Approximate Models for Nonlinear Process Control

G. Sentoni
Planta Piloto de Ingeniería Química, Universidad Nacional del Sur, CONICET, 12 de Octubre 1842 (8000), Bahía Blanca, Argentina

O. Agamennoni and A. Desages
Dept. Ingeniería Eléctrica, Universidad Nacional del Sur, CIC, Bahía Blanca, Argentina

J. Romagnoli
Engineering and Chemistry Dept., The University of Sydney, CONICET, Sydney, Australia

A methodology is presented to obtain approximate models from input–output data, particularly oriented to implement a model-predictive control scheme. Causal, time-invariant nonlinear discrete systems with a certain type of continuity condition called fading memory are dealt with. To synthesize the nonlinear model a finite-dimensional linear dynamic part (discrete Laguerre polynomials) is used, followed by a nonlinear nonmemory map (single hidden-layer perceptron). Results of the application to approximate and control a binary distillation column are presented.

Introduction

A methodology is presented to obtain approximate models from input–output data, particularly oriented to implement a model predictive control (MPC) scheme. In this strategy, the system behavior is optimized in some sense over a time horizon. The feature of the predictive control schemes is the explicit use of a system-dynamic model to evaluate the effects of the future control actions over the system output. Choosing the horizon large enough ensures stability, that is, larger than the maximum time required to steer the system to the origin from any state. In many systems this horizon may be large. Therefore models have to predict over large horizons without degrading the prediction performance. Particular applications that use linear models are the algorithmic model control (Garcia et al., 1989) and dynamic matrix control (Hernandez and Arkun, 1990). However, nonlinear behavior is the rule rather than the exception in the dynamic of physical systems. Most physical devices have nonlinear characteristics outside a limited linear range. Chemical processes have values that saturate, connecting lines whose time delays vary with flow rate, reacting mixtures that obey power law, and separation units that are very sensitive to input changes and disturbances. These are the reasons a nonlinear model is used in the implementation of the MPC. One possible way to deal with this nonlinear model is following the classic modeling route. However, there are some drawbacks in this approach:

1. The assumptions made in deriving the model can be too restrictive or not very realistic. Therefore, the model might not capture the essentials of the behavior of the process.
2. Some limits may be completely unknown or very difficult to estimate.
3. Models may be very complex, leading to a large set of differential equations.

Another common approach is to build the model directly from the observed behavior of the process itself. This leads to a way of modeling called *system identification*. The bulk of the work in this field starts with representing the process as a black box. Great effort has been made to use conventional ways in the development of nonlinear system approximation methods (De Figueiredo and Dwyer, 1980; Leontaritis and Billings, 1985; Korenberg and Paarmann, 1991; Pottmann et al., 1993; Boutayeb and Darouach, 1995) and by novel approaches like neural networks (Moody and Darken, 1988; Lapedes and Farber, 1989; Narendra and Parthasarathy, 1991; Lin Lin et al., 1995). Multilayer neural networks have been widely applied (Lippman, 1987; Hush and Horne, 1993) in a great variety of fields, including system identification. Implementations in the field of model predictive control using neural networks models are in Donat et al. (1990), Hernández

Correspondence concerning this article should be addressed to G. Sentoni.

(1990), Temeng et al. (1992), and Ishida and Zhan (1995). A wavelet-based model is used in Elias-Juarez (1992), while the use of recurrent neural networks may be found in Williams and Zipser (1989), Narendra and Parthasarathy (1991), and Chen and Weigand (1994).

Among all the methods, there are only a few available results proving that a specific identification model may approximate a certain kind of nonlinear system. Korenberg and Paarmann (1991) have proposed a method for identifying finite sums of parallel cascades, each composing a linear dynamic and nonlinear static. This scheme is valid for any discrete, causal, finite-memory systems that continuously map small input changes into small output changes. Boyd and Chua (1985) have shown that every nonlinear operator with fading memory can be approximated by a linear time-invariant operator followed by a nonlinear polynomial readout map. For discrete systems the linear dynamic part is chosen to be a tapped delay time of the inputs. Also, Sandberg (1991) showed that discrete-time causal time-invariant systems having approximately finite memory can be uniformly approximated arbitrarily well by a tapped delay time of the inputs followed by a multilayered perceptron. This scheme is difficult to implement when the system memory is large due to the large amount of input–output mapping involved. For example, if the system memory were 40, the tapped delay line would take 40 past inputs. This implies that the nonlinear mapping would be $R^{40} \rightarrow R$. The results of Stone (1982) suggest that if the function is not known and the dimension is high (e.g., higher than 10), the only option is to assume a high degree of smoothness. If the function to be approximated is not sufficiently smooth, the amount of needed information would be totally impractical. Then we need another way of synthesizing the linear dynamic part of the model, which leads us to a manageable approximation problem of the nonlinear without memory map.

In this article we will present a model structure comprising a finite set of discrete Laguerre polynomials and a neural network. We will show that every discrete nonlinear system with fading memory may be approximated arbitrarily by this structure. The use of neural networks for approximating a nonlinear nonmemory map is supported by the results of Cybenko (1989) and Funahashi (1989), who proved that any continuous function can be uniformly approximated by a neural network model with only one hidden layer. In this way, a small dimension map is achieved in the nonlinear part by increasing the complexity of the linear part. The resulting model is nonlinear, easily identifiable from input–output data and can predict the future response of the system over arbitrarily large time horizons. This feature makes this approximation structure especially suited for nonlinear predictive control schemes. We will also discuss strategies to determine significant parameters: the pole of the Laguerre polynomial, the number of them, the number of the neurons in the hidden layer, and how to bound the time horizon.

The article is organized as follows. In the next section we state notations and definitions. In the third section a method to develop approximate nonlinear models based on discrete Laguerre systems and neural networks is discussed. In the fourth section a nonlinear model predictive control strategy using the approximate model is developed. In the fifth section an illustrative example, consisting of a binary distillation column, is presented. We conclude in the sixth section with some general remarks.

Notation and Definitions

Here we state the notations and definitions that will be needed for our presentation. For this purpose, let I_n denote the n-dimensional unit cube $[0,1]^n$ and $C(I_n)$ the space of continuous functions on I_n, with the supremum norm $\|f\|$. $\mathbf{Z}$ will denote the integers, $\mathbf{Z}_+$; ($\mathbf{Z}_-$) the nonnegative (nonpositive) integers. Past (future) time will be denoted by nonpositive (nonnegative) integers. Let l^∞ be the space of bounded sequences (i.e., functions $\mathbf{Z} \rightarrow \boldsymbol{R}$) with the norm:

$$\|u\|_\infty := \sup_k |u(k)|. \tag{1}$$

A function $F: l^\infty(\mathbf{Z}_-) \rightarrow R$ is called a *functional* on $l^\infty(\mathbf{Z}_-)$, and a function $N: l^\infty \rightarrow l^\infty$ is called an *operator*. We will usually drop the parenthesis around the arguments of functionals and operators, writing, for example, Fu for $F(u)$ and $N(u)$ for $N(u)(t)$. U_τ will denote the τ samples delay operator defined by

$$(U_\tau u)(k) = :u(k-\tau). \tag{2}$$

Time Invariant. An operator N is *time-invariant* (TI) if $U_\tau N = NU_\tau$ for all $\tau \in \mathbf{Z}$.

Casuality. N is causal if $u(\tau) = v(\tau)$ $\tau \leq k$ implies $Nu(k) = Nv(k)$.

Fading Memory. An operator $N: l^\infty \rightarrow l^\infty$ has fading memory on a subset K of l^∞ if there is a decreasing sequence $w: \mathbf{Z}_+ \rightarrow (0,1]$, $\lim_{k \rightarrow \infty} w(k) = 0$, such that for each $u, v \in K$ and given $\epsilon > 0$ there is a $\delta > 0$ such that

$$\sup_{k \leq 0} |u(k) - v(k)| w(-k) < \epsilon \Rightarrow |Nu(0) - Nv(0)| < \delta. \tag{3}$$

Intuitively, an operator has *fading memory* if two input signals that are close in the *recent past* (but not necessarily close in the *remote past*), yield present outputs that are close. For dynamical systems, fading memory is related to the notion of a unique steady state.

