State-Space Nonlinear Process Modeling: Identification and Universality

Guillermo B. Sentoni and Lorenz T. Biegler
Dept. of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

John B. Guiver and Hong Zhao
Aspen Technology, Inc., Pittsburgh, PA 15275

A practical approach was used to identify nonlinear models from available data. Knowing the true dynamics of a process allows implementation of a better and more efficient control scheme. Since this knowledge is not always available, a valid alternative to obtain a model is to identify the system. For that purpose, the DABNet (Decoupled A-B Net) structure was used that is composed of a decoupled linear dynamic system followed by a nonlinear static map. The linear dynamic system was initially spanned by a set of discrete Laguerre systems and then cascaded with a single hidden layer Perceptron. A model reduction technique (linear balancing) used on the hidden nodes of the neural network as part of the identification process makes it possible not only to identify the main time constants, but reduce the dimensionality of the Perceptron input. It was proven that the DABNet structure can approximate every nonlinear, causal, discrete time invariant, multiple-input single-output system with fading memory. The final model consists of a linear state-space system whose states, decoupled by input, are mapped by a neural network. Results concerning the application of the methodology to the approximation of a CSTR and a polymer process are presented.

Introduction

This article presents a practical approach for identifying nonlinear dynamic models from process data. There are two main approaches to obtaining a dynamic process model. The first one is the first-principle modeling approach, which is usually costly and highly dependent on the availability of process knowledge. The alternative is the empirical modeling approach, in which the model is identified from the data. While first-principle models are appropriate for many purposes (simulation, operator-training, and so on), the identification approach is the most feasible and practical for control purposes for most industrial processes. However, in the identification of a general nonlinear model it is essential to deal with the issues of ensuring *long-term prediction capabilities, a feasible and practical identification procedure, and model suitability for certain kinds of processes.* We address these problems in order to make nonlinear empirical models more reliable.

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

The first important issue to be considered is ensuring the long-term prediction capabilities of the model. This is evident in implementations of model predictive control (MPC) techniques, one of the most successful process control approaches (Morari and Lee, 1997). In MPC, a dynamic model is used to predict the behavior of the system over a future time horizon in response to a sequence of input changes. The controller is in fact a minimization problem with constraints (Oliveira and Biegler, 1996), which plans a control path for which the prediction path is optimal with respect to a set criterion such as set point mismatches and suppression moves. Stability is achieved by choosing the horizon large enough, that is, larger than the maximum time required to steer the system to the origin from any state (Michalska and Mayne, 1993). In many systems this horizon may be large, *so models should be able to predict over large horizons without degrading the prediction performance.*

A second important point is getting a feasible and practical identification procedure. Most industrial processes that apply

MPC employ linear step-response models or finite impulse response models (FIR). Although these approaches are widely and successfully used in industry, they require extensive plant testing, which often results in overparameterization of the models (Qin and Badgwell, 1997). For example, the dynamics of a first-order, single input/single output (SISO) process, which can be represented with only three parameters (gain, time constant, and dead-time) in a parametric form, typically requires 30 to 120 coefficients. Alternative MPC implementations using neural networks models are described in Hernández and Arkun (1990), Temeng et al. (1992), and Ishida and Zhan (1995). A wavelet-based model is used in Elias-Juarez and Kantor (1990), while the use of recurrent neural networks may be found in Williams and Zipser (1989), Narendra and Parthasarathy (1991), and Chen and Weigand (1994). Recurrent neural networks provide a reasonable solution to long-term prediction problems, but training such models is very difficult. *So, efficient and feasible training methods are needed.*

A third important issue is to prove the approximation capabilities of the models. Great efforts have been made in the development of nonlinear system approximation methods using conventional means (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). 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. *So, determining the theoretical capabilities of a feasible model and to which systems they apply is very important.*

In this article, we present a methodology to obtain models for a class of nonlinear systems in terms of a linear dynamic part followed by a nonlinear static map using input-output data. The linear dynamic system is initially spanned by a set of discrete Laguerre systems, and then cascaded with a single hidden layer Perceptron as seen in Figure 1a. In previous work (Sentoni et al., 1996), it was shown that a nonlinear combination of discrete Laguerre systems is able to approximate every single-input, nonlinear discrete system having fading memory. The *fading memory* concept (Boyd and Chua, 1985) is an important part of the characterization and is still the subject of current research (Shamma and Zhao, 1993; Matthews and Moschytz, 1994; Partington and Mäkilä, 1996, Sentoni, 1997). After the initial span of the linear layer by Laguerre systems, a model reduction technique (a linear balancing, Zhou et al., 1996) is performed on the hidden nodes of the neural network as part of the identification process.

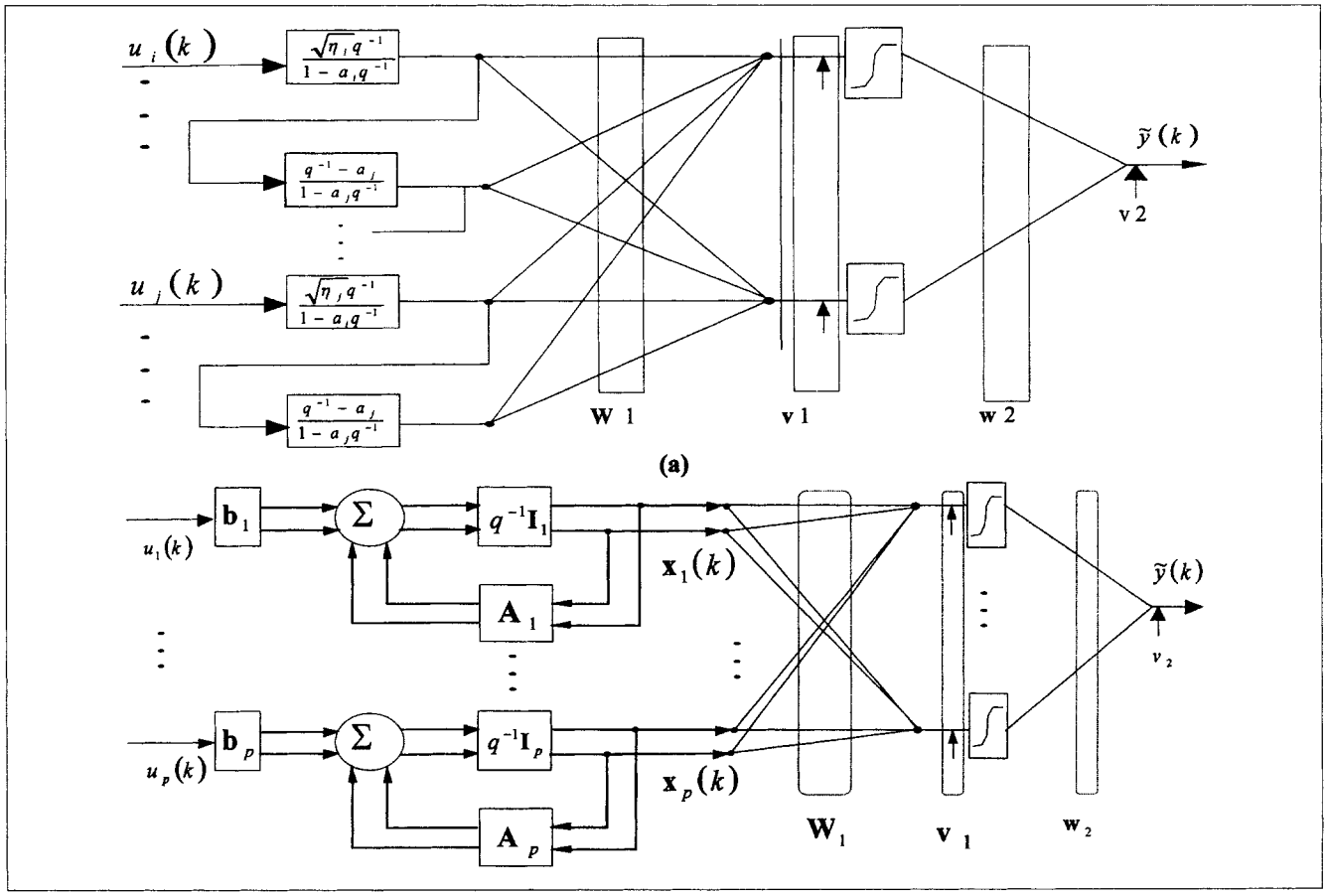

Figure 1. Proposed DABNet structure.

(a) Laguerre' systems initialization; (b) final reduced structure.

The balancing is performed in such a way that the linear state-space representation is decoupled by blocks. Doing so, it is possible not only to identify the main time constants, but also to reduce the dimensionality of the Perceptron input space. In addition, the noncontrollable/nonobservable modes are removed from the model. The final *DABNet* model (Decoupled A-B matrices Neural Network) consists of a sparse linear state-space system whose states, decoupled by blocks, are mapped by a neural network. This decoupled representation provides insight into the final model, especially since, in a practical identification scheme, the individual state-space matrices themselves can be represented as loosely coupled first- and second-order sections. The *DABNet* structure is well suited to be embedded into an MPC strategy, providing better long-term predictions than a NARX model in which the output errors are fed back into model input and amplified (Tsoi and Back, 1994). Besides, *DABNet* models show all of the following features: long-term prediction capabilities with minimal parameterization, practical identification that derive information for process control, and ability to identify a large class of real world processes.

