Hierarchical Fuzzy Models within the Framework of Orthonormal Basis Functions and Their Application to Bioprocess Control

Chemical Engineering Science

Vol. 58, p. 4259-4270, 2003

Unformatted Manuscript
Abstract

Fuzzy models within the framework of orthonormal basis functions (OBF Fuzzy Models) have been introduced in previous works and shown to be a very promising approach to the areas of nonlinear system identification and control, since they exhibit several advantages over those dynamic model topologies usually adopted in the literature. As fuzzy models, however, they exhibit the dimensionality problem which is the main drawback to the application of neural networks and fuzzy systems to the modeling and control of large-scale systems. This problem has successfully been dealt with in the literature by means of hierarchical structures composed of submodels connected in cascade. In the present paper a hierarchical fuzzy model within the OBF framework is presented. A data-driven hybrid identification method based on genetic and gradient-based algorithms is described in details. A model-based predictive control scheme is also presented and applied to control of a complex industrial process for ethyl alcohol (ethanol) production.

1 Introduction

In most of the (black-box) modeling strategies developed for nonlinear dynamic system identification, the task of determining a suitable regression vector using the relevant past terms of the discrete-time input/output signals of the process to be modeled is a key problem. In the context of fuzzy modeling [3], for instance, which is the main
subject of the present paper, the selection of adequate regressors among a set of candidates is usually addressed by means of trial-and-error procedures or heuristic methods. These methods, however, are often tedious, time-consuming and/or computationally demanding.

An alternative to get around this problem when modeling stable dynamic systems is the use of orthonormal basis functions, such as the Laguerre and Kautz functions [8, 30]. These functions have successfully been used for linear and nonlinear system modeling and control [20, 28, 29, 19, 18, 6, 23], especially because the corresponding dynamic models exhibit several useful features, such as: i) Absence of output recursion / feedback of prediction errors, often leading to superior performances over long-range horizon predictions. The absence of output recursion also results in a natural decoupling between multiple outputs (multivariable case) and desirable statistical characteristics for the numerical estimation of linear-in-parameter models via least squares methods [14]; ii) There is no need for previous knowledge about the relevant past terms of the system signals; iii) It is possible to increase the representability of the models merely by increasing the number of functions; iv) The representation of a stable system is assuredly stable; v) Tolerance to unmodeled dynamics; vi) Ability to deal with time delays.

Orthonormal Basis Function (OBF) Fuzzy Models were introduced by Oliveira et al. [24], who showed that an OBF fuzzy model outperformed an OBF Volterra model in the modeling of a laboratory-scale tank using fewer parameters and the same (least squares) estimation algorithm. A similar result was obtained by Campello et al. [14] in a comparative study involving four kinds of OBF models. Campello and Amaral [12] showed that the OBF fuzzy models presented in the papers referred to above are particular realizations of a more general and interpretable formulation, while being capable of approximating to desired accuracy a wide class of nonlinear dynamic systems. As fuzzy models, however, they suffer from the well-known problem called “Curse of Dimensionality”, i.e., an exponential increase in the number of parameters and data needed to provide the model with a desired accuracy as a function of the dimension of its input space.

In order to get around the curse of dimensionality problem in fuzzy control, Raju et al. [27] proposed a hierarchical structure in which a set of subsystems connected in cascade is used instead of a single fuzzy system. In this hierarchical structure the number of fuzzy rules increases linearly (instead of exponentially) with the dimension of the input space, thus allowing the application of fuzzy control to large-scale systems [5]. Recently, many authors have applied this structure to fuzzy modeling of complex systems (e.g. see [31, 21] and references therein).

In the present paper, a hierarchical fuzzy model within the OBF framework previously mentioned is presented. More specifically, it is composed of a Laguerre state-space dynamic followed by a static hierarchical structure composed of a series of fuzzy subsystems connected in cascade. The aim is to bring together the advantages of both the hierarchical and OBF frames into a single fuzzy model. The hierarchical structure considered here uses a special kind of fuzzy subsystem, called Simplified Relational Structure [25], which is under certain conditions completely equivalent to a Radial Basis Function (RBF) neural network [9] (see appendix A). Its formulation is, however, easier to manipulate since it deals separately with each input variable.

A data-driven hybrid identification method based on genetic and gradient-based algorithms is proposed for
estimating the free design parameters of the OBF hierarchical fuzzy model. In a nutshell, a genetic algorithm is used to generate an adequate initial model to be fine-tuned by a conjugate gradient algorithm. The conjugate gradient algorithm of Fletcher and Reeves [7] is used to carry out the fine-tuning procedure. This algorithm is well suited to large-scale problems since it does not demand the computation of any Hessian matrix or its inverse, thus having relatively small memory and processing requirements. Also, it ensures the convergence of the optimization procedure (with a second-order rate).

A model-based predictive control scheme is also presented in the paper. It is based on the use of the Generalized Predictive Controller (GPC) [16] combined with a linearization of the OBF hierarchical fuzzy model at each sampling instant. The linearization procedure is required since the GPC algorithm relies on a linear model of the process to be controlled when computing its optimal predictive control law.

A biotechnological application is considered to illustrate the performance of the proposed modeling and control schemes in a real-world problem. Biotechnology has become increasingly important in the activities of contemporary society as a “clean” and safe technology when compared to traditional chemical processes. Moreover, it provides extremely useful and valuable products in several industrial areas. Biotechnological processes are characterized by their complex dynamics, such as strong nonlinearities, especially because the main driving force of these processes are microorganisms (cells) that are very sensitive to any environmental variations in the fermentation broth. The particular case study considered here is a typical large-scale industrial plant to produce ethanol from sugar cane syrup. The process operational conditions are those typically found in the Brazilian distilleries.

