Adaptive fuzzy identification and predictive control for industrial processes

Jérôme Mendes, Rui Araújo, Francisco Souza
Institute for Systems and Robotics (ISR-UC), and
Department of Electrical and Computer Engineering (DEEC-UC),
University of Coimbra, Pólo II, PT-3030-290 Coimbra, Portugal
jermendes@isr.uc.pt, rui@isr.uc.pt, fasouza@isr.uc.pt

Abstract

This paper proposes a method for adaptive identification and control for industrial applications. The learning of a T-S fuzzy model is performed from input/output data to approximate unknown nonlinear processes by a hierarchical genetic algorithm (HGA). The HGA approach is composed by five hierarchical levels where the following parameters of the T-S fuzzy system are learned: input variables and their respective time delays, antecedent fuzzy sets, consequent parameters, and fuzzy rules. In order to reduce the computational cost and increase the algorithm's performance an initialization method is applied on HGA. To deal with nonlinear plants and time-varying processes, the T-S fuzzy model is adapted online to maintain the quality of the identification/control. The identification methodology is proposed for two application problems: (1) the design of data-driven soft sensors, and (2) the learning of a model for the Generalized predictive control (GPC) algorithm. The integration of the proposed adaptive identification method with the GPC results in an effective adaptive predictive fuzzy control methodology. To validate and demonstrate the performance and effectiveness of the proposed methodologies, they are applied on identification of a model for the estimation of the flour concentration in the effluent of a real-world wastewater treatment system; and on control of a simulated continuous stirred tank reactor (CSTR) and on a real experimental setup composed of two coupled DC motors. The results are presented, showing that the developed evolving T-S fuzzy model can identify the nonlinear systems satisfactorily and it can be used successfully as a prediction model of the process for the GPC controller.

Keywords:
Fuzzy identification, Hierarchical genetic algorithm, Predictive fuzzy control, Fuzzy generalized predictive control.

1. Introduction

Identification and model based control of industrial processes have been a focus in many engineering approaches that require accurate process models, such as soft sensors design or model predictive controller design, respectively.

Data-driven soft sensors (DDSS) are inferential models that use on-line available sensor measures for on-line estimation of variables which cannot be automatically measured at all, or can only be measured at high cost, sporadically, or with large time delays (e.g., laboratory analysis). These models are based on measurements which are recorded and provided as historical data. The models themselves are empirical predictive models. They are valuable tools to many industrial applications such as refineries, pulp and paper mills, wastewater treatment systems (Fortuna et al., 2006).

Model predictive control (MPC) is a popular control approach that is based on the use of a model of the process to predict the future behavior of the system over a prediction horizon. MPC is widely used in practice due to its high-quality control performance (Camacho & Bordons, 1998).

A weak point common in both methodologies, soft sensors design and model predictive control, it is their assumption of the knowledge of an accurate model of the process to be identified/controlled. The majority of physical systems contain complex nonlinear relations, which are difficult to model with conventional techniques. This assumption may present problems because many complex plants are difficult to be mathematically modelled based on physical laws, or have large uncertainties and strong nonlinearities. Several types approaches to modelling nonlinear plants can be considered to be used in DDSSs and MPCs. A suitable option, is the application of models based on fuzzy logic systems. This is theoretically supported by the fact that fuzzy logic systems are universal approximators (Wang & Mendel, 1992; Kosko, 1994). Takagi-Sugeno (T-S) fuzzy models (Takagi & Sugeno, 1985) are suitable to model a large class of nonlinear systems and have gained much popularity because of their rule consequent structure which is a mathematical function.

Some biologically inspired algorithms, such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), have been proved efficient in optimization problems. GAs are search methods that are inspired on natural evolution, selection, and survival of the fittest in the biological world. PSO is inspired in the social behavior of living organisms such as bird flocking or fish schooling. ACO is a multiagent approach that simulates the foraging behavior of ants. All algorithms could be used to design the T-S fuzzy models. However, because GAs provide a robust search with the ability to find near optimal solutions in complex and large search spaces ((Cordón et al., 2001), (Herrera, 2008)), GAs are a useful soft computing technique to design T-S fuzzy models. Other advantages in the use of GAs in the design of T-S fuzzy models are: GAs are simple to implement, they have the possibility of using different types of solution encoding (e.g., for different parts of the model), and they are adaptive, which means that they have the ability to learn, accumulating relevant knowledge to solve optimization problems (Kasabov, 1996).

In off-line training algorithms the discrete-time T-S fuzzy model can be obtained from input-output data collected from a plant. However, such collected dataset can be limited and the obtained T-S fuzzy model may not provide adequate accuracy in parts or the whole operating areas of the plant. Moreover, the behavior and model of the plant may be changing over time. This motivates the introduction of adaptive methodologies to solve these problems.

In (Han et al., 2012) a self-organizing radial basis function neural network model predictive control method is proposed for controlling the dissolved oxygen concentration in a wastewater treatment process. In (Wu et al., 2012) a GPC strategy with closed-loop model identification for burn-through point control in the sintering process is proposed. In (Li et al., 2010) a novel fuzzy-modeling approach is proposed, which it is able to determine the right number of rules automatically. A fuzzy c-regression model clustering algorithm is applied to identify the premise parameters, and afterwards the orthogonal least squares method is exploited to identify the consequent parameters. In (Cazarez-Castro et al., 2010) a hybrid architecture, which combines Type-1 or Type-2 fuzzy logic system and genetic algorithms for the optimization of the membership function parameters is presented. In (Kayadelen, 2011) the potential of genetic expression programming and adaptive neuro-fuzzy computing paradigm is studied to forecast the safety factor for liquefaction of soils. In (Hung & Lin, 2012) it is developed a novel evolutionary algorithm named the partial solutions consideration based self-adaptive evolutionary algorithm (PSC-SEA) to adjust the parameters of a neuro-fuzzy network.

The methods of (Han et al., 2012; Wu et al., 2012; Li et al., 2010; Cazarez-Castro et al., 2010; Kayadelen, 2011; Hung & Lin, 2012) have the limitation of not being able to perform automatic selection of variables and delays: preselection is performed. The selection of the most adequate input variables *and* respective time delays is crucial since the use of the correct variables with the correct delays can lead to better prediction accuracy because they can contain more information about the output than incorrect variables and/or variables with incorrect delays (Souza et al., 2010).

An approach using methods for learning T-S fuzzy models is proposed by (Mendes et al., 2012): a hierarchical evolutionary approach with five levels to optimize the parameters of T-S fuzzy systems is introduced. In the first level, the input variables and respective delays are chosen with the goal of attaining the highest possible prediction accuracy of the T-S fuzzy model. The selection of variables and delays is performed jointly with the learning of the fuzzy model, which increases the global optimization performance. The second level encodes the membership functions. The individual rules are defined at the third level. The population of the set of rules is defined in the fourth level, and a population of fuzzy systems is treated at the fifth level. The least squares method is used to determine the parameters of the rule consequents. The present paper proposes an improvement over the previous work (Mendes et al., 2012).

The main advancements of is this work in comparison with (Mendes et al., 2012) are (1) the application of an initialization method on the hierarchical evolutionary approach, (2) the use of an adaptive approach of the fuzzy consequent parameters, and (3) the integration of the T-S fuzzy model learned by the proposed identification method into an adaptive fuzzy GPC controller. GAs are usually initialized with random population elements. This sort of approach increases the tuning/search difficulty of the GA, since a set of totally random populations can lead to a very exhausting optimality search, requiring more iterations to attain convergence. Therefore, in order to reduce the computational cost and increase the algorithm's performance, an initialization method is applied. This work uses an initialization method based on a fuzzy c-means (FCM) clustering algorithm (Celikyilmaz & Trksen, 2009; Dovžan & Škrjanc, 2011). Another characteristic of the proposed methodology is that, when dealing with nonlinear plants, time-varying processes, disturbances or varying operating regions and parameters of the model, the fuzzy model adapts itself to new process conditions in order to maintain the quality of the identification/control. Other small modifications are introduced on levels 2 and 5. On level 2, only gaussian membership functions are used, and on level 5 a different t-norm is used.

To validate and demonstrate the performance and effectiveness of the proposed algorithm, it is applied on an identification problem, and on two control problems. First, a nonlinear system identification application problem is analyzed and quantitatively compared with the work (Mendes et al., 2012): the estimation of the flour concentration in the effluent of a real-world wastewater treatment plant. Then, the performance of the proposed adaptive predictive fuzzy identification and control methodology is demonstrated on two setups: a simulated CSTR plant, and a real-world experimental setup composed of two coupled DC motors. Moreover, the identification performance is quantitatively compared with two adaptive approaches: a recursive partial least squares (RPLS) ((Dayal & MacGregor, 1997)) and a recently proposed incremental local learning soft sensing algorithm (ILLSA) for adaptive soft sensors ((Kadlec & Gabrys, 2011)).