In the Nonlinear Model Representation section the main results concerning the model structure are presented, while the universality proof is left for the Appendix. The bulk of the identification process and a summary of the methodology are presented. The proposed methodology is applied to the approximation of typical process units: a CSTR and a continuous polymerization process. Conclusions and the future work arising from this study are presented.

Nonlinear Model Representation

A process can be represented by an operator that is a function, which maps input signals into output signals. If we are interested in discrete systems, those signals become discrete sequences. $\boldsymbol{Z}$ denotes the integers and $\boldsymbol{R}^p$ denotes the space of real p-vectors, $p > 0$, with the norm $|\boldsymbol{x}| = \max_{1 \le j \le p} |x_j|$. In this context, a sequence is an element of $l_p^\infty(\boldsymbol{Z})$, the space of bounded p-valued vector sequences defined on $\boldsymbol{Z}$, with the norm $\|\boldsymbol{u}\|_\infty := \sup_{k \in z} |\boldsymbol{u}(k)|$. This space becomes $l_p^\infty(\boldsymbol{Z}_+)$ for sequences defined only for the non-negative integers $\boldsymbol{Z}_+$, or $l_p^\infty(\boldsymbol{Z}_-)$ for sequences defined for the non-positive integers $\boldsymbol{Z}_-$. To clarify the notation, matrices are denoted in bold uppercase while functions are denoted in italic lowercase if they are scalars or in bold if vector valued. Thus, $\boldsymbol{u}$ represents a vector or a vector sequence in accordance with the context, while $\boldsymbol{\mu}(k)$ represents a value of the sequence at the particular time k. Moreover, an *operator* $N: l_p^\infty(\boldsymbol{Z}) \to l_p^\infty(\boldsymbol{Z})$ is a function mapping sequences into sequences while a *functional* $F: l_p^\infty(\boldsymbol{Z}_-) \to \boldsymbol{R}$ is a function mapping sequences into a number. When we talk about identifying a process, we are implicitly talking about approximating its associated operator.

We are interested in approximating the class of nonlinear, causal, discrete time-invariant, multiple-input, single-output fading memory systems using the structure shown in Figure 1b. Intuitively, a system has fading memory (see Appendix for a formal definition) if two 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, the fading memory concept is related to the uniqueness of steady state. For this kind of system, we propose the structure shown in Figure 1b which is initially spanned using Laguerre systems as depicted in Figure 1a. It is well known that a great variety of linear systems can be approximated by a linear combination of Laguerre systems (Gu et al., 1989). The next theorem follows and extends this idea.

Theorem: DABNet Structure

Let $\boldsymbol{K} := \{\boldsymbol{u} \in l_p^\infty(\boldsymbol{Z}) | 0 < \|\boldsymbol{u}\| \le c_1 < \infty\}$ be any ball in $l_p^\infty(\boldsymbol{Z})$, and suppose N is any time-invariant operator $N: l_p^\infty(\boldsymbol{Z}) \to l^\infty(\boldsymbol{Z})$ with fading memory on $\boldsymbol{K}$. Then, given any $\epsilon > 0$ there are a set of Laguerre operators $\{L_1^1(\cdot), \ldots, L_{m_1}^1(\cdot), \ldots, L_1^p(\cdot), \ldots, L_{m_p}^p(\cdot)\}$ and a single hidden layer Perceptron $NN(\cdot)$ characterized by $(w2, \nu 2, \boldsymbol{W}1, \nu 1)$ such that for all $\boldsymbol{u} \in \boldsymbol{K}$

$$\|N\boldsymbol{u} - \hat{N}\boldsymbol{u}\| < \epsilon, \tag{1}$$

where the *DABNet* model $\hat{N}$ is given by $\hat{N}\boldsymbol{u}(k) = NN(L_1^1\boldsymbol{u}(k), \ldots, L_{m_1}^1\boldsymbol{u}(k), \ldots, L_1^p\boldsymbol{u}(k), \ldots, L_{m_p}^p\boldsymbol{u}(k))$.

Proof: For a proof of this theorem, see Appendix A.

This theorem ensures that a certain class of nonlinear systems can be approximated by a "nonlinear combination" of Laguerre systems to any degree of accuracy $\epsilon > 0$. This approximation holds over a bounded input signal domain $\boldsymbol{K}$ in good agreement with any physical system. The nonlinear combination is performed by a single hidden layer Perceptron $NN(\cdot)$ characterized by the parameters $(w2, \nu 2, \boldsymbol{W}1, \nu 1)$. The use of this neural network allows the utilization of order reduction techniques for pruning the dimension on the linear part, initially spanned by the Laguerre systems.

Laguerre systems

The advantage of the Laguerre models is that they provide a stable parameterization for a linear system by using orthonormal basis functions generated with the same pole. For a system with p inputs, a set of Laguerre systems can be defined in the complex z-plane as

$$L_i^j(\cdot) = \frac{\sqrt{\eta_j}\, z^{-d_j+1}}{z - a_j} \left[\frac{1 - a_j z}{z - a_j}\right]^i, \; i{:}0, \ldots, \infty, \; j{:}1{:}, \ldots, p, \tag{2}$$

where $L_i^j(z) = Z$-transform of $l_i^j(k)$, the i-th order discrete Laguerre system for the j-th input component, a_j is the generating pole associated to the j-th input, such that $|a_j| < 1$, η_j is $1 - a_j^2$, just a constant, and d_j is 1, unless there is a pure time delay for the j-th input for which $d_j > 1$.

It is well known that a linear combination of Laguerre systems (Eq. 2) can approximate a great variety of linear systems (Heuberger et al., 1993; Wahlberg, 1991; Gu et al., 1989). 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 a_j, closely related to its main time constant τ_j (Zervos et al., 1988)

$$a_j = 1 - T/\tau_j \tag{3}$$

The dominant time constant τ_j is relating the output with the

j-th input while T is the sampling time. In this approach, the identification structure may be easily increased by augmenting the number of Laguerre systems. Besides, the identification task is robust with respect to the choice of sampling time T, as well as to the choice of model order (Wahlberg, 1991). Furthermore, the identification process is not very sensitive with respect to this parameter (Wahlberg, 1994). Notice that the set (Eq. 2) has an infinite number of terms, and for a practical application this number should be finite. To do this, we take m_j Laguerre systems relating the output to the j-th input generated with the pole a_j for j:1: , ..., p. Next, we need to express this set in a state-space form for then applying a model order reduction technique. Here, we represent a linear dynamic system in a general way by the quadruple of matrices $[\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}]$ defining its dynamics. Therefore, the quadruple $[\boldsymbol{A}_j, \boldsymbol{b}_j, \boldsymbol{C}_j, \boldsymbol{0}]$ denotes the m_j systems associated with the j-th input expressed in state space form. Those matrices can be expressed as the composition of a low-pass section, $\sqrt{\eta_j}(z-a_j)^{-1}$, in series with (m_j-1) all-pass sections $(1-za_j)/(z-a_j)$. Possible realizations are $[a_j, \sqrt[4]{a_j^2\eta_j}, \sqrt[4]{a_j^2\eta_j}, 0]$ for the low-pass and $[a_j, \sqrt{\eta_j}, \sqrt{\eta_j}, -a_j]$ for the all-pass systems, respectively. Then, the state-space representation $[\boldsymbol{A}_j, \boldsymbol{b}_j, \boldsymbol{C}_j, \boldsymbol{0}]$ is obtained by performing simple systems operations (see, for example, p. 65 of Zhou et al., 1996). The outputs of the whole set of the Laguerre systems $\hat{\boldsymbol{y}}(k)$ are the outputs of the following system expressed in state-space from

$$\begin{aligned}\hat{\boldsymbol{x}}(k+1) &= \hat{\boldsymbol{A}}\hat{\boldsymbol{x}}(k) + \hat{\boldsymbol{B}}\boldsymbol{u}(k)\\ \hat{\boldsymbol{y}}(k) &= \hat{\boldsymbol{C}}\hat{\boldsymbol{x}}(k)\end{aligned}, \tag{4}$$

where $m = \sum_{j=1}^{p} m_j$, $\hat{\boldsymbol{A}}$:$m \times m$, $\hat{\boldsymbol{B}}$:$m \times p$ and $\hat{\boldsymbol{C}}$:$m \times m$. Notice the particular structure of the $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$ matrices

$$\hat{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \boldsymbol{A}_j & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{A}_p \end{bmatrix},$$

$$\hat{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \boldsymbol{b}_j & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{b}_p \end{bmatrix} \tag{5}$$

showing a balanced realization with independent states $\hat{\boldsymbol{x}}_j(k)$:$1 \times m_j$ (decoupled by blocks). In a linear system, these states are linearly combined to obtain a linear model. In the *DABNet* structure, these states are nonlinearly mapped (preserving this structure) by a single hidden layer Perceptron.