The paper is organized as follows. In the next section, a brief review of OBF models is provided. This review focus on models implemented using Laguerre functions, which are those considered in the present paper and the most usually adopted in the literature as well. In section 3, the “curse of dimensionality” problem is addressed. The discussions take place in the context of the simplified fuzzy systems which are used as the subsystems of the hierarchical model described in section 4. The development of the hierarchical model within the OBF framework is then presented in section 5. Section 6 describes the predictive control scheme based on this model. The ethanol production process under interest is detailed in section 7. The proposed modeling and control schemes are applied to this process in sections 8 and 9, respectively, and the concluding remarks are then presented in section 10.

2 Laguerre Function-Based Models

Models based on Laguerre functions can generically be represented as\(^1\) (see [24, 14, 12] and references therein):

\[
I(k + 1) = AI(k) + Bu(k) \tag{1}
\]

\[
\hat{y}(k) = Hf(I(k)) \tag{2}
\]

\(^1\)For simplicity and without loss of generality Single-Input/Single-Output (SISO).
where \( u(k) \) and \( \hat{y}(k) \) are the input and the output of the model at discrete-time instant \( k \), respectively, \( \mathcal{H} \) is a generic memoryless operator and \( \mathbf{l}(k) = [l_1(k) \cdots l_n(k)]^T \) is the vector of Laguerre states. Equation (1) constitutes the state-space representation of the first \( n \) Laguerre transfer functions, where \( n \) is the model order. Matrix \( A \) and vector \( b \) are given by:

\[
A = \begin{bmatrix}
    p & 0 & 0 & \cdots & 0 \\
    1 - p^2 & p & 0 & \cdots & 0 \\
    (-p)(1 - p^2) & 1 - p^2 & p & \cdots & 0 \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    (-p)^{n-2}(1 - p^2) & (-p)^{n-3}(1 - p^2) & \cdots & p
\end{bmatrix}
\]

(3)

\[
b = \sqrt{1 - p^2} \begin{bmatrix}
    1 - p & (-p)^2 & \cdots & (-p)^{n-1}
\end{bmatrix}^T
\]

(4)

where \( p \in \{\Re : -1 < p < 1\} \) is the stable pole which parameterizes the Laguerre functions. The selection of this pole is not crucial because the completeness of the Laguerre basis ensures that an increase in the number \( n \) of functions enhances the representability of the model for any \( p \). The pole \( p \), however, can be selected to minimize the approximation error associated with a given number of functions. Analytical solutions to this problem have been presented in the literature in the context of linear \([19]\) and Volterra \([13]\) models within the same framework given by equations (1) to (4). Disregarding the particular type of model, which is determined solely by the specific form of the operator \( \mathcal{H} \) in (2), the simplest yet effective way to select this pole is by using \textit{a priori} knowledge of the dominant dynamic of the system, for example considering the shape of its time or frequency responses \([29, 14]\).

As mentioned above, a particular type of Laguerre model is characterized by a specific form of the operator \( \mathcal{H} \) in (2). For instance, \( \mathcal{H} \) can be a linear or affine combination of the states \( l_i(k) \) (Linear Laguerre model) \([19, 23]\), a Volterra polynomial (Wiener-Volterra model) \([18, 13]\), an artificial neural network \([6, 14]\) (OBF neural model), as well as a fuzzy system (OBF fuzzy model) \([24, 14, 12]\). Campello \textit{et al}. \([14]\) showed that an OBF fuzzy model outperformed all the others in the modeling of the same bioprocess considered in the present paper, but in a monovariable setting. However, as a fuzzy model, it exhibits the dimensionality problem described in section 1. To avoid this problem, which may be critical when a large number \( n \) of Laguerre functions is required (especially in the modeling of multivariable systems), the use of hierarchical fuzzy models is proposed in this paper to implement the operator \( \mathcal{H} \) in equation (2). This strategy results in a hierarchical fuzzy model within the framework of orthonormal basis functions (OBF hierarchical fuzzy model). As stated before, the hierarchical model considered here is composed of a series of submodels connected in cascade. Before discussing this model and its OBF version in details, the specific fuzzy system adopted to implement its hierarchical submodels is presented in section 3.
3 Simplified Fuzzy Relational Systems

A Multi-Input/Single-Output (MISO) simplified fuzzy relational system [25] possesses a generic structure given by 
\[ \hat{y} = \mathcal{F}(x_1, \cdots, x_n), \]
where \( \mathcal{F} \) is a nonlinear operator which maps the inputs \( x_i \) (\( i = 1, \cdots, n \)) into the output \( \hat{y} \). Specifically, the model output \( \hat{y} \) is governed by the following equation:

\[ \hat{y} = \Psi^T \Omega \]  
(5)

where \( \Omega (m \times 1) \) is the parameter vector and \( \Psi (m \times 1) \) is the fuzzy input vector. Vector \( \Psi \) is given by the Kronecker Product (\( \otimes \)) of the individual fuzzy inputs, i.e.,

\[ \Psi = X_1 \otimes X_2 \otimes \cdots \otimes X_n \]  
(6)

which in turn are derived from the nonfuzzy inputs \( x_i \), as follows:

\[ X_i = [ X_{i1}(x_i) \quad X_{i2}(x_i) \quad \cdots \quad X_{ic_i}(x_i) ]^T \]  
(7)

where \( X_{ij}(\cdot) \) is the \( j \)th fuzzy set of the \( i \)th input variable (with \( c_i \) fuzzy sets). It is possible to demonstrate that the fuzzy model given by the equations presented above is completely equivalent to an RBF neural network with Gaussian activation functions whenever Gaussian fuzzy sets are used (see appendix A).