Method to Approximate Nonlinear Discrete Systems

In this section we will develop a methodology for approximating nonlinear discrete systems. It is well known that a great variety of systems can be approximated by a linear combination of Laguerre systems. The identification structure may be easily increased in this simple approach. Besides, the identification task is robust with respect to the choice of sampling interval, as well as to the choice of model order (Wahlberg, 1994, 1991). The main feature in those schemes is that the linear system can be expressed as a linear combination of orthonormal systems, generated with the same pole. The only necessary previous knowledge of the systems is reflected in choosing this pole, closely related to its main time constant. Furthermore, the identification process is not so sensible with respect to this parameter. The next theorem follows and extends this idea.

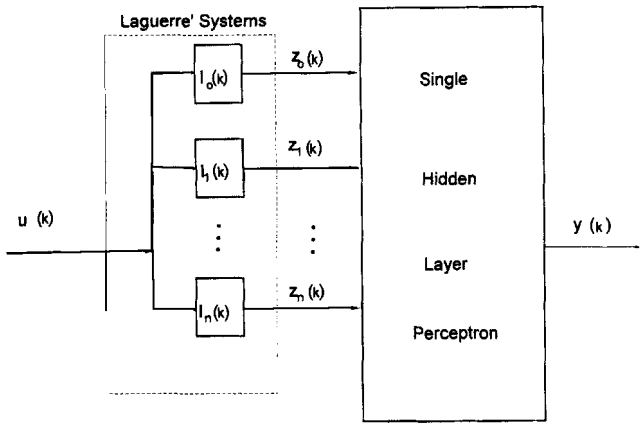

Figure 1. Proposed structure consisting of Laguerre system with neural network.

Theorem. Let K be any ball in l^∞, $K:=\{u \in l^\infty | \|u\| \le M_1, M_1 > 0\}$, and suppose N is any TI operator: $l^\infty \to l^\infty$ with fading memory on K. Then, there are a set of M Laguerre operators $\{L_0(\cdot), L_1(\cdot), \ldots, L_{M-1}(\cdot)\}$ and a neural network $NN: I^M \to R$ such, that for all $u \in K$,

$$\|Nu - \hat{N}u\| \le \epsilon, \tag{4}$$

where $\hat{N}$ is given by

$$\hat{N}u(k) = NN(L_0u(k), L_1u(k), \ldots, L_{M-1}u(k)). \tag{5}$$

Proof. See the Appendix.

This theorem ensures that a certain class of nonlinear systems can be approximated by a "nonlinear combination" of Laguerre polynomials. This idea can be traced back to Wiener (1956), who used a polynomial for this nonlinear combination. In Figure 1, we can see the approximation structure composed of a set of discrete time Laguerre systems (linear dynamic part) and a neural network (nonlinear nonmemory map). Why not use a polynomial? There are some reasons to prefer a neural network.

- In a polynomial the number of coefficients may blow up when the number of variables increases.
- It is very easy to overfit the data set. Some mechanism exists in neural networks to deal with overfitting.
- When the degree of the polynomial is high, it is not easy to deal with stability due to the bad conditioning of the data matrix.

These problems are overcome by many neural networks, for example, the multilayer perceptron. One of the main disadvantages is the character of the black box function estimator of the neural networks. The parameters of the multilayer perceptron structure are the number of hidden layers and the number of neurons in each hidden layer. We are faced with an architectural definition of the Laguerre–neural network structures. This implies

- Pole evaluation of the Laguerre polynomials.
- Number of Laguerre elements.
- Selection of the neural network structure.

We continue with a discussion on how to select the just mentioned parameters, but first, we give a brief introduction to the discrete Laguerre polynomials.

Discrete-time Laguerre systems

Application of continuous Laguerre networks in system identification apparently goes back to Wiener (1956). Discrete versions of the Laguerre expansions can be found in Wahlberg (1994, 1991) and the reference cited therein. The whole set can be expressed in the complex Z plane by

$$L_n(z) = \frac{K}{z-a}\left[\frac{1-az}{z-a}\right]^n, \quad n:0, \ldots, \infty, \tag{6}$$

where

$Ln(z)$ = z-transform of $l_n(k)$, the nth order discrete Laguerre polynomial
a = generating pole, such that $|a| < 1$
T = sampling time. When the system is discrete, $T = 1$
$K = \sqrt{(1-a^2)T}$, just a constant

Some of their relevant properties are:

(a) This set of polynomials is exponentially stable. We can express the polynomials in a transformed field as $L_n(z) = \sum_{k=0}^{\infty} l_n(k) z^{-k}$ for $|a| < 1$, which implies that the series is absolutely convergent for $|z| > |a|$. Particularly choosing $z = z_1$, such that $|a| < |z_1| < 1$, which implies that $\sum_{k=0}^{\infty} |l_n(k) z_1^{-k}| < \infty$. Then, since the series is absolutely convergent, there exists a finite constant A, such that $|l_n(k) z_1^{-k}| < A$. A suitable choice to bind $|l_n(k)|$ is to take $A = n+1$, and $r = |z_1|$, $|a| < r < 1$. Then,

$$|l_n(k)| < (n+1) r^k. \tag{7}$$

With this selection of r, $|l_n(k)|$ must approach zero at least exponentially as $n \to \infty$.

(b) This set of discrete Laguerre polynomials is orthonormal in the sense that

$$\sum_{k=0}^{\infty} l_p(k) l_q(k) = \begin{cases} 0 & p \ne q \\ 1 & p = q. \end{cases} \tag{8}$$

(c) Furthermore, the span of the functions $l_n(k)$ is dense in $l^2(\mathbf{Z}_+)$. The pole a of the expansion can be estimated according to

$$a = 1 - \frac{T}{t_d}, \tag{9}$$

where t_d expresses the dominating time constant. We will assume that a rough estimate of t_d is available or can be obtained (Zervos et al., 1988). The estimation of a defines the whole set of the polynomials.

Up to this point we have only determined the pole of the Laguerre systems. The remaining problem in our modeling approach is the evaluation of the number of Laguerre systems and the neural network architecture. These topics are closely related, and in the following section we will see how to estimate them.

Evaluation of the nonlinear no memory map

In the present work we will use the multilayer perceptron neural network. The multilayer perceptron is made up of one or more hidden layers between input and output layers. The layers are composed of computing units called *nodes*, which are interconnected together. The inputs propagate throughout the net in the conventional way (Rumelhart and McClelland, 1986).

From Figure 1 it can be appreciated that there exists a one-to-one correspondence between the Laguerre systems and the neural network inputs. The number of these systems is the same as the number of inputs of the neural network. As a consequence it is only necessary to evaluate the number of inputs and hidden nodes of the neural network. These parameters completely define the neural network structure.