A single hidden layer Perceptron $NN(\cdot)$ characterized by $(\boldsymbol{w}2, \nu 2, \boldsymbol{W}1, \boldsymbol{\nu}1)$, perhaps the most used neural network, completes the *DABNet* structure by mapping the states of Eq. 4 into the output $\tilde{\boldsymbol{y}}(k) = NN[\hat{\boldsymbol{x}}(k)]$

$$\tilde{\boldsymbol{y}}(k) = \boldsymbol{w}2\Phi[\boldsymbol{W}1\hat{\boldsymbol{C}}\hat{\boldsymbol{x}}(k) + \boldsymbol{\nu}1] + \nu 2 \tag{6}$$

$\boldsymbol{W}1$:$[n1 \times m]$ is the input weight matrix with $n1$ neurons; $\boldsymbol{\nu}1$:$[n1 \times 1]$ is the input bias vector; $\boldsymbol{w}2$:$[1 \times n1]$ is the output weight vector, and $\nu 2$ is the output bias. The activation function $\Phi(s_1, s_2, \ldots, s_{n1}) = (\phi(s_1), \phi(s_2), \ldots, \phi(s_{n1}))$ is any permitted sigmoidal activation function $\phi(\cdot)$ applied component-wise. This completes the description of the proposed structure.

Identification and Model Reduction

In this section, we reduce the dimension of the linear dynamic part of the *DABNet* structure by keeping the useful decoupled matrix structure. In a first step, the model is adjusted using Laguerre systems because little knowledge is required from a dynamic point of view. This is very important for avoiding a complex nonlinear fitting algorithm like traditional ones for these kind of structures (Boutayeb and Darouach, 1995). However, this is at a small added cost: the dimension of the state vector $\hat{x}(k)$ is bigger than if were using all the (unknown) dynamic information. What can be done to reduce the dimension of this vector? Let us rewrite the set of equations for the *DABNet* model

$$\hat{\boldsymbol{x}}(k+1) = \hat{\boldsymbol{A}}\hat{\boldsymbol{x}}(k) + \hat{\boldsymbol{B}}\boldsymbol{u}(k) \tag{7a}$$

$$\tilde{\boldsymbol{y}}(k) = \boldsymbol{w}2\Phi(\hat{\boldsymbol{W}}1\hat{\boldsymbol{x}}(k) + \boldsymbol{\nu}1) + \nu 2 \tag{7b}$$

where $\hat{\boldsymbol{W}}1 = \boldsymbol{W}1\hat{\boldsymbol{C}} = [\hat{\boldsymbol{W}}1_1 \cdots \hat{\boldsymbol{W}}1_j \cdots \hat{\boldsymbol{W}}1_p]$, $\hat{\boldsymbol{W}}1_j$:$n1 \times m_j$ for $n1$ hidden neurons and m_j Laguerre systems. Notice that the system $[\hat{\boldsymbol{A}}_j, \hat{\boldsymbol{b}}_j, \hat{\boldsymbol{W}}1_j, \boldsymbol{0}]$ forms a linear system with one output for every hidden neuron, over which the sigmoidal activation function is applied. Because this system is linear, all the theory of linear model reduction can be applied. In that way, the final model is obtained after balancing and truncating the linear system $[\hat{\boldsymbol{A}}_j, \hat{\boldsymbol{b}}_j, \hat{\boldsymbol{W}}1_j, \boldsymbol{0}]$ to reduce the dimension of the system. The objective of this step is to reduce the dimension of the vector $\boldsymbol{x}_j(k)$, retaining the highest possible quantity of information in the reduced system. One way to do this is to generate a balanced realization of the linear system $[\hat{\boldsymbol{A}}_j, \hat{\boldsymbol{b}}_j, \hat{\boldsymbol{W}}1_j, \boldsymbol{0}]$ (Zhou et al., 1996) and then truncate it.

The observability Grammian $\boldsymbol{P}_j$ and the controllability Grammian $\boldsymbol{Q}_j$ (For j:1, ..., p) are matrices that solve the following discrete-time Lyapunov equations

$$\boldsymbol{P}_j - \left(\hat{\boldsymbol{A}}_j\right)^T \boldsymbol{P}_j \hat{\boldsymbol{A}}_j = \left(\hat{\boldsymbol{W}}1_j\right)^T \hat{\boldsymbol{W}}1_j \tag{8}$$

$$\boldsymbol{Q}_j - \hat{\boldsymbol{A}}_j \boldsymbol{Q}_j \left(\hat{\boldsymbol{A}}_j\right)^T = \hat{\boldsymbol{b}}_j \left(\hat{\boldsymbol{b}}_j\right)^T \tag{9}$$

uniquely for the linear system $[\hat{\boldsymbol{A}}_j, \boldsymbol{b}_j, \hat{\boldsymbol{W}}1_j, \boldsymbol{0}]$ (Zhou et al., 1996). Additionally, there exists a similarity transformation $\boldsymbol{T}_j$ (Zhou et al., 1996) in which the system is in balanced form, that is

$$\boldsymbol{P}_{j_{\text{bal}}} = \boldsymbol{Q}_{j_{\text{bal}}} = \Sigma_j = \text{diag}(\sigma_{j1}, \ldots, \sigma_{jm_j}); \tag{10}$$

where $\sigma_1 \geq \ldots \geq \sigma_m$ are the Hankel singular values. Now, a

balanced realization can be computed simply as

$$A_{j_{\text{bal}}} = T_j \hat{A}_j (T_j)^T, \quad \hat{b}_{j_{\text{bal}}} = T_j \hat{b}_j, \quad \hat{W}1_{j_{\text{bal}}} = \hat{W}1_j (T_j)^T. \quad (11)$$

Small Hankel values ($\sigma_{jr} \gg \sigma_{jr+1}$) indicate that the system can be reduced to an r-order model by direct truncation as

$$\bar{A}_j = [I_r 0] A_{j_{\text{bal}}}, \quad \bar{b}_j = [I_r 0] \bar{b}_{j_{\text{bal}}}, \quad \bar{W}1_j = W1_{j_{\text{bal}}} \begin{bmatrix} I_r \\ 0 \end{bmatrix} \quad (12)$$

The final reduced system

$$\bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{B}u(k)$$
$$\bar{y}(k) = \bar{w}2\Phi(\bar{W}1\bar{x}(k) + \mathbf{8}I\bar{v}1) + \bar{v}2. \quad (13)$$

where $\bar{W}1 = [\bar{W}1_1 \ldots \bar{W}1_j \ldots \bar{W}1_p]$. The matrices $\bar{A}$ and $\bar{B}$, having the same structure as $\hat{A}$ and $\hat{B}$ in Eq. 5, are built up using $\bar{b}_j$ $\bar{A}_j$ from Eq. 12.

Proposed strategy: summary

(1) *Expand the System into a Laguerre Basis*

- Select m_j Laguerre systems (Eq. 2) for each input, typically $5 \le m_j \le 10$ are enough.
- Select each pole of the Laguerre basis to be determined as a_j (Eq. 3).
- Calculate the state-space representation $[\hat{A}, \hat{B}, \hat{C}, \mathbf{0}]$ for the whole set of Laguerre systems (Eq. 4).

(2) *Train the Neural Network*

- Given N samples $(u(k), y(k))$ drawn from the system, evaluate the Laguerre states $\hat{x}(k)$ (Eq. 7a).
- With $\hat{x}(k)$ as input and the output $y(k)$, fit the model to get the parameters $(w2, b2, \hat{W}1, b1)$ (Eq. 7b).

(3) *Balance the Model over the Hidden Nodes*

- Find P_j and Q_j Grammians by solving the Lyapunov Eqs. 8 and 9.
- Find T_j that makes the system balanced $P_{j_{\text{bal}}} = Q_{j_{\text{bal}}}$, and the singular values $(\sigma_{j_1}, \ldots, \sigma_{j_{m_j}})$ (Eq. 10).
- Find $A_{j_{\text{bal}}}$, $\hat{b}_{j_{\text{bal}}}$, $\hat{W}1_{j_{\text{bal}}}$ that balance the system (Eq. 11).

(4) *Truncate the Model*

- If $\sigma_{j_r} \gg \sigma_{j_{r+1}}$ (typically $\sigma_{j_r} > 10\sigma_{j_{r+1}}$), reduce to an r-order model by direct truncation using Eq. 12.

(5) *With the new dynamics $(\bar{A}, \bar{B})$, retrain the neural network for the best fit to get the model*

$$\bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{B}u(k)$$
$$\bar{y}(k) = \bar{w}2\Phi(W1\ \bar{x}(k) + \bar{v}1) + \bar{v}2$$

Practical considerations about modeling and balancing

As a practical methodology, the Laguerre basis can be optionally replaced by a set of first-/second-order sections that cover the practical frequency ranges for each input. In addition, each reduced A-B section can itself be decomposed into standard first- and second-order sections. This can be done from a practical point of view and also, approximately, from a theoretical point of view. In practice it is typically unnecessary to consider multiple poles greater than order two, and, therefore, a Jordan Canonical Form comprising first- and second-order sections can be used for each input section. Moreover, a more reduced model can be obtained even if the balancing and reduction procedure is applied to $[\hat{A}, \hat{B}, \hat{W}1, \mathbf{0}]$, that is, considering the global dynamics. However, in this way the structure in the matrices $\bar{A}$ and $\bar{B}$ is lost: these matrices become full matrices.

