International Journal of Systems Science
volume 35, number 4, 10 April 2004, pages 211–221

Taylor & Francis
Taylor & Francis Group

Identification of nonlinear discrete-time systems using raised-cosine radial basis function networks

A. F. Al-Ajlouni†, R. J. Schilling‡* and S. L. Harris§

An effective technique for identifying nonlinear discrete-time systems using raised-cosine radial basis function (RBF) networks is presented. Raised-cosine RBF networks are bounded-input bounded-output stable systems, and the network output is a continuously differentiable function of the past input and the past output. The evaluation speed of an n-dimensional raised-cosine RBF network is high because, at each discrete time, at most 2^n RBF terms are nonzero and contribute to the output. As a consequence, raised-cosine RBF networks can be used to identify relatively high-order nonlinear discrete-time systems. Unlike the most commonly used RBFs, the raised-cosine RBF satisfies a constant interpolation property. This makes raised-cosine RBF highly suitable for identifying nonlinear systems that undergo saturation effects. In addition, for the important special case of a linear discrete-time system, a first-order raised-cosine RBF network is exact on the domain over which it is defined, and it is minimal in terms of the number of distinct parameters that must be stored. Several examples, including both physical systems and benchmark systems, are used to illustrate that raised-cosine RBF networks are highly effective in identifying nonlinear discrete-time systems.

1. Introduction

Neural networks provide an effective framework for the identification and control of nonlinear discrete-time systems (Chen *et al.* 1990, Narendra and Parthasarathy 1990, Liu *et al.* 1999, Fang and Chow 2000, Chen and Chen 2001). In recent years, considerable effort has focused on the use of radial basis function (RBF) networks (Poggio and Girosi 1990, Chen and Billings 1992). RBF networks tend to have improved training characteristics in comparison with standard feedforward neural networks due to their localized nature and the fact that they are linear in the weights (Moody and Darken 1989, Gorinevski 1996). They can also be made compact by having the number of points about which the RBFs are centred grow and shrink to obtain a minimal network (Platt 1991, Chen *et al.* 1992, Kadirkamanathan and Niranjan 1993, Yingwei *et al.* 1997, Liu *et al.* 1999).

A variety of radial basis functions have been investigated and used including Gaussian functions (Schagen 1986), thin-plate splines (Duchon 1977), multiquadratics (Hardy 1971) and inverse multiquadratic functions (Powell 1987). In this paper, we make use of a network structure based on a raised-cosine radial basis function (Schilling *et al.* 2001) to identify discrete-time nonlinear systems. The class of nonlinear discrete-time systems considered consists of bounded-input bounded-output (BIBO) stable systems whose output depends continuously on the past input and past output. Raised-cosine RBF networks are BIBO-stable, and the network output is a continuously differentiable function of the past input and past output.

A raised-cosine RBF network consists of a sum of RBF terms, with each term centred about a grid point and scaled by a coefficient function. The coefficient functions are either zeroth- or first-degree polynomials of the state vector. Evaluation speed of an *n*-dimensional raised-cosine RBF network is very high because, at

Received 10 March 2001. Revised 22 January 2004. Accepted 19 March 2004.

† Hijjawi Faculty of Engineering Technology, Department of Communication Engineering, Yarmouk University, Irbid, Jordan.
‡ Electrical and Computer Engineering Department, Clarkson University, Potsdam, NY 13699, USA.
§ Chemical Engineering Department, Clarkson University, Potsdam, NY 13699, USA.
* To whom all correspondence should be addressed.
email: robert.schilling@clarkson.edu

International Journal of Systems Science ISSN 0020–7721 print/ISSN 1464–5319 online © 2004 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/00207720410001703213

each discrete time, at most 2^n RBF terms are nonzero and contribute to the output. As a consequence of this property, raised-cosine RBF networks can be used to identify relatively high-order nonlinear discrete-time systems. Simple analytical expressions are available for the weights of a raised-cosine RBF network that cause the network output and its gradient to be exact on the grid over which the network is defined. For high-precision grids, these weights effectively mould the input–output behaviour of the RBF network to that of the system being identified. For low-precision grids, the computed weights serve as an effective initial guess for a learning algorithm based on the least mean square (LMS) steepest descent method. Unlike other popular RBFs such as the Gaussian RBF, the raised-cosine RBF satisfies a constant interpolation property. This allows raised-cosine RBF networks to efficiently model nonlinear systems that undergo saturation effects when the system is overdriven with a large input. The constant interpolation property also makes the raised-cosine RBF network particularly effective for the important special case of a linear discrete-time system. For linear systems, a first-order RBF network is exact on the domain over which it is defined, and it is minimal in terms of the number of distinct parameters that must be stored. A number of simulation examples are presented that illustrate that raised-cosine RBF networks can be successfully used to identify a variety of nonlinear discrete-time systems.

2. Problem formulation

Consider a nonlinear discrete-time system or plant of the following general form.

$$y_p(k) = f[y_p(k-1), \ldots, y_p(k-N), u(k-1), \ldots, u(k-M)], \quad k \geq 0. \tag{1}$$

For convenience, (1) will be referred to as the system S_f. A more compact alternative formulation of S_f is obtained by introducing a *state vector* consisting of past outputs and past inputs.

$$x_p(k) \triangleq [y_p(k-1), \ldots, y_p(k-N), u(k-1), \ldots, u(k-M)]^T. \tag{2}$$

Thus, $x_p(k)$ is an $n \times 1$ vector denoting the state of S_f at time k where $n = N + M$. The output of S_f at time k is then

$$y_p(k) = f[x_p(k)], \quad k \geq 0. \tag{3}$$

The nonlinear system S_f corresponds to the most general class of systems (Model IV) introduced in (Narendra and Parthasarathy 1990). More specialized classes of systems include systems that are linear in the past outputs (Model I), linear in the past inputs (Model II), and nonlinear but separable between the past inputs and outputs (Model III).

The technique presented in this paper can be applied to a slightly more general system in which the current input, $u(k)$, also appears on the right-hand side of (1). However, almost all of the benchmark examples of systems cited in the literature fall into the more restrictive form of (1). Models of this form are particularly suitable for real-time control applications because the plant output at time k can be precomputed during sampling interval $k-1$, thereby reducing the latency in updating the control signal $u(k)$.

Assumptions: Two assumptions are made regarding the nonlinear discrete-time system to be identified:

(1) The function $f : R^n \rightarrow R$ is *continuous*.
(2) The system S_f is BIBO-*stable*.

