International Journal of Control
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713393989

Recursive hybrid algorithm for non-linear system identification using radial basis function networks

S. Chen [a]; S. A. Billings [b]; P. M. Grant [a]
[a] Department of Electrical Engineering, University of Edinburgh, Edinburgh, U.K.
[b] Department of Control Engineering, University of Sheffield, Sheffield, SI, U.K.

Online Publication Date: 01 May 1992
To cite this Article: Chen, S., Billings, S. A. and Grant, P. M. (1992) 'Recursive hybrid algorithm for non-linear system identification using radial basis function networks', International Journal of Control, 55:5, 1051 - 1070
To link to this article: DOI: 10.1080/00207179208934272
URL: http://dx.doi.org/10.1080/00207179208934272

INT. J. CONTROL, 1992, VOL. 55, NO. 5, 1051-1070

Recursive hybrid algorithm for non-linear system identification using radial basis function networks

S. CHEN†, S. A. BILLINGS‡ and P. M. GRANT†

Recursive identification of non-linear systems is investigated using radial basis function networks. A novel approach is adopted which employs a hybrid clustering and least squares algorithm. The recursive clustering algorithm adjusts the centres of the radial basis function network while the recursive least squares algorithm estimates the connection weights of the network. Because these two recursive learning rules are both linear, rapid convergence is guaranteed and this hybrid algorithm significantly enhances the real-time or adaptive capability of radial basis function models. The application to simulated real data are included to demonstrate the effectiveness of this hybrid approach.

1. Introduction

Neural networks have been widely used in many areas of signal processing. A particular network structure, namely the multi-layer perceptron, has been employed in non-linear systems identification (Chen *et al.* 1990 a, 1990 d). This application is based on the excellent approximation properties of the multi-layer perceptron (Cybenko 1989, Funahashi 1989). On-line or adaptive identification of multi-layer perceptron models can be achieved using the back-propagation algorithm (Rumelhart *et al.* 1986) or the recursive prediction error algorithm (Chen *et al.* 1990 d). Multi-layer perceptrons, however, are highly non-linear in the parameters and suffer drawbacks of slow convergence and unpredictable solutions during learning.

The radial basis function (RBF) network offers a viable alternative to the two-layer perceptron in signal processing applications. The RBF method has traditionally been used for strict interpolation in multi-dimensional space (Powell 1985, 1987). The original RBF model required that there be as many RBF centres as data points, which is rarely practical because the number of data points is usually very large. Broomhead and Lowe (1988) removed the strict interpolation restriction. Their interpretation of the RBF network allows the use of less RBF centres than data points and, therefore, provides a more suitable basis for the application to signal processing.

Modelling non-linear systems using RBF networks has certain attractive advantages. The general approximation capabilities of the RBF network provides the theoretical foundation of representing complex processes. Furthermore, the response of the RBF network is linear with respect to the connection weights of the network. Provided that the other parameters, the RBF centres, can be chosen appropriately, the linear least squares method can therefore be employed to

Received 23 February 1991. Revised 21 June 1991.

† Department of Electrical Engineering, University of Edinburgh, Mayfield Road, Edinburgh EH9 3JL, U.K.

‡ Department of Control Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, U.K.

estimate these weights. The performance of an RBF network critically depends upon the chosen centres. The RBF centres should suitably sample the network input domain and reflect the data distribution. For off-line system identification, blocks of data are usually available. Broomhead and Lowe (1988) suggested that the centres are randomly chosen from data points. This is clearly an unsatisfactory method for selecting centres. A better approach based on the orthogonal least squares algorithm (Chen *et al.* 1990 b, 1990 c) has been developed to fit RBF models. This algorithm not only provides the least squares estimate for the RBF weights but also selects appropriate centres automatically from the data set. Moreover, the information regarding how many centres are required to fit the data adequately is revealed during the identification procedure and the selection of centres is directly linked to the reduction of the error variance.

For on-line or adaptive applications of RBF models, however, some kind of recursive identification algorithm is required. A simple solution is to fix the RBF centres first and to update only the RBF weights in real-time using the recursive least squares or least mean squares algorithm. This can only work well if variations in the underlying system are small. It is advantageous to update RBF centres and weights simultaneously because this will significantly improve both the modelling capability and the tracking property. Moody and Darken (1989) suggested using an n-means clustering technique to adjust centres in real-time and derived a hybrid clustering and least mean squares algorithm. In the present study we adapt this idea to non-linear system identification using RBF models. In order to improve the convergence properties further, we propose a hybrid clustering and Givens least squares algorithm.

The n-means clustering algorithm partitions the data set into n clusters and obtains n cluster centres by minimizing the total squared error incurred in representing the data set by the n cluster centres (Duda and Hart 1973). In general, only a local minimum of the total squared error is found in this way. This is, however, sufficient for the current application of the RBF network. The value of the total squared error is unimportant in the present application. The aim is to allocate the RBF centres in only those regions where the network input data exists and to reflect data patterns by the positions of the centres. This n-means clustering algorithm can be implemented recursively so that, if the data distribution is changing, the distribution of centres can follow the variations of data patterns. The reason to choose the Givens least squares algorithm (Gentleman 1973) is that it has superior numerical properties over the ordinary least squares algorithm and it is inherently recursive. The version of the Givens method employed in this study does not require the square root computation (Gentleman 1973). A further advantage is that the Givens algorithm can be implemented using systolic arrays (Gentleman and Kung 1981).

The paper is organized as follows. Section 2 introduces a brief summary of RBF networks applied to model single-input single-output non-linear systems. The derivation of the hybrid clustering and Givens least squares algorithm is given in § 3, and application to a simulated time series process, a liquid level system and a heat exchanger are included in § 4. The results obtained using two-layer perceptrons trained by a parallel recursive prediction error algorithm (Chen *et al.* 1990 d) are also given in § 4 as comparison. The extension to multi-input multi-output systems is straightforward and this is discussed in § 5. Finally, some concluding remarks are given in § 6.

2. Modelling non-linear systems using RBF networks

Many single-input single-output non-linear systems can be described in terms of some non-linear functional expansion of lagged inputs and outputs as follows

$$y(t) = f_s(y(t-1), \ldots, y(t-n_y), u(t-1), \ldots, u(t-n_u)) + e(t) \tag{1}$$

where $y(t)$, $u(t)$ and $e(t)$ are the system output, input and white noise respectively; n_y and n_u are the lags of the output and input respectively; and $f_s(\cdot)$ is some non-linear function. Most of the discussion in the current study is based on this system representation.

The assumption for the system representation (1) is that the noise source is white and enters the system additively. In general, however, the noise source may be correlated and can enter the system in a more complicated manner. These possibilities can be accommodated in the following more general system representation

$$y(t) = f_s(y(t-1), \ldots, y(t-n_y), u(t-1), \ldots, u(t-n_u), e(t-1), \ldots, e(t-n_e)) + e(t) \tag{2}$$

where n_e is the lag of the noise, and $e(t)$ is white. The system (2) is known as the NARMAX model (Leontaritis and Billings 1985, Chen and Billings 1989). The identification procedure developed in the present study can be extended to this general system.

The RBF network depicted in Fig. 1 is a two-layer processing structure. The first layer consists of an array of computing units. Each unit contains a parameter vector called a centre, and this calculates the Euclidean distance between the centre and the network input vector. The unit then passes the result through a non-linear function. The second layer is essentially a linear combiner. The overall response of such a network is a mapping f_r: $\boldsymbol{R}^m \rightarrow \boldsymbol{R}$, that is

$$f_r(\boldsymbol{v}) = \sum_{i=1}^{n} \theta_i \phi(\|\boldsymbol{v} - \boldsymbol{c}_i\|) \tag{3}$$

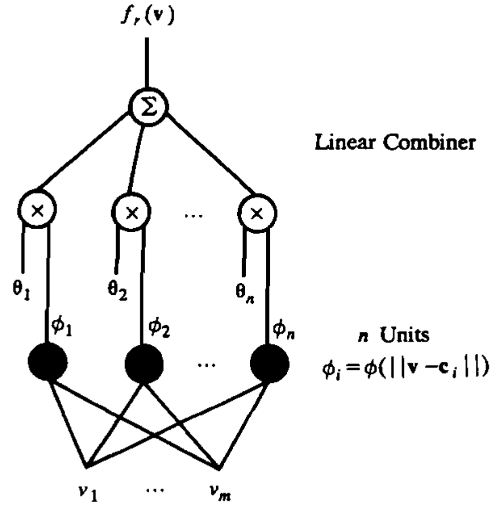


Figure 1. Radial basis function network.

where $\boldsymbol{v} \in \boldsymbol{R}^m$ is the network input vector; $\phi(\cdot)$ is a function from $\boldsymbol{R}^+$ to $\boldsymbol{R}$; $\|\cdot\|$ denotes the euclidean norm; $\boldsymbol{c}_i \in \boldsymbol{R}^m$, $1 \leqslant i \leqslant n$, are the RBF centres; θ_i, $1 \leqslant i \leqslant n$, are the connection weights; and n is the number of computing units in the first layer.