Application Examples

In this section, we illustrate the feasibility of the identification methodology applied to typical process units. Some experimental conditions apply to all of the cases and will be stated here. The input sequence uses a probability of switching factor P_s (Hernández and Arkun, 1993; see also Mane and Doyle, 1997; Barker, 1993) that corresponds to the probability that the input changes at the end of any sampling interval. If the input is changed, its new values are drawn from a uniform distribution with a probability P_s. This probability P_s is adjusted to provide a signal that excites the system evenly over the frequency range of interest. The output signals are integrated using a Gear method taking into account a zero-order hold in the input. In all the experiments the input/output signals have been scaled between $[-1, 1]$ previous to the identification and rescaled after the identification process has taken place. Every particular condition is stated within the proper example. In the first example, we identify a continuous-stirred tank reactor.

Continuous-stirred tank reactor

The following set of equations is a first-principle model of a two-tank pH neutralization process depicted in Figure 2. This is a highly nonlinear process that has been widely used as a benchmark problem for nonlinear process modeling (Zhao et al., 1997). The equations defining the mass balance, the acetate balance, and the sodium balance in tank 1, as well as the mass balance, the acetate balance, and the sodium

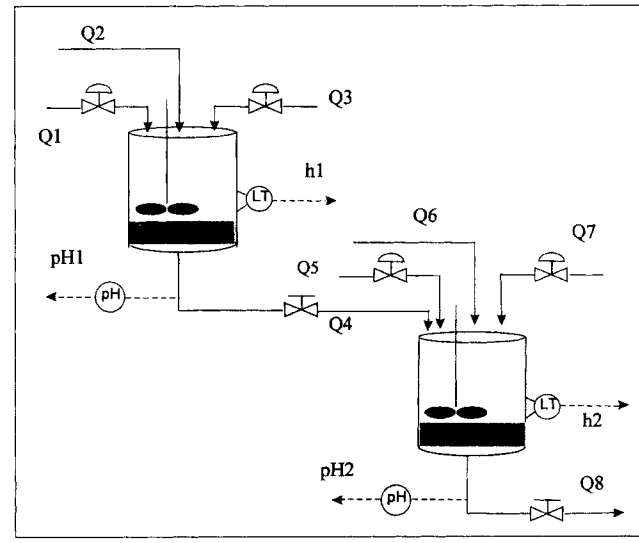

Figure 2. pH neutralization system.

Table 1. pH Process Nominal Conditions

Tank volume	V	1,000 L
Tank level	h_1, h_2	10 cm
Valve coefficient 1	$cV1$	188.5 $cm^{-1/2}$/min
Valve coefficient 2	$cV2$	351.3 $cm^{-1/2}$/min
Acid flow rate 1	Q_1	41 L/min
Acid flow rate 2	Q_2	40 L/min
Base flow rate 3	Q_3	515 L/min
Acid flow rate 5	Q_5	50 L/min
Acid flow rate 6	Q_6	31 L/min
Base flow rate 7	Q_7	515 L/min
HAC concentration	C_A	0.32 mol/L
N_aOH concentration	C_B	0.05 mol/L
HAC dissoc. constant	K_A	1.8×10^{-5}
N_aOH dissoc. constant	K_B	1.0
H_2O dissoc. constant	K_w	1.0×10^{-14}
Effluent pH value	pH_i	i: 1,2 = 7.0
Sampling interval	t_s	0.1 min

balance in tank 2 are

$$A_1 dh_1/dt = Q_1 + Q_2 + Q_3 - c_{V1}\sqrt{h_1} \tag{14a}$$

$$Vd\alpha_1/dt = Q_1 C_A + Q_2 C_A - c_{V1}\alpha_1\sqrt{h_1} \tag{14b}$$

$$Vd\beta_1/dt = Q_3 C_B - c_{V1}\beta_1\sqrt{h_1} \tag{14c}$$

$$A_2 dh_2/dt = Q_5 + Q_6 + Q_7 + c_{V1}\sqrt{h_1} - c_{V2}\sqrt{h_2} \tag{14d}$$

$$Vd\alpha_2/dt = Q_5 C_A + Q_6 C_A + c_{V1}\alpha_1\sqrt{h_1} - c_{V2}\alpha_2\sqrt{h_2} \tag{14e}$$

$$Vd\beta_2/dt = Q_7 C_B + c_{V1}\beta_1\sqrt{h_1} - c_{V2}\beta_2\sqrt{h_2} \tag{14f}$$

where the state variables are

- $\alpha_{1,2}$: acetate concentration coefficients for tanks 1,2;
- $\beta_{1,2}$: sodium concentration coefficients for tanks 1,2; and
- $h_{1,2}$: height in tanks 1,2

Q_i are the volumetric flow rates shown in Figure 2, C_A and C_B are the acetate and sodium inlet concentrations, respectively, and c_{v1} and c_{v2} are the valve constants for tanks 1 and 2, respectively. The whole system is then completely defined with the addition of the acetic acid and water equilibrium equations and the electroneutrality equations $[AC^-]_i[H^+]_i = K_A[HAC]_i$, $[H^-]_i[OH^-]_i = K_W$, $\beta_i + [H^+]_i = [OH^+]_i + [AC^-]_i$ for both tanks ($i = 1, 2$) and the $\alpha_i = [HAC] + [AC^-]$ and $\beta_i = N_a^+$ values. The values for the constants and parameters for this model can be found in Table 1. Finally, the definition for the pH is

$$\text{pH} = \log_{10}[H^+] \tag{15}$$

Experimental Conditions. The aim of this example is to identify a nonlinear dynamic model of the pH value in the 2nd tank as a function of acid and base flow rates in the 1st tank, while maintaining constant the values of the variables in the 2nd tank. This is a very nonlinear problem as can be appreciated from Figure 3 that illustrates the steady state of the pH level vs. the input stream Q_1. As usual, two data sets were generated to identify the system: a training and test set using a probability $P_s = 0.04$. The outputs were generated by integrating those inputs using the set of Eqs. 20 with a sampling time of $T = 0.1$ s. The former set was used to fit the model, while the latter one was used to validate the model. The DABNet model is built to reproduce pH_2, the pH value in the 2nd tank, using Q_1, Q_2, and Q_3 as input variables and keeping constant the variables Q_5, Q_6 and Q_7. As a way to show the relative robustness of this approach against choosing bad values for the Laguerre poles, we set them at $a_1 = a_2 = a_3 = 0.5$ (just in the middle of the valid range), while the dead times were selected to be: $d_1 = 2$, $d_2 = 2$, $d_3 = 1$. Six Laguerre basis terms were used for each input for the initial expansion of the linear dynamic part. For the neural network, we arbitrarily chose 10 hidden nodes without attempting to optimize these parameters. After the application of the methodology, the system was reduced to a second-order system per input. Figure 4 shows the approximation results for

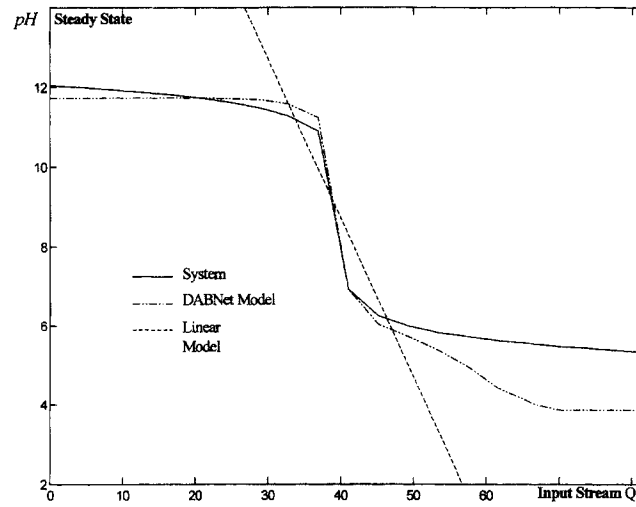

Figure 3. Steady-state behavior of the pH neutralization system, the DABNet model, and the linear model.

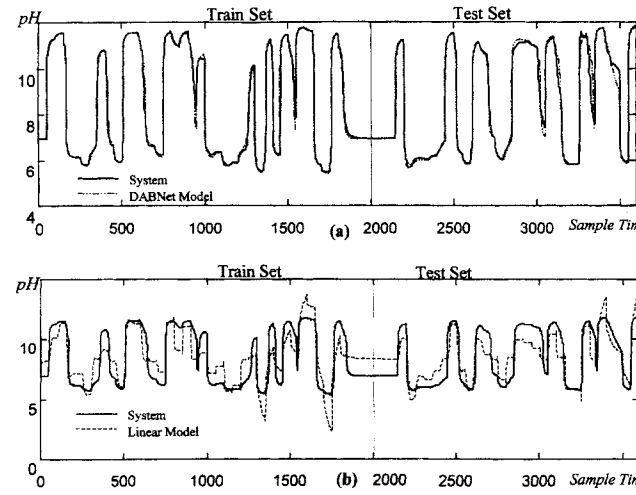

Figure 4. Approximation of the pH discriminated by train and test set for the CSTR example.