Evaluation of the Number of Inputs. As we have shown, the number of Laguerre systems and of the network inputs are the same (see Figure 1). To evaluate this number we will use the method suggested in Xiangdong and Haruhiko (1993). We have the following input–output formulation:

$$y(k) = f(Z_k) = f([z_1(k), \ldots, z_M(k)]), \tag{10}$$

where Z_k is a row vector $[1 \times M]$ where each component $z_j(k) = L_{j-1}u(k)$ is the input signal, 1 filtered by the $(j-1)$th Laguerre system. We will drop the parenthesis and note y_k instead of $y(k)$. $f(\cdot)$ represents the neural network, a continuous and smooth multivariable function over some region, with partial derivatives assumed to be bounded:

$$|f_j| = |\partial f/\partial z_j| \le K, \qquad j{:}1, \ldots, M, \tag{11}$$

where M, K are positive values. Our objective is to reconstruct the nonlinear $f(\cdot)$ from the data pairs (Z_k, y_k), k:1, ..., N assuming there are enough available. Let $Z{:}[N \times M]$ be the matrix with each row composed by

$$Z_k = [z_1(k), \ldots, z_M(k)], \qquad k{:}1, \ldots, N. \tag{12}$$

The Lipschitz quotient q_{ij} can be defined as

$$q_{ij} = \|\delta y\|/\|Z_i - Z_j\|, \qquad (i \ne j), \tag{13}$$

where q_{ij} is a ratio of two quantities, these being $\|Z_i - Z_j\|$, the distance of two points in the input space, and $\|\delta y\|$, the difference between $f(Z_i)$ and $f(Z_j)$. As the function $f(\cdot)$ is continuous, the Lipschitz condition states that the quotient must be bounded:

$$0 \le q_{ij} \le L. \tag{14}$$

Sensitivity analysis can be applied for the input–output formulation:

$$\delta y = f_1 \delta z_1 + f_2 \delta z_2 + \ldots + f_M \delta z_M, \tag{15}$$

where $\delta y = f(Z_i) - f(Z_j)$ and f_i are the partial derivatives of $f(z)$ with respect to each component i of z. Then,

$$q_{ij} = \frac{|\delta y|}{\sqrt{(\delta z_1)^2 + (\delta z_2)^2 + \ldots + (\delta z_m)^2}} = \frac{|f_1 \delta z_1 + f_2 \delta z_2 + \ldots + f_M \delta z_M|}{\sqrt{(\delta z_1)^2 + (\delta z_2)^2 + \ldots + (\delta z_M)^2}}. \tag{16}$$

According to the previous definitions, taking $\delta = \max_i |\delta_i|$ and $|f_i| \le K$, the bound in the Lipschitz quotients q_{ij} is obtained by using Schwartz's inequality:

$$q_{ij}^M \le K \frac{|\delta z_1| + |\delta z_2| + \ldots + |\delta z_M|}{\sqrt{(\delta z_1)^2 + (\delta z_2)^2 + \ldots + (\delta z_M)^2}} \le K\sqrt{M}, \tag{17}$$

where the superscript M in q_{ij}^M represents the number of Laguerre systems to be used. Let M_0 denote the desired number of Laguerre systems. Hence if all z_i, i:1, ..., M_0 variables are included in the reconstruction of function $f(Z)$ and given that $f(Z)$ satisfies $|f_i| \le K$, the Lipschitz quotient for all data pairs is (Z_i, y_i). Two possible situations arise in this scene:

(1) $M < M_0$, that is, the number of Laguerre systems M is less than the desired M_0.

If one of the input variables is not included (e.g., z_M) in the reconstruction of the unknown function, the distance in the input space becomes $\sqrt{(\delta z_1)^2 + (\delta z_2)^2 + \ldots + (\delta z_{M-1})^2}$. However, the distance of the two corresponding points in the output space remains the same. Lipschitz quotient $q_{ij}^{M_0-1}$ is

$$q_{ij}^{M_0-1} = \frac{|\delta y|}{\sqrt{(\delta z_1)^2 + (\delta z_2)^2 + \ldots + (\delta z_{M_0-1})^2}} = \frac{|f_1 \delta z_1 + f_2 \delta z_2 + \ldots + f_{M_0} \delta z_{M_0}|}{\sqrt{(\delta z_1)^2 + (\delta z_2)^2 + \ldots + (\delta z_{M_0-1})^2}}. \tag{18}$$

It is easy to see that the Lipschitz quotient $q_{ij}^{M_0-1}$ (with missing z_{M_0}) will be much larger than $q_{ij}^{M_0}$ (in which all correct input variables are included).

(2) $M > M_0$, that is, the number of Laguerre systems M is greater than the desired M_0.

When a redundant variable is included, from the preceding analysis it is easy to see that $q_{ij}^{M_0+1}$ will be only slightly smaller than $q_{ij}^{M_0}$, but not significantly.

On the basis of the previous discussion, we should be able to identify the optimal number of inputs, using the information of the Lipschitz quotients. Doing so, we will define the following index:

$$q^M = \left(\prod_{k=1}^{p} \sqrt{M}\, q^M(k) \right)^{1/p}, \tag{19}$$

where $q^M(k)$ is the kth largest Lipschitz quotient with the variables $(z_1, z_2, \ldots, z_M)$ among all q_{ij}^M $(i \ne j; i,j = 1, 2, \ldots, N)$. This index is a geometric mean of the p largest values from the q_{ij}^M set, where p is a positive integer (usually selected as $p = aN$, $a \in [0.01, 0.02]$). From the previous analysis

it was shown that if M_0 is the number of optimal variables, q^{M_0+1} is very close to q^{M_0}; however, q^{M_0-1} is much larger than q^{M_0}. Moreover q^{M_0-2} is much larger than q^{M_0-1} and q^{M_0+2} is very close to q^{M_0+1}. Hence if one plots the curve of q^M vs. M, one can observe that beginning from some value M_0, q^M enters a saturated range. However, q^M with $M < M_0$ increases significantly. Taking γ as a cutoff constant, M_0 can be selected as the number of inputs that make

$$M_0 = \min_M [q^M/q^{M+1} < \gamma]. \tag{20}$$

We have just estimated the value of the a pole of the Laguerre systems and the number of these systems. Moreover we know that this number is the same as the number of neural network inputs. We also know that we need a single hidden-layer perceptron (see theorem). In the following section we will introduce a methodology to estimate the number of hidden-layer nodes. This is the final parameter that completely defines the Laguerre–neural network structure.

Evaluation of the Number of Neurons of the Hidden Layer. The hidden-layer neuron's number is evaluated by using a singular-value-based methodology (Sentoni et al., 1993; Nebot et al., 1993). Previous results (Sartori and Antsaklis, 1991; Huang and Huang, 1991) show that for n different training patterns, it is sufficient to use $n-1$ hidden-layer neurons. Coming from pattern classification studies, this estimation usually gives an overestimation in the number of hidden-layer neurons when it is applied to function estimation. Moreover, given this first approximation in the number of hidden-layer neurons, unaccounted factors exist that can influence that number. These factors cannot be evaluated before applying the training algorithm. We will discuss a procedure to obtain a realistic bound in the number of such neurons. This technique can be included in the *pruning techniques*. Thus it is possible to use the following approach.

Let us assume a neural network with only one hidden layer. Let us have M inputs and N training patterns. Let Z:$[N \times M]$ as in Eq. 12, the matrix with each row consisting of an input pattern. This network maps input–output values in the following way:

$$Y = g(ZW_1)W_2, \tag{21}$$

where W_1 and W_2 are the weight matrices for the hidden and the output layer, respectively, and Y is a column vector containing the predicted output patterns. The activation function is applied componentwise:

$$g(z) = (1 - e^{-2x})/(1 + e^{-2x}). \tag{22}$$

We begin the training process, and after some time we stop it and propagate the Z matrix through the hidden layer. We obtain:

$$X = g(ZW_1), \tag{23}$$

where X is $[N \times H]$, H being the number of neurons in the hidden layer and N the number of training patterns. Any matrix X:$[N \times H]$ can be expressed using a singular-value decomposition, as $X = U' \cdot \Sigma \cdot V$, with U:$[N \times H]$, V:$[H \times H]$ and $\Sigma = \text{diag}\{\sigma_i\}$, $\sigma_1 \geq \ldots \geq \sigma_H$. If a matrix X has rank $= k < H$, then $\sigma_{k+i} = 0$, $i = 1 \ \ldots \ H = k$. In practice σ_{k+i} will be significantly smaller than σ_k. This implies that $H - k$ linearly dependent columns exist in X. Then it is possible to determine the condition number of the matrix X, and analyzing the values of the σ_i, to adjust the number of neurons that the hidden layer should have. Thus the number of singular values of X different from zero help us to estimate the number of hidden-layer neurons. Values of σ_i close to zero imply excessive numbers of nodes in this layer. The methodology has been applied in our experiments with remarkable success.

We have finished introducing our approximation results. Now we will present the necessary formulas for another well-founded approximation method: the cascade–parallel model, which will be used for purposes of comparison.