The aim in the present study is to use the RBF network response $f_r(\cdot)$ to capture or to approximate the underlying dynamics $f_s(\cdot)$ in (1). Define $m = n_y + n_u$ and let

$$\boldsymbol{v}(t) = [y(t-1) \dots y(t-n_y) u(t-1) \dots u(t-n_u)]^{\mathrm{T}} \tag{4}$$

The idea then becomes one that uses

$$\hat{y}(t) = f_r(\boldsymbol{v}(t)) \tag{5}$$

as the one-step-ahead predictor for $y(t)$. In the present study, $\phi(\cdot)$ is chosen as the so-called thin-plate-spline function

$$\phi(v) = v^2 \log(v) \tag{6}$$

This choice of $\phi(\cdot)$ provides good modelling capability and is discussed, for example, by Powell (1987). Other choices of $\phi(\cdot)$ can also be employed.

Whether a non-linear model is adequate can be tested using the following two model validity approaches. Define the one-step-ahead prediction error or residual

$$\varepsilon(t) = y(t) - \hat{y}(t) \tag{7}$$

The first model validation method computes the following correlation functions (Billings and Voon 1986).

$$\left.\begin{array}{rl} \Psi_{\varepsilon\varepsilon}(k) & k \neq 0 \\ \Psi_{u\varepsilon}(k) & \text{for all } k \\ \Psi_{\varepsilon(\varepsilon u)}(k) & k \geqslant 0 \\ \Psi_{u^{2'}\varepsilon}(k) & \text{for all } k \\ \Psi_{u^{2'}\varepsilon^2}(k) & \text{for all } k \end{array}\right\} \tag{8}$$

where $\varepsilon u(t) = \varepsilon(t+1)u(t+1)$, $u^{2'}(t) = u^2(t) - \overline{u^2(t)}$ and $\overline{u^2(t)}$ is the time average or mean value of $u^2(t)$. In general, if the correction functions (8) are within the 95% confidence bands, $\pm 1{\cdot}96/N^{1/2}$, the model is regarded as adequate, where N is the number of data samples. The tests (8) were developed based on the facts that $\Psi_{\varepsilon\varepsilon}(k)$ and $\Psi_{u\varepsilon}(k)$ alone are not sufficient to validate non-linear models and may even give misleading information regarding the adequacy of the fitted model. The higher order correlation tests are thus included (Billings and Voon 1986). The alternative approach is the chi-squared statistical test (Bohlin 1978, Leontaritis and Billings 1987). Define an η-dimensional vector valued function

$$\Omega(t) = [\omega(t)\omega(t-1) \dots \omega(t-\eta+1)]^{\mathrm{T}} \tag{9}$$

where $\omega(t)$ is some function of the past inputs, outputs and prediction errors. The chi-squared statistic is defined as

$$\zeta = N\mu^{\mathrm{T}}(\Gamma^{\mathrm{T}}\Gamma)^{-1}\mu \tag{10}$$

with

$$\mu = N^{-1} \sum_{t=1}^{N} \Omega(t)\varepsilon(t)/\sigma_\varepsilon \tag{11}$$

and

$$\Gamma^{\mathrm{T}}\Gamma = N^{-1} \sum_{t=1}^{N} \Omega(t)\Omega^{\mathrm{T}}(t) \tag{12}$$

where σ_ε^2 is the variance of $\varepsilon(t)$. If the values of ζ for several different choices of $\omega(t)$ are within the 95% acceptance region, the model is regarded as adequate. The theoretical justification for this chi-squared test can be found in (Bohlin 1978, Leontaritis and Billings 1987).

3. Hybrid clustering and Givens least squares algorithm

For on-line identification applications using the RBF network, some recursive rules are essential to update the centres and weights. The centres should suitably sample the network input domain and should be able to track the changing patterns of data. Moody and Darken (1989) suggested the n-means clustering procedure as a good updating rule for the RBF centres. The n-means clustering technique is well documented in many pattern classification text books (e.g. Duda and Hart 1973). Because the response of the network is linear with respect to its weights, it is natural to consider the recursive least squares method for adjusting the weights. These observations suggest the recursive identification algorithm for RBF models should have a hybrid structure consisting of

(*a*) recursive n-means clustering sub-algorithm for adjusting the RBF centres;

(*b*) recursive least squares sub-algorithm for updating the RBF weights.

Details of these sub-algorithms are now given.

Given initial centres $c_i(0)$, $1 \leqslant i \leqslant n$, and an initial learning rate for the centres $\alpha_c(0)$, at each sample t the recursive n-means clustering algorithm consists of the following computational steps.

(i) Compute distances and find a minimum distance

$$a_i(t) = \|v(t) - c_i(t-1)\|, \quad 1 \leqslant i \leqslant n$$
$$k = \arg[\min\{a_i(t), 1 \leqslant i \leqslant n\}]$$

(ii) Update centres and re-compute kth distance

$$c_i(t) = c_i(t-1), \quad 1 \leqslant i \leqslant n \text{ and } i \neq k$$
$$c_k(t) = c_k(t-1) + \alpha_c(t)(v(t) - c_k(t-1))$$
$$a_k(t) = \|v(t) - c_k(t)\|$$

The initial centres are often chosen randomly. The learning rate should be $\alpha_c(t) < 1$, and should slowly decrease to zero. In the present appliction $\alpha_c(t)$ is computed according to

$$\alpha_c(t) = \alpha_c(t-1)/(1 + \text{int}[t/n])^{1/2} \tag{13}$$

where int $[\cdot]$ denotes the integer part of the argument. Other computing rules can also be applied to $\alpha_c(t)$.

The convergence properties of the n-means clustering procedure were studied by MacQueen (1967). The n-means clustering is based on a linear learning rule, thus guaranteeing rapid convergence. It is also an unsupervised procedure using only the network input data. No desired response is required and the procedure will not be affected by the learning rule used for the weights. Notice the similarities between the n-means clustering and Kohonen self-organizing algorithm (Kohonen 1987).

The recursive least squares algorithm is based on a recursive solution of the normal equation. Define the hidden layer output vector at t as

$$\Phi(t) = [\phi_1(t) \dots \phi_n(t)]^{\mathrm{T}} = [\phi(a_1(t)) \dots \phi(a_n(t))]^{\mathrm{T}} \tag{14}$$

and the connection weight vector at t as

$$\Theta(t) = [\theta_1(t) \dots \theta_n(t)]^{\mathrm{T}} \tag{15}$$

The weighted normal equation can then be written as

$$(\boldsymbol{X}_t^{\mathrm{T}} \boldsymbol{W}_t \boldsymbol{X}_t)\Theta(t) = \boldsymbol{X}_t^{\mathrm{T}} \boldsymbol{W}_t \boldsymbol{y}_t \tag{16}$$

where

$$\boldsymbol{X}_t = \begin{bmatrix} \Phi^{\mathrm{T}}(1) \\ \vdots \\ \Phi^{\mathrm{T}}(t) \end{bmatrix} \tag{17}$$

$$\boldsymbol{y}_t = [y(1) \dots y(t)]^{\mathrm{T}} \tag{18}$$

and $\boldsymbol{W}_t$ is a $t \times t$ diagonal matrix defined recursively by

$$\boldsymbol{W}_t = \begin{bmatrix} \lambda(t)\boldsymbol{W}_{t-1} & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{W}_1 = 1 \tag{19}$$

$\lambda(t)$ is the usual forgetting factor at t. The recursive least squares algorithm solves (16) to give

$$\Theta(t) = (\boldsymbol{X}_t^{\mathrm{T}} \boldsymbol{W}_t \boldsymbol{X}_t)^{-1} \boldsymbol{X}_t^{\mathrm{T}} \boldsymbol{W}_t \boldsymbol{y}_t \tag{20}$$

It is well known that if the number of parameters n is large, the least squares problem may become ill-conditioned and the use of Givens transformations (Gentleman 1973) to solve the recursive least squares problem has numerical advantages over the algorithms based directly on the normal equation. $\boldsymbol{W}_t^{1/2}\boldsymbol{X}_t$ can be decomposed into

$$\boldsymbol{W}_t^{1/2}\boldsymbol{X}_t = \boldsymbol{Q}(t)\boldsymbol{S}(t) \tag{21}$$

where

$$\boldsymbol{S}(t) = \begin{bmatrix} 1 & s_{12}(t) & s_{13}(t) & \dots & s_{1n}(t) \\ 0 & 1 & s_{23}(t) & \dots & s_{2n}(t) \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & s_{n-1n}(t) \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix} \tag{22}$$

and $\boldsymbol{Q}(t)$ is a $t \times n$ matrix with orthogonal columns that satisfy

$$\boldsymbol{Q}^{\mathrm{T}}(t)\boldsymbol{Q}(t) = \boldsymbol{D}(t) = \mathrm{diag}\,\{d_1(t), \dots, d_n(t)\} \tag{23}$$

