorithm are shown in Fig. 3, in comparison with the system output.

TABLE II
COMPARISON OF MODELING ACCURACY FOR TWO-DIMENSIONAL SIMULATED TIME SERIES MODELING

D-optimality	MSE over training data		MSE over testing data		number of terms	
weighting β	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt
1e-6	0.09275	0.07764	0.09635	2.53132	19	94
1e-4	0.09311	0.07762	0.09607	0.41540	13	93
1e-2	0.09338	0.08966	0.09750	0.09379	13	25
1e+0	0.09395	0.09360	0.09667	0.09627	13	14

TABLE III
COMPARISON OF MODELING ACCURACY FOR ENGINE DATA SET

D-optimality	MSE over training data		MSE over testing data		number of terms	
weighting β	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt
1e-8	0.000459	0.000336	0.000488	0.000872	22	60
1e-7	0.000442	0.000345	0.000484	0.000831	27	58
1e-6	0.000441	0.000345	0.000479	0.000838	25	57
1e-5	0.000452	0.000429	0.000499	0.000517	22	24
1e-4	0.000586	0.000445	0.000606	0.000497	20	22
1e-3	0.000478	0.000503	0.000501	0.000536	20	19
1e-2	0.000884	0.000883	0.000982	0.000987	16	16
1e-1	0.004951	0.004951	0.005050	0.005052	12	12



(a)

(b)

Fig. 3. System output $y(k)$ (solid) superimposed on (a) model one-step prediction $\hat{y}(k)$ (dashed) and (b) model iterative output $\hat{y}_d(k)$ (dashed). The model was selected by the LROLS + D-optimality algorithm with $\beta = 10^{-5}$.

V. CONCLUSION

A locally regularized OLS algorithm with the D-optimality design has been proposed for nonlinear system identification using the kernel regression model. It has been demonstrated that combining regularization with the D-optimality experimental design provides a state-of-art procedure for constructing very sparse 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 computational requirements of this iterative model selection procedure are very simple and its implementation straightforward.

APPENDIX A

The regularized least squares solution for $\mathbf{g}$ is obtained by setting $\partial J_R / \partial \mathbf{g} = \mathbf{0}$, that is

$$\mathbf{W}^T \mathbf{y} = \left(\mathbf{W}^T \mathbf{W} + \mathbf{\Lambda}\right) \mathbf{g}. \tag{38}$$

Now

$$\begin{aligned}\mathbf{y}^T \mathbf{y} - 2\mathbf{g}^T \mathbf{\Lambda} \mathbf{g} &= (\mathbf{W}\mathbf{g} + \mathbf{e})^T (\mathbf{W}\mathbf{g} + \mathbf{e}) - 2\mathbf{g}^T \mathbf{\Lambda} \mathbf{g} \\ &= \mathbf{g}^T \mathbf{W}^T \mathbf{W} \mathbf{g} + \mathbf{e}^T \mathbf{e} + \mathbf{g}^T \mathbf{W}^T \mathbf{e} \\ &\quad + \mathbf{e}^T \mathbf{W} \mathbf{g} - 2\mathbf{g}^T \mathbf{\Lambda} \mathbf{g}.\end{aligned} \tag{39}$$

Noting (38)

$$\begin{aligned}\mathbf{g}^T \mathbf{W}^T \mathbf{e} - \mathbf{g}^T \mathbf{\Lambda} \mathbf{g} &= \mathbf{g}^T \mathbf{W}^T (\mathbf{y} - \mathbf{W}\mathbf{g}) - \mathbf{g}^T \mathbf{\Lambda} \mathbf{g} \\ &= \mathbf{g}^T (\mathbf{W}^T \mathbf{y} - \mathbf{W}^T \mathbf{W} \mathbf{g} - \mathbf{\Lambda} \mathbf{g}) = \mathbf{0}.\end{aligned} \tag{40}$$

Similarly, $\mathbf{e}^T \mathbf{W} \mathbf{g} - \mathbf{g}^T \mathbf{\Lambda} \mathbf{g} = \mathbf{0}$. Thus, $\mathbf{y}^T \mathbf{y} - 2\mathbf{g}^T \mathbf{\Lambda} \mathbf{g} = \mathbf{g}^T \mathbf{W}^T \mathbf{W} \mathbf{g} + \mathbf{e}^T \mathbf{e}$, or

$$\mathbf{e}^T \mathbf{e} + \mathbf{g}^T \mathbf{\Lambda} \mathbf{g} = \mathbf{y}^T \mathbf{y} - \mathbf{g}^T \mathbf{W}^T \mathbf{W} \mathbf{g} - \mathbf{g}^T \mathbf{\Lambda} \mathbf{g}. \tag{41}$$