(a) Approximation of the pH by the DABNet model; (b) approximation of the pH by the linear model.

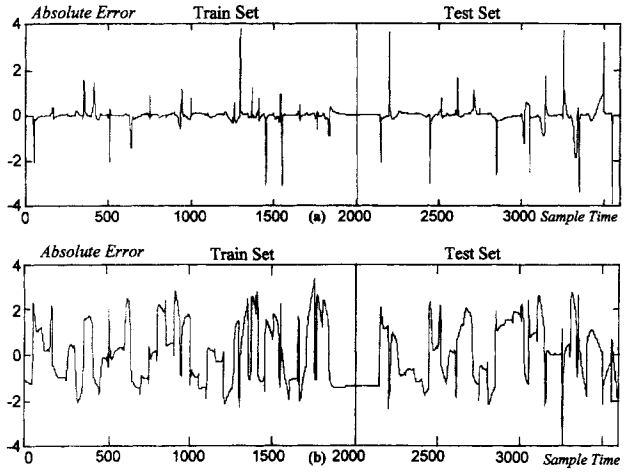

Figure 5. Absolute error of the pH approximation discriminated by train and test set in the CSTR example.

(a) DABNet model. The norm of the error is: 12.383 for the train set and 18.580 for the test set; (b) linear model. The norm of the error is: 60.223 for the train set and 50.186 for the test set.

the DABNet model and for a linear model. The linear model was adjusted using a partial least-squares (PLS) strategy, being composed by a six state linear state-space model. Figure 4a shows the DABNet approximation and the real output of the system for the training and test sets, while Figure 5a shows the training and test errors for the final DABNet model. For comparison purposes, Figure 4b shows the linear model approximation and the real output of the system, while Figure 5b shows the training and test errors for the linear model. Note that the nonlinear model performs much better than the linear one.

It is interesting to address that the steady-state curve of the DABNet fits fairly closely the steady-state curve of the system even though it is very nonlinear. The small discrepancy in the low pH range (pH < 6) is due to the lack of information used to train the model in this range. As can be appreciated from Figure 4, the data used to fit the model was for a pH range greater than 6. Here the DABNet model is extrapolating in this zone.

Continuous polymerization reactor process

In this example, we will apply the proposed methodology to identify the conversion factor of a continuous polymerization reactor shown in Figure 6. The process consists of a polymerization solution of methyl-methacrylate (MMA), initiated by azo-bis-isobutotyronitrile (AIBN), in which toluene is used as the solvent (Soroush and Kravaris, 1992). This model is based on the standard free-radical reaction kinetic mechanism, and, under standard assumptions, the reaction rates for the monomer and initiator are given by

$$R_m = -C_m \xi_0 (k_p + k_{f_m}) \tag{16}$$

where $\xi_0 = (2 f k_i C_i / k_t)^{0.5}$. The overall propagation rate constants for the gel and glass effects are

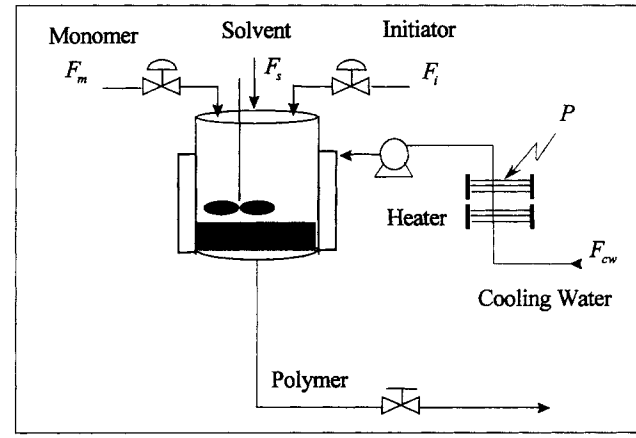

Figure 6. Polymerization system.

$$k_t = \begin{cases} k_{t_0} \alpha_{t_{11}} \exp(\alpha_{t_{12}} V_f - \alpha_{t_{13}} (T - 273.2)), \\ \qquad \text{if } V_f > \alpha_{v_1} - \alpha_{v_2} (T - 273.2) \\ k_{t_0} \alpha_{t_{21}} \exp(\alpha_{t_{22}} V_f), \\ \qquad \text{if } V_f \le \alpha_{v_1} - \alpha_{v_2} (T - 273.2) \end{cases} \tag{17}$$

$$k_p = \begin{cases} k_{p_0}, & \text{if } V_f > 0.05 \\ k_{p_0} \alpha_{p_{21}} \exp(\alpha_{p_{22}} V_f), & \text{if } V_f \le 0.05 \end{cases} \tag{18}$$

where the α parameters are constants in Table 3 and M_i is the molecular weight of component i. The total free volume $V_f = V_{f_p} \phi_p + V_{f_m} \phi_m + V_{f_s} \phi_s$ is defined by the contributions of specific free volume of the monomer $V_m = 0.025 + \alpha_m (T - T_{gm})$, the polymer $V_{f_p} = 0.025 + \alpha_p (T - T_{gp})$, and the solvent $V_{f_s} = 0.025 + \alpha_s (T - T_{gs})$. The volume fractions are $\phi_m = C_m M_m / \rho_m / (\mu_1 / \rho_p + C_m M_m / \rho_m + C_s M_s / \rho_s)$ for the monomer, $\phi_p = \mu_1 / \rho_p / (\mu_1 / \rho_p + C_m M_m / \rho_m + C_s M_s / \rho_s)$ for the polymer, and $\phi_s = C_s M_s / \rho_s / (\mu_1 / \rho_p + C_m M_m / \rho_m + C_s M_s / \rho_s)$ for the solvent. The values of the parameters for the MMA polymerization system are given in Table 2. Under the assumption of $F_i \ll F_m$, and the fact that the volume remains constant during operation (because of the experimental design), the volumetric flow rate of the outlet stream (F) is related to the volumetric flow rate of the monomer stream (F_m) by $F = F_m (1 + \bar{\epsilon} x_p)$, where x_p is the conversion factor given in terms of the mass concentration of polymer chains (μ_1)

Table 2. Parameters of System

$T_{cw} = 2.9520 \times 10^2$	K
$T_\infty = 2.9520 \times 10^2$	K
$T_{ms} = 2.9520 \times 10^2$	K
$c_w = 4.2000 \times 10^0$	$kJ \cdot kg^{-1} \cdot K^{-1}$
$\rho_w = 1.0000 \times 10^3$	$kg \cdot m^3$
$P_{\max} = 3.0000 \times 10^0$	$kJ \cdot s^{-1}$
$F_{cw\max} = 2.5500 \times 10^{-4}$	$m^3 \cdot s^{-1}$
$F_{i\max} = 2.3000 \times 10^{-8}$	$m^3 \cdot s^{-1}$
$F_m = 2.8000 \times 10^{-7}$	$m^3 \cdot s^{-1}$
$V = 1.2000 \times 10^{-3}$	m^3
$\alpha_1 = 3.8000 \times 10^{-3}$	s^{-1}
$\alpha_2 = 8.0000 \times 10^{-4}$	s^{-1}
$\alpha_3 = 3.7000 \times 10^{-4}$	s^{-1}
$\alpha_4 = 6.6400 \times 10^{-2}$	$K \cdot kJ^{-1}$

Table 3. Gel and Glass Effect Model Parameters

$\alpha_{t_{11}} = 1.0575 \times 10^{-1}$	$\alpha_{t_{12}} = 1.7150 \times 10^{1}$
$\alpha_{t_{13}} = 1.7150 \times 10^{-2}$	$\alpha_{t_{21}} = 2.3000 \times 10^{-6}$
$\alpha_{t_{22}} = 7.5000 \times 10^{1}$	$\alpha_{p_{21}} = 7.1000 \times 10^{-5}$
$\alpha_{p_{22}} = 1.7153 \times 10^{-2}$	$\alpha_{v_1} = 1.8560 \times 10^{-1}$
$\alpha_{v_{22}} = 2.9650 \times 10^{-4}$	$\alpha_p = 4.8000 \times 10^{-4}$
$\alpha_s = 1.0000 \times 10^{-3}$	$\alpha_m = 1.0000 \times 10^{-3}$

$$x_p = \frac{\mu_1}{\mu_1 + M_m C_m}. \tag{19}$$

The volume expansion factor $\bar{\epsilon}$ is determined by $\bar{\epsilon} = \phi_{m_{m_s}}(\bar{\rho}/\rho_p - 1)$, where $\phi_{m_{m_s}} = C_m M_m/\rho_m$ is the volume fraction of the monomer in the inlet stream. The other kinetic, operating and physical parameters for the system MMA, AIBN, and toluene are in Tables 4 and 5. Species balances for the monomer, initiator, solvent, and dead polymer (assuming perfect mixing) and an energy balance for the reactor (assuming constant reacting mixture heat capacity c and overall heat-transfer coefficient U, $c_{ms} = c$, $\rho_{ms} = \rho$, and $Td\rho/dt \ll \rho dT/dt$) give the set of ordinary differential equations