$\Theta(t)$ can be obtained by solving the triangular system

$$\boldsymbol{S}(t)\Theta(t) = \boldsymbol{z}(t) \tag{24}$$

where $z(t)$ is an n-dimensional vector given by

$$z(t) = \boldsymbol{D}^{-1}(t)\boldsymbol{Q}^{\mathrm{T}}(t)\boldsymbol{W}_t^{1/2}\boldsymbol{y}_t \tag{25}$$

Givens least squares algorithm can be employed to derive (24) and thus to solve for $\Theta(t)$. The algorithm is initialized by setting

$$\left.\begin{aligned} z(0) &= 0 \\ \sigma_\varepsilon(0) &= 0 \\ \boldsymbol{S}(0) &= \boldsymbol{I} \\ \boldsymbol{D}(0) &= \boldsymbol{I}/\rho \end{aligned}\right\} \tag{26}$$

where $\boldsymbol{I}$ is the $n \times n$ identity matrix and ρ is a large positive scalar. The forgetting factor $\lambda(t)$ is usually computed according to the rule (Ljung and Söderström 1983)

$$\lambda(t) = \lambda_0 \lambda(t-1) + 1 - \lambda_0. \tag{27}$$

λ_0 and $\lambda(0)$ are chosen to be less than but close to 1. At each sample t, the computational procedure is as follows.

(*a*) Perform Givens transformations

$$\begin{bmatrix} \boldsymbol{D}^{1/2}(t-1) & [\boldsymbol{S}(t-1) \quad z(t-1)] \\ & [0 \ldots 0 \quad \sigma_\varepsilon(t-1)] \\ \delta^{1/2} & [\Phi^{\mathrm{T}}(t) \quad y(t)] \end{bmatrix} \rightarrow \begin{bmatrix} \boldsymbol{D}^{1/2}(t) & [\boldsymbol{S}(t) \quad z(t)] \\ & [0 \ldots 0 \quad \sigma_\varepsilon(t)] \\ & [0 \ldots 0 \quad 0] \end{bmatrix} \tag{28}$$

where δ is initialized to $1/\lambda(t)$.

(*b*) Solve the triangular system (24) for $\Theta(t)$.

Explicit formulations for the Givens transformations will now be given. First let $l = 1 + n$ and introduce and $l \times l$ diagonal matrix $\tilde{\boldsymbol{D}}(t)$ as

$$\tilde{\boldsymbol{D}}(t) = \begin{bmatrix} \boldsymbol{D}(t) & 0 \\ 0 & \sigma_\varepsilon^2(t) \end{bmatrix} = \operatorname{diag}\{\tilde{d}_1(t), \ldots, \tilde{d}_l(t)\} \tag{29}$$

Next define an $l \times l$ upper triangular matrix $\tilde{\boldsymbol{S}}(t)$ as

$$\tilde{\boldsymbol{S}}(t) = \begin{bmatrix} \boldsymbol{S}(t) & z(t) \\ 0 \ldots 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \tilde{s}_{12}(t) & \tilde{s}_{13}(t) & \ldots & \tilde{s}_{1l}(t) \\ 0 & 1 & \tilde{s}_{23}(t) & \ldots & \tilde{s}_{2l}(t) \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & \tilde{s}_{l-1l}(t) \\ 0 & \ldots & 0 & 0 & 1 \end{bmatrix} \tag{30}$$

and denote

$$\delta^{1/2}[\Phi^{\mathrm{T}}(t)\ y(t)] = (\delta^{(0)})^{1/2}[x_1^{(0)}(t) \ldots x_l^{(0)}(t)] \tag{31}$$

The Givens transformations (28) can then be written more concisely as

$$\begin{bmatrix} \tilde{\boldsymbol{D}}^{1/2}(t-1) & [\tilde{\boldsymbol{S}}(t-1)] \\ (\delta^{(0)})^{1/2} & [x_1^{(0)}(t) \ldots x_l^{(0)}(t)] \end{bmatrix} \rightarrow \begin{bmatrix} \tilde{\boldsymbol{D}}^{1/2}(t) & [\tilde{\boldsymbol{S}}(t)] \\ & [0 \ldots 0] \end{bmatrix} \tag{32}$$

Assume that after $i - 1$ Givens transformations have been performed,

$$(\delta^{(0)})^{1/2}(x_1^{(0)}(t), \ldots, x_l^{(0)}(t)) \tag{33}$$

is transferred to

$$0, \ldots, 0, (\delta^{(i-1)})^{1/2} x_i^{(i-1)}(t), \ldots, (\delta^{(i-1)})^{1/2} x_l^{(i-1)}(t) \tag{34}$$

Then the ith Givens transformation transfers

$$\left.\begin{array}{llll} 0, \ldots, 0, & \tilde{d}_i^{1/2}(t-1), & \tilde{d}_i^{1/2}(t-1)\tilde{s}_{ii+1}(t-1), \ldots, & \tilde{d}_i^{1/2}(t-1)\tilde{s}_{il}(t-1) \\ 0, \ldots, 0, & (\delta^{(i-1)})^{1/2}x_i^{(i-1)}(t), & (\delta^{(i-1)})^{1/2}x_{i+1}^{(i-1)}(t), \quad \ldots, & (\delta^{(i-1)})^{1/2}x_l^{(i-1)}(t) \end{array}\right\} \quad (35)$$

into

$$\left.\begin{array}{llll} 0, \ldots, 0, & \tilde{d}_i^{1/2}(t), & \tilde{d}_i^{1/2}(t)\tilde{s}_{ii+1}(t), \ldots, & \tilde{d}_i^{1/2}(t)\tilde{s}_{il}(t) \\ 0, \ldots, 0, & 0, & (\delta^{(i)})^{1/2}x_{i+1}^{(i)}(t), \ldots, & (\delta^{(i)})^{1/2}x_l^{(i)}(t) \end{array}\right\} \quad (36)$$

where

$$\begin{aligned} \tilde{d}_i(t) &= \tilde{d}_i(t-1) + \delta^{(i-1)}(x_i^{(i-1)}(t))^2 \\ c &= \tilde{d}_i(t-1)/\tilde{d}_i(t) \\ b &= \delta^{(i-1)}x_i^{(i-1)}(t)/\tilde{d}_i(t) \\ \delta^{(i)} &= c\,\delta^{(i-1)} \end{aligned} \quad (37)$$

and

$$\left.\begin{aligned} x_k^{(i)}(t) &= x_k^{(i-1)}(t) - x_i^{(i-1)}(t)\tilde{s}_{ik}(t-1) \\ \tilde{s}_{ik}(t) &= c\tilde{s}_{ik}(t-1) + bx_k^{(i-1)}(t) \end{aligned}\right\} k = i+1, \ldots, l \quad (38)$$

For time-varying or non-stationary systems, in order to provide a continual tracking capability, the time-decreasing learning rate $\alpha_c(t)$ in (13) can be replaced by a constant learning rate α_c and a constant forgetting factor $0 < \lambda < 1$ can be employed instead of $\lambda(t)$ given in (27). For certain applications, it is vital to reduce computational load as much as possible, and the least squares sub-algorithm within the hybrid structure may be replaced by the least mean squares sub-algorithm at the cost of convergence speed.

4. Application examples

The hybrid clustering and Givens least squares algorithm derived in the previous section was used to identify three systems.

Example 1

This is a simulated time-series process. 1500 samples of data were generated by

$$\begin{aligned} y(t) &= (0{\cdot}8 - 0{\cdot}5\exp(-y^2(t-1)))y(t-1) \\ &\quad -(0{\cdot}3 + 0{\cdot}9\exp(-y^2(t-1)))y(t-2) + 0{\cdot}1\sin(3{\cdot}1415926y(t-1)) + e(t) \end{aligned}$$

where the noise $e(t)$ was a gaussian white sequence with mean zero and variance 0·04. The structure of the RBF model was defined by $m = n_y = 2$ and $n = 30$. The parameters in the hybrid algorithm were chosen to be

$$\rho = 1000, \quad \lambda_0 = 0{\cdot}99, \quad \lambda(0) = 0{\cdot}95 \quad \text{and} \quad \alpha_c(0) = 0{\cdot}9.$$

Initial centres were randomly selected from the region $[-2, 2] \times [-2, 2]$.