APPENDIX B

Following [9], it can be shown that the log model evidence for $\mathbf{h}$ and ε is approximated as

$$\begin{aligned}\log\left(p(\mathbf{y}|\mathbf{h}, \varepsilon)\right) \approx & \sum_{i=1}^{n_M} \frac{1}{2} \log(h_i) - \frac{n_M}{2} \log(\pi) \\ & - \frac{N}{2} \log(2\pi) + \frac{N}{2} \log(\varepsilon) \\ & - \sum_{i=1}^{n_M} \tfrac{1}{2} h_i g_i^2 - \tfrac{1}{2} \varepsilon \mathbf{e}^T \mathbf{e} - \tfrac{1}{2} \log\left(\det(\mathbf{B})\right) \\ & + \frac{n_M}{2} \log(2\pi)\end{aligned} \tag{42}$$

where $\mathbf{g}$ is set to the maximum *a posterior* probability solution, and the "Hessian" matrix $\mathbf{B}$ is diagonal and is given by

$$\begin{aligned}\mathbf{B} &= \mathbf{H} + \varepsilon \mathbf{W}^T \mathbf{W} \\ &= \text{diag}\left\{h_1 + \varepsilon \mathbf{w}_1^T \mathbf{w}_1, \ldots, h_{n_M} + \varepsilon \mathbf{w}_{n_M}^T \mathbf{w}_{n_M}\right\}.\end{aligned} \tag{43}$$

Setting $\partial \log(p(\mathbf{y}|\mathbf{h}, \varepsilon))/\partial\varepsilon = 0$ yields the recalculation formula for ε

$$\varepsilon \mathbf{e}^T \mathbf{e} = N - \sum_{i=1}^{n_M} \frac{\varepsilon \mathbf{w}_i^T \mathbf{w}_i}{h_i + \varepsilon \mathbf{w}_i^T \mathbf{w}_i}. \tag{44}$$

Setting $\partial \log(p(\mathbf{y}|\mathbf{h}, \varepsilon))/\partial h_i = 0$ yields the recalculation formula for h_i

$$h_i = \frac{\varepsilon \mathbf{w}_i^T \mathbf{w}_i}{g_i^2 (h_i + \varepsilon \mathbf{w}_i^T \mathbf{w}_i)}. \tag{45}$$

Note $\lambda_i = h_i/\varepsilon$ and define

$$\gamma = \sum_{i=1}^{n_M} \gamma_i \quad \text{with } \gamma_i = \frac{\varepsilon \mathbf{w}_i^T \mathbf{w}_i}{h_i + \varepsilon \mathbf{w}_i^T \mathbf{w}_i} = \frac{\mathbf{w}_i^T \mathbf{w}_i}{\lambda_i + \mathbf{w}_i^T \mathbf{w}_i}. \tag{46}$$

Then, the recalculation formula for λ_i is

$$\lambda_i = \frac{\gamma_i}{N - \gamma} \frac{\mathbf{e}^T \mathbf{e}}{g_i^2}, \qquad 1 \le i \le n_M. \tag{47}$$

APPENDIX C

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

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

The last stage of the procedure is simply $\mathbf{w}_{n_M} = \boldsymbol{\phi}_{n_M}^{(n_M - 1)}$. The elements of $\mathbf{g}$ are computed by transforming $\mathbf{y}^{(0)} = \mathbf{y}$ in a similar way

$$\left.\begin{aligned} &g_l = \mathbf{w}_l^T \mathbf{y}^{(l-1)} \Big/ \left(\mathbf{w}_l^T \mathbf{w}_l + \lambda_l\right) \\ &\mathbf{y}^{(l)} = \mathbf{y}^{(l-1)} - g_l \mathbf{w}_l \end{aligned}\right\}, \qquad 1 \le l \le n_M. \tag{49}$$

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

$$\mathbf{\Phi}^{(l-1)} = \left[\mathbf{w}_1 \cdots \mathbf{w}_{l-1} \boldsymbol{\phi}_l^{(l-1)} \cdots \boldsymbol{\phi}_{n_M}^{(l-1)}\right]. \tag{50}$$