$$\frac{dC_m}{dt} = R_m(C_m, C_i, C_s, T) + \left(C_{m_{m_s}} - (1 + \bar{\epsilon}x_p)C_m\right)/\tau \tag{20a}$$

$$\frac{dC_i}{dt} = R_i(C_m, C_i, C_s, T) + \left(F_i C_{i_{i_s}} - (1 + \bar{\epsilon}x_p)C_i\right)/V \tag{20b}$$

$$\frac{dC_s}{dt} = \left(F_m C_{s_{m_s}} + F_i C_{i_{i_s}} - F_m(1 + \bar{\epsilon}x_p)C_s\right)/V \tag{20c}$$

$$\frac{d\mu_1}{dt} = -M_m R_m(C_m, C_i, C_s, T) - ((1 + \bar{\epsilon}x_p)\mu_1)/\tau \tag{20d}$$

$$\frac{dT}{dt} = \alpha_0 k_p \xi_0 C_m + \alpha_1(T_j - T) + (T_{m_s} - (1 + \bar{\epsilon}x_p)T)/\tau \tag{20e}$$

$$\frac{dT}{dt} = \alpha_2(T_j - T) + \alpha_3(T_\infty - T_j) + \alpha_4(P - F_{cw}\rho_{cw}(T_j - T)) \tag{20f}$$

where the states variables are

- C_m: concentration of the monomer;
- C_i: concentration of the initiator;
- C_s: concentration of the solvent;
- μ_1: mass concentration of the polymer chains;
- T: reactor temperature; and
- T_j: jacket temperature.

Experimental Conditions. The aim of this example is to identify a nonlinear dynamic model of the conversion factor of the polymer using the power given to the heater, the cooler flow rate, and the initiator flow rate as inputs. The rest of the variables, that is, the monomer flow rate, the solvent flow rate, and so on, are held constant at their steady-state values. This is a very nonlinear problem as can be appreciated from Figure 7 that illustrates the steady-state conversion factor x_p vs. the power P. As usual, two data sets were generated to identify the system: a training and test set using a probability $P_s = 0.04$. The former set was used to fit the model, while the latter one was used to validate the model. The outputs were generated integrating those inputs using the set of Eqs. 20 with a sampling time of $T = 1{,}000$ s. The DABNet model is

Table 4. Operating Conditions

$C_{i_{is}} = 2.6340 \times 10^{-1}$	kmol·m^{-3}
$C_{s_{is}} = 9.9860 \times 10^{0}$	kmol·m^{-3}
$C_{s_{ms}} = 5.8810 \times 10^{0}$	kmol·m^{-3}
$C_{m_{ms}} = 3.9840 \times 10^{0}$	kmol·m^{-3}

Table 5. Physical and Kinetics Parameters

$k_l = Z_l \exp\left(\frac{-E_l}{RT}\right)$	$l = f_m, i, t_0, P_0$
$Z_{t_0} = 9.8000 \times 10^{7}$	m^3·kmol^{-1}·s^{-1}
$E_{t_0} = 2.9442 \times 10^{3}$	kJ·kmol^{-1}
$Z_{P_0} = 4.9167 \times 10^{5}$	m^3·kmol^{-1}·s^{-1}
$E_{P_0} = 1.8283 \times 10^{4}$	kJ·kmol^{-1}
$Z_{f_m} = 4.6610 \times 10^{9}$	m^3·kmol^{-1}·s^{-1}
$E_{f_m} = 7.4479 \times 10^{4}$	kJ·kmol^{-1}
$Z_i = 1.0533 \times 10^{15}$	s^{-1}
$E_i = 1.2877 \times 10^{5}$	kJ·kmol^{-1}
$T_{gp} = 3.8720 \times 10^{2}$	K
$T_{gm} = 1.6720 \times 10^{2}$	K
$T_{gs} = 1.1320 \times 10^{2}$	K
$c = 1.8150 \times 10^{0}$	kJ·kg^{-1}·K^{-1}
$f = 5.8000 \times 10^{-1}$	
$m = 8.6460 \times 10^{0}$	kg
$R = 8.3450 \times 10^{0}$	kJ·kmol^{-1}·K^{-1}
$\bar{\rho}_m = 8.8570 \times 10^{2}$	kg·m^{-3}
$\bar{\rho}_s = 8.1550 \times 10^{2}$	kg·m^{-3}
$\rho_p = 1.1660 \times 10^{3}$	kg·m^{-3}
$M_m = 1.0012 \times 10^{2}$	kJ·kmol^{-1}
$M_s = 9.2140 \times 10^{2}$	kJ·kmol^{-1}
$M_i = 1.6421 \times 10^{2}$	kJ·kmol^{-1}
$\Delta H_P = 5.7800 \times 10^{4}$	kJ·kmol^{-1}

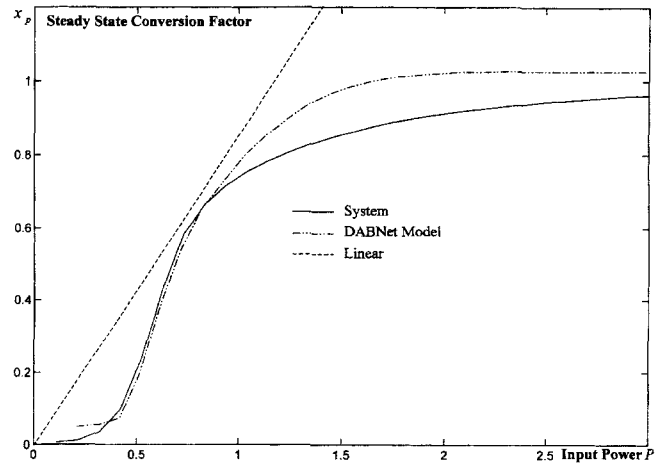

Figure 7. Steady-state behavior of the polymerization system, the DABNet model, and the linear model.

built to reproduce the conversion factor x_p using three inputs: the heater input power P, the cooling water inlet flow rate F_{cw}, and the initiator flow rate stream F_i (see Eq. 20), while keeping the others constant. As a way to show the relative robustness of the approach against choosing bad values for the Laguerre poles, those were selected to be $a_1 = a_2 = a_3 = 0.5$ (just in the middle of the valid range), while the dead times were selected to be: $d_1 = 1$, $d_2 = 1$, $d_3 = 1$. Six Laguerre basis terms were used for each input for the initial expansion of the linear dynamic part. For the neural network, we arbitrarily chose to use 10 hidden nodes without attempting to optimize these parameters. After the application of the methodology, the system was reduced to a second-order system for each input. Figure 8 shows the approximation results for the DABNet model and for a linear model. The linear model was adjusted using a PLS strategy, being composed by a six-state linear state-space model. Figure 8a shows the DABNet approximation and the real output of the system for the training and test sets, while Figure 9a shows the training and test errors for the final DABNet model. For comparison purposes, Figure 8b shows the approximation of the linear model and the real output, while Figure 9b shows the train and test error for the linear model. Note again that the nonlinear model performs better than the linear one. Similar considerations to the CSTR case apply to this example with respect to the steady-state curve.

Note that the steady-state curve of the DABNet also fits closely the steady-state curve in this case. The small discrepancy in this example, however, can be found in the conversion factor range higher than 0.8. This behavior is also explained by the lack of information used to train the model in this range. As can be appreciated from Figure 7, the data used to fit the model were sparse for a conversion factor greater than 0.8. However, the DABNet model can fit this curve better than any linear model. It is obvious that no matter where a straight line is located, it cannot approximate the entire range.

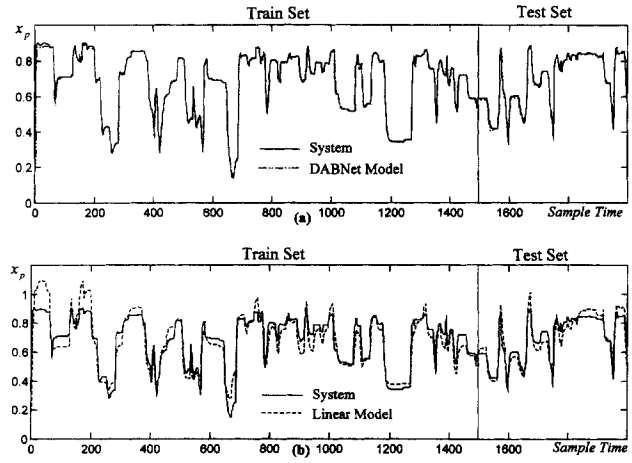

Figure 8. Conversion factor approximations discriminated by train and test set for the polymer example.

(a) Approximation of x_p by the DABNet model; (b) approximation of x_p by the linear model.