The evolution of the mean square error (variance of the residuals) obtained using the hybrid algorithm is plotted in Fig. 2. During the recursive identification procedure, the mean square error was reduced from an initial 12 dB to the noise floor, approaching -14 dB. The distribution of the observations and the final RBF centres are depicted in Fig. 3. Several chi-squared tests for the final RBF network

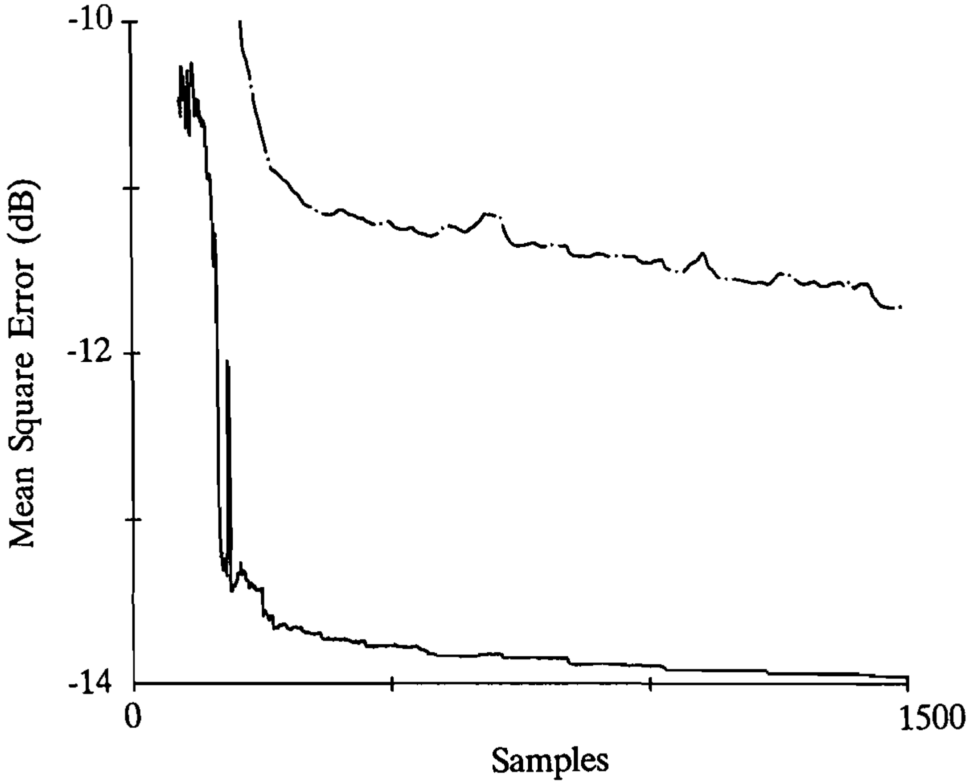


Figure 2. Evolution of mean square error (Example 1). Solid: RBF network, dashed: multi-layer perceptron.

were calculated and they were all within the 95% confidence band. Two typical chi-squared tests and the autocorrelations of $\varepsilon(t)$ are shown in Figs 4 and 5 respectively. The model validity tests confirm that this RBF network is an adequate model for the time series.

It can easily be verified that without the noise $e(t)$ this simulated system generates a stable limit cycle as illustrated in Fig. 6. The identified RBF network was used to produce iteratively the network output

$$\hat{y}_d(t) = f_r(v_d(t))$$

where $v_d(t) = [\hat{y}_d(t-1)\hat{y}_d(t-2)]^T$. The iterative network outputs produce a similar limit cycle as can be seen from Fig. 7. These two limit cycles have approximately a period of 5, in the sense that every five samples approximately complete a circle (2π phase angle) in the state space. However, the amplitudes of response appear to change randomly. One hundred samples of the autonomous system outputs and the iterative network outputs are shown in Fig. 8. Even though the RBF network was identified using the noisy system observations, the iterative network outputs closely matches the response of the autonomous system. This demonstrates that the identified RBF model does capture the underlying dynamics of the system.

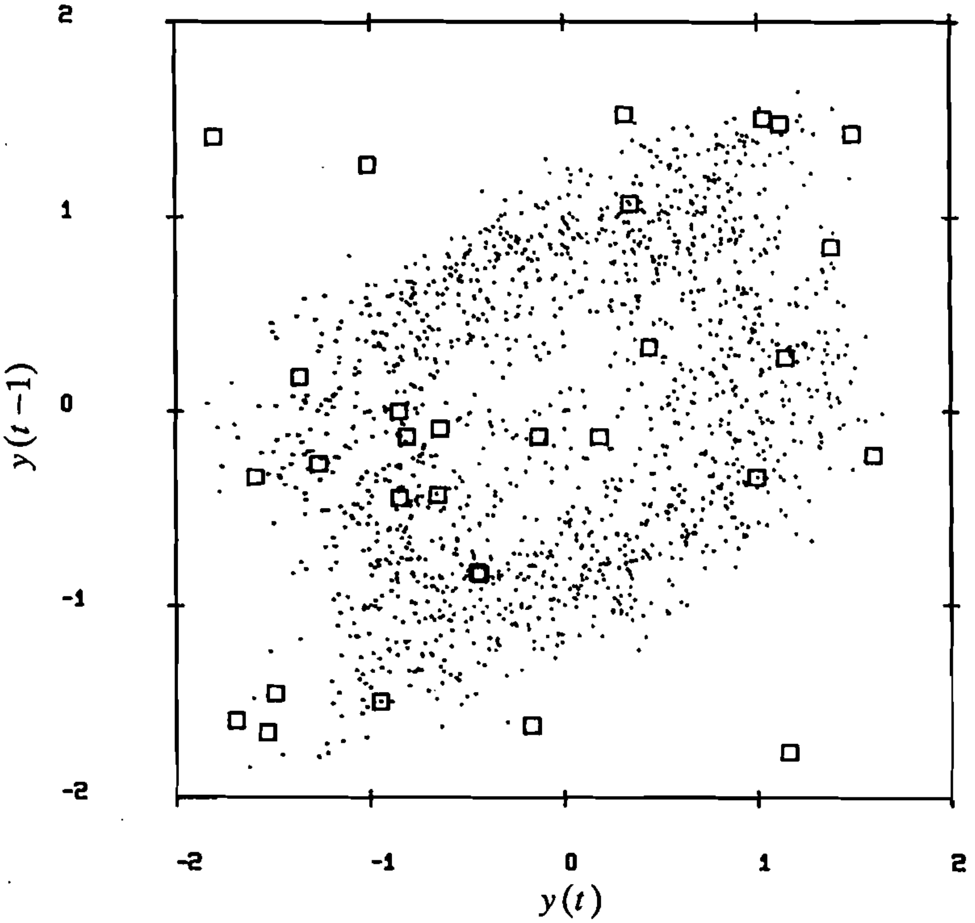


Figure 3. Distribution of observations and RBF centres. 1500 observation samples, □: position of RBF centre.

The rate of convergence for the hybrid identification algorithm shown in Fig. 2 is quite remarkable. As a comparison, a two-layer perceptron was fitted to the data using the parallel recursive prediction error algorithm (Chen *et al.* 1990 d). This parallel recursive prediction error algorithm is known to have much better convergence properties over the back-propagation algorithm. The two-layer perceptron had a structure of $m = n_y = 2$ and 7 hidden nodes, giving rise to a total of 29 parameters. The evolution of the mean square error for this neutral network model is also depicted in Fig. 2. It is clear that the RBF model trained by the hybrid algorithm achieved much faster convergence.

Example 2

The process considered is a liquid level system. The system consists of a DC water pump feeding a conical flask which in turn feeds a square tank. The system input is the voltage to the pump motor and the system output is the water level in the canonical flask: 1000 samples of data generated in an experiment are shown in