The paper is organized as follows. Section 2 presents nonlinear systems modeling by T-S fuzzy models, and a method for nonlinear systems modeling using fuzzy c-means clustering algorithm. The hierarchical genetic fuzzy system proposed in this paper is described in Section 3. In Section 4 a brief overview of the fuzzy GPC is presented. In Section 5, results of the proposed identification and control methodology are presented and analyzed. Finally, Section 6 makes concluding remarks.

2. Fuzzy c-Means clustering

This Section presents the modelling of the T-S fuzzy model and an initialization method to reduce the computational cost and increase the performance of the GAs. As an initialization method, the fuzzy c-means clustering algorithm (Celikyilmaz & Trksen, 2009), (Dovžan & Škrjanc, 2011) is employed on the T-S fuzzy model learning methodology.

2.1. Modelling using T-S fuzzy models

Takagi-Sugeno fuzzy models with simplified linear rule consequents are universal approximators capable of approximating any continuous nonlinear system (Ying, 1997). For more details about T-S fuzzy models, references Ref. (Takagi & Sugeno, 1985; Wang, 1997), are recommended. With a T-S fuzzy model, the global operation of the nonlinear system can be accurately approximated into several local affine models. In general, a nonlinear system can be described by a T-S fuzzy model defined by the following fuzzy rules:

$$
\begin{aligned}
R_i: \quad & \text{IF } x_1(k) \text{ is } A_1^i, \text{ and } \dots \text{ and } x_n(k) \text{ is } A_n^i \\
& \text{THEN } y_i(k) = \theta_{i1}x_1(k) + \cdots + \theta_{in}x_n(k), \\
& \quad i = 1, 2, \dots, c, \qquad (1)
\end{aligned}
$$

where R_i $(i = 1, 2, \dots, c)$ represents the i-th fuzzy rule, c is the number of rules, $x_1(k), \dots, x_n(k)$ are the input variables of the T-S fuzzy system. A_j^i $(j = 1, 2, \dots, n)$ are linguistic terms characterized by fuzzy membership functions $\mu_{A_j^i}(x_j)$ which describe the local operating regions of the plant. $\theta_{i1}, \cdots, \theta_{in}$ are model parameters of $y_i(k)$. From (1), $y(k)$ can be rewritten as

$$
\begin{aligned}
y(k) &= \sum_{i=1}^{c} \bar{\omega}^i[\mathbf{x}(k)]\mathbf{x}(k)\boldsymbol{\theta}_i, \\
&= \boldsymbol{\Psi}(k)\boldsymbol{\Theta}, \qquad (2)
\end{aligned}
$$

where for $i = 1, \dots, c$, and assuming Gaussian membership functions,

$$
\begin{aligned}
\mathbf{x}(k) &= [x_1(k), \dots, x_n(k)], && (3) \\
\mu_{A_j^i}(x_j) &= \exp\left(-\frac{(x_j - v_{ij})^2}{\sigma_{ij}}\right), && (4) \\
\bar{\omega}^i[\mathbf{x}(k)] &= \frac{\prod_{j=1}^{n} \mu_{A_j^i}(x_j)}{\sum_{i=1}^{c}\prod_{j=1}^{n} \mu_{A_j^i}(x_j)}, && (5) \\
\boldsymbol{\theta}_i &= [\theta_{i1} \dots, \theta_{in}]^T, && (6) \\
\boldsymbol{\Theta} &= \left[\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, \dots, \boldsymbol{\theta}_c^T\right]^T, && (7) \\
\boldsymbol{\Psi}(k) &= \left[\left(\bar{\omega}^1[\mathbf{x}(k)]\right)\mathbf{x}(k), \dots,\right. \\
& \quad \left.\left(\bar{\omega}^c[\mathbf{x}(k)]\right)\mathbf{x}(k)\right]. && (8)
\end{aligned}
$$

where v_{ij} and σ_{ij} represent the center and width of the membership function, respectively, which need to be defined/learned.

2.2. Fuzzy c-means

The objective of the fuzzy c-means (FCM) clustering algorithm is the partitioning of the dataset X into a predefined number of clusters, c. In the fuzzy clustering methods, the objects can belong to multiple clusters, with different degrees of membership.

Consider n samples which compose an observation l (one sample of each input variable), and they are grouped as an n-dimensional vector $\mathbf{x}_l = [x_{l1}, \dots, x_{ln}]^T$, where $\mathbf{x}_l \in \mathbb{R}^n$. A set of L observations is then denoted as

$$
\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{L1} & x_{L2} & \dots & x_{Ln} \end{bmatrix}. \qquad (9)
$$

The fuzzy partition of the set $\mathbf{X}$ into c clusters, is a family of fuzzy subsets $\{A^i \mid 1 \leqslant i \leqslant c\}$. The membership functions of these fuzzy subsets are defined as $\mu_i(l) = \mu_{A^i}(\mathbf{x}_l)$, and form the fuzzy partition matrix $\mathbf{U} = [u_{il}] = [\mu_i(l)] \in \mathbb{R}^{c\times L}$. The i-th row of the matrix $\mathbf{U}$ contains the values of the membership function of the i-th fuzzy subset A^i for all the observations belonging to the data matrix $\mathbf{X}$. The partition matrix has to meet the following conditions (Dovžan & Škrjanc, 2011): The membership degrees are real numbers in the interval $\mu_i(l) \in [0, 1], 1 \leqslant l \leqslant L$; The total membership of each sample in all the clusters must be equal to one $\sum_{i=1}^{c} \mu_i(l) = 1$; And none of the fuzzy clusters is empty, neither do any contain all the data $0 < \sum_{l=1}^{L} \mu_i(l) < L, 1 \leqslant i \leqslant c$.

FCM clustering tries to minimize the following objective function, which has a pre-defined number of clusters, c, and includes a fuzziness parameter, η:

$$
J(\mathbf{X}, \mathbf{U}, \mathbf{V}) = \sum_{i=1}^{c}\sum_{l=1}^{L} (\mu_i(l))^{\eta} d_{il}^2(\mathbf{x}_l, \mathbf{v}_i), \qquad (10)
$$

Nomenclature

$\boldsymbol{\theta}_i$	Model parameter vector of the i-th fuzzy rule.	A_j^i	Linguistic term characterized by fuzzy membership functions $\mu_{A_j^i}(x_j)$.
$\mathbf{C}_i$	Covariance matrix of the i-th fuzzy rule.	c	Number of rules.
$\mathbf{I}$	Identity matrix.	d_{il}	Euclidean distance (l^2-norm).
$\mathbf{U}$	Fuzzy partition matrix.	J_q	Fitness function of the Level q.
$\mathbf{V}$	Matrix of cluster centroid vectors.	L	Number of observations.
$\mathbf{v}_i$	Cluster centroid vectors.	n	Number of input variables.
$\mathbf{X}$	Data matrix.	N_p	Output horizon.
$\mathbf{x}$	Input variable vector.	N_u	Control horizon.
$\mathbf{x}_l$	Input variable vector at observation l.	p	p-step ahead prediction.
Δ	Difference operator.	p_m	Mutation probability.
η	Overlapping factor or the fuzziness parameter.	$r(k+p)$	Future reference trajectory.
$\hat{y}(k+p)$	An p-step ahead prediction of the system.	R_i	The i-th fuzzy rule.
$\lambda(z^{-1})$	Weighting polynomial.	$u(\cdot)$	Process input.
μ_i	Fuzzy partition of fuzzy subsets i.	v_{ij}	Center of the membership function of the i-th fuzzy rule and of the input variable j.
σ_{ij}	Width of the membership function of the i-th fuzzy rule and of the input variable j.	x_j	Input variable j of the T-S fuzzy system.
θ_{ij}	Model parameter of the i-th fuzzy rule and of the input variable j.	$y(\cdot)$	Process output.
φ_i	Forgetting factor of the fuzzy rule i.		

where $\mathbf{V}$ is a matrix of cluster centroid vectors $\mathbf{v}_i = [v_{i1}, \cdots, v_{in}]^T$, $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_c]^T \in \mathbb{R}^{c \times n}$, d_{il} is the Euclidean distance (l^2-norm) between the observation $\mathbf{x}_l$ and the cluster centroid $\mathbf{v}_i$, and the overlapping factor or the fuzziness parameter that influences the fuzziness of the resulting partition is denoted as η. The partition can range from a hard partition ($\eta = 1$) to a completely fuzzy partition ($\eta \to \infty$).