Neither assumption is particularly restrictive in terms of limiting potential applications. Suppose that, in the normal operation of the system S_f, the input is restricted to the following interval where b is a 2×1 vector of input bounds:

$$b_1 \leq u(k) \leq b_2, \quad k \geq 0. \tag{4}$$

Since S_f is BIBO stable, there exists a corresponding 2×1 vector of output bounds a such that

$$a_1 \leq y_p(k) \leq a_2, \quad k \geq 0. \tag{5}$$

The BIBO stability assumption effectively restricts the domain of the function f to the following compact subset of R^n.

$$X = [a_1, a_2]^N \times [b_1, b_2]^M. \tag{6}$$

Thus the problem of representing or modelling the nonlinear discrete-time system S_f reduces to one of representing a continuous function f on a compact set X. In this paper, it is proposed that this be done using a raised-cosine radial basis function (RBF) neural network.

$$y(k) = g[x(k)]. \tag{7}$$

The system in (7) will be referred to as the RBF network S_g. Here, $x(k)$ is an $n \times 1$ vector representing the state of the network at time k, and $y(k)$ is the network output. Let $e(k)$ denote the error between the system S_f and the RBF network S_g at time k.

$$e(k) = y_p(k) - y(k). \tag{8}$$

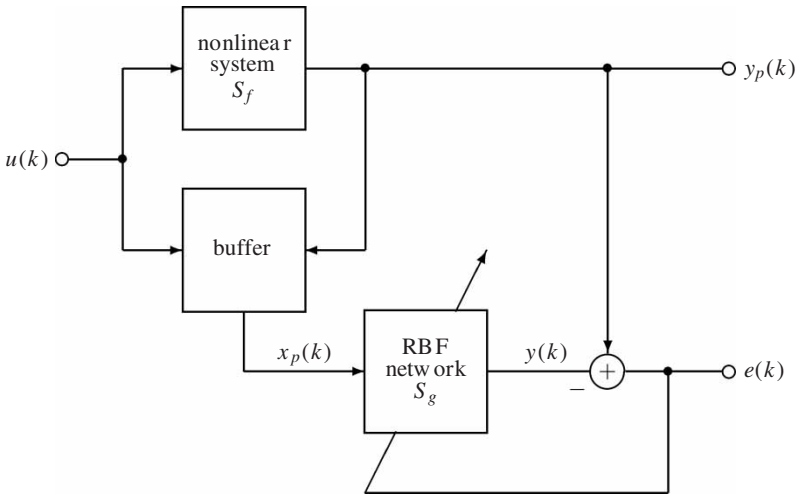

Figure 1. Training a RBF network S_g using the system state x_p.

Given measurements of the input $u(k)$ and system output $y_p(k)$, the problem is to adapt the network parameters so as to minimize the square of the error $e(k)$. This will be achieved using the block diagram shown in figure 1. Note that during the learning or adaptation process, the RBF network S_g uses the system state, $x_p(k)$, rather than the network state, $x(k)$. However, once the learning process is terminated, the RBF network is independent of S_f with the output $y(k)$ computed as in (7). This technique of using the system state to train the RBF network has been referred to as the series-parallel method (Narendra and Parthasarathy 1990).

3. Raised-cosine RBF network

A raised-cosine RBF neural network has been introduced as a representation for approximating continuous-time nonlinear systems (Schilling *et al*. 2001). In this paper, it is demonstrated that the same basic structure is highly effective when applied to the problem of real-time identification of nonlinear discrete-time systems. The general form of a first-order raised-cosine RBF network is as follows.

$$\begin{aligned} y(k) &= g[x(k)] \\ &= [w + Vx(k)]^T \phi[x(k)]. \end{aligned} \tag{9}$$

Here, w is an $r \times 1$ *weight vector*, V is an $r \times n$ *linear weight matrix*, and $\phi(x)$ is an $r \times 1$ vector of radial basis functions. Since the coefficient of $\phi(x)$ is a first-degree polynomial in x, the RBF network in (9) is referred to as a first-order network. When the weight matrix $V = 0$, (9) reduces to a zeroth-order RBF network.

3.1. *Grid points*

Each component of the vector $\phi(x)$ is an RBF centred about a particular grid point. Let $d_j \geq 2$ be the number of grid points along dimension j for $1 \leq j \leq n$. Then, the total number of grid points is

$$r = d_1 d_2 \cdots d_n. \tag{10}$$

If the grid points are distributed uniformly along dimension j, then from (6) the mth grid value along dimension j is

$$x_{jm} = \begin{cases} a_1 + \dfrac{(m-1)(a_2 - a_1)}{d_j - 1}, & 1 \leq j \leq N \\ b_1 + \dfrac{(m-1)(b_2 - b_1)}{d_j - 1}, & N+1 \leq j \leq N+M. \end{cases} \tag{11}$$

In order to arrange the r points of the n-dimensional grid $\{x_{jm}\}$ into a one-dimensional array $\{x^i\}$, it is helpful to introduce a vector index $q \in R^n$ whose elements specify the grid point subscript along each dimension. That is, element q_j takes on integer values in the range $1 \leq q_j \leq d_j$ for $1 \leq j \leq n$. If $x[q]$ denotes the qth grid point, then the jth element of $x[q]$ is defined

$$x_j[q] = x_{jq_j}, \quad 1 \leq j \leq n, \quad 1 \leq q_j \leq d_j. \tag{12}$$

The r grid points can be stacked into a long one-dimensional array. Given the vector index q of grid point $x[q]$, the corresponding scalar index i is computed as follows.

$$\begin{aligned} i = q_1 &+ d_1(q_2 - 1) + d_1 d_2(q_3 - 1) + \cdots \\ &+ d_1 d_2 \cdots d_{n-1}(q_n - 1). \end{aligned} \tag{13}$$

The ith grid point can be expressed either as x^i or as $x[q]$ with (13) used to go back and forth between the two representations. Suppose v^i denotes the ith row of the linear weight matrix V. Then the RBF network output in (9) can be expressed explicitly in terms of the scalar index i as follows.

$$y(k) = \sum_{i=1}^{r} [w_i + v^i x(k)] \phi_i[x(k)]. \tag{14}$$

3.2. *Raised-cosine RBF*

To define the vector of RBF functions $\phi(x)$, it is helpful to start with the special case of a one-dimensional raised-cosine RBF centred at $z = 0$.