The model given by equations (5), (6), and (7) follows the conventional structure of fuzzy models (FM) and feedforward neural networks (NN) shown in figure 1-a. The main problem with this structure is discussed in the sequel. Consider, for simplicity and without loss of generality, that \( c_i = c \) for \( i = 1, \cdots, n \) in (7). Then, it can be seen from equation (6) that the number of elements of both vectors \( \Psi \) and \( \Omega \) in (5) is given by \( m = c^n \). This is the number of fuzzy rules associated with the model or, alternatively, the number of neurons in the equivalent RBF network. This is also the number of parameters to be estimated (synaptic weights in the RBF networks) if the fuzzy sets/activation functions are kept constant. On the other hand, if the centers and widths of the fuzzy sets/activation functions can be varied, then the number of parameters to be estimated becomes \( \mu = c^n + 2nc \).

From these equations, it is clear that the number of fuzzy rules/neurons \( m \) and parameters \( \mu \) needed to cover the input space with a given “density” is an exponential function of the number of inputs \( n \). Hierarchical models are presented in the next section as an alternative to avoid this dimensionality problem.

4 Hierarchical Models

As outlined in section 1, an approach to overcome the dimensionality problem of the conventional (nonhierarchical) models is the use of the hierarchical structure shown in figure 1-b, where \( n - 1 \) submodels (processing blocks) with two-dimensional input spaces are connected in cascade. Although submodels with different numbers of inputs
could also be used, the configuration with just two inputs will be considered hereafter because it leads to the most parsimonious model [27]. In this case, the number of fuzzy rules/neurons in each submodel is $c^2$. Consequently, the total number of fuzzy rules/neurons in the model is $m = (n-1)c^2$ ($n \geq 2$). This is also the number of parameters to be estimated if the fuzzy sets/activation functions are kept constant. Otherwise, the number of free design parameters becomes $\mu = (n-1)c^2 + 2nc + 2(n-2)c = (n-1)c^2 + 4(n-1)c$. These equations show that the rate of growth in the number of fuzzy rules/neurons $m$ and parameters $\mu$ as a function of the number of inputs $n$ is constant. This is a significant advantage when compared to the behavior of the nonhierarchical structure.

### 4.1 Formulation

Based on the formulation presented in section 3, the equations which describe the model shown in figure 1-b are:

\[
\hat{y}_i = \Psi_i^T \Omega_i = \sum_{l=1}^{c} \Psi_{il} \Omega_{il}, \quad i = 1, \ldots, n-1
\]
(8)

\[
\Psi_i = [ \Psi_{i1} \ldots \Psi_{il} ]^T = \begin{cases} X_{i+1} \otimes \hat{y}_{i-1}, & i = 2, \ldots, n-1 \\ X_{i+1} \otimes X_i, & i = 1 \end{cases}
\]
(9)

\[
X_j = [ X_{j1}(x_j) \ X_{j2}(x_j) \ \cdots \ X_{jc}(x_j) ]^T, \quad j = 1, \ldots, n
\]
(10)

\[
\hat{y}_h = [ \mathcal{Y}_{h1}(\hat{y}_h) \ \mathcal{Y}_{h2}(\hat{y}_h) \ \cdots \ \mathcal{Y}_{hc}(\hat{y}_h) ]^T, \quad h = 1, \ldots, n-2
\]
(11)

where $X(\cdot)$ and $\mathcal{Y}(\cdot)$ are the fuzzy sets associated with the inputs $x(\cdot)$ and hidden (intermediate) outputs $\hat{y}(\cdot)$, respectively. When Gaussian fuzzy sets are used, for instance, it follows that:
where $\theta_j$ (or $\phi_h$) and $\sigma_j$ (or $\phi_h$) are the center and the width of the $l$th fuzzy set associated with the $j$th input $x_j$ (or $h$th hidden output $\hat{y}_h$), respectively. The formulation presented above describes the hierarchical model as a kind of layered fuzzy neural network. Accordingly, data-driven neuro-fuzzy modeling techniques can be used for estimating its unknown parameters [5, 4]. The identification of hierarchical models is discussed in the sequel.

### 4.2 Data-Driven Identification

Consider a set of $N$ input/output data pairs, i.e. \{ $x_1(k), \cdots, x_n(k), y(k)$ \}$_{k=1}^{N}$, measured from a system to be modeled. Then, a hierarchical model of the system can be estimated by solving the following optimization problem:

$$\min_{\Gamma} J \triangleq \frac{1}{2} \sum_{k=1}^{N} (y(k) - \hat{y}(k))^2$$

where $\Gamma$ denotes the set of all free design parameters of the model. Problem (13) can be solved using nonlinear optimization techniques [7], namely, gradient-based algorithms, as usual in neuro-fuzzy modeling approaches. These techniques require the computation of the gradient vector of the cost function $J$ with respect to the set of parameters $\Gamma$. The set $\Gamma$ related to the hierarchical models considered here is composed of the parameter vectors $\Omega(\cdot)$ in equation (8) as well as the centers ($\theta_j$ and $\phi_h$) and widths ($\sigma_j$ and $\phi_h$) of the fuzzy sets in (12). The derivatives of $J$ in (13) with respect to these parameters can be computed by applying the chain rule to equations (8), (9), (10), and (11). The corresponding analytical equations can be found in [11, 21].

It is well-known that gradient-based algorithms, such as the conjugate gradient algorithm of Fletcher and Reeves, are subject to getting stuck in local minima when solving nonconvex optimization problems. This characteristic is undesired since it makes the estimation procedure of the hierarchical fuzzy models (and many other architectures of neural nets and fuzzy systems) sensitive to the initialization of their parameters. In order to get around this problem, stochastic search methods like Genetic Algorithms (GAs) [22] can be used. In practice, owing to their demanding computational requirements, GAs are commonly used as part of hybrid schemes in which they are responsible for finding suitable rough solutions to be refined by gradient-based methods. Such a hybrid scheme will be illustrated in section 8.3 in the modeling of the ethanol production plant mentioned in the introduction.