In order to find the fuzzy clusters in the dataset $\mathbf{X}$, equation (10) must be minimized. If the derivative of the objective function is taken with respect to the cluster centers $\mathbf{V}$ and to the membership values $\mathbf{U}$, then optimum membership values are calculated as follows (Dovžan & Škrjanc, 2011):

$$\mu_i(l) = \left(d_{il}^2 \sum_{q=1}^{c} \left(d_{ql}^2\right)^{1/(\eta-1)} \right)^{-1}, \tag{11}$$

where

$$d_{il}^2 = \left(\mathbf{x}_l - \mathbf{v}_i\right)^T \left(\mathbf{x}_l - \mathbf{v}_i\right), \tag{12}$$

and

$$\mathbf{v}_i = \frac{\sum_{l=1}^{L} \mu_i^{\eta}(l)\mathbf{x}_l}{\sum_{l=1}^{L} \mu_i^{\eta}(l)}. \tag{13}$$

To finalize the identification of the premise parameters, the $\boldsymbol{\sigma}_i = [\sigma_{i1}, \cdots, \sigma_{in}]^T$, $i = 1, \ldots, c$, can be easily calculated using $\mathbf{U} = [\mu_i(l)]$, as follows:

$$\sigma_{ij} = \sqrt{\frac{2 \sum\limits_{l=1}^{L} \mu_i(l)(x_{lj} - v_{ij})^2}{\sum\limits_{l=1}^{L} \mu_i(l)}}, \quad j = 1, \ldots, n. \tag{14}$$

2.3. Recursive least squares method with adaptive directional forgetting

In off-line training algorithms the T-S fuzzy model can be obtained from input-output data collected from a plant. However, such collected dataset(s) can be limited, the obtained T-S fuzzy models may not provide adequate accuracy, the system can be nonlinear and/or time-varying, and can have varying operating points and parameters of the model. Adaptive methodologies should be applied to solve these problems.

Thus, in the proposed methodology, after the learning of the antecedent parameters (Section 2.2), the consequent parameters are given by a recursive least squares (RLS) method, with the adaptive directional forgetting approach of (Kulhavý, 1987; Bobál et al., 2005) here adapted for the T-S fuzzy model.

At each iteration, l, the vector of parameter estimations

(6), is updated using

$$\boldsymbol{\theta}_i(l) = \boldsymbol{\theta}_i(l-1) + \frac{\mathbf{C}_i(l-1)\psi_i^T(l)}{1+\xi_i}\left(y_i(l) - \psi_i(l)\boldsymbol{\theta}_i(l-1)\right), \tag{15}$$

where $\psi_i(l) = \left(\bar{\omega}^i[\mathbf{x}(l)]\right)\mathbf{x}(l)$, $\xi_i = \psi_i(l)\mathbf{C}_i(l-1)\psi_i^T(l)$, $\mathbf{C}_i(l)$ is the covariance matrix of the fuzzy rule i and $y_i(l) = \left(\bar{\omega}^i[\mathbf{x}(l)]\right)y(l)$.

The covariance matrix is also updated at each iteration, l, using

$$\mathbf{C}_i(l) = \mathbf{C}_i(l-1) - \frac{\mathbf{C}_i(l-1)\psi_i^T(l)\psi_i(l)\mathbf{C}_i(l-1)}{\varepsilon_i^{-1} + \xi_i}, \tag{16}$$

where $\varepsilon_i = \varphi_i(l-1) - \frac{1-\varphi_i(l-1)}{\xi_i}$ and $\varphi_i(l-1)$ is the forgetting factor at $l-1$ iteration of the fuzzy rule i.

The adaptation performed on the forgetting factor is obtained using (Kulhavý, 1987; Bobál et al., 2005)

$$\varphi_i(l) = \frac{1}{1 + (1+\rho)\left\{ln(1+\xi_i) + \left[\frac{(\nu_i(l)+1)\gamma_i}{1+\xi_i+\gamma_i} - 1\right]\frac{\xi_i}{1+\xi_i}\right\}}, \tag{17}$$

where $\nu_i(l) = \varphi_i(l-1)(\nu_i(l-1)+1)$, $\gamma_i = \frac{(y_i(l)-\psi_i(l)\boldsymbol{\theta}_i(l-1))^2}{\tau_i(l)}$, $\tau_i(l) = \varphi_i(l-1)\left[\tau_i(l-1) + \frac{(y_i(l)-\psi_i(l)\boldsymbol{\theta}_i(l-1))^2}{1+\xi_i}\right]$, and ρ is positive constant.

The initial values of $\varphi_i(0)$, $\tau_i(0)$ and $\nu_i(0)$ should be set between zero and one.

2.4. Initialization algorithm

To construct a T-S fuzzy system of the form (2) the antecedent parameters ($\mathbf{v}_i$ and $\boldsymbol{\sigma}_i$), and the consequent parameters ($\boldsymbol{\theta}_i$) are necessary. The antecedent parameters are given by the fuzzy c-means algorithm, and the consequent parameters are given by recursive least squares method with adaptive directional forgetting.

The initialization algorithm is presented in Algorithm 1.

3. Hierarchical genetic fuzzy system

In this section is explained the hierarchical genetic algorithm that will learn all parts of the T-S fuzzy model to identify nonlinear systems. This approach is constituted by five hierarchical populations, where each population represents different species. The first level is responsible to select a set of input variables and respective time delays. The second level is constituted by all the antecedent membership functions of the T-S fuzzy system. The individual rules are treated at the third level. A set of the fuzzy rules are handled in the fourth level, and the population of fuzzy systems is evolved at the fifth level. The hierarchical architecture is illustrated in Fig. 1. The detailed description of each level is given below.

Level 1*:* the population is formed by a set of input variables and respective delays that is used in the T-S fuzzy model. The chromosome of Level 1 is represented by a binary encoding, where each allele corresponds to each input variable and respective delay (see Fig. 1). The length of

---

Algorithm 1 Initialization algorithm.

1. Obtain a dataset $\mathbf{X}$ (9) and define number of clusters c, degree of fuzziness η, and the stop conditions $\epsilon > 0$ and Max. Initialize the partition matrix $\mathbf{U}$, randomly;
2. Find initial cluster centers using (13) with the membership values of initial partition matrix $\mathbf{U}$;
3. For $t = 1, \cdots, Max$ do:
   (a) Using (11), calculate membership values at iteration t, $\mu_i^{(t)}(l)$, of each input data object $\mathbf{x}_l$ in cluster i, using the cluster center vector $\mathbf{v}_i^{(t-1)}(l)$, from iteration $(t-1)$;
   (b) Calculate cluster center of each cluster i at iteration t, $\mathbf{v}_i^{(t)}(l)$, by (13), using the membership values (11) at iteration t, $\mu_i^{(t)}(l)$;
   (c) If termination condition is satisfied, e.g., $|\mathbf{v}_i^{(t)}(l) - \mathbf{v}_i^{(t-1)}(l)| \leqslant \epsilon$, then save the last iteration of the matrices $\mathbf{U}$ and $\mathbf{V}$ and go to Step 4;
4. Compute the parameters $\boldsymbol{\sigma}_i$ using (14);
5. Compute the consequent parameters $\boldsymbol{\theta}_i$, by initializing its components to small values (e.g., 10^{-10}), and then using the recursive least squares method with adaptive directional forgetting (Section 2.3), using recursion (15) for $l = 1, \ldots, L$;

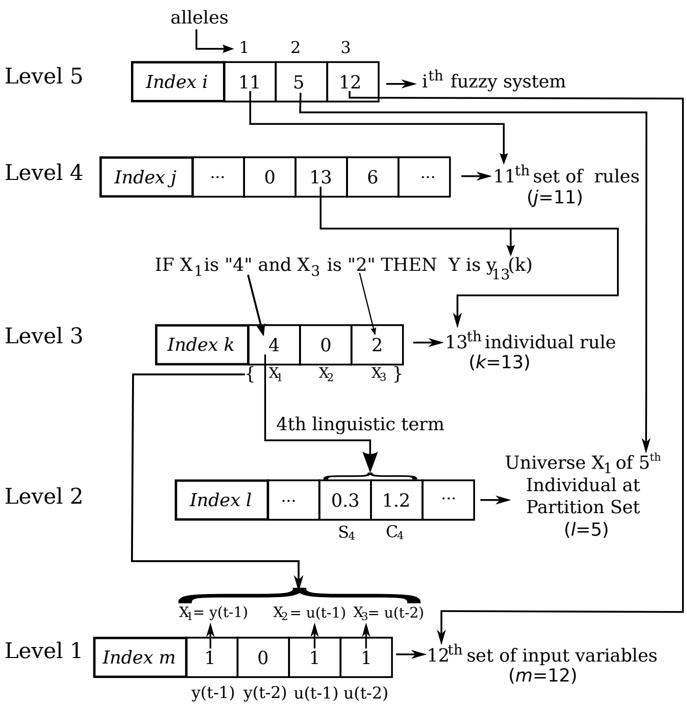


Figure 1: Encoding and hierarchical relations among the individuals of the different levels of the genetic hierarchy.

the chromosome is given by the total number of pairs of system variables and respective delays that are considered as possible candidates to be used as inputs for the T-S model. In the example of Fig. 1 the selected pairs correspond to $x_1 = y(t-1)$, $x_2 = u(t-1)$, and $x_3 = u(t-2)$.

Level 2*:* contains the representation of all antecedent membership functions. All alleles use real encoding to represent the parameters of the Gaussian membership function $\boldsymbol{\sigma}_i$ and $\mathbf{v}_i$, (14) and (13) respectively. In the example of Fig. 1 it is represented the parameters of the 4th membership function of variable x_1 on the 13th individual rule. Thus, the 4th Gaussian membership function of variable x_1 is represented by the width $S_4 = 0.3$ and the center $C_4 = 1.2$.