$$\psi(z) \triangleq \begin{cases} \dfrac{1 + \cos(\pi z)}{2}, & |z| \le 1 \\ 0, & |z| > 1. \end{cases} \tag{15}$$

Note that $\psi(z)$ is a continuous function with compact support. The scalar RBF in (15) is normalized to a grid of unit radius. To adjust $\psi(z)$ to the grid size along dimension j, one can simply divide by the grid spacing along dimension j.

$$\Delta x_j = \frac{x_{jd_j} - x_{j1}}{d_j - 1}, \quad 1 \le j \le n. \tag{16}$$

The ith RBF is centred about grid point x^i, and the n-dimensional raised-cosine RBF is constructed as a product of n scalar RBFs, one for each dimension. Thus, the ith raised-cosine RBF can be expressed as follows.

$$\phi_i(x) = \prod_{j=1}^{n} \psi\left(\frac{x_j - x_j^i}{\Delta x_j}\right), \quad 1 \le i \le r. \tag{17}$$

A plot of a two-dimensional raised-cosine RBF centred at the origin is shown in figure 2. Like the scalar version, it is a continuous function with compact support. The raised-cosine RBFs in (17) have a number of interesting and useful properties (Schilling *et al.* 2001). Both the network function $g(x)$ in (9) and its gradient $\nabla g(x)$ are continuous. Thus, the RBF network output is a continuously differentiable function of the past input and past output. As a consequence of their

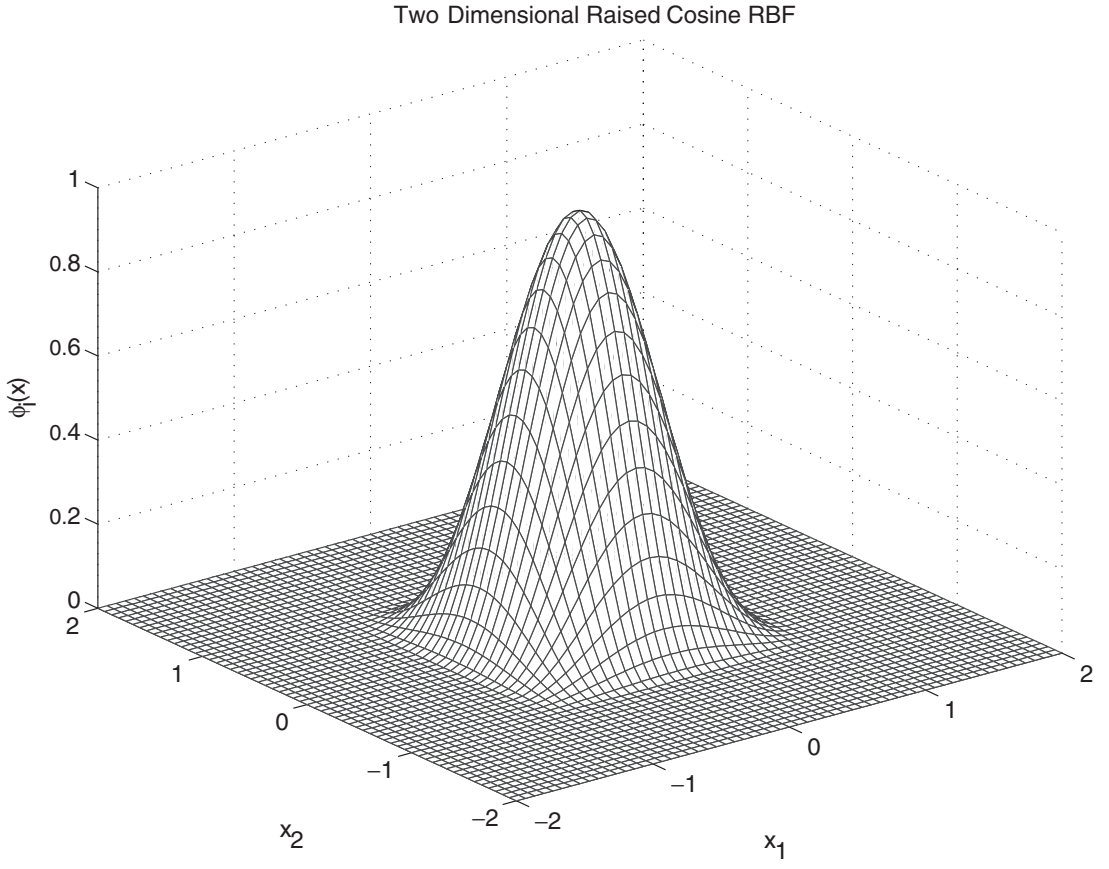

Figure 2. Two-dimensional raised-cosine RBF centred at the origin.

compact support, the raised-cosine RBFs in (17) satisfy the following basic orthogonality property on the grid.

$$\phi_i(x^j) = \begin{cases} 1, & j = i \\ 0, & j \neq i. \end{cases} \tag{18}$$

This leads directly to the following simple, but highly effective, expressions for initial guesses for the network parameters. Recalling that v^i is the ith row of V, let

$$v^i = \nabla f(x^i), \qquad 1 \leq i \leq r \tag{19}$$

$$w_i = f(x^i) - v^i x^i, \quad 1 \leq i \leq r. \tag{20}$$

Using the orthogonality property in (18), it can be shown that the RBF network output and its gradient are then exact on the set of grid points.

$$g(x^i) = f(x^i), \qquad 1 \leq i \leq r \tag{21}$$

$$\nabla g(x^i) = \nabla f(x^i), \quad 1 \leq i \leq r. \tag{22}$$

For a sufficiently high precision grid, the weights in (19) and (20) effectively mould the input–output behaviour of the RBF network to that of the system being identified. Of course, it may not be possible to directly evaluate $f(x)$ and $\nabla f(x)$ at prescribed values of x, in which case w and V can be set to zero or to small random values. The network then can be trained by minimizing $e^2(k)$ using a numerical search method where $e(k)$ is the error defined in (8). If $\mu > 0$ is the step size, the least mean square (LMS) method based on the steepest descent (Widrow and Stearns 1985) results in a learning algorithm with the following weight update formulas:

$$w(k+1) = w(k) + 2\mu e(k)\phi[x(k)] \tag{23}$$

$$V(k+1) = V(k) + 2\mu e(k)\phi[x(k)]x^T(k). \tag{24}$$

The evaluation speed of the raised-cosine RBF network is also quite high. This is a consequence of the fact that for each $x \in X$, there are at most 2^n nonzero terms in (14). The grid points of the nonzero terms form a hypercube in X, and the vector indices of the vertices of this hypercube can be quickly identified and then converted to scalar indices using (13). Thus, the computational effort for evaluating the RBF network output grows only as 2^n, and it is independent of the precision d of the grid. Note from (15) and (17) that $|\phi_i(x)| \leq 1$. Hence, from (14), it follows that if $x(k)$ is bounded, the RBF network output $y(k)$ is also bounded. That is, the RBF network, S_g, is BIBO-stable.