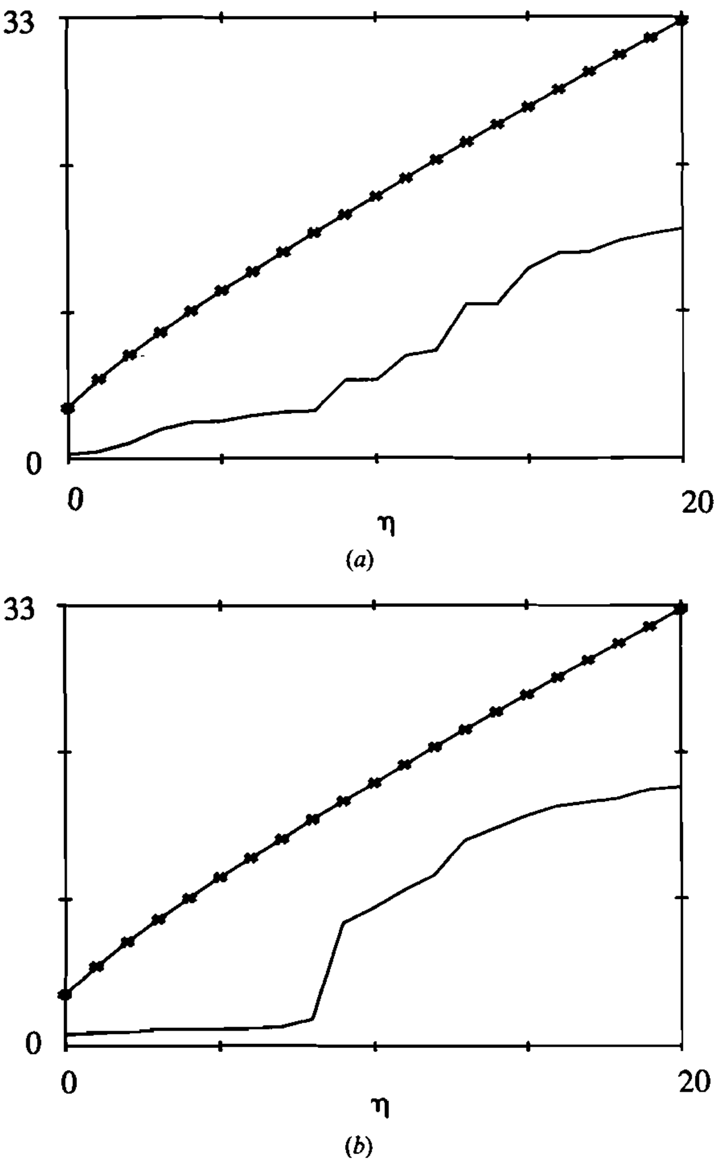


Figure 4. Chi-squared tests (Example 1): (*a*) $\omega(t) = \varepsilon^2(t-1)y(t-1)$; (*b*) $\omega(t) = \varepsilon(t-1)y^2(t-1)$; -✱-95% confidence band.

Fig. 9. The RBF model had a structure of $m = n_y + n_u = 3 + 5$ and $n = 40$. The parameters for the identification algorithm were chosen to be

$$\rho = 1000, \quad \lambda_0 = 0{\cdot}99, \quad \lambda(0) = 0{\cdot}95 \quad \text{and} \quad \alpha_c(0) = 0{\cdot}6$$

Random initial centres were used.

The evolution of the mean square error is depicted in Fig. 10. The mean square error was reduced from the initial value of 2 dB to the final value of −26 dB. The correlation tests for the identified model are shown in Fig. 11. It is observed that at

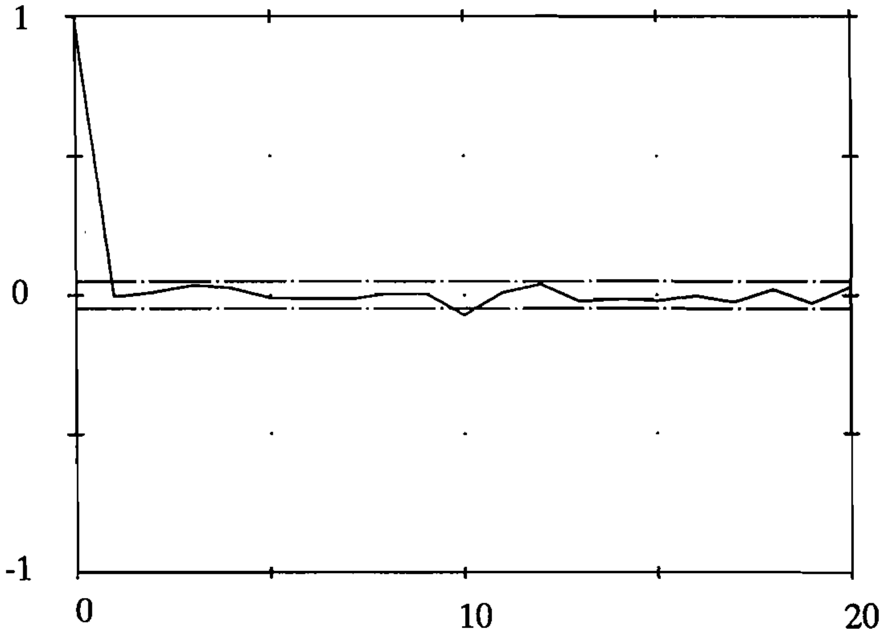


Figure 5. Autocorrelations of residuals (Example 1). — · — 95% confidence band.

three points the values of $\Phi_{u\varepsilon}(k)$ are slightly outside the 95% confidence bands. Several chi-squared tests were also computed. Again, only at a few isolated points are statistics slightly outside the confidence band, and this was judged a good result, considering that the data was from a real system and the identification algorithm was a recursive one.

A two-layer perceptron with a structure of $m = n_y + n_u = 3 + 5$ and 5 hidden nodes was also employed to identify this system using the parallel recursive prediction error algorithm. The network had a total of 51 parameters. The evolution of the mean square error is shown in Fig. 10, where it is seen that the reduction in the mean square error was much slower compared with that of the RBF model.

Example 3

The data was generated from a heat exchanger and contains 996 samples as shown in Fig. 12. A description of this process and the experiment design was given by Billings and Fadzil (1985). The dimension of the RBF centres was chosen to be $m = n_y + n_u = 6 + 6$ and the number of centres was $n = 90$. The parameters for the recursive algorithm were

$$\rho = 1000, \quad \lambda_0 = 0{\cdot}99, \quad \lambda(0) = 0{\cdot}95 \quad \text{and} \quad \alpha_c(0) = 0{\cdot}5$$

Initial centres were set randomly.

During the identification procedure, the mean square error was reduced from the initial 16 dB to the final $-13{\cdot}5$ dB, and the evolution of the mean square error is plotted in Fig. 13. The correlation tests and several chi-squared tests were

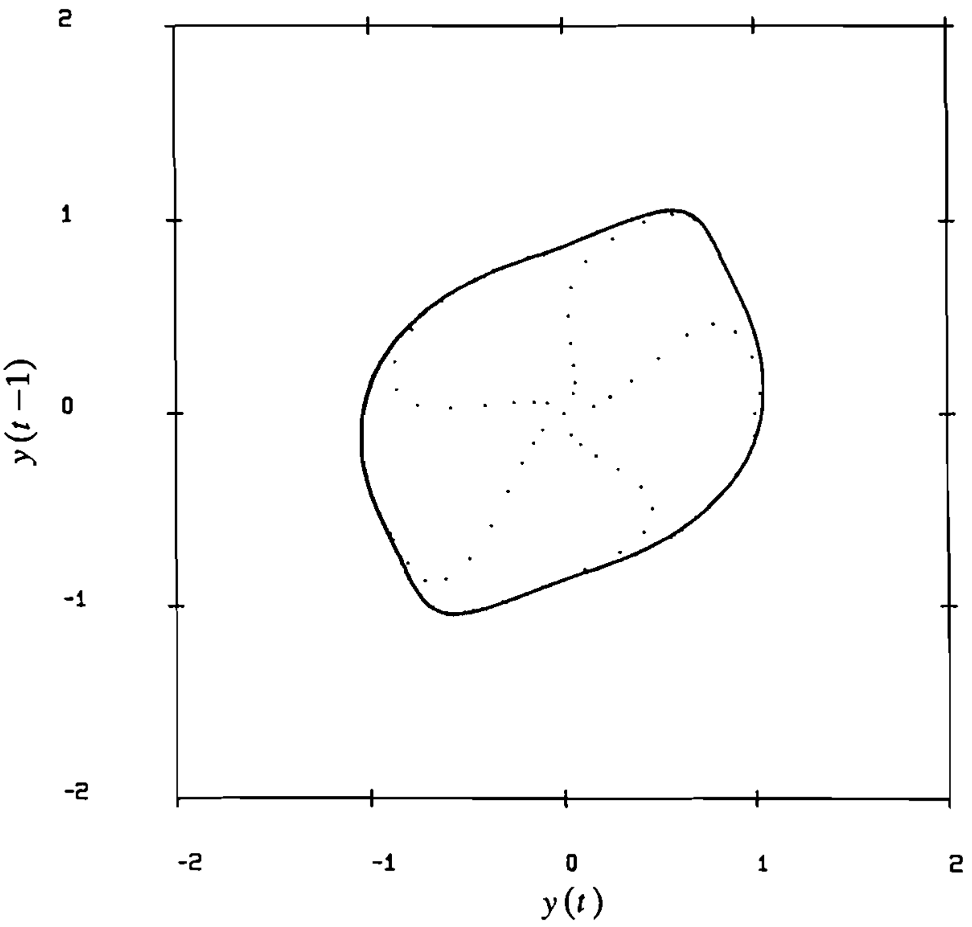


Figure 6. Limit cycle generated by autonomous system response. 1500 samples, initial condition: $y(0)=0{\cdot}1$ and $y(-1)=0{\cdot}01$.

computed, and they were all within the 95% confidence bands. Six chi-squared tests are shown in Fig. 14.