Level 3*:* is formed by a population of individual rules. The length of the chromosome is determined by the maximum number of antecedent variables. The chromosome is represented by integer encoding where each allele is formed by the index that identifies the corresponding antecedent membership function (defined at Level 2). Null index values indicate the absence of membership function for the corresponding variable (i.e. the absence of the variable) in the rule. In the example of Fig. 1, Level 3 of the GA hierarchy is illustrated by describing the 13th individual rule. As can be seen, in this rule x_1 is represented by its 4th membership function, x_2 is not used, and x_3 is represented by its 2nd membership function. Thus, Fig. 1 includes the illustration of the 13th individual rule which is the following:

$$R_{13}: \quad \text{IF } x_1(k) \text{ is ``4'' and } x_3(k) \text{ is ``2''} \\ \text{THEN } y_{13}(\boldsymbol{\theta}_{13}, [x_1, x_3]). \qquad (18)$$

The linguistic terms "4" and "2" are defined in Level 2 (only term "4" is illustrated at the Level 2 of Fig. 1).

Level 4*:* each individual of the population corresponds to a set of fuzzy rules, where each allele contains the index of the corresponding individual rule that is being included in the set. Null values indicate that the corresponding allele does not contribute to the inclusion of any rule to the set of fuzzy rules. The chromosome is represented by integer encoding. The length of the chromosome is determined by the maximum number of fuzzy rules. In the example of Fig. 1, Level 4 of the GA hierarchy is illustrated by the 11th set of fuzzy rules that contains the 13th and 6th individual rules, where these rules are described/represented in Level 3 of the hierarchy (but only the 13th rule is illustrated at the Level 3 of Fig. 1).

Level 5*:* each individual represents a fuzzy system. The chromosome is represented by integer encoding. The first allele represents a jth set of fuzzy rules specified at Level 4. Allele 2 contains a lth partition set individual at Level 2, and Allele 3 represents the mth set of input variables and delays at Level 1. In the example of Fig. 1, the ith fuzzy system at Level 5 uses the 11th set of fuzzy rules at Level 4, the 5th partition set of Level 2, and the 12th set of selected input variables and delays.

Algorithm 2 Proposed initialization of the hierarchical methodology.

1. Compute the antecedent membership functions parameters ($\boldsymbol{\sigma}_i$ (14) and $\mathbf{v}_i$ (13)) by the FCM (Algorithm 1);
2. Initialize populations of all levels:
    (a) Level 1: initialize the first individual with ones (i.e. to use all the input variables defined in dataset used in Algorithm 1);
    (b) Level 2: initialize all individuals with the antecedent membership functions computed in step 1;
    (c) Level 3: initialize the first c individuals with the antecedent part of the fuzzy rules which is learned by the FCM Algorithm. The way this is done is by initializing the first individual with ones, the second individual with twos, until the cth individual with c's;
    (d) Level 4: initialize the first individual with indexes of the first c individuals of Level 3;
    (e) Level 5: initialize the first individual of Level 5 with ones;
    (f) The remaining individuals of Levels 1, 3, 4, and 5 are randomly initialized;

The proposed initialization of the hierarchical methodology is presented in Algorithm 2.

The main steps to learn/improve the T-S fuzzy model parameters are presented in Algorithm 3, which also defines the fitness functions used to evaluate each individual for Levels 1 to 5. Each level of the genetic hierarchy is evolved separately as an independent genetic algorithm. However, the fitness functions of Levels 1 to 5 depend on the populations of all the levels, then evolution of each level also influences the evolution of all other levels.

As can be seen in Algorithm 3, the following genetic operators are used:

Selection: the Roulette Wheel selection method is used. The principle of roulette selection consists in a linear search of individuals through a roulette wheel, where the wheel slots are weighted in proportion to the individuals fitness values. In each generation, with the selection operator, two parents from the population are chosen for crossing.

Crossover: the Single Point crossover technique is used. The process consists of taking the two parents selected from the selection operator and producing two offspring solutions (childs) from them. For the first child, the crossover process generates a random point of crossover, R_r, and the child will receive the alleles from 1 to R_r from the first parent and the rest of the alleles are received from second parent. The second child is constituted by the remaining alleles of the parents.

Mutation: is used to maintain the diversity of the population and to prevent the algorithm from being trapped in local minima. After crossover, each of the two chromo-

Algorithm 3 Hierarchical methodology algorithm.

1. Set Generation ← 1;
2. Initialize populations of all levels with Algorithm 2;
3. Compute the consequent parameters of the T-S fuzzy model for all individuals at Level 5 by initializing the components of $\boldsymbol{\theta}_i$ to small values (e.g., 10^{-10}), and then using the recursive least squares method with adaptive directional forgetting (Section 2.3), using recursion (15) for $l = 1, \ldots, L$;
4. Compute the fitness of each individual, from Level 5 to Level 1:
   (a) Level 5: the fitness function is $J_5(i) = 1/MSE(i)$, where $MSE(i) = \frac{1}{L}\sum_{l=1}^{L}(y_k - \hat{y}_k)^2$ is the mean square error of the ith fuzzy system, $\hat{y}_k$ is the predicted output pattern and y_k is the target output pattern;
   (b) Level 4: the fitness function is $J_4(j) = \max(J_5(b), \cdots, J_5(d))$, where $b, \cdots, d$ are the fuzzy systems at Level 5 that contain rule-base j (set of fuzzy rules);
   (c) Level 3: the fitness function is $J_3(k) = \max(J_4(m), \cdots, J_4(p))$, where $m, \cdots, p$ are the rule-bases at Level 4 that contain individual rule k;
   (d) Level 2: the fitness function is $J_2(l) = \max(J_5(x), \cdots, J_5(z))$, where $x, \cdots, z$ are the fuzzy systems at Level 5 that contain partition set l;
   (e) Level 1: the fitness function is $J_1(m) = \max(J_5(e), \cdots, J_5(h))$, where $e, \cdots, h$ are the fuzzy systems at Level 5 that contain the selection number m of inputs and delays;
5. If the stop condition does not hold, do for each level:
   (a) Generation ← Generation + 1;
   (b) Apply the evolutionary operators to form a new population: selection, crossover and mutation;
   (c) Replace the current population with the new evolved population;
   (d) Return to step 3.

somes resulting from the crossover operator is subject to mutation with probability p_m. In binary-encoded chromosomes, the Flip Bit mutation technique is used, where the value of a random allele is inverted. In real and integer encoded chromosomes Uniform mutation is used, where the value of one randomly selected allele of the chromosome is replaced by a uniform random value selected between the upper and lower bounds defined for that allele.

Replacement: the Weakest Individuals replacement technique is the last operator used. It consists in taking two individuals with the weakest fitness from the old generation, and replacing them by the two new individuals that result from the application of the selection-crossover-mutation sequence of operators, in order to form the new population.

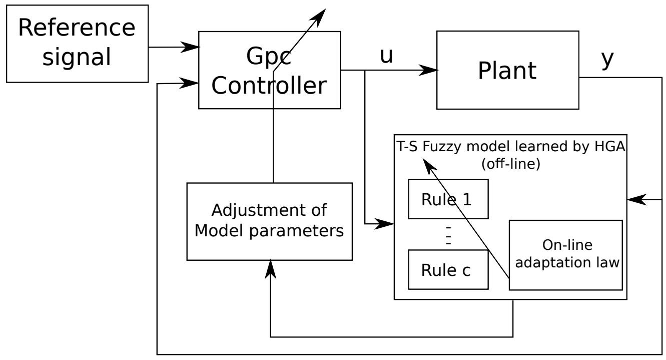


Figure 2: A generic schematic diagram of the AFGPC control architecture.

For more details about these operators see (Sivanandam & Deepa, 2007).

4. Adaptive fuzzy predictive control law

After having studied the identification algorithm (Sections 2 and 3), in this section the control algorithm is explained. A diagram of the adaptive fuzzy generalized predictive control (AFGPC) approach is presented in Fig. 2. As can be seen, the control scheme consists of the plant, the controller, and the adaptive T-S fuzzy model. The controller is composed of a model-based predictive controller that integrates a T-S fuzzy model learned off-line, according to the methodology presented in Section 3 on Algorithm 3. Then, the model parameters (consequent parameters) are adjusted on-line by an adaptation law studied in Section 2.3. The main steps of the control architecture are presented, at the end of this section, on Algorithm 4.

A large class of nonlinear processes can be represented by a model of the following type:

$$\begin{aligned} y(k) &= f[y(k-1), y(k-2), \ldots, y(k-n_y), \\ &\quad u(k-d-1), \ldots, u(k-d-n_u)], \end{aligned} \tag{19}$$

where $u(\cdot) : \mathbb{N} \to \mathbb{R}$ and $y(\cdot) : \mathbb{N} \to \mathbb{R}$ are the process input and output, respectively, $n_u \in \mathbb{N}$ and $n_y \in \mathbb{N}$ are the orders of the input and output, respectively, and $d \in \mathbb{N}$ is the time-delay of the system. In the discrete-time nonlinear SISO plant (19), $f(\cdot) : \mathbb{R}^{n_y+n_u} \to \mathbb{R}$ represents a nonlinear mapping which is assumed to be unknown. $f(\cdot)$ is approximated by a T-S fuzzy system.