Another important qualitative property of the raised-cosine RBF network is the constant interpolation property (Schilling *et al*. 2001).

$$\sum_{i=1}^{r} \phi_i(x) = 1, \quad x \in X. \tag{25}$$

This property (which is not shared by a Gaussian RBF) is useful when the desired output is constant over certain regions (Lee and Chung 1999). This can occur, for example, when a nonlinear system undergoes saturation effects as a result of being overdriven by an excessively large input. The constant interpolation property is illustrated graphically using a two-dimensional RBF network with four grid points in figure 3.

3.3. *Linear discrete-time systems*

In view of the constant interpolation property, the first-order raised-cosine RBF network is particularly effective in representing linear discrete-time systems. If the system S_f is linear, then from (3) there exists an $n \times 1$ coefficient vector c such that

$$y_p(k) = c^T x_p(k). \tag{26}$$

Suppose $d_j = 2$ for $1 \leq j \leq n$. Let the weight vector be $w = 0$, and let the rows of the linear weight matrix V be defined

$$v^i = c^T, \quad 1 \leq i \leq r. \tag{27}$$

Next, let $x(k) \in X$ be arbitrary. Using (14), (27), and the constant interpolation property in (25), the RBF network output is

$$\begin{aligned} y(k) &= \sum_{i=1}^{r} [w_i + v^i x(k)]\phi_i[x(k)] \\ &= \sum_{i=1}^{r} c^T x(k)\phi_i[x(k)] \\ &= c^T x(k) \sum_{i=1}^{r} \phi_i[x(k)] \\ &= c^T x(k). \end{aligned} \tag{28}$$

Thus, the RBF network is exact on X when the system S_f is linear. Note from (27) that the number of distinct parameters that have to be stored to compute w and V in this case is n. Thus, the RBF network is also optimal for a linear system in the sense that the number of distinct parameters that must be stored is the minimum possible. When the system S_f is linear, the RBF network in (9) simplifies to

$$y(k) = x^T(k)V^T\phi[x(k)]. \tag{29}$$

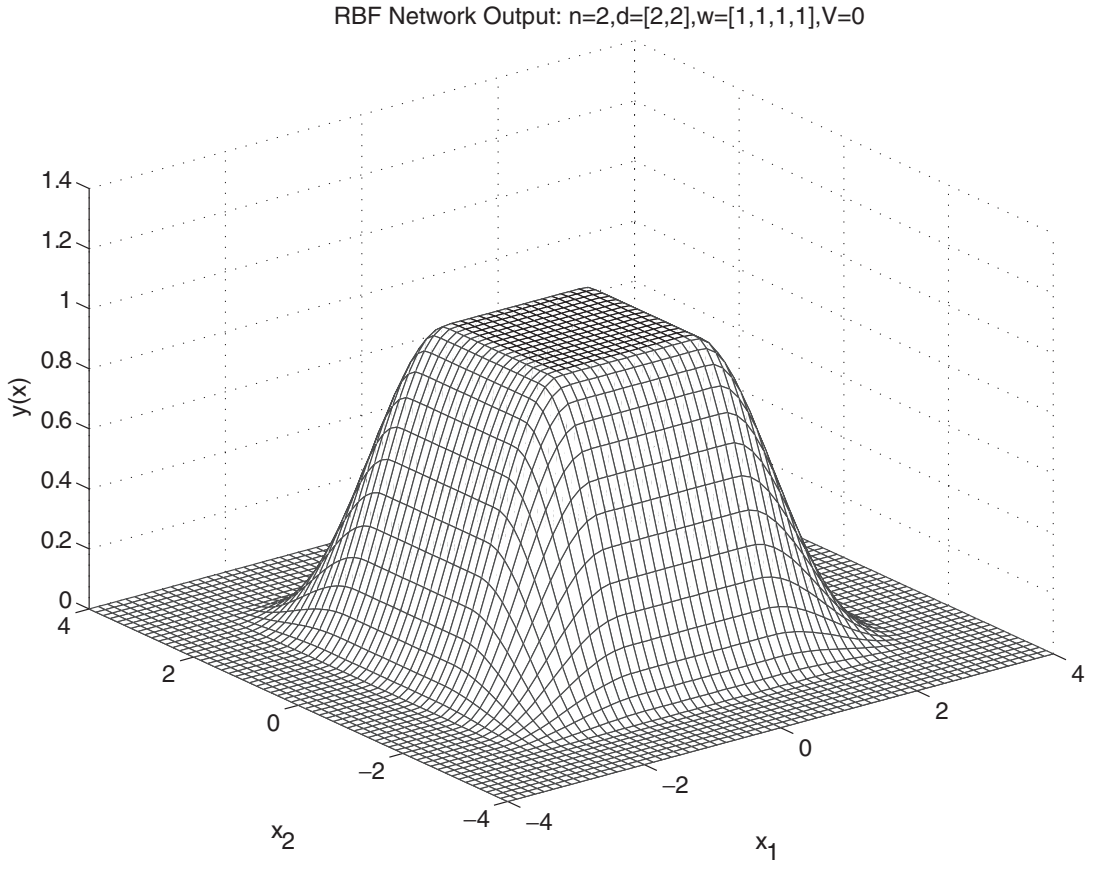

Figure 3. Constant interpolation property of raised-cosine RBF network: $n = 2$, $d = [2, 2]^T$, $w = [1, 1, 1, 1]^T$, $V = 0$.