The data set was also fitted to a two-layer perceptron trained by the parallel recursive prediction error algorithm. The network structure was defined by $m=n_y+n_u=6+6$ and 6 hidden nodes, giving rise to a total of 85 parameters. Two passes of the data set were performed and the resulting mean square error sequences are plotted in Fig. 13. Initially the network parameters were set randomly. Possibly because the network parameters were in a region where gradients of the mean square error were almost flat, the mean square error remained very large and almost flat. After about 600 samples of updating, the parameter vector moved out of the original region of flat gradients and the mean square error began descending quickly. The parameter vector obtained at the end of the first pass was then used as the initial value and the algorithm was initialized a second time. The mean square error rapidly improved at first after a few recursions of the second pass and then became flat again, most likely because the parameter vector was

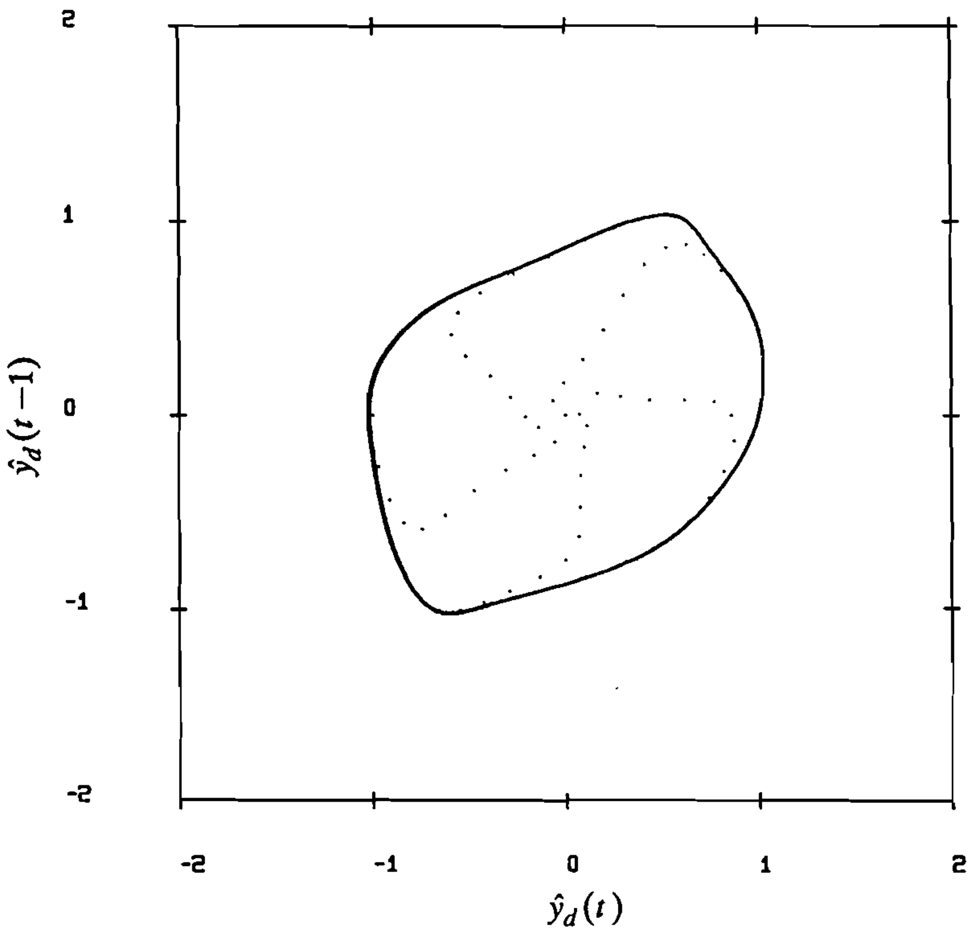


Figure 7. Limit cycle generated by iterative network response. 1500 samples, initial condition: $y(0) = 0{\cdot}1$ and $y(-1) = 0{\cdot}01$.

trapped at a local minimum of the mean square error. These problems highlight the difficulties associated with highly non-linear-in-the-parameter models trained by gradient learning algorithms, and also serve to demonstrate advantages of the linear learning rules for the RBF networks.

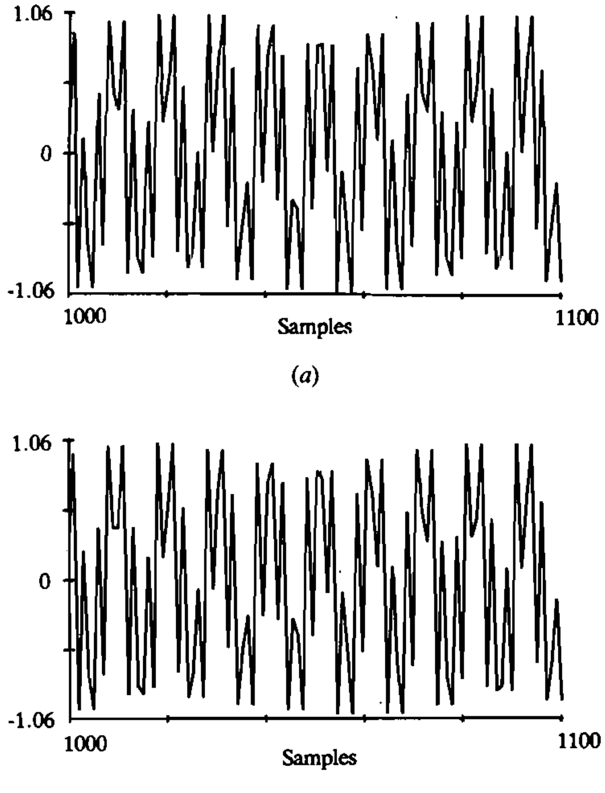
5. Extensions of the recursive hybrid algorithm

The hybrid algorithm of § 3 can be extended to the multi-input multi-output system. Consider

$$\mathbf{y}(t) = \mathbf{f}_s(\mathbf{y}(t-1), \ldots, \mathbf{y}(t-n_y), \mathbf{u}(t-1), \ldots, \mathbf{u}(t-n_u)) + \mathbf{e}(t) \tag{39}$$

and assume that the dimension of $\mathbf{y}(t)$ is p. A p-output RBF network is required to model the above system, and the network input vector is given by

$$\mathbf{v}(t) = [\mathbf{y}^{\mathrm{T}}(t-1) \ldots \mathbf{y}^{\mathrm{T}}(t-n_y)\mathbf{u}^{\mathrm{T}}(t-1) \ldots \mathbf{u}^{\mathrm{T}}(t-n_u)]^{\mathrm{T}} \tag{40}$$



(*a*)

(*b*)

Figure 8. (*a*) Autonomous system response; and (*b*) iterative network response. Initial condition: $y(0) = 0{\cdot}1$ and $y(-1) = 0{\cdot}01$.

The hidden layer of the RBF network remains unchanged, and the output layer of the network contains p linear combiners. Each of these linear combiners is defined as

$$\hat{y}_i(t) = f_{ri}(v(t)) = \sum_{j=1}^{n} \theta_{ij}\phi(\|v(t) - c_j\|), \quad 1 \leqslant i \leqslant p \tag{41}$$

p independent least squares estimators can be employed to identify the connection weights of these linear combiners. The above discussion can obviously be applied to the multi-input multi-output NARMAX model

$$y(t) = f_s(y(t-1), \ldots, y(t-n_y), \quad u(t-1), \ldots, u(t-n_u), \\ e(t-1), \ldots, e(t-n_e)) + e(t) \tag{42}$$

6. Conclusions

A hybrid clustering and Givens least squares algorithm has been developed for the recursive identification of non-linear systems using a radial basis function network. This hybrid algorithm combines the supervised least squares method with an unsupervised clustering technique. The centres of the radial basis function

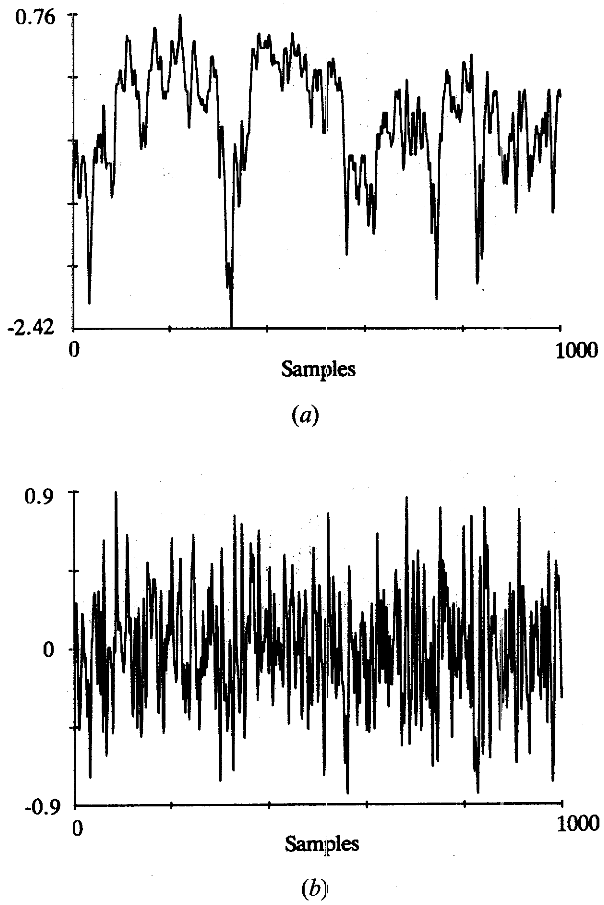


Figure 9. System outputs and inputs (Example 2): (*a*) outputs $y(t)$; (*b*) inputs $u(t)$.