### 4.3 Parameter Initialization

First, it is assumed that the input and output variables are normalized within a certain interval, such as $[-1, 1]$, in order to avoid numerical problems during the model optimization procedure. Under this assumption, the fuzzy sets associated with a given input or hidden output can be initialized empirically with a homogeneous distribution within the normalization interval, which means equally spaced centers and widths equal to the distance between
two consecutive centers. The fuzzy sets of the input variables could optionally be initialized using data-driven fuzzy clustering techniques [3].

The parameter vectors \( \Omega(\cdot) \) should be initialized randomly with zero mean and absolute values small enough so that the initial values of the hidden outputs belong (at least approximately) to the normalization interval.

5 OBF Hierarchical Fuzzy Model

An OBF hierarchical fuzzy model is characterized by the implementation of the operator \( \mathcal{H} \) in (2) using a cascaded structure like that in figure 1-b. In the case of SISO models, as in equations (1) and (2), a natural choice of the hierarchical disposal of the Laguerre states – as inputs to the cascaded fuzzy model – is the order of the states itself, i.e., \( x_1 = l_1, x_2 = l_2, \ldots, x_n = l_n \). The reason is the convergence of the Laguerre series, which suggests that the first Laguerre states should be placed on the first hierarchical levels. Indeed, the higher the hierarchical level, the larger the number of free design parameters (degrees of freedom) associated with the mapping from the corresponding input of the model into its output.

In the case of MISO models, there are multiple possibilities. The most straightforward are those exemplified in figures 2-a and 2-b, where \( l_1 = [l_{11} \ l_{12}]^T \) and \( l_2 = [l_{21} \ l_{22}]^T \) are the (2nd order) vectors of Laguerre states associated with the inputs \( u_1 \) and \( u_2 \) of the model, respectively.

Figure 2: OBF hierarchical models with two inputs having two Laguerre states each.

If no hierarchical distinction between the Laguerre states associated with different inputs is desired, then an alternative structure constituted by the composition of multiple SISO OBF hierarchical models can be adopted. In this case, the final aggregation of the individual model outputs can be accomplished by means of a fuzzy system, a neural network or simply an algebraic operator (e.g. sum).

\(^{3}\)MIMO models within the OBF framework are composed of multiple MISO models because of the natural decoupling of their outputs.
6 OBF Model-Based Predictive Control

The control strategy proposed here consists basically of the Generalized Predictive Controller (GPC) [16, 15, 10], which is widespread in the field of chemical engineering applications, combined with the on-line linearization of an OBF hierarchical fuzzy model of the process to be controlled. The idea is to derive a linearized model at each sampling instant to be used to compute the optimal predictive control law of the GPC algorithm, which is based on the formulation of a linear model of the process. In the OBF model given by equations (1) and (2), the state-space dynamic in (1) is linear, in such a way that the linearization is necessary only with respect to the static mapping $\mathcal{H}$ in (2) (hierarchical fuzzy model). The linearized model output at a certain state $\mathbf{l}_0$ is derived using a first-order Taylor expansion, as follows:

$$\hat{y}' = \mathcal{H}'(\mathbf{l}, \mathbf{l}_0) = \mathcal{H}(\mathbf{l}_0) + (\mathbf{l} - \mathbf{l}_0)^T \nabla \mathcal{H}(\mathbf{l}_0)$$

where $\nabla \mathcal{H}(\mathbf{l})$ is the gradient of the fuzzy mapping $\mathcal{H}$ with respect to the state vector $\mathbf{l}$. The respective derivatives are presented in appendix B. It can be noted that the model output in (14) is not truly linear, but affine with respect to the Laguerre states since the terms $\mathbf{l}_0$, $\mathcal{H}(\mathbf{l}_0)$ and $\nabla \mathcal{H}(\mathbf{l}_0)$ are constant terms. A linear model can easily be obtained by using an auxiliary output given by $\bar{y}' = \hat{y}' - \mathcal{H}(\mathbf{l}_0) + \mathbf{l}_0^T \nabla \mathcal{H}(\mathbf{l}_0) = \mathbf{l}_0^T \nabla \mathcal{H}(\mathbf{l}_0)$, which is clearly linear with respect to the state vector $\mathbf{l}$.

The linearization procedure is advantageous for several reasons. The main advantage is that it allows the use of the standard GPC algorithm, whose globally optimal control law is derived by solving a simple convex optimization problem. In case the nonlinear model is used, the optimization problem becomes nonconvex and its solution demands the use of more sophisticated algorithms (e.g. Sequential Quadratic Programming – SQP). Besides much more complex computationally, these algorithms are not globally optimal. Hence, the designer has no guarantee regarding the quality of the suboptimal solutions provided by the algorithm. The linearization procedure, on the other hand, is transparent in the sense that the designer knows that the longer the prediction horizon of the controller the highest the possibility of moving away from the linearization point and, accordingly, degrading the quality of the linearized model predictions. This is very similar to what happens when classical auto-regressive linear models (e.g. CARIMA) are used for predictive control, that is, the loss of accuracy of the recursive predictions over long-range horizons due to the feedback of prediction errors.