For the GPC controller, system (19) can be described by a T-S fuzzy model defined by the following fuzzy rules:

$$\begin{aligned} R_i : \quad & \text{IF } x_1(k) \text{ is } A_{i1}, \text{ and } \ldots \text{ and } x_n(k) \text{ is } A_{in} \\ & \text{THEN } y_i(k) = a_i(z^{-1})y(k-1) \\ & +b_i(z^{-1})u(k-d-1), \quad i = 1, \ldots, c, \end{aligned} \tag{20}$$

where c is the number of rules, and $n = n_y + n_u$,

$$\begin{aligned} a_i(z^{-1}) &= a_{1i} + a_{2i}z^{-1} + \ldots + a_{n_y i}z^{-(n_y-1)}, \\ b_i(z^{-1}) &= b_{1i} + b_{2i}z^{-1} + \ldots + b_{n_u i}z^{-(n_u-1)}, \end{aligned} \tag{21}$$

and $\mathbf{x}(k) = [x_1(k), \ldots, x_n(k)] = [y(k-1), \ldots, y(k-n_y), u(k-d-1), \ldots, u(k-d-n_u)]$ is the vector of input variables of the T-S fuzzy system. Thus, from (20) $y(k)$ can be rewritten as

$$\begin{aligned} y(k) &= \sum_{i=1}^{c} \bar{\omega}^i[\mathbf{x}(k)] \big[a_i(z^{-1})y(k-1)+ \\ &\quad b_i(z^{-1})u(k-d-1)\big], \qquad (22)\\ &= \sum_{i=1}^{c} \bar{\omega}^i[\mathbf{x}(k)]\mathbf{x}(k)\boldsymbol{\theta}_i, \\ &= \boldsymbol{\Psi}(k)\boldsymbol{\Theta}, \qquad (23) \end{aligned}$$

where for $i = 1, \ldots, c$,

$$\bar{\omega}^i[\mathbf{x}(k)] = \frac{\prod_{j=1}^{n} A_{ij}^h(x_j)}{\sum_{i=1}^{c} \prod_{j=1}^{n} A_{ij}^h(x_j)}, \qquad (24)$$

$$\boldsymbol{\theta}_i = \left[a_{1i}, \ldots, a_{n_y i}, b_{1i}, \ldots, b_{n_u i}\right]^T, \qquad (25)$$

$$\boldsymbol{\Theta} = \left[(\boldsymbol{\theta}_1)^T, (\boldsymbol{\theta}_2)^T, \ldots, (\boldsymbol{\theta}_c)^T\right]^T, \qquad (26)$$

$$\begin{aligned} \boldsymbol{\Psi}(k) &= [(\bar{\omega}_1[\mathbf{x}(k)])\,\mathbf{x}(k), \ldots, \\ &\quad (\bar{\omega}_c[\mathbf{x}(k)])\,\mathbf{x}(k)]. \qquad (27) \end{aligned}$$

4.1. Predictive control law

It is assumed that the plant model is of the form (23), which can be rewritten as follows:

$$\bar{a}(z^{-1})y(k) = \bar{b}(z^{-1})u(k-d-1), \qquad (28)$$

where

$$\bar{a}(z^{-1}) = 1 - \bar{a}_1 z^{-1} - \ldots - \bar{a}_{n_y} z^{-n_y}, \qquad (29)$$

$$\bar{b}(z^{-1}) = \bar{b}_1 + \bar{b}_2 z^{-1} + \ldots + \bar{b}_{n_u} z^{-(n_u-1)}, \qquad (30)$$

$$\bar{a}_t = \sum_{i=1}^{c} \bar{\omega}^i[\mathbf{x}(k)]a_{ti}, \;\; t = 1, \ldots, n_y, \qquad (31)$$

$$\bar{b}_m = \sum_{i=1}^{c} \bar{\omega}^i[\mathbf{x}(k)]b_{mi}, \;\; m = 1, \ldots, n_u. \qquad (32)$$

The GPC control law is obtained so as to minimize the following cost function

$$\begin{aligned} J(k) &= \sum_{p=d+1}^{N_p} [\hat{y}(k+p|k) - r(k+p)]^2 + \\ &\quad \sum_{p=d+1}^{d+N_u} \left[\lambda(z^{-1})\Delta u(k+p-d-1|k)\right]^2, \qquad (33) \end{aligned}$$

where $\hat{y}(k+p|k)$ is an p-step ahead prediction of the system on instant k, $r(k+p)$ is the future reference trajectory, $\Delta = 1 - z^{-1}$, and $\lambda(z^{-1}) = \lambda_0 + \lambda_1 z^{-1} + \ldots + \lambda_{N_p+n_u-1} z^{-(N_p+n_u-1)}$ is a weighting polynomial. N_p and N_u are the output and control horizons, respectively. Consider the following Diophantine equation (34):

$$1 = \Delta e_p(z^{-1})\bar{a}(z^{-1}) + z^{-p} f_p(z^{-1}), \qquad (34)$$

$$e_p(z^{-1}) = 1 + e_{p,1} z^{-1} + \ldots + e_{p,p-1} z^{-(p-1)}, \qquad (35)$$

$$f_p(z^{-1}) = f_{p,0} + f_{p,1} z^{-1} + \ldots + f_{p,n_y} z^{-n_y}, \qquad (36)$$

where $e_p(z^{-1})$ and $f_p(z^{-1})$ can be obtained by dividing 1 by $\Delta\bar{a}(z^{-1})$ until the remainder can be factorized as $z^{-p} f_p(z^{-1})$. The quotient of the division is the polynomial $e_p(z^{-1})$. A simple and efficient way to obtain polynomials $e_p(z^{-1})$ and $f_p(z^{-1})$ is to use recursion of the Diophantine equation as demonstrated in (Camacho & Bordons, 1998). Polynomials $e_{p+1}(z^{-1})$ and $f_{p+1}(z^{-1})$ can be obtained from polynomials of $e_p(z^{-1})$ and $f_p(z^{-1})$, respectively. Polynomials $e_{p+1}(z^{-1})$ are given by

$$e_{p+1}(z^{-1}) = e_p(z^{-1}) + z^{-p} e_{p+1,p}, \qquad (37)$$

where $e_{p+1,p} = f_{p,0}$. The coefficients of polynomial $f_{p+1}(z^{-1})$ can be obtained recursively as follows:

$$f_{p+1,i} = f_{p,i+1} - f_{p,0}\,\Delta\bar{a}_{i+1}, \quad i = 0, \ldots, n_y - 1, \qquad (38)$$

where $f_{p,n_y} = 0$. Polynomial $g_{p+1}(z^{-1})$ is expressed as:

$$\begin{aligned} g_{p+1}(z^{-1}) &= e_{p+1}(z^{-1})\bar{b}(z^{-1}), \qquad (39)\\ &= \left[e_p(z^{-1}) + z^{-p} f_{p,0}\right]\bar{b}(z^{-1}), \qquad (40)\\ &= g_p(z^{-1}) + z^{-p} f_{p,0}\bar{b}(z^{-1}), \qquad (41) \end{aligned}$$

where the coefficients of $g_{p+1}(z^{-1})$ are given by $g_{p+1,j} = g_{p,j}$ for $j = 0, \ldots, p-1$, and

$$g_{p+1,p+i} = g_{p,p+i} + f_{p,0}\bar{b}_i, \quad i = 0, \ldots, n_u, \qquad (42)$$

where $g_{p,p+n_u} = 0$. $e_p(z^{-1})$, $f_p(z^{-1})$, and $g_p(z^{-1})$ are recursively computed for $p = d+1, \ldots, N_p$. To initialize the recursion (34), $p = d+1$, and

$$e_{d+1}(z^{-1}) = 1, \qquad (43)$$

$$\begin{aligned} f_{d+1}(z^{-1}) &= z(1 - \tilde{a}(z^{-1})), \\ &= \tilde{a}_1 + \tilde{a}_2 z^{-1} + \ldots + \tilde{a}_{n_y+1} z^{-n_y}, \qquad (44) \end{aligned}$$

where

$$\tilde{a}(z^{-1}) = \Delta\bar{a}(z^{-1}) = 1 - \tilde{a}_1 z^{-1} - \ldots - \tilde{a}_{n_y+1} z^{-(n_y+1)}.$$

Thus,

$$g_{d+1}(z^{-1}) = e_{d+1}(z^{-1})\bar{b}(z^{-1}) = \bar{b}(z^{-1}). \qquad (45)$$

Multiplying (28) by $\Delta z^p e_p(z^{-1})$ yields

$$\begin{aligned} &\Delta z^p e_p(z^{-1})\bar{a}(z^{-1})y(k) = \\ &\Delta z^p e_p(z^{-1})\bar{b}(z^{-1})u(k-d-1). \qquad (46) \end{aligned}$$