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

Step 1) For $l \le j \le n_M$, compute

$$\left.\begin{aligned} g_l^{(j)} &= \left(\boldsymbol{\phi}_j^{(l-1)}\right)^T \mathbf{y}^{(l-1)} \Big/ \left(\left(\boldsymbol{\phi}_j^{(l-1)}\right)^T \boldsymbol{\phi}_j^{(l-1)} + \lambda_j\right), \\ [\text{crerr}]_l^{(j)} &= \left(\left(g_l^{(j)}\right)^2 \left(\left(\boldsymbol{\phi}_j^{(l-1)}\right)^T \boldsymbol{\phi}_j^{(l-1)} + \lambda_j\right)\right. \\ &\quad \left. + \beta \log\left(\left(\boldsymbol{\phi}_j^{(l-1)}\right)^T \boldsymbol{\phi}_j^{(l-1)}\right)\right) \Big/ \left(\mathbf{y}^T \mathbf{y}\right) \end{aligned}\right\}.$$

Step 2) Find

$$[\text{crerr}]_l = [\text{crerr}]_l^{(j_l)} = \max\left\{[\text{crerr}]_l^{(j)}, \; l \le j \le n_M\right\}.$$

Then, the j_lth column of $\mathbf{\Phi}^{(l-1)}$ is interchanged with the lth column of $\mathbf{\Phi}^{(l-1)}$, the j_lth column of $\mathbf{A}$ is interchanged with the lth column of $\mathbf{A}$ up to the $(l-1)$th row, 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 orthogonalization as indicated in (48) to derive the lth row of $\mathbf{A}$ and to transform $\mathbf{\Phi}^{(l-1)}$ into $\mathbf{\Phi}^{(l)}$. Calculate g_l and update $\mathbf{y}^{(l-1)}$ into $\mathbf{y}^{(l)}$ in the way shown in (49).

The selection is terminated at the n_s stage when the criterion (30) is satisfied and this produces a subset model containing n_s significant regressors. The algorithm described here is in its standard form. A fast implementation can be adopted, as shown in [20], to reduce complexity.

REFERENCES

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

[2] S. A. Billings and S. Chen, "Extended model set, global data and threshold model identification of severely nonlinear systems," *Int. J. Control*, vol. 50, no. 5, pp. 1897–1923, 1989.

[3] C. J. Harris, X. Hong, and Q. Gan, *Adaptive Modeling, Estimation and Fusion From Data: A Neurofuzzy Approach*. New York: Springer-Verlag, 2002.

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

[5] 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, pp. 302–309, Mar. 1991.

[6] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, pp. 55–67, 1970.

[7] C. M. Bishop, "Improving the generalization properties of radial basis function neural networks," *Neural Computation*, vol. 3, no. 4, pp. 579–588, 1991.

[8] 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.

[9] D. J. C. MacKay, "Bayesian interpolation," *Neural Comput.*, vol. 4, no. 3, pp. 415–447, 1992.

[10] S. Chen, "Kernel-based data modeling using orthogonal least squares selection with local regularization," in *Proc. 7th Annu.Chinese Automation and Computer Science Conf. U.K.*, Nottingham, U.K., Sept. 22, 2001, pp. 27–30.

[11] ——, "Locally regularised orthogonal least squares algorithm for the construction of sparse kernel regression models," in *Proc. 6th Int. Conf. Signal Processing*, vol. 2, Beijing, China, Aug. 26–30, 2002, pp. 1229–1232.

[12] ——, "Local regularization assisted orthogonal least squares regression," *Int. J. Control*, 2003, submitted for publication.

[13] A. C. Atkinson and A. N. Donev, *Optimum Experimental Designs*. Oxford, U.K.: Clarendon, 1992.

[14] 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, pp. 1245–1250, Sept. 2002.

[15] ——, "Experimental design and model construction algorithms for radial basis function networks," *Int. J. Syst. Sci.*, 2003, submitted for publication.

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

[17] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 716–723, Dec. 1974.

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

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

[20] S. Chen and J. Wigger, "Fast orthogonal least squares algorithm for efficient subset model selection," *IEEE Trans. Signal Processing*, vol. 43, pp. 1713–1715, July 1995.

Stabilization of Nonlinear Systems With Moving Equilibria

A. I. Zecevic and D. D. Siljak

***Abstract*—This note provides a new method for the stabilization of nonlinear systems with parametric uncertainty. Unlike traditional techniques, our approach does not assume that the equilibrium remains fixed for all parameter values. The proposed method combines different optimization techniques to produce a robust control that accounts for uncertain parametric variations, and the corresponding equilibrium shifts. Comparisons with analytical gain scheduling are provided.**

***Index Terms*—Linear matrix inequalities, moving equilibria, nonlinear optimization, parametric stability, robustness.**

I. Introduction

In the analysis of nonlinear dynamic systems, it is common practice to separately treat the existence of equilibria and their stability. The traditional approach has been to compute the equilibrium of interest, and then introduce a change of variables that translates the equilibrium to the origin. This methodology has been widely applied to systems that contain parametric uncertainties, and virtually all control schemes developed along these lines implicitly assume that the equilibrium remains fixed for the entire range of parameter values [1]–[5].