Cascade–parallel model: Korenberg approach

The Korenberg cascade–parallel method (Korenberg and Paarmann, 1991) is suitable for any nonlinear, causal, finite-memory, discrete, time-invariant system. It consists of parallel connection sections composed of linear dynamic systems cascaded with nonlinear static systems, Figure 2. Mathematically the model may be represented as follows:

$$y[k] = \sum_{i=1}^{I} z_i[k] + e[k], \tag{24}$$

where $z_i[k]$ is the output of the ith linear/nonlinear cascade path. Suppose that we have calculated up to an $(i-1)$th cascade, then:

1. Calculate $y_{i-1}[k]$, $n = 0, \ldots, N$, the residue remaining $\left(y[k] - \sum_{j=1}^{i-1} z_j[k]\right)$ after estimating the $(i-1)$th cascade (where $y_0[k] = y[k]$).
2. **If** $y_{i-1}[k]$, the remaining residue is small enough **then Stop.**

else approximate the residue $y_{i-1}[k]$ by an ith cascade.

3. For $m = 0, \ldots, R$, let $h_i[m]$ randomly set equal to one of

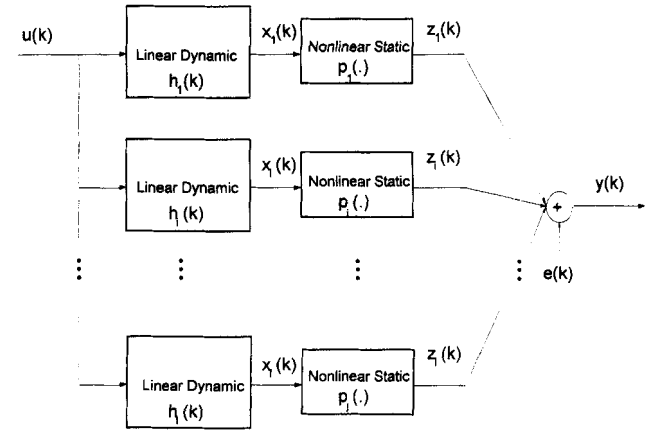

Figure 2. Parallel–cascade approximation structure.

$$h_i[m] = \varphi_{xy_{-1}}[m] = \sum_{k=R}^{k=N} y_{i-1}[k]u[k-m]$$

or

$$h_i[m] = \varphi_{xxy_{-1}}[m, A] \pm E\delta[m-A]$$

$$= \sum_{k=R}^{k=N} u[k-A]u[k-m]y_{i-1}[k] \pm E\delta[m-A].$$

- the sign of the delta term is chosen at random.
- the constant A is randomly selected from 0, ..., R.
- E is made to tend to zero, for example, $E = \overline{y_{i-1}^2[k]}/y^2[k]$.

4. Calculate $x_i[k] = \sum_{m=0}^{R} h_i[m]u[k-m]$.

5. Next a polynomial $p_i(\cdot)$ with input x_i is best fitted to y_{i-1} over the interval $n = 0, \ldots, R$. This determines the nonlinearity in the cascade, and then the output z_i can be calculated.

6. Set $i = i + 1$ and **goto** 1.

We have stated our approximation methodology and we have briefly introduced the Korenberg approach. We will finish this section with a brief summary of the complete Laguerre–neural network methodology.

Summary of the Laguerre–neural network modeling methodology

We will assume that there are enough data pairs $(u(k), y(k))$, $k = 1, \ldots, N$ available for the approximation process. Then:

1. Evaluate the a pole of Laguerre systems, using Eq. 9.
2. Evaluate the number M_0 of Laguerre systems by using the Lipschitz quotients.

Filter the available training data with the Laguerre systems evaluated in step 1 to complete the Z matrix as in Eq. 12. Then evaluate M_0 using Eq. 19. Now Z is $[N \times M_0]$ with $Z(k) = [z_1(k), \ldots, z_{M_0}(k)]$, k:1, ..., N. See Figure 3 for a typical plot.

3. Evaluate the number of hidden layer neurons using the singular value decomposition (SVD) technique.

Begin the training process with the set $(Z(k), y_k)$, k:1, ..., N evaluated in step 2. After some time, stop the training process and propagate the Z matrix through the hidden layer to obtain $X = g(ZW_1)$. X is $[N \times H]$, H being the number of neurons in the hidden layer. According to the criterion given in the section on evaluating the number of neurons in the hidden layer based in the singular values decomposition of X, select the suitable quantity of neurons from the hidden layer.

4. Continue training the single hidden-layer perceptron and validate the Laguerre–neural network structure.

The preceding procedural description gives the steps to obtain the approximate models. Step 4 will not be treated here because it is a common topic in the neural network literature and a subject for discussion in itself. In the next section we will show how to use the Laguerre–neural network approximate models in a nonlinear predictive control scheme.

Nonlinear Predictive Control Using Approximate Models

The dynamics of physical systems are usually dominated by nonlinear behavior. Most physical devices have nonlinear characteristics outside a limited linear range. In most chemical processes, understanding the nonlinear characteristics is important for controller design. For example, chemical processes have valves that saturate, connecting lines whose time delays vary with flow rate, reacting mixtures that obey power law, and separation units that are very sensitive to input changes and disturbances. Also there are phenomena such as incomplete mixing whose effects are not well understood. The combination of such effects may be unpredictable. This is the main motivation for using nonlinear models of the systems.

The feature of the predictive control schemes is the explicit use of a system dynamic model to evaluate the effects of the future control actions over the system output. Particular applications are algorithmic model control (Garcia et al., 1989) and dynamic matrix control (Hernández and Arkun, 1990). In both cases, the control action at time t comes from minimizing a performance index that may involve states, inputs, outputs, and possibly some restrictions over them (Garcia et al., 1989). Donat et al. (1990), Hernández and Arkun (1990), and Temeng et al. (1992) implemented neural networks models. In Elias-Juarez and Kantor (1990), a wavelets-based model is used. In these implementations, the next predicted output is evaluated on the basis of lagged values of previously predicted outputs. Among other authors (Williams and Zipser, 1989; Narendra and Parthasarathy, 1991), Chen and Weigand (1994) have used recurrent neural networks for the model. In this work a recurrent neural network is used to model the nonlinear function that relates states with its derivatives. It is assumed that all the states are measurable. If this were not the case, nonlinear observers could be used (Michalska and Mayne, 1995). However, these approaches have the disadvantage of great complexity and a time-consuming training algorithm.

Mayne and Michalska (1993) treat the problem of receding horizon control of nonlinear systems expressed in state variables. It is stated as a problem of control wherein the system at state x and time t is obtained by determining on-line the control $\hat{u}$, which solves an optimal control problem over the interval $[t, t+T]$, and setting the current control equal to $\hat{u}(t)$. Repeating this calculation continuously yields a feedback control (since the control action $\hat{u}(t)$ depends on the current state x). The finite-horizon-constrained optimal control problem is usually posed as that of minimizing a quadratic function over the interval $[t, t+T]$, subject to the constraint $x(t+T) = 0$. The parameter T is chosen to be larger than the maximum time required to steer the system to the origin from any state. By doing so, the constraint $x(t+T) = 0$ assures stability. Furthermore, as suggested in Mayne and Michalska (1993), the control can be discretized in time, thereby reducing the optimal control problem to a finite-dimensional nonlinear programming problem.

The problem formulation is stated as:

$$\min \sum_{j=1}^{T} \|y_{t+j} - yd_{t+j}\|^2 Q_j + \sum_{j=1}^{T_1} \|\Delta u_{t+j-1}\|^2 R_j,$$

$$\text{st } \Delta u_j = 0, \quad j = T_1, \ldots, T, \quad (25)$$

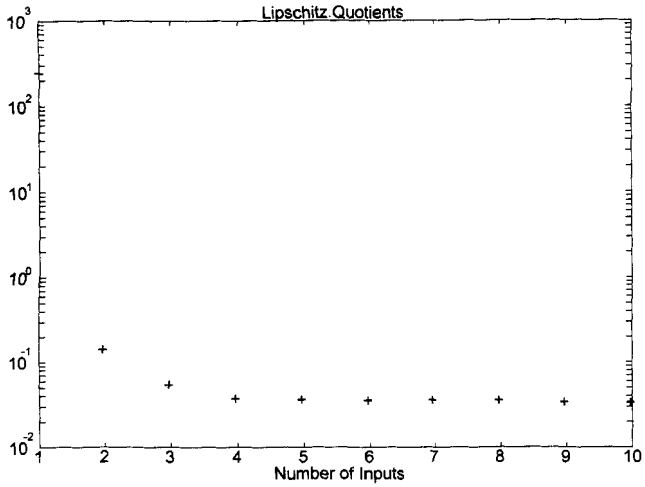

Figure 3. Lipschitz number as function of the number of inputs for distillation example.