7 Industrial Process for Ethanol Production

7.1 Introduction

As stated previously, an industrial plant for the production of ethanol is considered. Because of difficulties encountered when working directly with the plant in operating mode, especially because of the high costs involved in interruptions in its operation for tests, it has been chosen to work with a simulator whose kinetic parameters have
already been validated in the real plant. This simulator was developed by Andrietta [1, 2], who modeled the set of biochemical reactions of the process by means of a set of phenomenological nonlinear ordinary differential equations (ODEs). The resulting phenomenological model, composed of 24 coupled ODEs and 20 algebraic equations, presents excellent agreement with actual data measured from the plant [1, 2]. By using this validated simulator, an optimized operation region was determined and further implemented in the actual plant so as to achieve satisfactory productivity values without affecting its operational and economic feasibility. Within this pre-optimized operation region, controllers should be applied to act on the manipulated variable(s) of the process in order to optimize its real time yield, even in the presence of disturbances. This is the main objective of this work.

7.2 Plant Description

The fermentative process for ethanol production is illustrated in figure 3. The system is a typical large-scale industrial process composed of four tank reactors (fermenters) arranged in series and operated with cell recycling to produce ethanol from sugar cane syrup. Each reactor has an external system of heat exchangers with independent control loops (PI controllers) whose objective is to maintain the temperature of the reactants (fermentation broth) constant at an ideal level for the fermentation process.

![Figure 3: Schematic illustration of the industrial plant for ethanol production.](image)

The process is fed with a mixture composed of sugars (Total Reducing Sugars - TRS) as well as sources of nitrogen and mineral salts, called feed medium. The feed medium is converted into ethanol by a fermentation process carried out using the yeast *Saccharomyces cerevisae*. A set of centrifuges splits the outlet fermented medium, which is formed of a mixture of water, CO₂, sugars, microorganisms (30-45kg/m³ of cells), and alcohol, into two phases. The heavy phase contains most of the cells (160-200kg/m³) whereas the light phase contains...
at most 3kg/m³ of cells and is 9-12% alcohol. The light phase is sent to the distillation unit, where the alcohol is extracted. The heavy phase is submitted to an acid treatment and dilution before being recycled into the first reactor. The recycle rate $t_r$ [dimensionless] relates the feed medium flow rate $F_0$ [m³/h] with the cell recycle flow rate $F_r$ [m³/h] and, accordingly, with the real inlet feed flow rate in the first tank $F_0$ [m³/h], as shown in figure 3. A recycle rate of 0.3, for instance, implies $F_0 = 0.7F_0$ and $F_r = 0.3F_0$. This is precisely the pre-optimized industrial operation condition of the actual plant. More details on this process can be found in [1, 2], and a set of trials illustrating its strongly nonlinear behavior is presented in [17].

Considering all the above-mentioned characteristics, the fundamental objective of the study of the fermentative process for ethanol production is to generate models and controllers so as to maximize its efficiency. In the present context, this means to regulate the ethanol concentration in the outlet of the fourth tank.

### 7.3 Manipulated, Disturbance, and Output Variables

The input variables of the process are:

- **Feed Medium Flow Rate** ($F_0$ [m³/h]): This is the main input variable of the process. It is a manipulated input. The universe of discourse of this variable is the interval [50,150]. This interval is conservative in terms of the economic and operational viability of the plant. It represents the upper and lower bounds for the feed flow and comprises the limitations related to valve operation and tank volumes as well.

- **TRS Concentration in the Feed Medium** ($S_0$ [kg/m³]): The nominal value of this variable under real operational conditions is 180kg/m³. However, since it depends on the sugar cane used, it is important to take into account possible disturbances of up to +/− 5% around this value. In this case, this variable becomes a measurable disturbance belonging to the interval [170,190].

The output variable according to the control objective discussed in session 7.2 is:

- **Outlet Ethanol Concentration in the Fourth Tank** ($P_4$ [kg/m³]).

### 8 OBF Hierarchical Fuzzy Modeling of the Ethanol Production Process

#### 8.1 Data Generation and Sampling

The strategy of dealing with a validated simulator of the actual process makes it possible to generate identification data as desired. Thus, a representative data set containing the input and output signals of the process related to 2 months (1440 hours) of its simulated operation was generated. In these data, the manipulated input $F_0$ is a sequence of steps, each of which with period of 10h (long enough so that the process can nearly reach the steady state if $S_0$ is kept constant) and random amplitude uniformly distributed within the operational interval [50,150]. Similarly,
the disturbance input \( S_0 \) is also a sequence of steps with period of 20h and random amplitude uniformly distributed within the interval [170,190], i.e., about \( \pm 5\% \) around its nominal value (\( S_0 = 180\text{kg/m}^3 \)).

The data set was sampled with a sampling period of \( T = 30\text{min} \) [21]. This value is large enough to avoid numerical problems during the model estimation phase and gives rise to a set of 2880 discrete-time data which will be used in the sequel, one half intended for the estimation of an OBF hierarchical model of the process and the other half intended for the validation of this model.

### 8.2 Structure Selection

Structure here refers to three distinct concepts: 

- \( i) \) The Laguerre pole associated with each input of the OBF model;
- \( ii) \) The hierarchical order of the Laguerre states as inputs to the hierarchical fuzzy model;
- \( iii) \) The internal structural configuration of the hierarchical fuzzy model itself.

The Laguerre poles were selected based on the time response of the process. Specifically, preliminary experiments showed that its dominant dynamic with respect to the manipulated input \( F_a \) can roughly be approximated by a real discrete pole of 0.65. Analogously, the dominant dynamic with respect to the disturbance input \( S_0 \) can roughly be approximated by a real discrete pole of 0.9. For this reason, Laguerre poles \( p = 0.65 \) and \( p = 0.9 \) were selected to implement the state-space dynamics related to the inputs \( F_a \) and \( S_0 \) of the OBF model, respectively.