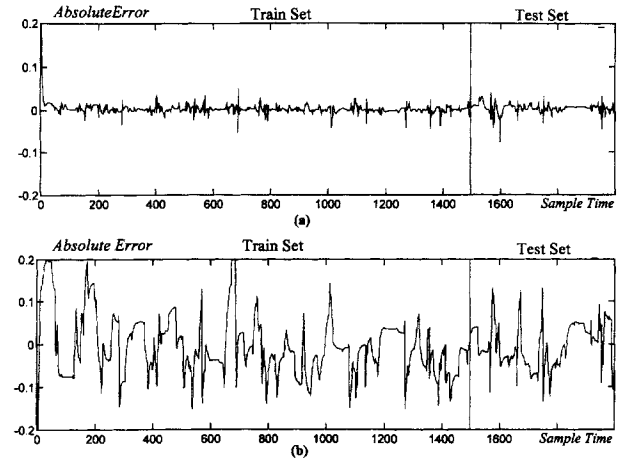

Figure 9. Absolute error of the x_p approximation discriminated by train and test set in the polymer example.

(a) DABNet model. The norm of the error is: 1.263 for the train set and 0.251 for the test set; (b) linear model. The norm of the error is: 2.862 for the train set and 1.122 for the test set.

Conclusions

We have shown a practical approach for identifying nonlinear models from available data. For that purpose, we use the DABNet (Decoupled A-B Net) structure, which is composed of a decoupled linear dynamic system followed by a nonlinear static map. The linear dynamic system is initially spanned by a set of discrete Laguerre systems, and then cascaded with a single hidden layer Perceptron. A model reduction technique (linear balancing) is performed on the hidden nodes of the neural net. In that way, it is possible to identify the main time constants and to reduce the dimensionality of the Perceptron input. We proved that the DABNet structure is able to approximate every nonlinear, causal, discrete time invariant, multiple-input, single-output system with fading memory. The final model consists of a linear state-space system whose states, decoupled by input, are mapped by a neural network. Results concerning the approximation of a pH neutralization and a polymer process were presented, showing the feasibility of this approach in the approximation of typical process units.

Literature Cited

Barker, A. H., "Design of Multi-Level Pseudorandom Signals for System Identification," in *Perturbation Signal for System Identification*, Prentice Hall, New York (1993).

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

Boutayeb, M., and M. Darouach, "Recursive Identification Method for MISO Wiener-Hamerstein Model," *IEEE Trans. on Automatic Control*, **40**(2), 287 (1995).

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

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

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

García, C. E., D. M. Prett, and M. Morari, "Model Predictive Control: Theory and Practice—A Survey," *Automatica*, **25**(3), 335 (1989).
Gu, G., P. P. Khargonekar, and E. B. Lee, "Approximation of Infinite-Dimensional Systems," *IEEE Trans. Automatic Control*, **34**(6), 610 (1989).
Hernández, H., and Y. Arkun, "Neural Network Modeling and an Extended DMC Algorithm to Control Nonlinear Systems," *Proc. ACC 90*, 2454 (1990).
Hernández, H., and Y. Arkun, "Control of Nonlinear Systems using Polynomial ARMA Models," *AIChE J.*, **39**(3), 446 (1993).
Heuberger, P. S., P. M. Van den Hoof, and O. H. Bosgra, "A Generalized Orthonormal Basis for Linear Dynamical Systems," *Proc. of the 32nd Conf. on Dec. & Con.*, San Antonio, TX, 2850 (Dec. 1993).
Ishida, M., and J. Zhan, "Neural Model Predictive Control of Distributed Parameter Cristal Growth Process," *AIChE J.*, **41**(10), 2333 (1995).
Kolmogorov, A. N., and S. V. Fomin, *Introductory Real Analysis*, Dover, New York (1980).
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," *Technical Report*, 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. of Control*, **41**, 303 (1985).
Lin Lin, J. J., D. S. Wong, and S. W. Yu, "Optimal Multiloop Feedback Design Using Simulated Annealing and Neural Network," *AIChE J.*, **41**(2), 430 (1995).
Mane, B. R., and F. J. Doyle, III, "Polymerization Reactor Control using Autoregressive-plus Volterra-Based MPC," *AIChE J.*, **43**(7), 1763 (1997).
Matthews, M. B., and G. S. Moschytz, "The Identification of Nonlinear Discrete-Time Fading-Memory Systems Using Neural Network Models," *IEEE Trans. Circuits and Systems II*, **41**(11), 740 (1994).
Michalska, H., and D. Q. Mayne, "Robust Receding Horizon Control of Constrained Non-Linear Systems," *IEEE Trans. Automatic Control*, **38**(11), 1623 (1993).
Moody, J., and C. Darken, "Fast Learning in Networks of Locally Tuned Processing Units," *Neural Computation*, **1**, 281 (1988).
Morari, M., and J. H. Lee, "Model Predictive Control: Past, Present and Future," *PSE-ESCAPE '97 Meeting*, Trondheim, Norway (May 25–29, 1997).
Narenda, S., and K. Parthasarathy, "Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks," *IEEE Trans. on Neural Networks*, **2**(2), 252 (1991).
Oliveira, N. M. C., and L. T. Biegler, "Constraint Handling and Stability Properties of Model-Predictive Control," *AIChE J.*, **40**(7), 1138 (1996).
Partington, J. R., and P. M. Mäkilä, "Modeling of Linear Fading Memory Systems," *IEEE Trans. Automatic Control*, **41**(6), 899 (1996).
Pottmann, M., H. Unbehauen, and D. E. Seborg, "Application of a General Multi-Model Approach for Identification of Highly Nonlinear Processes—A Case Study," *International Journal of Control*, **57**(1), 97 (1993).
Qin, S. J., and T. A. Badgwell, "An Overview of Industrial Model Predictive Control Technology," *Preprints of CPC-V*, Tahoe, 122 (1996).
Sentoni, G., O. Agamennoni, A. Desages, and J. Romagnoli, "Approximates Models for Nonlinear Process Control," *AIChE J.*, **42**(8), 2240 (1996).
Sentoni, G., "LNet: A Structure for Nonlinear Discrete System Approximation," PhD Thesis, Electrical Engineering Dept., Universidad Nacional del Sur, Bahía Blanca, Argentina (1997).
Shamma, J. S., and R. Zhao, "Fading-Memory Feedback Systems and Robust Stability," *Automatica*, **29**, 191 (1993).
Soroush, M., and C. Kravaris, "Multivariable Nonlinear Control of a Continuous Polymerization Reactor," *Proc. ACC'92*, San Francisco, **I**, 607 (1992).
Temeng, K. O., P. D. Schnelle, Su Hong-Te, and T. McAvoy, "Neural Model Predictive Control of an Industrial Packed Bed Reactor," AIChE Meeting (1992).
Tsoi, A. C., and A. D. Back, "Locally Recurrent Globally Feedforward Networks: A Critical Review of Architectures," *IEEE Trans. on Neural Networks*, **5**(2), 229 (1994).
Wahlberg, Bo, "System Identification Using Laguerre Models," *IEEE Trans. on Automatic Control*, **36**(5), 551 (1991).
Wahlberg, Bo, "System Identification Using Kautz Models," *IEEE Trans. on Automatic Control*, **39**(6), 1276 (1994).
Williams, R. J., and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Computation*, **102**, 270 (1989).
Zervos, C., P. Belanger, and G. Dumont, "On PID Controller Tuning using Orthonormal Series Identification," *Automatica*, **24**, 165 (1988).
Zhao, H., J. Guiver, and C. Klimasauskas, "NeuCOP II: A Nonlinear Multivariable Process Modeling, Control and Optimization Package," AIChE Meeting, Houston, TX (1997).
Zhou, K., J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice Hall, Englewood Cliffs, NJ (1996).

Appendix

To prove the DABNet structure theorem, three Lemmas and a restatement on the Stone-Weierstrass theorem need to be presented. Moreover, recall the relation between an operator and a functional and that past (future) time is denoted by nonpositive (non-negative) integers. Then, the truncation operator P: $l_p^\infty(\mathbf{Z}) \to l_p^\infty(\mathbf{Z}_-)$ is defined as

$$P[\boldsymbol{u}](k) := \begin{cases} \boldsymbol{u}(k), & k \le 0 \\ \mathbf{0}, & k > 0 \end{cases}$$

while with every sequence $\boldsymbol{u}$ in $l_p^\infty(\mathbf{Z}_-)$, we associate a causal extension sequence $\boldsymbol{u}_e$ in $l_p^\infty(\mathbf{Z})$ defined as

$$\boldsymbol{u}_e(k) = \begin{cases} \boldsymbol{u}(k), & k \le 0 \\ \boldsymbol{u}(0), & k > 0 \end{cases}$$

With each time-invariant causal operator N, we associate a functional F on $l_p^\infty(\mathbf{Z}_-)$ defined by $F\boldsymbol{u} = N\boldsymbol{u}_e(0)$. Therefore, the operator N can be recovered from its associated functional F via

$$N\boldsymbol{u}(k) = FPQ^{-k}\boldsymbol{u}. \tag{A1}$$

where Q, the delay operator, is defined as $Q^\tau[\boldsymbol{u}](k) := \boldsymbol{u}(k-\tau)$. Then, N is continuous if and only if F is; so, the above equations establish a one to one correspondence between any time invariant causal continuous operator and its functional F on $l_p^\infty(\mathbf{Z}_-)$.