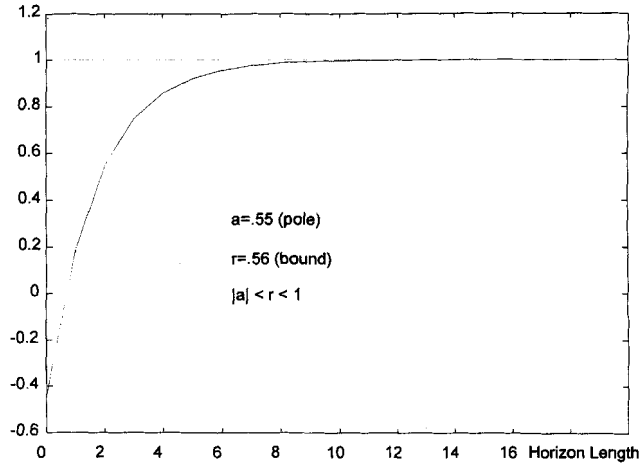

Figure 4. Bound in the time horizon T for distillation example.

where $\{u_t\}$, $\{y_t\}$, $\{yd_t\}$ are the input, output, and reference sequences. The control action Δu_t at time t is chosen as the first element Δu_t^o of the minimizing sequence $\{\Delta u_t^o, \ldots, \Delta u_{t+n_2-1}^o\}$. The sequences Q_j and R_j were fixed to unitary values. The remaining problem is how to select the control horizon T in a convenient way to achieve some degree of stability. Mayne and Michalska (1993) pointed out that T must be selected to be larger than the settling time of the system. In our model formulation, the dynamic part is linear and it is represented by the set of the Laguerre systems. Then the major dynamic behavior is due to the highest order Laguerre system $l_n(k)$. A suitable way to bound the time horizon T is to consider $l_n(k)$ approximated by its first N terms. This implies that the contribution of the terms greater than N is negligible. Let $G_n u$ and $G_n^N u$ be the response of $l_n(k)$ to the input signal u, considering the infinite impulse response and the N-terms impulse response of $l_n(k)$, respectively. By the assumption of fading memory and taking $G_n \upsilon = 0$ in Eq. A7 and Eq. A8, it can be shown that:

$$\frac{|G_n^N u|}{|G_n u|} \leq 1 - 2r^N + r^{N+1}, \text{ with } \quad |a| < r < 1. \qquad (26)$$

It is easy to see that the preceding relation tends to one when N increases. This shows a bound to the major dynamic behavior considering only the first N terms of $l_n(k)$. Then, taking the time horizon $T > N$, this is enough to ensure stability. Figure 4 shows the plot of this relation when the pole is $a = 0.55$ bounded by $r = 0.56$, for N varying from 0 to 20. For $N > 14$ the curve is very close to one, meaning that it is enough to take the control time horizon T greater than 14 to deal with stability. In the next section we will use all our previous results to approximate and control a binary distillation column.

Application to a Binary Distillation Column

In this section we will apply the strategy discussed earlier in the approximation and control of a binary distillation column. Furthermore, we will compare our approximation results with the Korenberg approach. For the control part, we will contrast our results with a model predictive strategy using a linear model.

Nonlinear input – output model: Approximation results

A model of the column was available in the MATLAB environment, assuming constant molar overflow and a given relative volatility profile. In this specific example the input and the output are the reflux flow rate and the distillate composition of the column, respectively. Figure 5 shows the plot of the top composition steady states for different reflux flow rates. This figure shows the typical behavior of a distillation unit with its marked saturated region. Three regions can be distinguished in the plot: below 0.95, between 0.95 and 0.99, and over 0.99. The system can be approximated by linear systems in the region below 0.95 (see l_1) and over 0.99 (see l_2), as can be seen from the inset in Figure 5. Moreover, it can be easily appreciated from the inset that the major nonlinear behavior is in the range [0.96, 0.99]. No matter how a line can be placed tangent in this range, it could not approximate the systems over it. It is precisely in this range where the overhead product lost in the bottom is minimized. This is the

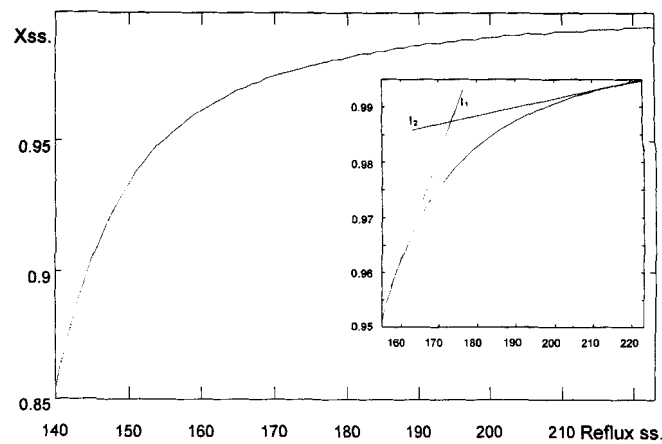

Figure 5. Top composition steady states for different reflux values.

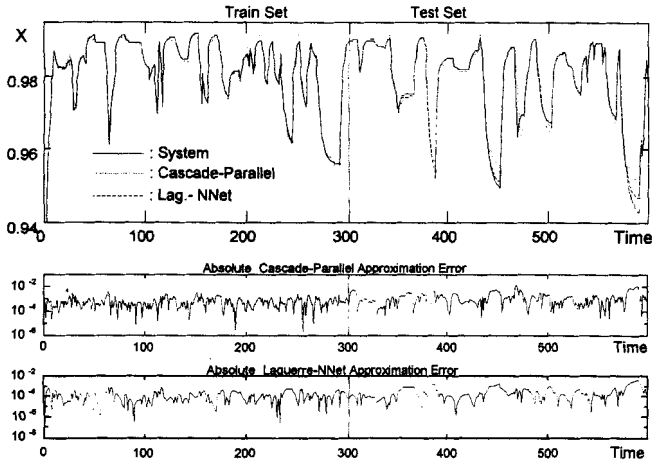

Figure 6. (a) System top composition, Korenberg approximation, and Laguerre–neural network approximation; (b) Korenberg approximation error in the train set ($\|TrainSetError\|_2 = 0.0161$) and the test set ($\|TestSetError\|_2 = 0.0489$); (c) Laguerre–neural network approximation error for the train set ($\|TrainSetError\|_2 = 0.0034$) and the test set ($\|TestSetError\|_2 = 0.0122$).

reason we want good control in this range, so we choose to approximate the top composition in between 0.89 and 1 (including the major nonlinear section). In this way we generate a pseudorandom binary sequence (PRBS) of different fixed-length input steps, with its amplitude varying according to a normal distribution. Then they were conveniently scaled to produce an output into the chosen range. The reboiler heat function was kept fixed to the equilibrium point. The response of the system to this set of inputs was obtained by integrating the set of differential equations, taking into account a zero-order hold in the input. In other words, the inputs were held fixed by a time of $dt = 0.01$, being the sample time at which the discrete model approximates the continuous system. Two different data sets were generated. One of them, called the *training set*, was used to get the approximated models by both methods. The other one, called the *testing set*, was used for testing the quality of the approximation. The following two sections will compare our approximation results with those obtained with Korenberg's approach.