The hierarchical order of the Laguerre states was defined as in figure 2-a in such a way that each state associated with \( F_a \) is placed on a hierarchical level just above the level of the respective state associated with \( S_0 \). The reason is that the model requires a more accurate representation of the dynamics related to \( F_a \) since this is the manipulated variable to be used to control the process.

Once the Laguerre poles and the hierarchical order have been defined, the model can be completely specified in terms of the number \( n \) of Laguerre states per input variable (\( F_a \) and \( S_0 \)) as well as the number \( c \) of (Gaussian) fuzzy sets per Laguerre state. The value \( c = 3 \) is adopted. This value is quite common in the context of fuzzy logic applications [26] and, combined with an amount of \( n = 2 \) Laguerre states per input variable, is capable of providing a satisfactory model of the ethanol production process, as will be shown in section 8.3.

It is important to remark that all the configuration parameters specified above could optionally be included into a GA (section 4.2) to be optimized in an automated manner together with the remaining parameters of the model. This strategy would, however, increase significantly the complexity and the computational requirements of the GA.

### 8.3 Model Estimation and Validation

As mentioned in section 4.2, a hybrid two-phase algorithm is used to optimize the remaining unknown parameters of the model, which are composed of the vectors \( \Omega(\cdot) \) in equation (8) as well as the centers \( (\theta(\cdot) \text{ and } \phi(\cdot)) \) and widths \( (\sigma(\cdot) \text{ and } \phi(\cdot)) \) of the fuzzy sets in (12). In the first phase of the algorithm, an approximate model is obtained using a GA (detailed in appendix C). Since high accuracy is not required in this phase, only the vectors \( \Omega(\cdot) \) (randomly
initialized within \([-1, 1]\), as described in section 4.3) need to be optimized. Indeed, the homogeneous distribution of the fuzzy sets described in section 4.3 is itself a suitable preliminary solution with respect to their parameters (centers and widths).

Previous experiments with the GA, using simple testing systems, showed that it is not very sensitive to its settings if a representative amount of individuals, such as two or three times the number of parameters to be optimized, is used in the population. For instance, an adequate tuning is: \(P_r = 0.95, P_m = 0.1, N_c = 1\) and \(N_d = 20\%\) of \(N_I\). In the present application, the vectors \(\Omega(\cdot)\) to be optimized by the GA represent a total of 27 parameters. So, an amount of \(N_I = 100\) individuals is selected.

The GA is executed for \(N_g = 100\) generations, which represent the first phase of the hybrid algorithm. In the second phase, the final model generated by the GA is fine-tuned for 100 training epochs using the Fletcher and Reeves algorithm. In this phase, the total number of optimized parameters is 63 (27 rules), since it includes the fuzzy sets of the model. Note that a nonhierarchical model with the same number of inputs (i.e., 4) and fuzzy sets per input (i.e., 3) would require the optimization of 105 parameters (81 rules). It is worth remarking that in case the number of fuzzy sets per input and/or the number of inputs (Laguerre states) were larger, this difference would be even more expressive.

The evolution of the Mean Squared Error (MSE) between the output data of the process and the synthetic data of the model is presented in figure 4. In this figure, the first 100 iterations refer to the GA and the remaining ones refer to the conjugate gradient algorithm, which is responsible for reducing the validation error from 0.225 at iteration 100 to 0.089 at the final iteration, i.e., a reduction of about 60\%. The term “synthetic data” denotes a kind of simulation in which the model does not rely on past values of the system output to predict its future behavior. This is an intrinsic characteristic of the OBF models since their state-space formulation given by equations (1) and (2) does not depend on past values of the output (auto-regressive terms). It depends on the input signal only. In this sense, this simulation is quite different from the well-known one-step-ahead prediction.

The simulation performance of the final model using the validation data is shown in figure 5 (MSE=0.089). The figure illustrates an adequate performance of the model, especially considering the long prediction horizon involved in the simulation.
9 Predictive Control of the Ethanol Production Process

The control scheme is implemented as described in section 6. The constrained variational GPC algorithm is used [15, 10] so that the manipulated variable $F_a$ is maintained within its operational interval, that is, $[50, 150]$. Before closing the control loop, it is advisable that the manipulated input $F_a$ be kept constant during a transitory period necessary to the convergence of the Laguerre states from their initial values (set equal to zero) to their correct values. The initial Laguerre states are usually unknown since they depend on the past history of the input signals, but they always converge to their correct values because the state-space dynamic in equation (1) is assuredly stable. In the present application, the manipulated variable is kept constant at $F_a = 100\text{m}^3/\text{h}$ for a period of 10 hours reserved to the convergence of the states and beyond which the automatic control system is enabled.

Initially, it is assumed that there is no disturbance acting on the process, i.e., $S_0 = 180\text{kg}/\text{m}^3$. In this case, the controller parameters are tuned as $N_1 = 1$ (initial horizon), $N_u = 1$ (control horizon), $N_y = 4$ (prediction horizon) and $\lambda = 0.0001$ (weighting factor for the control signal). Figure 6 presents the closed-loop simulation of the ethanol production process and shows that the proposed scheme, based on the linearization of the OBF hierarchical fuzzy model, is able to control this nonlinear process with satisfactory performance.

Figure 6: Reference and closed-loop process output $P_4$ [kg/m$^3$] (above); Manipulated variable $F_a$ [m$^3$/h] (below).
For the sake of illustration, figure 7 displays the model prediction throughout the experiment.