4. Examples

In this section, a number of examples illustrating the identification of nonlinear discrete-time systems are presented. For all of the examples, the inputs used are random white-noise sequences uniformly distributed over $[b_1, b_2]$. The system under investigation is first driven by a *domain input*, $u_d(k)$, that is used to estimate the operating range of system. Let

$$y_{\min} = \min_{k=1}^{m}\{y_p(k)\} \tag{30}$$

$$y_{\max} = \max_{k=1}^{m}\{y_p(k)\}. \tag{31}$$

The domain X of the RBF network is then determined by computing output bounds that are slightly larger in each direction to take into account the fact that the domain input $u_d(k)$ is of finite length.

$$a_1 = y_{\min} - 0.05 \tag{32}$$

$$a_2 = y_{\max} + 0.05(y_{\max} - y_{\min}). \tag{33}$$

The closeness of the fit between the system and the RBF network is measured by using a separate m point *test input*, $u_t(k)$. The two test outputs are used to compute the following normalized root mean square error, where $e(k)$ is defined in (8):

$$E_{\text{rms}} \triangleq \frac{\sqrt{\sum_{k=0}^{m-1} e^2(k)}}{\sqrt{\sum_{k=0}^{m-1} y_p^2(k)}}. \tag{34}$$

The measure of error in (34) is normalized in the sense that if $w = 0$ and $V = 0$, then $E_{\text{rms}} = 1$.

4.1. *Example 1 (tunable plucked-string filter)*

The first example is a linear system designed to synthesize the sound produced by a stringed musical instrument (Steiglitz 1996). A block diagram of the tunable plucked-string filter is shown in figure 4. The objective for this type of system is to produce a magnitude response that consists of series of harmonically related resonances of decreasing amplitude with the first resonance tuned to the pitch of the instrument. The feedback delay block produces a comb filter with uniform resonance peaks. The FIR lowpass filter block causes the size of the peaks to decay gradually with increasing frequency. Finally, the IIR allpass filter block is used to finely tune the location of the

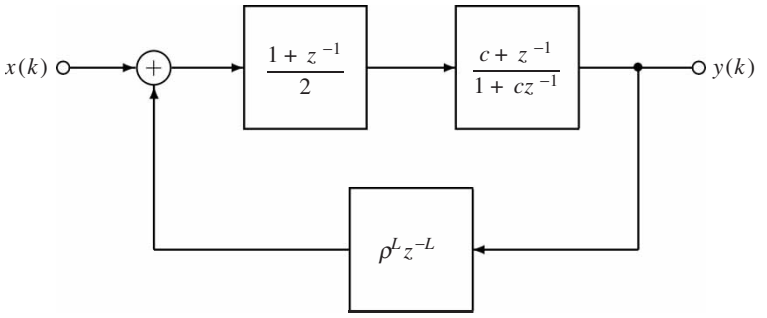

Figure 4. Tunable plucked-string filter.

first resonance. Analysis of figure 4 reveals that the overall transfer function of the tunable plucked-string filter is

$$H(z) = \frac{0.5[c + (1+c)z^{-1} + z^{-2}]}{1 + cz^{-1} - 0.5\rho^L[cz^{-L} + (1+c)z^{-(L+1)} + z^{-(L+2)}]}. \tag{35}$$

This example was chosen to illustrate that a first-order raised-cosine RBF network can be used to efficiently model a relatively high-order linear discrete-time system. Suppose the sampling frequency is $f_s = 20{,}000\,\text{Hz}$, and the desired location for the first resonance is $f_0 = 1440\,\text{Hz}$. Using the technique discussed in Jaffe and Smith (1983), appropriate values for the delay L and fine tuning parameter c are $L = 13$ and $c = 0.44$. Suppose the feedback attenuation factor is set to $\rho = 0.98$. In this case, $N = 15$, $M + 1 = 3$, and the order of the system is $n = N + M + 1 = 18$. Notice from (35) that the current input $u(k)$ appears on the right-hand side in this case, hence the use of $M + 1$ rather than M. Given that $n \gg 1$, the size of the grid is quite large at $r = 2^n = 262144$. However, because S_g is linear, the number of distinct parameter values that must be stored for the raised-cosine RBF network is small. In this case, $w = 0$, and the first row of V is

$$v^1 = [-c, 0, \ldots, 0, c\rho^L/2, (1+c)\rho^L/2, \rho^L/2]. \tag{36}$$

Thus, a total of n distinct parameter values must be stored, the remaining rows of V being copies of row one. Using (32) and (33), the domain of the RBF network was found to be

$$X = [-1.1644, 1.1125]^{15} \times [-1, 1]^3. \tag{37}$$

Plots of the magnitude responses of the tunable string filter S_g and the RBF network are shown in figure 5. The two are indistinguishable in this case because the RBF network is an exact representation on X.

4.2. *Example 2 (separable nonlinear system)*

As an example of a nonlinear system, consider the following discrete-time system introduced in Narendra and Parthasarathy (1990):

$$y_p(k) = \frac{y_p(k-1)}{1 + y_p^2(k-1)} + u^3(k-1). \tag{38}$$

This system is nonlinear in both the past inputs and the past outputs, but it is separable, which makes it a Model III type system. The objective of this example is to demonstrate the training of the RBF network. Using (32) and (33), the domain of the RBF network was determined to be

$$X = [-1.5945, 1.6157] \times [-1, 1]. \tag{39}$$

A zeroth-order network was trained with a white noise input, u_{tr}, consisting of 30,000 points uniformly distributed over $[-1, 1]$. The initial value for the weight vector was $w = 0$, and the step size was $\mu = 0.15$. The grid parameters used for the RBF network were $n = 2$ and $d = [8, 8]^T$. A 1000-point white noise test input was used to evaluate the network performance, which produced a normalized RMS error of $E_{\text{rms}} = 0.133$. The system output and the RBF network output are shown in figure 6. For clarity, only the first 100 points are displayed. The horizontal lines represent the RBF grid values for the output. The fit in figure 6 is clearly not exact, but it is quite reasonable considering the type of test input (broad spectral content), and the number of weights used, $r = 64$. This is in contrast to the model used in Narendra and Parthasarathy (1990) which consisted of two separate feedforward three-layer networks of type $N^3_{1-20-10-1}$ resulting in a total of 460 weights. Thus, the raised-cosine RBF network representation in figure 6 uses almost an order of magnitude fewer weights in this case. The number of