Parallel–Cascade Model. We followed the procedure described in the preceding section with the following assumptions. The number of parallel cascades was limited to 20. This was because of the incremental decrement of the error with higher numbers. The length of the impulse response of each linear dynamic system was limited to 30 because no improvement was obtained beyond this limit. The maximum order allowed for the polynomials was up to 15 to prevent numeric instability. An optimal-order search was implemented inside of this limit. Figure 6 shows the real and approximated outputs, while Figure 6b shows the error plot with a $\|TrainSetError\|_2 = 0.0161$ and a $\|TestSetError\|_2 = 0.0489$.

Laguerre–Single Neural Network Scheme. The first step in our approximation procedure is to evaluate a, that is, the pole of the Laguerre systems. It results in $a = 0.55$, since this was the dominant time constant.

We followed the procedure of the section on the evaluation of the number of inputs to evaluate the number of Laguerre systems. Using the input of the training set filtered by $l_0(k), \ldots, l_9(k)$, we evaluated the Lipschitz quotients shown earlier in Figure 3. It is easy to see that there are no significant variations in the value of the Lipschitz quotients when the number of inputs is greater than four. Now we only need to define the neural network structure. We know that only one hidden layer is sufficient, and that the number of inputs is the same as the quantity of Laguerre systems. Using the procedure detailed in the section on the evaluation of the hidden layer's neurons; the hidden layer neurons were chosen to be equal to five.

The last step in the approximation method is to train the neural network. We got the signal to train it by taking the input signal filtered by the four Laguerre systems. This data set was scaled onto the range $[-1, 1]$ to avoid bad conditioning in the neural network training process. We have used a sequential quadratic programming technique with analytic gradient evaluation to speed up the training time and to improve the generalization. However, this is not an appropriate algorithm for on-line training. Backpropagation schemes (Narendra and Parthasarathy, 1991) or the recursive prediction error algorithm (Billings et al., 1991) should be used in this case.

Finally, we had our model composed of five Laguerre systems followed by a Perceptron with five hidden-layer neurons. Figure 6 shows that the real and approximated output are almost the same, while Figure 6c shows the error plot with a $\|TrainSetError\|_2 = 0.0034$ and a $\|TestSetError\|_2 = 0.0122$.

Nonlinear model predictive control

Now we will use the approximate model embedded in the model predictive scheme. In this case, assuming the line-feed-level perturbation to be a measurable one, we were able to synthesize a misomodel, even though we have not yet demonstrated it. In this way our model was enlarged due to the inclusion of this second input, that is, of the line-feed-level perturbation: Doing so it was necessary to include two more Laguerre systems with a generating pole $a = 0$ driven by the second input: The final model was made up of two Laguerre sets: set one, $l_0(k)$ to $l_4(k)$ generated by $a = 0.55$ (associated with the reflux); and set two, $l_0(k)$ to $l_1(k)$ generated by $a = 0$ (associated with the line-feed-level perturbation). The single-layer perceptron had seven inputs, connected to both Laguerre sets, and nine hidden-layer neurons. A state space linear model was also identified using the System Identification Toolbox of Matlab. The optimal size was chosen to be of order 3, with two inputs and one output. Both models were adjusted with the same data set, and the approximation results of both models can be appreciated from Figure 7. It can be seen that the nonlinear Laguerre–neural network model greatly outperforms the linear one in the whole range of the approximation.

One of the relevant parameters of the MPC scheme is the length of the control horizon. Taking into account that the major dynamic is due to $l_4(k)$ with $a = 0.55$, and applying the ideas expressed in this section, we evaluated $T = 14$ (see Figure 4) and left $T_1 = T$. We tested our model and the linear model in a setting where the system was moved to three dif-

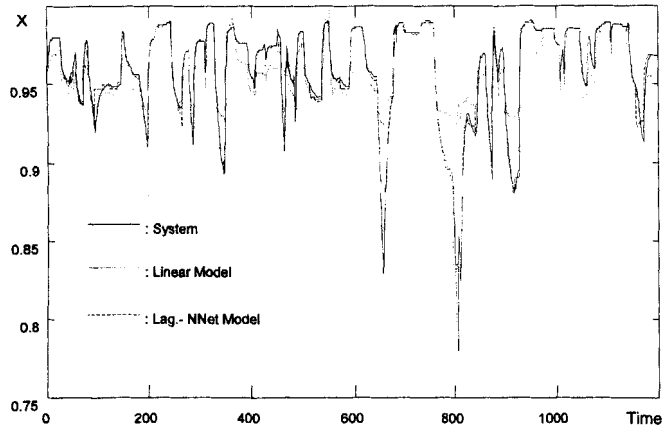

Figure 7. Top composition approximation for the Laguerre neural network and the linear MISO models.

Inputs: reflux and feed level perturbation. Output: top composition.

ferent setpoints. This was done in the region of the major nonlinear behavior of the system with its closed-loop performance specified by the following difference equation:

$$y_d(k) = 0.4u(k) + 0.6y_d(k-1). \tag{27}$$

The sequences Q_j (see Eq. 25) were fixed equal to 10 for both cases. In the linear model case, slowing down the control action by modifying the R_j sequence was the only way to stabilize the system. Figure 8 shows the reference, and the system's response by both the linear and the nonlinear models. The method to be followed was known beforehand, including three different setpoints which are 0.992, 0.962, and 0.977 for the top composition. The applied perturbation to the line-feed level can be seen in the bottom plot of the same figure, with the scale to its right. It is easy to see that the nonlinear model manages to control the system with better performance than the linear model does.

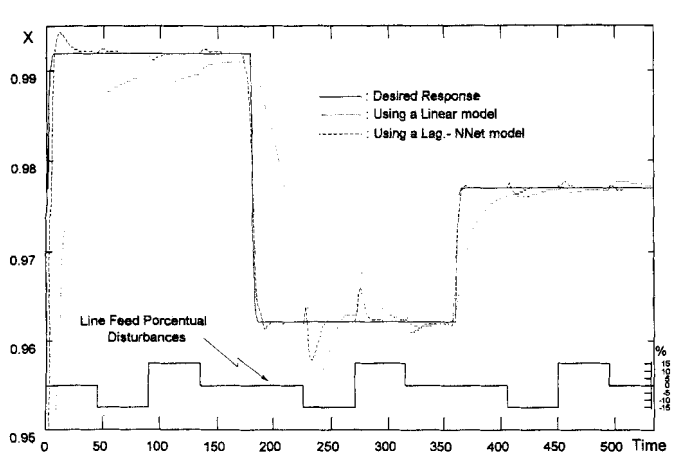

Figure 8. Model predictive control results.

Distillation column response for the MISO linear model and for the MISO Laguerre-neural network model.

Conclusions

In this article we have presented a methodology to obtain approximate nonlinear input-output models from available data and used these models in an MPC scheme. We have shown that any discrete nonlinear operator with fading memory can be approximated by a finite set of Laguerre systems with a single hidden-layer perceptron. The resulting model is nonlinear, easily identifiable from input-output data, and can predict the future response of the system over arbitrarily large time horizons. It is very common in control applications to deal with resonant systems. In those cases, the quantity of the required Laguerre systems for a given approximation would be high. The use of Kautz functions (Whalberg, 1995) should resolve the problem. For time-delayed systems simply including the discrete dead-time of the plant in the Laguerre bases would be sufficient. In this context, several important aspects of the model-building procedure within a model-based control scheme were discussed, such as the selection of the number of inputs, number of neurons, as well as the estimation of the size of the control time horizon.

Finally, applying the proposed methodology to approximate and control the binary distillation column was discussed, showing the feasibility of the approach to deal with typical operation units. The proposed scheme is applicable to a broad class of systems and is suitable for an adaptive implementation to cope with time-varying systems. Future lines of investigation include the formalizing for MISO and MIMO cases.

Literature Cited

Billings, S., H. B. Jammaludin, and S. Chen, "A Comparison of the Backpropagation and Recursive Prediction Error Algorithms for Training Neural Networks," *Mech. Syst. Signal Process.*, **5**, 233 (1991).

Boutayeb, M., and M. Darouach, "Recursive Identification Method for Miso Wiener-Hamerstein Model," *IEEE Trans. Automat. Contr.*, **AC-40**(2), 287 (1995).