Defining

$$\begin{aligned} g_p(z^{-1}) &= e_p(z^{-1})\bar{b}(z^{-1}), \qquad (47)\\ &= g_{p,0} + g_{p,1} z^{-1} + \ldots + \qquad (48)\\ &\quad g_{p,p+n_u-1} z^{-(p+n_u-1)}, \end{aligned}$$

and substituting (34) and (47) into (46) yields

$$y(k+p|k) = f_p(z^{-1})y(k) + g_p(z^{-1})\Delta u(k+p-d-1). \qquad (49)$$

Thus, the best prediction of $y(k+p|k)$ is

$$\hat{y}(k+p|k) = f_p(z^{-1})y(k) + g_p(z^{-1})\Delta u(k+p-d-1). \qquad (50)$$

Equation (50) can be rewritten as

$$\mathbf{y}(k) = \mathbf{G}\mathbf{u}(k) + \mathbf{F}(z^{-1})y(k) + \mathbf{L}(z^{-1}), \qquad (51)$$

where

$$\mathbf{y}(k) = \begin{bmatrix} \hat{y}(k+d+1) \\ \hat{y}(k+d+2) \\ \vdots \\ \hat{y}(k+N_p) \end{bmatrix}, \qquad (52)$$

$$\mathbf{u}(k) = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix}, \mathbf{F} = \begin{bmatrix} f_{d+1}(z^{-1}) \\ f_{d+2}(z^{-1}) \\ \vdots \\ f_{N_p}(z^{-1}) \end{bmatrix}, \qquad (53)$$

$$\mathbf{G} = \begin{bmatrix} g_{1,0} & 0 & \dots & 0 \\ g_{2,1} & g_{2,0} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_p,N_p-1} & g_{N_p,N_p-2} & \dots & g_{N_p,N_p-N_u} \end{bmatrix}, \qquad (54)$$

$$\mathbf{L} = \begin{bmatrix} \left[g_{d+1}(z^{-1}) - \bar{g}_{d+1}(z^{-1})\right] z\Delta u(k-1) \\ \left[g_{d+2}(z^{-1}) - \bar{g}_{d+2}(z^{-1})\right] z^2\Delta u(k-1) \\ \vdots \\ \left[g_{N_p}(z^{-1}) - \bar{g}_{N_p}(z^{-1})\right] z^{N_p}\Delta u(k-1) \end{bmatrix},$$

$$\bar{g}_p(z^{-1}) = g_{p,0} + g_{p,1}z^{-1} + \ldots + g_{p,p-d-1}z^{d+1-p}.$$

Using (51) and considering $\lambda(z^{-1})$ to be constant ($\lambda > 0$), (33) can be rewritten as

$$J_{eq}(k) = \left[\mathbf{F}y(k) + \mathbf{G}\mathbf{u}(k) + \mathbf{L} - \mathbf{R}\right]^T \left[\mathbf{F}y(k) + \mathbf{G}\mathbf{u}(k) + \mathbf{L} - \mathbf{R}\right] + \left[\lambda\mathbf{u}(k)\right]^2, \qquad (55)$$

where

$$\mathbf{R} = \left[r(k+d+1), \ldots, r(k+N_p)\right]^T. \qquad (56)$$

To minimize $J_{eq}(k)$ the following equation is solved

$$\frac{\partial J_{eq}(k)}{\partial[\Delta u(k)]} = 0. \qquad (57)$$

By minimizing $J_{eq}(k)$ using (57), the following optimum control increment is obtained (Camacho & Bordons, 1998):

$$\mathbf{u}^*(k) = \frac{\mathbf{G}^T(\mathbf{R} - \mathbf{F}y(k) - \mathbf{L})}{\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I}}, \qquad (58)$$

where $\mathbf{I}$ is the identity matrix.

As the control signal sent to the process is the first row of $\mathbf{u}^*(k)$, the $\Delta u^*(k)$ is given by:

$$\Delta u^*(k) = \mathbf{K}[\mathbf{R} - \mathbf{F}y(k) - \mathbf{L}], \qquad (59)$$

where $\mathbf{K}$ is the first row of matrix $(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T$,

$$\mathbf{K} = \begin{bmatrix}1 & 0 & 0 & \cdots & 0\end{bmatrix}_{1\times N_u} (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T. \qquad (60)$$

Algorithm 4 summarizes the design and operation of the adaptive fuzzy generalized predictive control method.

Algorithm 4 Adaptive fuzzy generalized predictive control algorithm.

1. Design control parameters: N_p, N_u, λ and d. Design the identification parameters (ρ, φ_i, τ_i, ν_i, for all $1 \leqslant i \leqslant c$) with the same values as the ones defined in Algorithm 3;
2. Use the fuzzy rule base (input variables, respective membership functions, the fuzzy rules and the final learned model parameters) learned in Algorithm 3 and initialize $u(0)$;
3. For/using each newly arriving online sample, do:
   (a) Compute $\bar{a}(z^{-1})$ and $\bar{b}(z^{-1})$ using (29) and (30), respectively;
   (b) Compute control signal $\Delta u(k)$ with (59);
   (c) Adapt the T-S fuzzy model parameters (a_{ji} and b_{ji} of (21)) by performing one iteration of recursion (15).

5. Experiments and results

In this section simulation and real-world results are presented to demonstrate the feasibility, performance and effectiveness of the proposed T-S design methodology in identification and in control. First, a nonlinear system identification application problem is analyzed and quantitatively compared with the work of (Mendes et al., 2012), named as HGA: the estimation of the flour concentration in the effluent of a real-world wastewater treatment system, where HGA has been shown to be superior when compared with 8 other methods. Then, the performance of the identification and control is studied in two experiments: a simulated CSTR plant and a real-world control of two coupled DC motors. The identification performance is, also, quantitatively compared with two adaptive approaches: a RPLS (Dayal & MacGregor, 1997), and a ILLSA for adaptive soft sensors (Kadlec & Gabrys, 2011). In both experiments, the mutation probability is $p_m = 10\%$, a maximum of 1500 generations is used, and the population of each species is fixed: 30 individuals for each of the Levels V, IV, III, II, and 50 individuals for Level I; and the first half of the dataset is used for training and the remaining data is used for evaluation.

5.1. Application to wastewater treatment system

In this section, the performance of the proposed identification methodology is studied. Specifically, a Soft Sensor application is studied. The objective of this experiment is to estimate the flour concentration in the effluent of a real-world urban wastewater treatment plant (WWTP). The dataset of plant variables that is available for learning consists of 11 input variables, $u_1 \ldots u_{11}$, and one target output

Table 1: Variables of the wastewater treatment plant dataset.

Variables	Description
u_1	Amount of chlorine in the influent;
u_2	Amount of chlorine in the effluent;
u_3	Turbidity in the raw water;
u_4	Turbidity in the influent;
u_5	Turbidity in the effluent;
u_6	Ph in the raw water;
u_7	Ph in the influent;
u_8	Ph in the effluent;
u_9	Color in the raw water;
u_{10}	Color in the influent;
u_{11}	Color in the effluent;
y	Flour in the effluent.

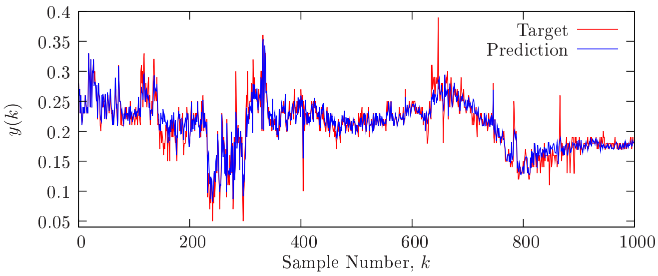


Figure 3: Modeling performance of the proposed algorithm for wastewater treatment system data set.

variable to be estimated, y. The variables correspond to physical values, such as pH, turbidity, color of the water and others. The input variables are measured on-line by plant sensors, and the output variable in the dataset is measured by laboratory analysis. The sampling interval is 2 [hours]. The plant variables are described in Table 1. To construct the dataset, the first three delayed versions of each variable were chosen as candidates for inputs of the T-S model. Specifically, the following combinations of process variables and delays are used as the candidates for inputs of the T-S model to predict $y(t)$: $[u_1(t-1), u_1(t-2), u_1(t-3), \ldots, u_{11}(t-1), u_{11}(t-2), u_{11}(t-3)]$. The number of clusters and the degree of fuzziness were chosen as $c = 13$, and $\eta = 2$, respectively.

Fig. 3 shows the predicted and desired (real) values of the target variable to be estimated, for the WWTP experiment. As can be seen in Fig. 3 the accuracy of the modeling is good.