![Figure 7: Process output $P_d$ [ kg/m$^3$ ] (solid line) and synthetic data of the model (dashed line).](image)

In what follows, the presence of disturbances acting on the process is considered. In the first experiment, $S_0$ is kept constant during each sampling period, but its value changes randomly at each new sampling instant in order to simulate the disturbance effects. These changes are governed by a Gaussian probability distribution which expresses the underlying statistical characteristics of this disturbance, i.e., its greatest probability of taking its nominal value or its neighboring values most of the time. To accomplish this, the respective Gaussian probability distribution function was centered on the nominal value $S_0 = 180\text{kg/m}^3$ and its standard deviation was set as $1/6$ of the operational interval $[170, 190]$ in such a way that the amplitudes of the randomly generated steps belong to this interval with a probability of approximately 99%. In this case, the controller is tuned as $N_1 = 1$, $N_u = 1$, $N_y = 3$ and $\lambda = 0.001$. Figure 8 presents the closed-loop simulation of the process. Note that, although the manipulated variable seems to have a high-frequency profile, the sampling interval is $T = 30\text{min}$. This interval is much longer than the dynamics of the corresponding flow control valve, which can thus be neglected without loss of accuracy.

Figure 8 shows that the controller is capable of adequately controlling the ethanol concentration even in the presence of the disturbance $S_0$, which strongly affects the process dynamics [17, 14]. However, the performance is not as good as if $S_0$ was constant, although this variable has been included into the model. Indeed, the behaviors of the controlled and manipulated variables are not as smooth as those illustrated in figure 6 (the maximum overshoot, for instance, is about 6.7% in figure 6 and about 19.2% in figure 8). The reasons are twofold. First, the model is not an exact representation of the process, as it can be seen in figure 7. Second, its linearization at each operating point introduces an additional approximation error into the control law.

Besides the same disturbance $S_0$ considered above, it is now assumed the presence of noise in the output sensor. Specifically, a relatively strong additive white noise, uniformly distributed between $[-0.1, 0.1]$, is added to the output signal before this signal is measured by the model/controller. The controller is tuned as $N_1 = 1$, $N_u = 1$, $N_y = 2$ and $\lambda = 0.01$. Figure 9 presents the closed-loop simulation of the process. As expected, the disturbance worsened the performance of the control scheme, although the controller is still capable of controlling the process.

These experiments highlight to which extent the use of appropriate auxiliary systems can improve the performance of the controlled process. In the present context, these systems refer to accurate instrumentation devices as well as an adequate previous treatment of the feed medium so that its relevant chemical characteristics (including
Figure 8: Reference and closed-loop process output $P_4$ [kg/m$^3$] (top); Manipulated variable $F_a$ [m$^3$/h] (middle); Disturbance variable $S_0$ [kg/m$^3$] (bottom).

Figure 9: Reference and closed-loop process output $P_4$ [kg/m$^3$] (above); Manipulated variable $F_a$ [m$^3$/h] (below).

16
S(t) can be kept homogeneous inside the storage tank and, accordingly, approximately constant with time.

10 Conclusions and Perspectives

An Orthonormal Basis Function (OBF) hierarchical fuzzy model has been proposed and applied to modeling of an industrial plant for ethyl alcohol (ethanol) production. This model has been identified using a hybrid data-driven method which consists of the use of a genetic algorithm to generate an adequate initial model to be fine-tuned by a conjugate gradient algorithm. Simulations have shown that the resulting model can adequately represent the process with a reduced number of free design parameters. The adequate performance of this model in long-range horizon predictions is reflected in satisfactory results presented by a model-based predictive controller which has also been implemented and applied to the same process. The proposed control scheme is based on the use of the well-known GPC algorithm combined with a linearization of the OBF hierarchical fuzzy model at each sampling instant. In future works the authors intend to extend the proposed control scheme by adopting some adaptive strategy based on the on-line estimation of the model parameters, e.g., using the Extended Kalman Filter.

Appendix A: Model Equivalence

Equation (5) can be rewritten explicitly as

\[ \hat{y} = \sum_{l_1=1}^{c_1} \cdots \sum_{l_n=1}^{c_n} \Psi_{l_1 \ldots l_n} \Omega_{l_1 \ldots l_n} \]  

(15)

where \( \Psi_{l_1 \ldots l_n} \) and \( \Omega_{l_1 \ldots l_n} \) denote elements of \( \Psi \) and \( \Omega \), respectively, in such a way that \( \Psi = [\Psi_{1 \ldots 1} \cdots \Psi_{c_1 \ldots c_n}]^T \) and \( \Omega = [\Omega_{1 \ldots 1} \cdots \Omega_{c_1 \ldots c_n}]^T \). Each element \( \Psi_{l_1 \ldots l_n} \) can be written from equations (6) and (7) as

\[ \Psi_{l_1 \ldots l_n} = X_{l_1} (x_1) X_{l_2} (x_2) \cdots X_{l_n} (x_n) \]  

(16)

Whenever Gaussian fuzzy sets are used, equation (16) can be rewritten from (12) as

\[ \Psi_{l_1 \ldots l_n} = \exp \left( -\frac{(x_1 - \theta_{l_1})^2}{\sigma_{l_1}^2} \right) \cdots \exp \left( -\frac{(x_n - \theta_{l_n})^2}{\sigma_{l_n}^2} \right) \]  

(17)

Since the product of \( n \) unidimensional Gaussian functions is an \( n \)-dimensional Gaussian function, equation (17) results in \( \Psi_{l_1 \ldots l_n} = \exp \left( - (x - \Theta_{l_1 \ldots l_n})^T \Lambda^{-1} (x - \Theta_{l_1 \ldots l_n}) \right) \), where \( x = [x_1 \cdots x_n]^T \), \( \Theta_{l_1 \ldots l_n} = [\theta_{l_1} \cdots \theta_{l_n}]^T \) and \( \Lambda = \text{diag}(\sigma_{l_1}^2, \cdots, \sigma_{l_n}^2) \). From the above equations it follows that the model output is given by a weighted sum of multivariate Gaussian functions, which is precisely the architecture of an RBF neural network [9]. In this case, the synaptic weights between the hidden layer and the output (summation) node are given by the parameters \( \Omega (\cdot) \).
Appendix B: Derivatives