It is important to note, however, that there are many practical applications where the fixed equilibrium assumption is not realistic. In fact, it is often the case that variations in the system parameters result in a moving equilibrium, whose stability properties can vary substantially. In some situations, the equilibrium could even disappear altogether, as in the case of heavily stressed electric power systems [6]–[8]. Much of the recent work involving moving equilibria has focused on analytical gain scheduling [9]–[12]. This approach assumes the existence of an exogenous scheduling variable, whose instantaneous value determines the appropriate control law (which may be nonlinear in general). Analytical gain scheduling will be discussed in some detail in Section IV, where it is compared with the method proposed in this note.

For our purposes, it is suitable to use the concept of parametric stability, which simultaneously captures the *existence* and the *stability* of a moving equilibrium [13]–[17]. This concept has been formulated in [14], where a general nonlinear dynamic system

$$\dot{x} = f(x, p) \tag{1}$$

was considered, with the assumption that a stable equilibrium state $x^e(p^*) \in R^n$ corresponds to the nominal parameter value $p = p^* \in R^l$. System (1) is said to be *parametrically stable* at p^* if there is a neighborhood $\Omega(p^*) \subset R^l$ such that

i) an equilibrium $x^e(p) \in R^n$ exists for any $p \in \Omega(p^*)$;
ii) equilibrium $x^e(p)$ is stable for any $p \in \Omega(p^*)$.

With this definition in mind, the main objective of this note will be to develop a strategy for the parametric stabilization of nonlinear systems. Our approach combines two different optimization techniques to produce a robust control that allows for unpredictable equilibrium shifts due to parametric variations. The resulting controller is linear, and the corresponding gain matrix is obtained using linear matrix inequalities (LMIs) [18]–[23]. The reference input values, on the other hand, are computed by a nonlinear constrained optimization procedure that takes into account the sensitivity of the equilibrium to parameter changes.

The note is organized as follows. In Section II, we provide a brief overview of the control design using linear matrix inequalities, and extend these concepts to systems with parametrically dependent equilibria. Section III is devoted to the problem of selecting an appropriate reference input, and the effects that this selection may have on the size of the stability region in the parameter space. The proposed control strategy is then compared with analytical gain scheduling in Section IV.

II. Parametric Stabilization Using Linear Matrix Inequalities

Let us consider a general nonlinear system described by the differential equations

$$\dot{x} = Ax + h(x) + Bu \tag{2}$$

where $x \in R^n$ is the state of the system, $u \in R^m$ is the input vector, A and B are constant $n \times n$ and $n \times m$ matrices, and h: $R^n \to R^n$ is a piecewise-continuous nonlinear function in x, satisfying $h(0) = 0$. The term $h(x)$ is assumed to be uncertain, but bounded by a quadratic inequality

$$h^T h \leq \alpha^2 x^T H^T H x \tag{3}$$

where $\alpha > 0$ is a scalar parameter and H is a constant matrix. In the following, it will be convenient to rewrite this inequality as:

$$\begin{bmatrix} x \\ h \end{bmatrix}^T \begin{bmatrix} -\alpha^2 H^T H & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ h \end{bmatrix} \leq 0. \tag{4}$$

If we assume a linear feedback control law $u = Kx$, the closed-loop system takes the form

$$\dot{x} = \hat{A}x + h(x) \tag{5}$$

where $\hat{A} = A + BK$. The global asymptotic stability of (5) can then be established using a Lyapunov function

$$V(x) = x^T P x \tag{6}$$

where P is a symmetric positive–definite matrix (denoted $P > 0$). As is well known, a sufficient condition for stability is for the derivative of $V(x)$ to be negative along the solutions of (5). Formally, this condition can be expressed as a pair of inequalities

$$P > 0, \quad \begin{bmatrix} x \\ h \end{bmatrix}^T \begin{bmatrix} \hat{A}^T P + P\hat{A} & P \\ P & 0 \end{bmatrix} \begin{bmatrix} x \\ h \end{bmatrix} < 0. \tag{7}$$

Defining $Y = \tau P^{-1}$ (where τ is a positive scalar), $L = KY$, and $\gamma = 1/\alpha^2$, the control design can now be formulated as an LMI problem in Y, L and γ [22]:

Manuscript received June 26, 2002; revised October 8, 2002 and December 13, 2002. Recommended by Associate Editor K. Gu. This work was supported by the National Science Foundation under Grant ECS-0099469.

The authors are with the Department of Electrical Engineering, Santa Clara University, Santa Clara, CA 95053 USA.

Digital Object Identifier 10.1109/TAC.2003.812791