Numerical results comparing the performance of the proposed method and the works RPLS (Dayal & MacGregor, 1997), ILLSA (Kadlec & Gabrys, 2011) and (Mendes et al., 2012) are presented in Table 2. The same parameters are used in both methods. As can be seen the largest value of the fitness function ($1/MSE$) in the test dataset is obtained with the method proposed in this paper. The proposed method selects a larger number of (variable, delay) pairs, but uses less fuzzy rules.

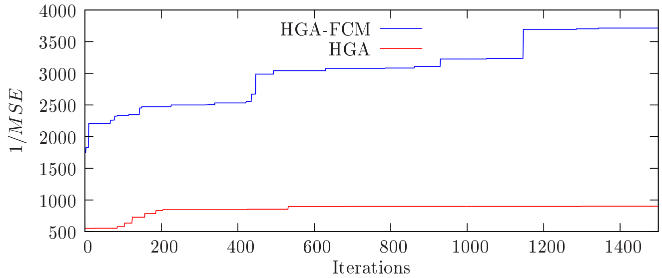


Figure 4: Evolution of the best fitness function value on Level V for all generations in the WWTP experiment.

Fig. 4 presents the evolution of the fitness functions on Level V for all generations of the proposed methodology, named as HGA-FCM (Hierarchical Genetic Algorithm with Fuzzy c-Means), and by the previous work (Mendes et al., 2012) (HGA). As can be seen, in the proposed work there is a good initialization performed by the Fuzzy c-Means algorithm that outperforms the initialization obtained by the work (Mendes et al., 2012), and afterwards, the evolution attained by the proposed hierarchical GA in also good. The proposed HGA-FCM method attains faster response and better results when compared to the results obtained by the HGA proposed in (Mendes et al., 2012) (Table 2).

5.2. Control of a continuous-stirred tank reactor (CSTR)

A Continuous Stirred Tank Reactor (CSTR) is a highly nonlinear process which is very common in chemical and petrochemical plants. In the process, a single irreversible, exothermic reaction is assumed to occur in the reactor. The CSTR for an exothermic irreversible reaction $A \to B$ is described by the following dynamic model based on a component balance for reactant A and on an energy balance (Morningred et al., 1992):

$$\frac{\partial C_A(t+d_c)}{\partial t} = \frac{q(t)}{V}\left(C_{A0}(t) - C_A(t+d_c)\right) - k_0 C_A(t+d_c)\exp\left(-\frac{E}{RT(t)}\right), \qquad (61)$$

$$\begin{aligned}\frac{\partial T}{\partial t} = {} & \frac{q(t)}{V}\left(T_0(t) - T(t)\right) - \frac{(-\Delta H)k_0 C_A(t+d_c)}{\rho C_p}\exp\left(-\frac{E}{RT(t)}\right) \\ & + \frac{\rho_c C_{pc}}{\rho C_p V} q_c(t)\left[1 - \exp\left(\frac{-hA}{q_c(t)\rho_c C_{pc}}\right)\right] \\ & \left(T_{c0}(t) - T(t)\right), \\ y(t) = {} & C_A(t), \quad u(t) = q_c(t). \end{aligned} \qquad (62)$$

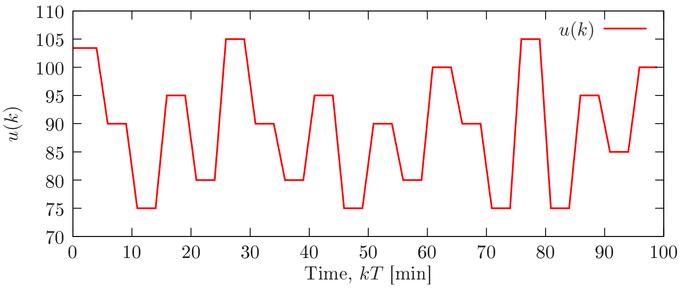
The objective is to control the measured concentration of $C_A(t)$ by manipulating the coolant flow rate $q_c(t)$. The plant variables and the respective nominal values are described in Table 3. The sampling period is assumed to be $T = 0.1\,[\text{min}]$, and the time delay is assumed to be $d_c = 5T = 0.5\,[\text{min}]$.

Table 2: Comparison results of the test dataset for the wastewater treatment system.

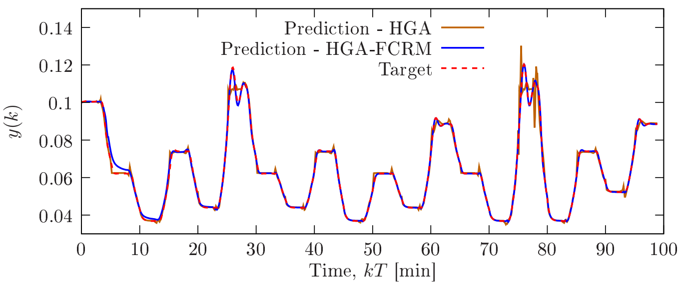
Method	Number of rules	Number of inputs	Inputs	$1/MSE$
RPLS (Dayal & MacGregor, 1997)	-	-	all candidate variables	840.9
ILLSA (Kadlec & Gabrys, 2011)	-	-	all candidate variables	1197.6
HGA (Mendes et al., 2012)	20	13	$u_3(t-1), u_4(t-2), u_4(t-3), u_6(t-3),$ $u_7(t-1), u_7(t-3), u_8(t-1), u_8(t-3),$ $u_9(t-1), u_9(t-2), u_{10}(t-2), u_{10}(t-3), u_{11}(t-2)$	901.2
HGA-FCM (proposed method)	10	17	$u_1(t-1), u_1(t-3), u_2(t-2), u_3(t-1),$ $u_3(t-2), u_5(t-2), u_6(t-2), u_6(t-3),$ $u_7(t-1), u_7(t-2), u_7(t-3), u_8(t-1),$ $u_8(t-2), u_8(t-3), u_9(t-3), u_{10}(t-1), u_{11}(t-3)$	5791.7

Table 3: Variables of the continuous stirred tank reactor (CSTR) (Morningred et al., 1992).

Variables-Description	Value
C_A - Product concentration	0.1 [mol/l]
T - Reactor temperature	438.54 [K]
q_c - Coolant flow rate	103.41 [l/min]
q - Process flow rate	100 [l/min]
C_{A0} - Feed concentration	1 [mol/l]
T_o - Feed temperature	350 [K]
T_{c0} - Inlet coolant temperature	350 [K]
V - CSTR volume	100 [l]
hA - Heat transfer term	7×10^5 [cal/min/K]
k_0 - Reaction rate constant	7.2×10^{10} [min^{-1}]
E/R - Activation energy term	1×10^4 [K]
$-\Delta H$ - Heat of reaction	-2×10^5 [cal/mol]
ρ, ρ_c - Liquid densities	1×10^3 [g/l]
C_p, C_{pc} - Specific heats	1 [cal/g/K]



(a)



(b)

Figure 5: CSTR plant: (a) control signal used to compile the process dataset, and (b) modeling performance of the proposed algorithm (HGA-FCM) and by the work (Mendes et al., 2012) (HGA).

5.2.1. *Identification*

As a first step, a dataset representative of the CSTR operation was constructed. The dataset was obtained by applying the control signal represented in Fig. 5(a): in order to represent a possible real dataset in industry, a sequence of step control signals was applied. The chosen variables for the dataset were $C_A(k-2)$, $C_A(k-4)$, $C_A(k-6)$, $C_A(k-8)$, $C_A(k-10)$, $C_A(k-12)$, $q_c(k-1)$, $q_c(k-3)$, $q_c(k-5)$, $q_c(k-7)$, $q_c(k-9)$, $q_c(k-11)$, $q_c(k-13)$), where k is the sample time. The number of clusters and the degree of fuzziness were chosen as $c = 20$, and $\eta = 2$, respectively.

Fig. 5(b) shows the comparison of the predicted values of the CSTR plant, $C_A(t)$, by the proposed methodology, HGA-FCM, and by the previous work (Mendes et al., 2012), HGA. As can be seen in Fig. 5(b) the modeling of the target variable, $C_A(t)$, is accurate and better than the one obtained by the HGA method proposed in (Mendes et al., 2012). Numerical results comparing the performance of the proposed method and the works RPLS (Dayal & MacGregor, 1997), ILLSA (Kadlec & Gabrys, 2011), and (Mendes et al., 2012) are presented in Table 4. Variables $[C_A(k-2), C_A(k-4), C_A(k-6), C_A(k-10), q_c(k-1)]$ and 20 fuzzy rules were chosen by the proposed methodology.

Table 4: Comparison results of the dataset test for the CSTR plant.