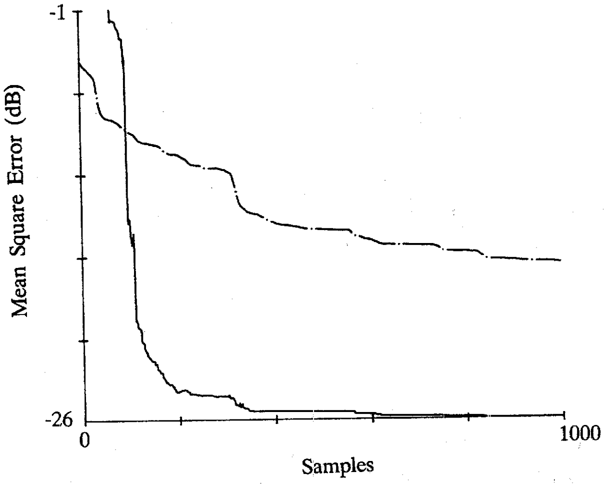


Figure 10. Evolution of mean square error (Example 2). Solid: RBF network, dashed: multi-layer perceptron.

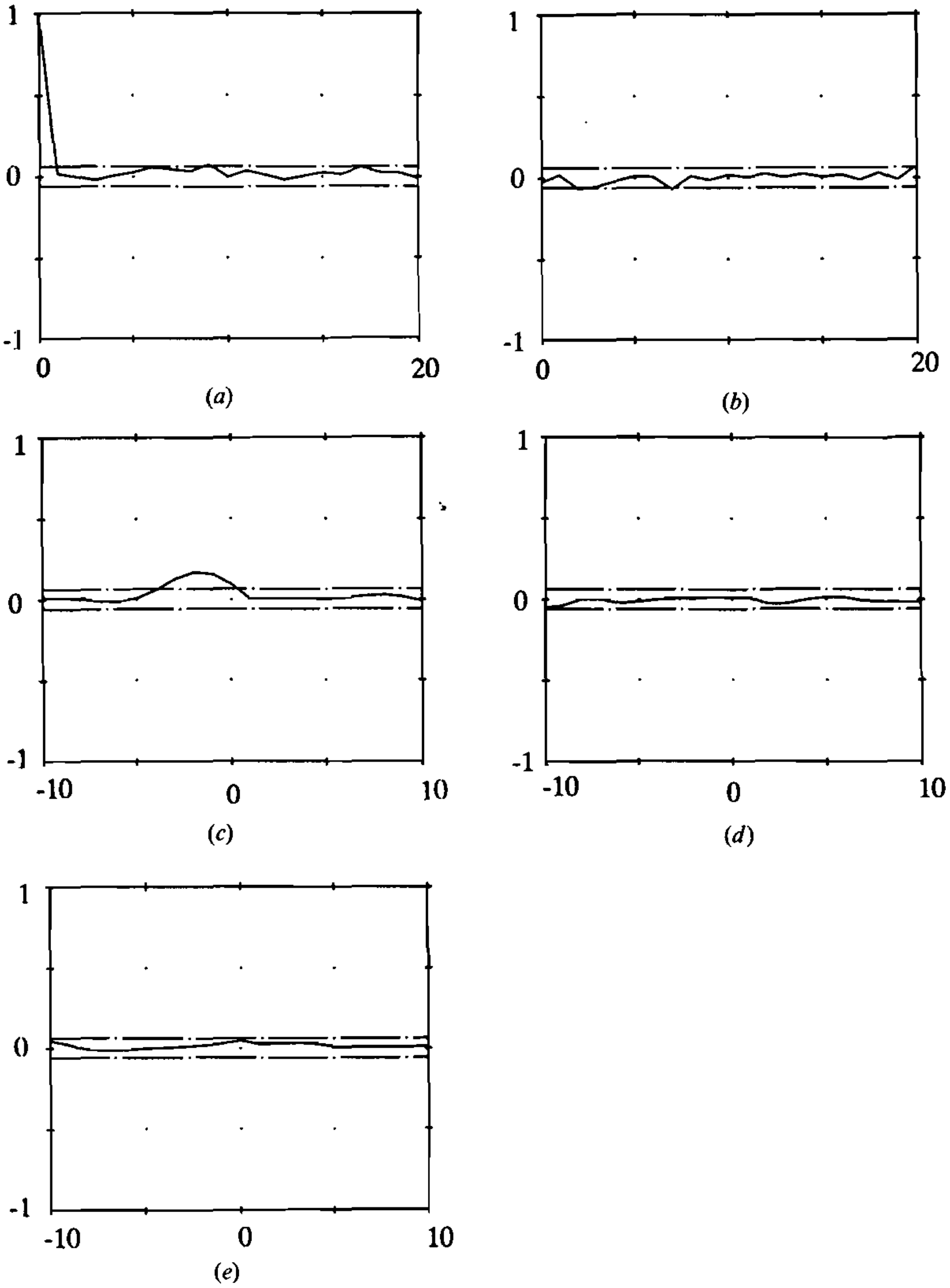


Figure 11. Correlation tests (Example 2): (*a*) $\Psi_{\varepsilon\varepsilon}(k)$; (*b*) $\Psi_{\varepsilon(\varepsilon u)}(k)$; (*c*) $\Psi_{u\varepsilon}(k)$; (*d*) $\Psi_{u^{2\prime}\varepsilon}(k)$; (*e*) $\Psi_{u^{2\prime}\varepsilon^2}(k)$; — · — 95% confidence band.

network are adjusted using the *n*-means clustering technique and the connection weights of the network are updated using the least squares principle. These two learning rules are implemented recursively and are thus appropriate for real-time or adaptive applications. Furthermore, they are linear learning rules, thus guaranteeing rapid convergence. Using three examples, a simulated non-linear time-series and

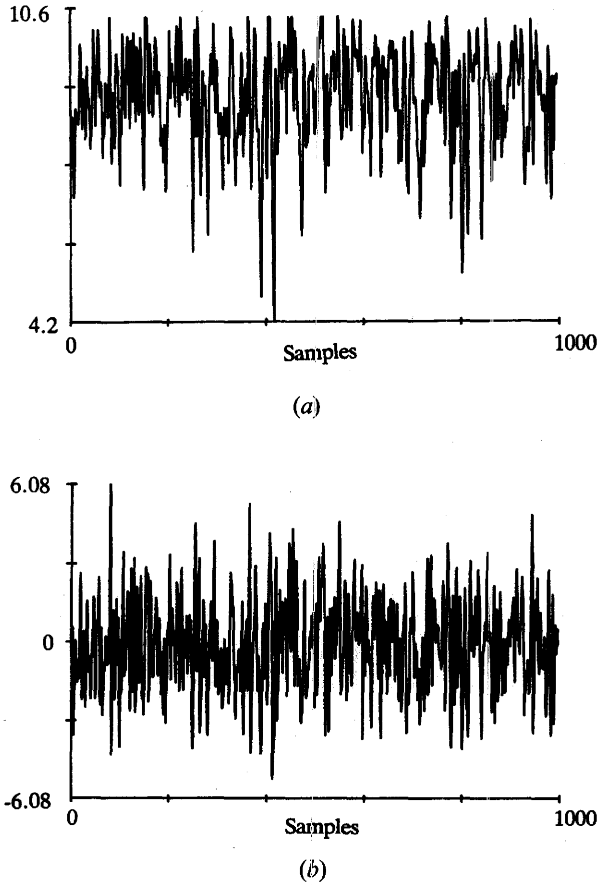


Figure 12. System outputs and inputs (Example 3): (*a*) outputs $y(t)$; (*b*) inputs $u(t)$.