Boyd, S., and L. Chua, "Fading Memory and the Problem of Approximating Nonlinear Operators with Volterra Series," *IEEE Trans. Circuits Syst.*, **CAS-32**(11), 1150 (1985).

Chen, Q., and W. A. Weigand, "Dynamic Optimization of Nonlinear Processes by Combining Neural Net Model with UDMC," *AIChE J.*, **40**(9), 1488 (1994).

Cybenko, G., "Approximation by Superposition of Sigmoidal Functions," *Math. Contr. Signal Syst.*, 303 (1989).

De Figueiredo, R., and T. Dwyer III, "A Best Approximation Framework and Implementation for Simulation of Large Scale Nonlinear Systems," *IEEE Trans. Circuits Syst.*, **CAS-27**(11), 1005 (1980).

Dieudonne, J., *Foundations of Modern Analysis*, Academic Press, New York (1969).

Donat, J. S., N. Bath, and T. McAvoy, "Optimizing Neural Net Based Predictive Control," Internal Document, Dept. of Chemical Engineering, Univ. of Maryland, College Park, MD (1990).

Elias-Juares, A., and J. C. Kantor, "On the Application of Wavelet to Model Predictive Control," *Proc. ACC 90*, p. 1582 (1990).

Funahashi, K., "On the Approximate Realization of Continuous Mapping by Neural Networks," *Math. Contr. Signal Syst.*, 183 (1989).

García, C. E., D. M. Prett, and M. Morari, "Model Predictive Control: Theory and Practice—A Survey," *Automatica*, **25**(3), 335 (1989).

Hernández, H. E., and Y. Arkun, "Neural Network Modeling and an Extended DMC Algorithm to Control Nonlinear Systems," *Proc. ACC 90*, 2454 (1990).

Huang, S. C., and Y. F. Huang, "Bounds on the Number of Hidden Neurons in Multilayer Perceptrons," *IEEE Trans. Neural Networks*, **NN-2**(1), 47 (1991).

Hush, D., and B. Horne, "Progress in Supervised Neural Networks," *IEEE Signal Processing Mag.*, **8** (Jan., 1993).
Ishida, M., and J. Zhan, "Neural Model Predictive Control of Distributed Parameter Cristal Growth Process," *AIChE J.*, **41**(10), 2333 (1995).
Korenberg, M. J., and L. D. Paarmann, "Orthogonal Approaches to Time-Series Analysis and System Identification," *IEEE Signal Processing Magazine*, 29, July (1991).
Lapedes, A., and R. Farber, "Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling," *Tech. Rep.*, Los Alamos National Laboratory, Los Alamos, NM (1989).
Leontaritis, I., and S. Billings, "Input-Output Parametrics Models for Nonlinear Systems, Part I: Deterministic Nonlinear Systems. Part II: Stochastic Nonlinear Systems," *Int. J. Contr.*, **41**, 303 (1985).
Lin Lin, J. J., D. S. H. Wong, and S. W. Yu, "Optimal Multiloop Feedback Design Using Simulated Annealing and Neural Network," *AIChE J.*, **41**(2), 430 (1995).
Lippman, R. P., "An Introduction to Computing with Neural Nets," *IEEE ASSP Mag.*, **4** (Apr., 1987).
Mayne, D. Q., and H. Michalska, "Robust Receding Horizon Control of Constrained Nonlinear Systems," *IEEE Trans. Automat. Contr.*, **AC-38**(11), 1623 (1993).
Michalska, H., and D. Q. Mayne, "Moving Horizon Observers and Observer-Based Control," *IEEE Trans. Automat. Contr.*, **AC-40**(6), 995 (1995).
Moody, J., and C. J. Darken, "Fast Learning in Networks of Locally Tuned Processing Units," *Neural Comput.*, **1**, 281 (1988).
Narendra, S., and K. Parthasarathy, "Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks," *IEEE Trans. Neural Networks*, **NN-2**(2), 252 (1991).
Nebot, E., G. Sentoni, and F. Masson, "Identification of a Flexible Manipulator Using Neural Networks," *Proc. IFAC Meeting*, Sydney, Australia, Vol. 8, p. 199 (1993).
Pottmann, M., H. Umbeahauen, and D. E. Seborg, "Application of a General Multi-model Approach for Identification of Highly Nonlinear Processes—A Case Study," *Int. J. Contr.*, **57**(1), 97 (1993).
Rumelhart, D. E., and J. L. McClelland, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, Vol. 1, *Foundations*, MIT Press, Cambridge, MA (1986).
Sandberg, I. W., "Approximation Theorems for Discrete Time Systems," *IEEE Trans. Circuits Syst.*, **AC-38**(5), 564 (1991).
Sartori, M. A., and P. J. Antsaklis, "A Simple Method to Derive Bounds on the Size and to Train Multilayer Neural Networks," *IEEE Trans. Neural Networks*, **NN-2**(4), 467 (1991).
Sentoni, G., O. Agamennoni, A. Desages, and J. Romagnoli, "Neural Network Modeling: Structural Analysis via Singular Value Decomposition," *Proc. APCChE/CHEMECA Meeting*, Melbourne, Australia, Vol. 3, p. 205 (1993).
Stone, J. A., "Optimal Global Rates of Convergence for Nonparametric Regression," *Ann. Stat.*, **10**, 1040 (1982).
Temeng, K. O., P. D. Schnelle, Su Hong-Te, and T. McAvoy, "Neural Model Predictive Control of an Industrial Packed Bed Reactor," *Proc. AIChE Meet.*, (1992).
Wahlberg, Bo, "System Identification Using Kautz Models," *IEEE Trans. Automat. Contr.*, **AC-39**(6), 1276 (1994).
Wahlberg, Bo, "System Identification Using Laguerre Models," *IEEE Trans. Automat. Contr.*, **AC-36**(5), 551 (1991).
Wiener, N., *The Theory of Prediction, Modern Math for Engrs*, Bechenbach, McGraw-Hill, New York (1956).
Williams, R. J., and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Comput.*, **102**, 270 (1989).
Xiangdong, He., and A. Haruhiko, "A New Method for Identifying Orders of Input-Outputs Models for Nonlinear Dynamic Systems," *Proc. ACC Meeting*, San Francisco, CA, p. 2520 (1993).
Zervos, C., P. R. Belanger, and G. A. Dumont, "On PID Controller Tuning using Orthogonal Series Identification," *Automatica*, **24**, 165 (1988).

Appendix

Relevant theorems

In the following we will state the two main theorems that will be used to prove our main approximation result: The Stone–Weirestrass theorem and the single-layer perceptron approximation theorem.

Stone–Weierestrass Theorem (Dieudonne, 1969; Boyd and Chua, 1985). Suppose E is a compact metric space and $\boldsymbol{G}$ a set of continuous functionals on E that separates points, that is, for any distinct $u, v \in E$ there is a $G \in \boldsymbol{G}$ such that $Gu \neq Gv$. Then for any continuous functional F on E and given $\epsilon > 0$, there is a polynomial $p: \boldsymbol{R}^M \to \boldsymbol{R}$ and $G_1, G_2, \ldots, G_M \in \boldsymbol{G}$ such that for all $u \in E$

$$|Fu - p(G_1u, G_2u, \ldots, G_Mu)| < \epsilon. \tag{A1}$$

Single-Layer Perceptron Approximation Theorem (Cybenko, 1989). Given any $f \in C(I_n)$, σ any continuous sigmoidal function, and $\epsilon > 0$, there is a finite sum of K terms $NN(x) = \sum_{j=1}^{K} \alpha_j \sigma(y_j^T x + \theta_j)$, with $y_j \in R^n$, θ_j, $\alpha_j \in R$ the parameters of the neural network, for which $|NN(x) - f(x)| < \epsilon$ for all $x \in I_n$.