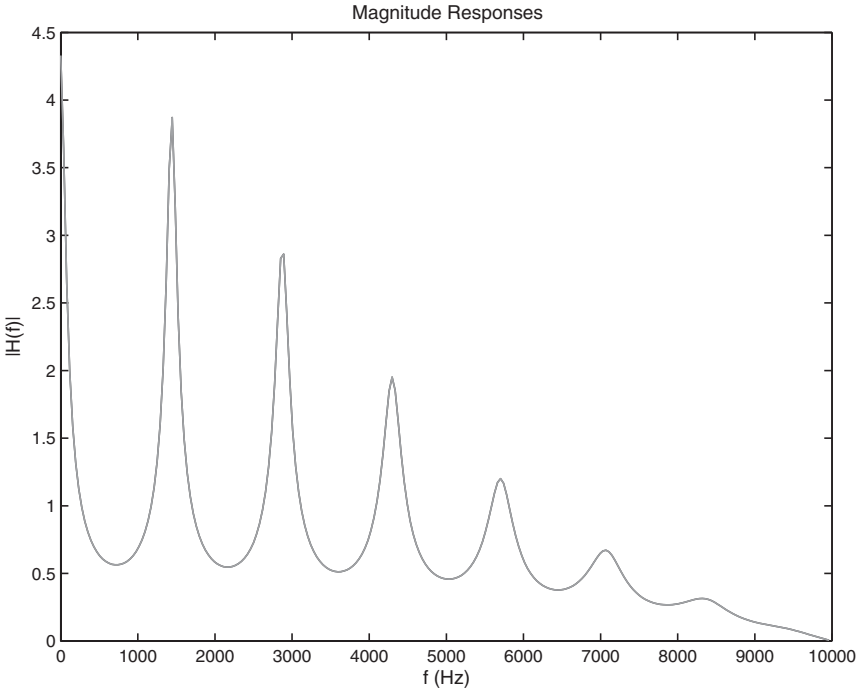

Figure 5. Magnitude responses of a plucked-string filter and first-order RBF network with $L = 13$, $c = 0.44$, and $\rho = 0.98$.

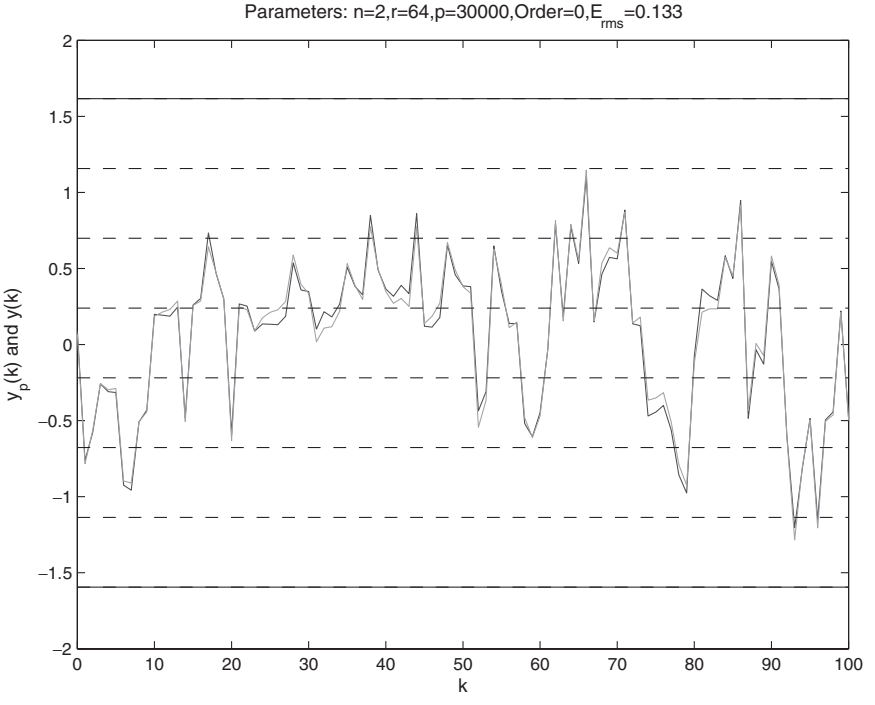

Figure 6. Outputs of a separable nonlinear system and trained zeroth-order RBF network with $n = 2$, $d = [8, 8]^T$, $E_{\text{rms}} = 0.133$.

weights might be reduced even further by using two separate raised-cosine RBF networks.

4.3. *Example 3 (continuous stirred tank reactor)*

The Van de Vusse reaction in a continuous stirred tank reactor (CSTR) can be modelled by the following nonlinear discrete-time system introduced in Hernandez and Arkun (1996):

$$y_p(k) = c_1 + c_2 u(k-1) + c_3 y_p(k-1) + c_4 u^3(k-1) + c_5 y_p(k-2)u(k-2)u(k-1). \quad (40)$$

Here, $y_p(k)$ denotes the product concentration at time $t = kT$, and $u(k)$ is a scaled reactant concentration at time $t = kT$, where the sampling interval is $T = 0.04$ h. The input has been normalized to $0 \le u(k) \le 1$, and the parameters of this system are

$$c = [0.558, 0.538, 0.116, -0.127, -0.034]^T. \quad (41)$$

This is a system of dimension $n = 4$ with $N = 2$ and $M = 2$ that is nonlinear in both the past inputs and the past outputs. Using (32) and (33), the domain of the RBF network was determined to be

$$X = [0.6151, 1.0872]^2 \times [0, 1]^2. \quad (42)$$

The grid parameters for a first-order RBF network were $n = 4$, $d = [3, 3, 3, 3]^T$. The initial weights were computed using (19) and (20), and the network was trained using a step size of $\mu = 0.1$ and 5000 random points uniformly distributed over $[0, 1]$. A separate 1000-point white-noise test input uniformly distributed over $[0, 1]$ was used to evaluate the network performance. The system output and the RBF network output are shown in figure 7 where, for clarity, only the first 50 points are displayed. The fit is quite close with a normalized RMS error of $E_{\text{rms}} = 0.005$. This was achieved using $r(n + 1) = 405$ weights.