Stone-Weierstrass theorem (Boyd, 1985)

Suppose $\boldsymbol{E}$ is a compact metric space and $\boldsymbol{G}$ is a set of continuous functionals on $\boldsymbol{E}$ that separates points, that is, for any distinct $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{E}$ there is a $G \in \boldsymbol{G}$ such that $G\boldsymbol{u} \neq G\boldsymbol{v}$. Then, for any continuous functional F on $\boldsymbol{E}$ and given any $\epsilon > 0$, there are functionals, $\{G_1^1, \ldots, G_{m_1}^1, \ldots, G_1^p, \ldots, G_{m_p}^p\} \subseteq \boldsymbol{G}$, $m = \sum_{i=1}^p m_i$ and a polynomial p: $\boldsymbol{R}^m \to R$, such that for all $\boldsymbol{u} \in \boldsymbol{E}$

$$|F\boldsymbol{u} - p\left(G_1^1\boldsymbol{u}, \ldots, G_{m_1}^1\boldsymbol{u}, \ldots, G_1^p\boldsymbol{u}, \ldots, G_{m_p}^p\boldsymbol{u}\right)| < \epsilon$$

The reason for the group indexing, which is not necessary for a general statement of the Stone-Weierstrass theorem, will become apparent in Lemma 2 when we consider each block with a Laguerre operator. In addition, the fading memory definition is stated and three lemmas are necessary before the theorem can be proved.

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

$$\sup_{k \le 0} \{|\boldsymbol{u}(k) - \boldsymbol{v}(k)| w(-k)\} < \delta \to |N\boldsymbol{u}(0) - N\boldsymbol{v}(0)| < \epsilon.$$

Lemma 1: The sub-space $K_- \equiv \{\boldsymbol{u} \in l_p^\infty(Z_-) | 0 < \|\boldsymbol{u}\| \le c_1\}$ is compact with the $\|\cdot\|_w$ norm defined for any decreasing sequence w: $\mathbf{Z}_+ \to (0,1]$, $\lim_{k\to\infty} w(k) = 0$

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

Proof: Let $\boldsymbol{u}^{(n)}$ be any sequence in K_-. We will find $\boldsymbol{u}^{(0)} \in K_-$ and a subsequence of $\boldsymbol{u}^{(n)}$ converging in the $\|\cdot\|_w$ norm to $\boldsymbol{u}^{(0)}$. It is well known that K_- is not compact in $l_p^\infty(\mathbf{Z}_-)$ with the usual supremum norm $\|\cdot\|_\infty$ (Kolmogorov, 1980). For each l, let $K_-[-l,0]$ be the restriction of K_- to $[-l,0]$. $K_-[-l,0]$ is uniformly bounded by c_1 and is composed of a finite set of values, hence, compact in $l_p^\infty[-l,0]$. Since $K_-[l,0]$ is compact for every l, we can find a subsequence $\boldsymbol{u}^{(n_m)}$ of $\boldsymbol{u}^{(n)}$ and a $\boldsymbol{u}^{(0)} \in K_-[-l,0]$ along with $\boldsymbol{u}^{(n_m)}$ converges:

$$\sup_{-l \le k \le 0} |\boldsymbol{u}^{(n_m)}(k) - \boldsymbol{u}^{(0)}(k)| \to 0 \quad \text{as} \quad m \to \infty. \tag{A3}$$

Now, for some $\epsilon > 0$, since $w(k) \to 0$ as $k \to \infty$, we can find a $k_0 > 0$ such that $w(k_0) \le \epsilon / c_1$. Since $\boldsymbol{u}^{(n_m)}, \boldsymbol{u}^{(0)} \in K_-$, we have that

$$\sup_{k \le -k_0} \{|\boldsymbol{u}^{(n_m)}(k) - \boldsymbol{u}^{(0)}(k)| w(-k)\} \le 2c_1 w(k_0) < \epsilon \tag{A4}$$

Now, from Eq. A3 we can find m_1 such that

$$\sup_{-k_0 < k \le 0} |\boldsymbol{u}^{(n_m)}(k) - \boldsymbol{u}^{(0)}(k)| < \epsilon \quad \text{for} \quad m > m_1 \tag{A5}$$

so by Eq. A3 and Eq. A5, we can conclude that $\|\boldsymbol{u}^{(n_m)} - \boldsymbol{u}^{(0)}\|_w < \epsilon$, $m > m_1$ which proves that K_- is compact.

Lemma 2. The set of functionals $\{G_i^j\}$ associated to the discrete Laguerre Operators are continuous with respect to the norm $\|\cdot\|_w$, that is, given any $\epsilon > 0$ there exists $\delta > 0$ such that $\|\boldsymbol{u} - \boldsymbol{v}\|_w < \delta \Rightarrow |G_i^j \boldsymbol{u} - G_i^j \boldsymbol{v}| < \epsilon$.

Proof. Every Laguerre system is associated with an operator that defines the input-output as: $L_i^j \boldsymbol{u}(k) = L_i^j u_j(k)$. With each of the Laguerre operators $L_i^j(\cdot)$, we associate a functional defined by $G_j^i u_i = \sum_{k=0}^\infty l_j^i(k) u_i(-k)$, such that the application of the functional to a sequence is performed as $G_i^j \boldsymbol{u} = G_i^j u_j$. Consider the functional $G_i^j(\cdot)$ associated with $L_i^j(\cdot)$, so given $\epsilon > 0$, choose $\delta > 0$ such that

$$|u_j - v_j| < \delta \quad \Rightarrow \quad |G_i^j u_j - G_i^j v_j| < \epsilon \tag{A6}$$

This is possible due to the continuity of the one-dimensional (1-D) Laguerre operators with respect to the weighted norm as shown in Sentoni et al. (1996). Therefore, from Eq. A6 and the definition of the functionals

$$\|\boldsymbol{u} - \boldsymbol{v}\|_w < \delta \Rightarrow |u_j - v_j|_w < \delta \Rightarrow |G_i^j \boldsymbol{u} - G_i^j \boldsymbol{v}| = |G_i^j u_j - G_i^j v_j| < \epsilon \tag{A7}$$

which proves Lemma 2.

Lemma 3. The functionals $\{G_i^j\}$ separate points in $l_p^\infty(\mathbf{Z}_-)$, that is, for any distinct $\boldsymbol{u}, \boldsymbol{v} \in l_p^\infty(\mathbf{Z}_-)$ there is a $G_i^j \in \boldsymbol{G}$ such that $G_i^j \boldsymbol{u} \ne G_i^j \boldsymbol{v}$.

Proof. Suppose $\boldsymbol{u}, \boldsymbol{v} \in l_p^\infty(\mathbf{Z}_-)$ are equal except for the jth component. Then, by the functionals' definition

$$G_i^j \boldsymbol{u} \ne G_i^j \boldsymbol{v} \Leftrightarrow G_i^j u_j \ne G_j^i v_j \tag{A8}$$

It known from 1-D theory (Sentoni et al., 1996) that for any distinct $u_j, v_j \in l^\infty(\mathbf{Z}_-)$ there is a G_i^j such that $G_i^j u_j \ne G_i^j v_j$; this result together with Eq. A8 proves Lemma 3.

Approximation theorem

Now given $\epsilon > 0$, Lemmas 1, 2, 3 together with the Stone-Weierstrass theorem imply that given any continuous functional F on K_-, there is a polynomial p: $\boldsymbol{R}^m \to \boldsymbol{R}$, such that for all $\boldsymbol{u} \in K_-$

$$|F\boldsymbol{u} - p\left(G_1^1 \boldsymbol{u}, \ldots, G_{m_1}^1 \boldsymbol{u}, \ldots, G_1^p \boldsymbol{u}, \ldots, G_{m_p}^p \boldsymbol{u}\right)| < \epsilon \tag{A9}$$

Because the Laguerre systems are continuous and act on a bounded space, the $G_i^j \boldsymbol{u}$ are bounded real intervals. So, the polynomial p can be replaced by any static model that acts as a universal approximator on a bounded input space, for example, a neural net. In other words Eq. A9 can be replaced by

$$|F\boldsymbol{u} - NN\left(G_1^1 \boldsymbol{u}, \ldots, G_{m_1}^1 \boldsymbol{u}, \ldots, G_1^p \boldsymbol{u}, \ldots, G_{m_p}^p \boldsymbol{u}\right)| < \epsilon$$

Using Eq. A1, a time-invariant causal operator N can be recovered from its associated functional, so let $\boldsymbol{u} \in K$ and $k \in \mathbf{Z}$, so $PQ^{-k}\boldsymbol{u} \in K_-$, hence

$$|FPQ^{-k}\boldsymbol{u} - NN\left(G_1^1 PQ^{-k}\boldsymbol{u}, W \ldots, G_{m_1}^1 PQ^{-k}\boldsymbol{u}, \ldots, G_1^p PQ^{-k}\boldsymbol{u}, \ldots, G_{m_p}^p PQ^{-k}\boldsymbol{u}\right)| < \epsilon.$$

Since the last equation is true for all $k \in \mathbf{Z}$, we conclude that for all $\boldsymbol{u} \in K_-$, $\|N\boldsymbol{u} - \hat{N}\boldsymbol{u}\| < \epsilon$. This completes the proof.

Manuscript received Oct. 20, 1997, and revision received Mar. 5, 1998.