Approximation theorem

We have to prove that any discrete nonlinear time-invariant operator having fading memory can be approximated by a finite set of discrete-time Laguerre polynomials followed by a single hidden-layer perceptron. First, we will make use of the Stone–Weierestrass theorem to prove that the continuous nonlinear nonmemory map is just a polynomial. Then, we will use the well-known single-layer approximation theorem that states that every continuous nonlinear nonmemory map can be approximated arbitrarily well in a compact set by a single hidden-layer perceptron. Doing so we have to prove the following Lemmas:

Lemma 1. The set of functional G_k associated with the discrete Laguerre operators are continuous with respect to weighted norm given by

$$\|u\|_w := \sup_{k \leq 0} |u(k)| w(-k). \tag{A2}$$

Proof. Consider the set of functionals $G := \{G_0, G_1, \ldots\}$, where $G_n u = \sum_{i=0}^{\infty} l_n(i) u(-i)$ is the functional associated with the operator $L_n(z)$. We will say that a weighting function $w'(n)$ dominates $w(n)$ if $w'(n) \geq w(n)$, $\forall n$. For purposes of the demonstration it is only necessary to rename the dominating weighting function $w'(n)$ as $w(n)$. Let $w(n) = 1/1 + n$ be the weighting function. Then

$$|G_n u - G_n v| = \left| \sum_{i=0}^{\infty} l_n(i) u(-i) - \sum_{i=0}^{\infty} l_n(i) v(-i) \right|, \tag{A3}$$

$$|G_n u - G_n v| \leq \sup_{i \geq 0} (|u(-i) - v(-i)| w(i)) \sum_{i=0}^{\infty} |l_n(i)| w(i)^{-1}, \tag{A4}$$

$$|G_n u - G_n v| \leq \|u(i) - v(i)\|_w \sum_{i=0}^{\infty} |l_n(i)| w(i)^{-1}. \tag{A5}$$

We have to bound the last term in Eq. A5. We know that Laguerre polynomials are exponentially stable, hence by Eq. 7

$$\sum_{i=0}^{\infty} |l_n(i)| w(i)^{-1} \le (n+1) \sum_{i=0}^{\infty} r^i (i+1), \quad \text{with} \quad |a| < r < 1. \tag{A6}$$

Let $S(r,N) = \sum_{i=0}^{N} r^i = (1-r^N)/(1-r)$. Taking the derivative of $(d/dr) \sum_{i=0}^{N} r^i = \sum_{i=0}^{N} (d/dr) r^i = r^{-1} \sum_{i=0}^{N} i r^i$ and since $(d/dr)S(r,N) = (1-r^{N-1})/(1-r)^2$, we have $\sum_{i=0}^{N} r^i(i+1) = (1 - 2r^N + r^{N+1})/(1-r)^2$. Using the last equation, we can bound:

$$|G_n^N u - G_n^N v| \le \|u(i) - v(i)\|_w (n+1) \frac{1 - 2r^N + r^{N+1}}{(1-r)^2}, \tag{A7}$$

where the functional $G_n u = \sum_{i=0}^{N} l_n(i) u(-i)$ uses only N-terms of the impulse response of $l_n(i)$. Taking the limit $N \to \infty$, that is, considering the infinite impulse response, gives

$$\lim_{N \to \infty} |G_n^N u - G_n^N v| = |G_n u - G_n v| \le \|u(i) - v(i)\|_w \frac{(n+1)}{(1-r)^2}. \tag{A8}$$

Then given any $\epsilon > 0$, taking $\|u(i) - v(i)\|_w \le \delta$, with $\delta = \epsilon(1-r)^2/(n+1)$ by Eq. A8 $|G_n u - G_n v| < \epsilon$, which proves Lemma 1.

Lemma 2. The G_k separate points in $l^\infty(\mathbf{Z}_-)$.

Proof. Suppose $u_1, u_2 \in l^\infty(\mathbf{Z}_-)$ such that $G_n u_1 = G_n u_2$ for all n. Let $u = u_1 - u_2$, so that $G_n u = 0$ for all n. We will show that $u = 0$, and this proves that the discrete Laguerre functional separates points in $l^\infty(\mathbf{Z}_-)$. The G_k can be expressed as:

$$G_n u = \sum_{i=0}^{\infty} (l_n(i) a^{-i/2})(u(-i) a^{i/2}). \tag{A9}$$

Note that $[u(-i)a^{i/2}] \in l^2(\mathbf{Z}_+)$ and $[l_n(i)a^{-i/2}] \in l^2(\mathbf{Z}_+)$. The span of the functions $[l_n(i)a^{-i/2}]$ is dense in $l^2(\mathbf{Z}_+)$, so we conclude that $[u(-i)a^{i/2}] = 0$, and hence $u = 0$. This proves that the discrete Laguerre functional separates points in $l^\infty(\mathbf{Z}_-)$.

Lemma 3. $K_- := \{u \in l^\infty(Z_-) | \|u\| \le M_1\}$ is compact with the weighted norm given by

$$\|u\|_w := \sup_{k \le 0} |u(k)| w(-k).$$

Proof. For a proof of this, see Boyd and Chua (1985).

Now given $\epsilon_1 > 0$, by Lemmas 1, 2, 3, and the Stone–Weierestrass theorem, we conclude that

$$|Fu - p(G_0 u, G_1 u, \ldots, G_{M-1} u)| < \epsilon_1. \tag{A10}$$

Finally, we have to prove that there exists a single hidden-layer perceptron $NN(\cdot)$ for which $|NN(x) - p(x)| < \epsilon_2$, $\epsilon_2 > 0$. Since $u \in K \Rightarrow \|u\| \le M_1$ it is easy to see that $x_i < \|L_{i-1} u\| \le c_i$, $c_i > 0$, i:1 $- M$. Taking $\xi_i \le (x_i/c_i)$, i:1 $- M$, $\xi \in I_n$ and $p(\xi) \in C(I_n)$. By the single-layer perceptron approximation theorem there exists a neural network $NN(\xi)$ for which $|NN(\xi) - p(\xi)| < \epsilon_2$ for all $\xi \in I_n$. Hence given $\epsilon > 0$, $\epsilon = \epsilon_1 + \epsilon_2$ there exist a single hidden-layer perceptron $N(\cdot)$ that approximates $p(\cdot)$ arbitrarily well, and

$$|Fu - NN(G_0 u, G_1 u, \ldots, G_{M-1} u)| < \epsilon. \tag{A11}$$

With each time-invariant operator N we associate a functional F on $l^\infty(\mathbf{Z}_-)$, defined by

$$Fu = N u_o(0), \tag{A12}$$

for $u \in l^\infty(\mathbf{Z}_-)$, where

$$u_o(t) = \begin{cases} u(t), & t \le 0 \\ u(0) & t > 0 \end{cases} \tag{A13}$$

is just a continuous extension of u to l^∞. The operator N can be recovered from its associated functional F through

$$Nu(t) = FPU_{-t} u, \tag{A14}$$

where $P: l^\infty \to l^\infty(\mathbf{Z}_-)$ truncates an element of $l^\infty(\mathbf{Z})$ into an element of $l^\infty(\mathbf{Z}_-)$:

$$Pu(t) = u(t), \qquad t \le 0. \tag{A15}$$

Then, N is continuous if and only if F is, so the preceding equations establish a one-to-one correspondence between time-invariant causal continuous operators N and continuous functionals F on $l^\infty(\mathbf{Z}_-)$.

Now let $y \in K$ and $t \in R$. Then $PU_{-t} u \in K_-$, hence

$$|FPU_{-t} u - p(G_0 PU_{-t} u, G_1 PU_{-t} u, \ldots, G_{M-1} PU_{-t} u)| = |Nu(t) - \hat{N}u(t)| < \epsilon.$$

Since the last equation is true for all $t \in R$, we conclude for all $u \in K$

$$\|Nu - \hat{N}u\| \le \epsilon. \tag{A16}$$

In other words, it is possible to approximate any nonlinear discrete time-invariant operator having fading memory on K, with a set of finite-dimensional discrete Laguerre-like systems followed by a single hidden-layer perceptron. This completes the proof.

Manuscript received July 21, 1995, and revision received Dec. 27, 1995.