4.4. *Example 4 (higher-order nonlinear system)*

As a final example, consider the following Model IV type system introduced in Narendra and Parthasarathy (1990).

$$y_p(k) = \frac{y_p(k-1)y_p(k-2)y_p(k-3)u(k-2)[y_p(k-3)-1] + u(k-1)}{1 + y_p^2(k-2) - y_p^2(k-3)}. \quad (43)$$

This is a nonlinear system of dimension $n = 5$ with $N = 3$ and $M = 2$. Using (32) and (33), the domain of the RBF network was determined to be

$$X = [-1.0576, 1.0727]^3 \times [-1, 1]^2. \quad (44)$$

The grid parameters used for the RBF network were $n = 5$ and $d = [4, 4, 4, 4, 4]^T$. A zeroth-order RBF network was used, and the weight vector w was computed using (19) and no training. A 1000-point white noise test input uniformly distributed over $[-1, 1]$ was used to evaluate the resulting RBF network performance. The system output and the RBF network output are shown in figure 8 where, for clarity, only the first 50 points are displayed. The fit is reasonably good with a normalized RMS error of $E_{\text{rms}} = 0.124$. This was achieved using $r = 1024$ weights.

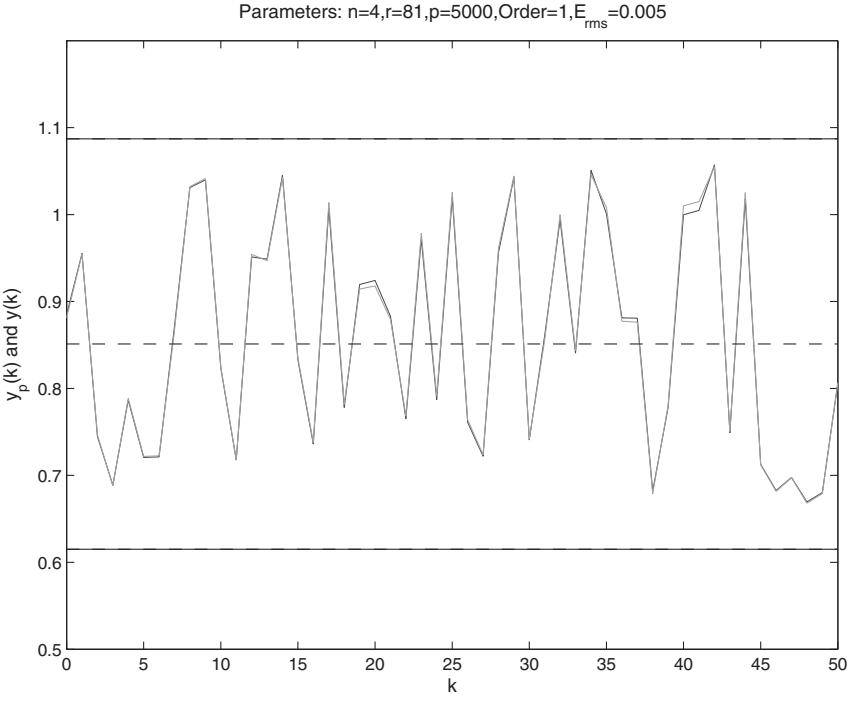

Figure 7. Outputs of a continuous tank stirred reactor and trained first-order RBF network with $n = 4$, $d = [3, 3, 3, 3]^T$, $E_{\text{rms}} = 0.005$.

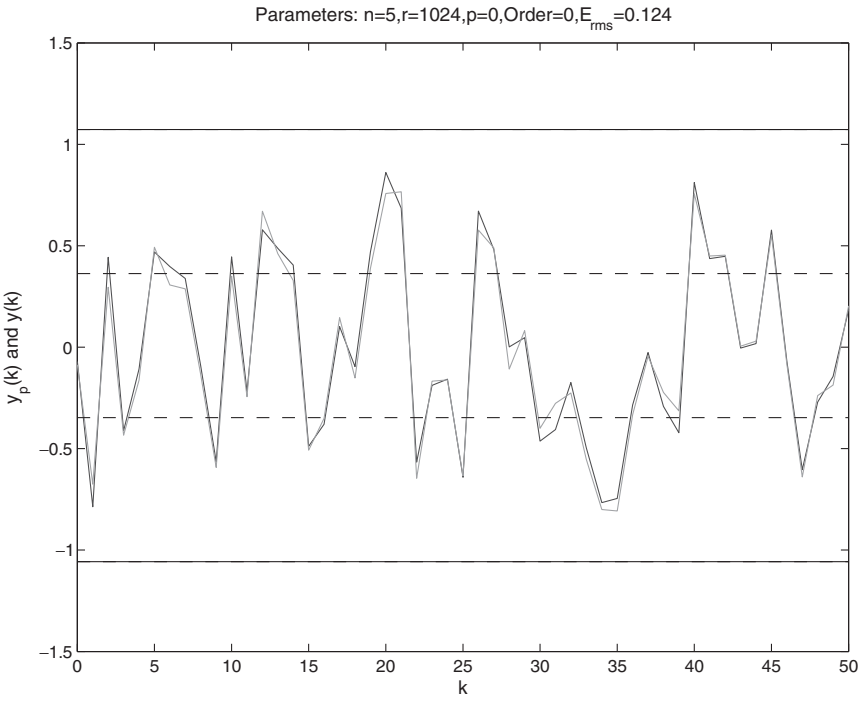

Figure 8. Outputs of a higher-order nonlinear system and zeroth-order RBF network with $n = 5$, $d = [4, 4, 4, 4, 4]^T$, $E_{\text{rms}} = 0.124$.

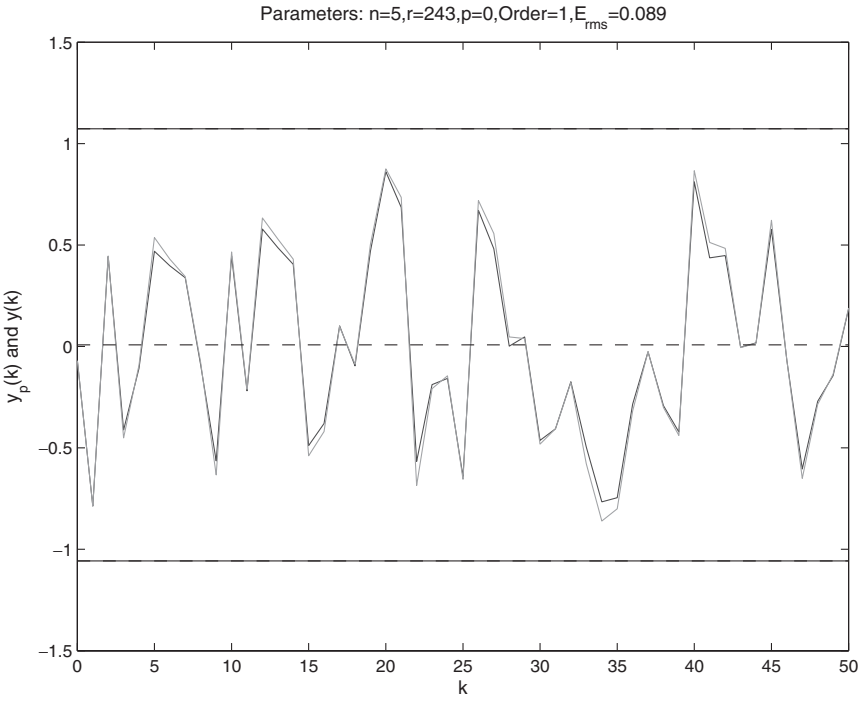

Figure 9. Outputs of a higher-order nonlinear system and first-order RBF network with $n = 5$, $d = [3, 3, 3, 3, 3]^T$, $E_{\text{rms}} = 0.089$.