The derivatives of the output of the hierarchical model with respect to its inputs can be derived by applying the chain rule to equations (8) through (11). They are given by:

\[
\frac{\partial \hat{y}}{\partial x_h} = \lambda_{h-1} \sum_{i=1}^{c} \left[ \left( \sum_{j=1}^{c} \Omega_{(h-1)(i-c+j)} \hat{y}_{(h-2)j} \right) \frac{\partial X_h(x_h)}{\partial x_h} \right]
\]

for \( h = 3, \cdots, n \) and

\[
\frac{\partial \hat{y}}{\partial x_2} = \lambda_{1} \sum_{i=1}^{c} \left[ \left( \sum_{j=1}^{c} \Omega_{1(i-c+j)} X_1 j \right) \frac{\partial X_2(x_2)}{\partial x_2} \right]
\]

\[
\frac{\partial \hat{y}}{\partial x_1} = \lambda_{1} \sum_{i=1}^{c} \left[ \left( \sum_{j=0}^{c-1} \Omega_{1(i+1)(j+1)} X_2 j+1 \right) \frac{\partial X_1(x_1)}{\partial x_1} \right]
\]

where \( X_{hj} \triangleq X_h(x_h), \hat{Y}_{hj} \triangleq Y_{hj}(\hat{y}_h) \), and \( \lambda(\cdot) \) is the following recursive function:

\[
\lambda_q = \lambda_{q+1} \sum_{j=0}^{c-1} \left[ \left( \sum_{l=0}^{c-1} \Omega_{(q+1)(l+1)(j+l+1)} X_{q+2} j+l+1 \right) \frac{\partial Y_{q} j}{\partial \hat{y}_q} \right]
\]

with \( q = 1, \cdots, n-2 \) and \( \lambda_{n-1} = 1 \). The terms \( \partial X_h(x_h) / \partial x_h \) and \( \partial Y_{q} j / \partial \hat{y}_q \) depend on the specific shape of the fuzzy sets \( X(\cdot) \) and \( Y(\cdot) \).

Appendix C: Genetic Algorithm

The GA considered in the present paper uses a direct real-valued coding to encode the free design parameters of the model into a chromosome/individual. In this strategy, which is often adopted when there is a large number of variables (parameters) to be optimized, the chromosome/individual is simply a vector where each real-valued element (gene) stands for an unknown parameter of the model [22]. The fitness criterion is set as \( F = 1/J \), where \( J \) is given in (13), in such a way that the more accurate the model the largest its fitness. The stopping criterion is selected simply as a fixed number of generations to be executed. The iterative algorithm is described below:

1. (Settings and Initialization) Select the GA parameters, i.e., the probability of reproduction \( (P_r) \), the probability of mutation \( (P_m) \), the number of individuals for elitism \( (N_e) \), the number of individuals for diversity \( (N_d) \), the total number of individuals in the population \( (N_i) \), and the desired number of generations \( (N_g) \). Set \( k = 0 \) and generate the initial population \( P(0) \) with \( N_i \) individuals. Each individual should be generated according to the parameter initialization guidelines presented in section 4.3.

2. (Stopping Condition) Stop if \( k > N_g \). Otherwise, go to the next step.

3. (Fitness) Compute the fitness \( F(I_j) \) of each individual \( I_j \) \( (j = 1, \cdots, N_i) \) of the current population \( P(k) \).
4. **(Probability of Selection)** Compute the probability of selection $P_s(I_j)$ of each individual $I_j$ ($j = 1, \cdots, N_I$) of the current population as $P_s(I_j) = F(I_j)/\sum_{l=1}^{N_I} F(I_l)$, in such a way that $P_s(I_j)$ is proportional to $F(I_j)$.

5. **(Elitism and Diversity)** Take those $N_e$ individuals with the best fitnesses and include a copy of them into an auxiliary population $\bar{P}(k)$. Do the same with those $N_d$ individuals having the smallest degree of similarity (Euclidean distance) with the individual corresponding to the best fitness.

6. **(Selection and Reproduction)** Execute the substeps below until $N_I - N_e - N_d$ new individuals are obtained to complete the auxiliary population $\bar{P}(k)$:
   - **(Selection)** Take two individuals out the current population $P(k)$ according to the probabilities of selection computed in step 4.
   - **(Reproduction)** Select a value $\xi \in [0, 1]$ at random. If $\xi > P_r$, then simply include a copy of the individuals selected in the previous step into the auxiliary population $\bar{P}(k)$. Otherwise, apply crossover to these individuals (parents) and include the two resulting individuals (offspring) into $\bar{P}(k)$. The (single point) crossover operation works as follows: Select a position of the chromosome at random. Take this position as a reference and assemble the new individuals using the left-hand portion of one parent and the right-hand portion of the other parent, and vice-versa.
   - Return the two original individuals (parents) to the current population $P(k)$.

7. **(Mutation and Generation Update)** Apply mutation to every individual of $\bar{P}(k)$, except the best one, as follows: Select a value $\xi \in [0, 1]$ at random for each gene of the individual. If $\xi \leq P_m$, then add a value $\delta$ to the current value of that gene. Otherwise, do nothing. The value $\delta$ is selected randomly, with Gaussian probability distribution of mean zero and variance equal to the variance of the corresponding gene with respect to the different individuals of $P(k)$. A lower bound for the variance is used to avoid early convergence of the algorithm and numerical troubles as well. Finally, set $P(k+1) = \bar{P}(k)$, $k = k + 1$ and go back to step 2.

**References**


---

$^3$1% of the respective variance associated with the initial population.