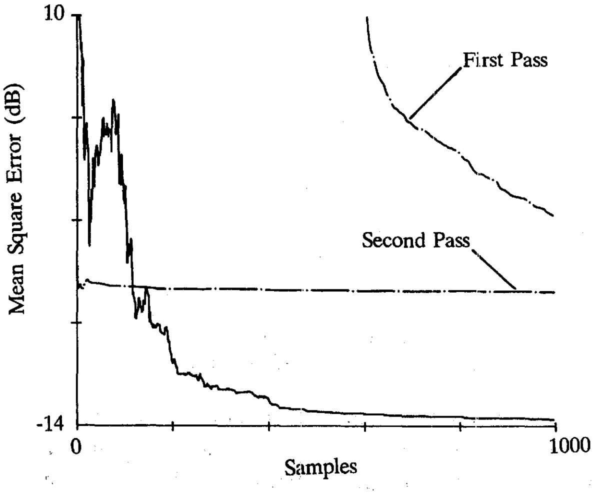
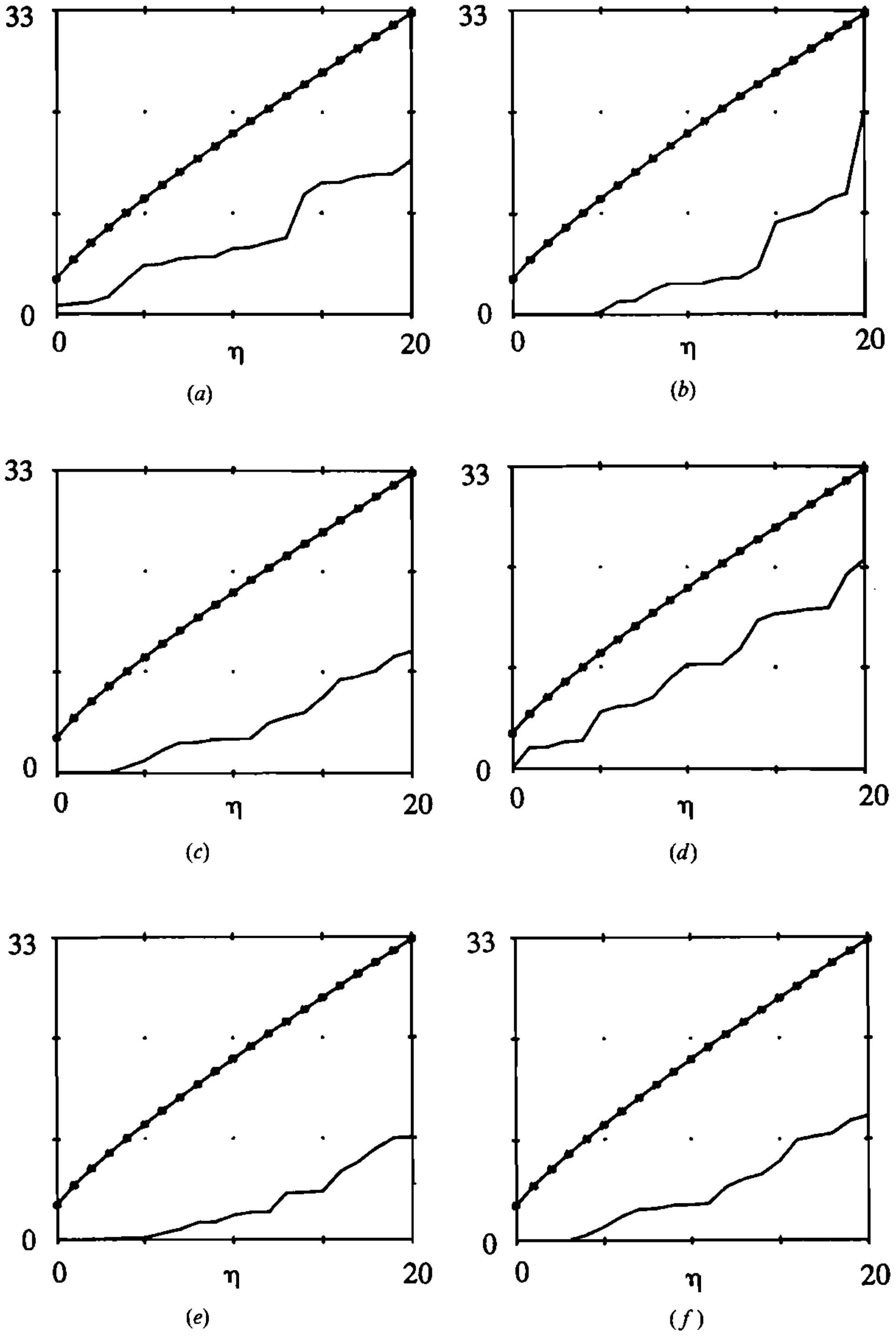


Figure 13. Evolution of mean square error (Example 3). Solid: RBF network; dashed: multi-layer perceptron.



(*a*) (*b*) (*c*) (*d*) (*e*) (*f*)

Figure 14. Chi-squared tests (Example 3): (*a*) $\omega(t) = \varepsilon(t-1)$; (*b*) $\omega(t) = u(t-1)$; (*c*) $\omega(t) = y(t-1)$; (*d*) $\omega(t) = \varepsilon^2(t-1)$; (*e*) $\omega(t) = u^2(t-1)$; (*f*) $\omega(t) = y^2(t-1)$; —◆— 95% confidence band.

two real processes, it has been shown that this hybrid approach offers a powerful on-line identification algorithm for radial basis function models.

REFERENCES

BILLINGS, S. A., and FADZIL, M. B., 1985, The practical identification of systems with nonlinearities. *Proceedings of the 7th IFAC Symposium on Identification and System Parameter Estimation*, York, U.K., pp. 155–160.

BILLINGS, S. A., and VOON, W. S. F., 1986, Correlation based model validity tests for non-linear models. *International Journal of Control*, **44**, 235–244.

BOHLIN, T., 1978, Maximum-power validation of models without higher-order fitting. *Automatica*, **4**, 137–146.

BROOMHEAD, D. S., and LOWE, D., 1988, Multivariable functional interpolation and adaptive networks. *Complex Systems*, **2**, 321–355.

CHEN, S., and BILLINGS, S. A., 1989, Representation of non-linear systems: the NARMAX model. *International Journal of Control*, **49**, 1013–1032.

CHEN, S., BILLINGS, S. A., and GRANT, P. M., 1990 a, Non-linear systems identification using neural networks. *International Journal of Control*, **51**, 1191–1214.

CHEN, S., BILLINGS, S. A., COWAN, C. F. N., and GRANT, P. M., 1990 b, Non-linear systems identification using radial basis functions. *International Journal of Systems Science*, **21**, 2513–2539; 1990 c, Practical identification of NARMAX models using radial basis functions. *International Journal of Control*, **52**, 1327–1350.

CHEN, S., COWAN, C. F. N., BILLINGS, S. A., and GRANT, P. M., 1990 d, Parallel recursive prediction error algorithm for training layered neutral networks. *International Journal of Control*, **51**, 1215–1228.

CYBENKO, G., 1989, Approximations by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, **2**, 303–314.

DUDA, R. O., and HART, P. E., 1973, *Pattern Classification and Scene Analysis* (New York: John Wiley).

FUNAHASHI, K., 1989, On the approximate realization of continuous mappings by neural networks. *Neural Networks*, **2**, 183–192.

GENTLEMAN, W. E., 1973, Least squares computations by Givens transformations without square roots. *Journal of the Institute of Mathematics and Applications*, **12**, 329–336.

GENTLEMAN, W. E., and KUNG, H. T., 1981, Matrix triangularization by systolic arrays. *Real-Time Signal Processing IV, SPIE Proceedings*, **298**, 19–26.

KOHONEN, T., 1987, *Self-Organization and Associative Memory* (Berlin: Springer Verlag).

LEONTARITIS, I. J., and BILLINGS, S. A., 1985, Input–output parametric models for non-linear systems—Part 1: deterministic non-linear systems; Part 2: stochastic non-linear systems. *International Journal of Control*, **41**, 303–344; 1987, Model selection and validation methods for non-linear systems. *International Journal of Control*, **45**, 311–341.

LJUNG, L., and SÖDERSTRÖM, T., 1983, *Theory and Practice of Recursive Identification* (Cambridge: MIT Press).

MACQUEEN, J., 1967, Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium on Mathematics Statistics and Probability, PI*, editors, LeCam, L. M., and Neymann, J., (University of California Press), pp. 281–297.

MOODY, J., and DARKEN, C., 1989, Fast-learning in networks of locally-tuned processing units. *Neural Computation*, **1**, 281–294.

POWELL, M. J. D., 1985, Radial basis functions for multivariable interpolation: a review. *IMA Conference on Algorithms for the Approximation of Functions and Data*, RMCS Shrivenham; 1987, Radial basis function approximations to polynomials. *Proceedings of the 12th Biennial Numerical Analysis Conference*, Dundee, pp. 223–241.

RUMELHART, D. E., HINTON, G. E., and WILLIAMS, R. J., 1986, Learning internal representations by error propagation, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, editors Rummelhart, D. E., and McClelland, J. L., (MIT Press), pp. 318–362.