For comparison, the same system was modelled using a first-order raised-cosine RBF network. In this case, the grid point precision was reduced to $d = [3, 3, 3, 3, 3]^T$, and a linear weight matrix V was included. The weights w and V were computed using (19) and (20), respectively, and no training was used. The network performance was evaluated using the same test input, and the results are displayed in figure 9, which shows the first 50 points of the system output and the RBF network output. In this case, the normalized

RMS error was reduced from 0.124 for the zeroth-order network to 0.089 for the first-order network. The number of weights was $r(n+1)=1458$, somewhat larger than for the zeroth-order case. Thus, in going from the zeroth-order RBF network to the first-order RBF network, a 27.8% improvement in accuracy was achieved at the expense of a 42.4% increase in parameter storage space.

5. Conclusions

An effective technique for identifying nonlinear discrete-time systems using raised-cosine RBF networks has been presented. The class of nonlinear discrete-time systems identified consisted of BIBO stable systems whose current output depends continuously on the past outputs and past inputs. Raised-cosine RBF networks are BIBO-stable, and the network output is a continuously differentiable function of the state. Evaluation speed of an n-dimensional raised-cosine RBF network is high because, at each discrete time, at most 2^n RBF terms are nonzero and contribute to the output. When the dependence on some of the state variables is known to be linear, the number of parameters of the raised-cosine RBF network can be reduced by employing a first-order network and setting the grid precision in selected dimensions to the minimum value of $d_i = 2$. For a linear discrete-time system, the first-order RBF network is exact on the domain over which it is defined, and it is minimal in terms of the number of distinct parameters that must be stored. Raised-cosine RBF networks were successfully used to identify a variety of discrete-time systems including both practical physical systems and benchmark systems previously cited in the literature.

References

CHEN, S., and BILLINGS, S. A., 1992, Neural networks for nonlinear dynamic system modelling and identification. *International Journal of Control*, **56,** 319–346.

CHEN, S., BILLINGS, S. A., and GRANT, P. M., 1990, Nonlinear system identification using neural networks. *International Journal of Control*, **51,** 1191–1214.

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,** 1051–1070.

CHEN, S.-C., and CHEN, W.-L., 2001, Adaptive radial basis function neural network control with variable variance parameters. *International Journal of Systems Science*, **32,** 413–424.

DUCHON, J., 1977, *Splines Minimizing Rotation-Invariant Semi-Norms in Sobelev Spaces* (New York: Springer).

FANG, Y., and CHOW, T. W. S., 2000, Synthesis of the sliding-mode neural network controller for unknown nonlinear discrete-time systems. *International Journal of Systems Science*, **31,** 401–408.

GORINEVSKI, D., 1996, On the persistency of excitation in radial basis function network identification of nonlinear systems. *IEEE Transactions on Neural Networks*, **6,** 1237–1244.

HARDY, R. L., 1971, Multiquadratic equations of tomography and other irregular surfaces. *Journal of Geophysical Research*, **76,** 1905–1915.

HERNANDEZ, E., and ARKUN, Y., 1996, Stability of nonlinear polynomial ARMA models and their inverse. *International Journal of Control*, **63,** 885–906.

JAFFE, D. A., and SMITH, J. O., 1983, Extensions of the Karplus–Strong plucked-string algorithm. *Computer Music Journal*, **7,** 56–69.

KADIRKAMANATHAN, V., and NIRANJAN, M., 1993, A function estimation approach to sequential learning with neural network. *Neural Computation*, **5,** 954–975.

LEE, C. C., and CHUNG, P. C., 1999, Robust radial basis function neural networks. *IEEE Transactions of Systems, Man and Cybernetics*, **29,** 674–685.

LIU, G. P., KADIRKAMANATHAN, V., and BILLINGS, S. A., 1999, Neural network-based variable structure control for nonlinear discrete systems. *International Journal of Systems Science*, **30,** 1153–1160.

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

NARENDRA, K. S., and PARTHASARATHY, K., 1990, Identification and control of dynamic systems using neural networks. *IEEE Transactions on Neural Networks*, **1,** 4–27.

PLATT, J. C., 1991, A resource allocating network for function interpolation. *Neural Computation*, **3,** 213–225.

POGGIO, T., and GIROSI, F., 1990, Networks for approximation and learning. *Proceedings of the IEEE*, **78,** 1481–1497.

POWELL, M. J. D., 1987, Radial basis functions approximations to polynomials. *Proceedings of the 12th Biennial Numerical Analysis Conference*, Dundee, pp. 223–241.

SCHAGEN, I. P., 1986, Sequential exploration of unknown multidimensional functions as an aid to optimization. *Numerical Analysis*, **4,** 337–347.

SCHILLING, R. J., CARROLL, J. J., and AL-AJLOUNI, A. F., 2001, Approximation of nonlinear systems with radial basis function neural networks. *IEEE Transactions on Neural Networks*, **12,** 1–15.

STEIGLITZ, K., 1996, *A Digital Signal Processing Primer with Applications to Digital Audio and Computer Music* (Menlo Park, CA: Addison-Wesley).

WIDROW, B., and STEARNS, S. D., 1985, *Adaptive Signal Processing* (Englewood Cliffs, NJ: Prentice-Hall).

YINGWEI, L., SUNDARARAJAN, N., and SARATCHANDRAN, P., 1997, Identification of time-varying nonlinear systems using minimal radial basis function neural networks. *IEE Proceedings Control Theory and Applications*, **144,** 202–208.