Method	$1/MSE$
RPLS (Dayal & MacGregor, 1997)	7.0871×10^4
ILLSA (Kadlec & Gabrys, 2011)	3.9814×10^5
HGA (Mendes et al., 2012)	1.0088×10^5
HGA-FCM (proposed method)	4.6195×10^5

Fig. 6 presents the evolution of the fitness functions on Level V for all generations. As can be seen, in the proposed method there is a good initialization performed by the FCM algorithm that outperforms the initialization obtained by the work (Mendes et al., 2012), and afterwards,

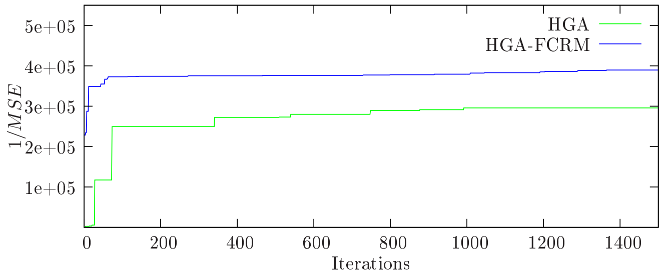


Figure 6: Evolution of the best fitness function value on Level V for all generations in the CSTR experiment.

the evolution attained by the proposed hierarchical GA in also good. The proposed HGA-FCM method attains faster response and better results when compared to the results obtained by the HGA proposed in (Mendes et al., 2012).

5.2.2. *Adaptive predictive fuzzy control of a simulated CSTR*

The model learned by HGA-FCM in Section 5.2.1 is used to initialize the prediction model of the adaptive fuzzy GPC controller.

The following controller parameters were chosen by the user: $N_p = 150$, $N_u = 1$, $\lambda = 0.05$, $d = 5$, $\rho = 0.999$, $\varphi_i = 1$, $\tau_i = \nu_i = 1 \times 10^{-9}$, for all $1 \leqslant i \leqslant c$. The reference input is

$$r(t) = \begin{cases} 0.12, & 0 < t \leqslant 5\,[\text{min}], \\ 0.07, & 5\,[\text{min}] < t \leqslant 10\,[\text{min}], \\ 0.1, & 10\,[\text{min}] < t \leqslant 15\,[\text{min}], \\ 0.08, & 15\,[\text{min}] < t \leqslant 20\,[\text{min}], \\ 0.11, & 20\,[\text{min}] < t \leqslant 15\,[\text{min}], \end{cases} \qquad (63)$$

and the load disturbance is defined as a change of the process flow rate q, where $q = 110$ for $13\,[\text{min}] \leqslant t \leqslant 17\,[\text{min}]$.

From the results presented in Fig. 7(a) and (b), it can be seen that the proposed controller is able to adequately (attain and) control the system output at the desired reference $r(t)$. When the load disturbance is applied at $13\,[\text{min}] \leqslant t \leqslant 17\,[\text{min}]$, there is an undershoot at $t = 13\,[\text{min}]$ and an overshoot at $t = 17\,[\text{min}]$ in the system response. As can be seen the controller eliminates this disturbance. By the results, it is concluded that the proposed controller methodology can control the process using only a dataset of the process to initialize the T-S fuzzy model.

5.3. *Real-world control of two coupled DC motors*

The experimental system consists of two similar DC motors coupled by a shaft (Fig. 8), where the first motor acts as an actuator, while the second motor is used as a generator and to produce nonlinearities and/or a time-varying load. The system exhibits noise, parasitic electro-magnetic

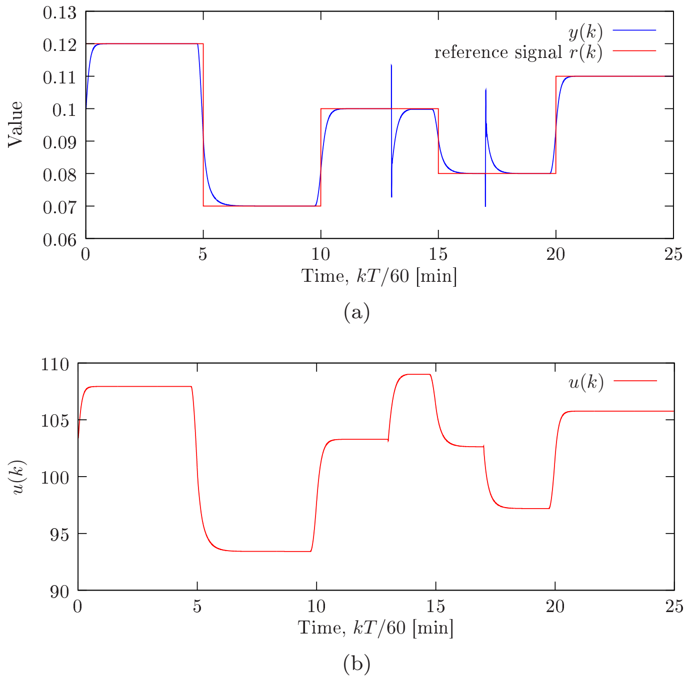


Figure 7: (a) Results of the proposed controller with HGA-FCM in presence of load disturbances in the CSTR process; and (b) respective applied command signal.

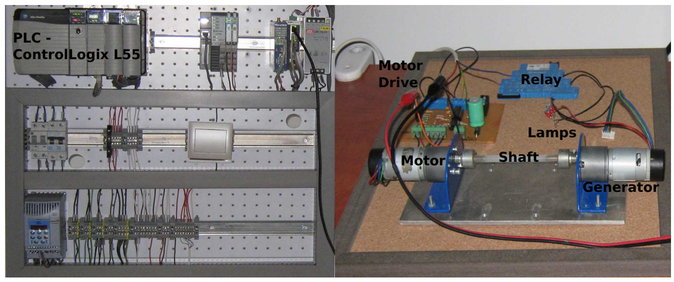
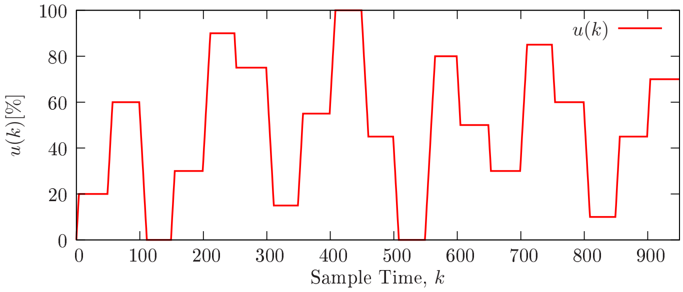


Figure 8: The experimental scheme of the DC motors.

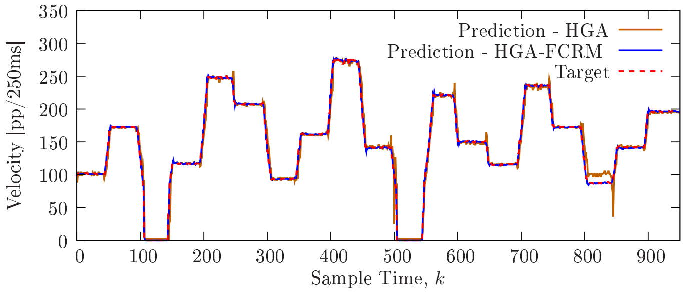
effects, friction and other phenomena commonly encountered in practical applications, that make the control task more difficult.

The voltage command signal to the DC motor is in the range of $[0, 12]$ [V]. The proposed control methodology runs on a PC that communicates by OPC[1] to a PLC[2] (ControlLogix L55 expanded with an analog I/O module for signal conditioning). The PLC provides the voltage command signal to the DC motor through the signal conditioning circuit. The velocity units are [pp/(0.25 seg)] (pulses per 250 ms). The generator has an electrical load composed of 2 lamps connected in parallel. When the lamps are connected in the generator circuit, the electrical load to the generator is increased (load resistance is decreased), and consequently the mechanical load that the

[1]OLE (Object Linking and Embedding) for Process Control.
[2]Programmable Logic Controller.



(a)



(b)

Figure 9: Motor dataset: (a) control signal used to compile the dataset on the DC motors process and (b) modeling performance of the proposed algorithm (HGA-FCM) and by the work (Mendes et al., 2012) (HGA).

generator applies to the motor also increases. Thus, it is possible to change the mechanical load to the motor, and consequently change its model. The main goal is to perform a velocity control where the load of the DC motor can be changed.

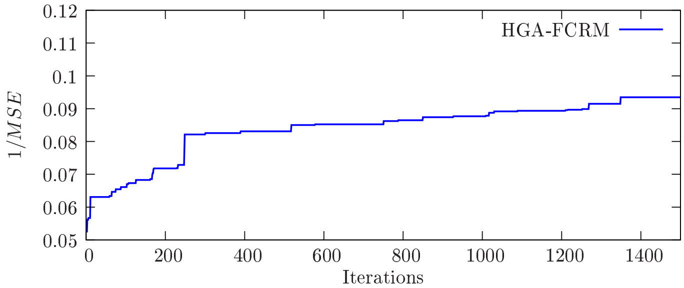
5.3.1. Identification

To identify the experimental setup, a dataset was constructed. The dataset was obtained by applying to the motor the control signal represented in Fig. 9(a). The variables chosen for the dataset were the first four delayed versions of the velocity $[y(k-1)$, $y(k-2)$, $y(k-3)$, $y(k-4)]$, and the command signal and its first three delayed versions $[u(k)$, $u(k-1)$, $u(k-2)$, $u(k-3)]$, where k is the sample time. The number of clusters and the degree of fuzziness were chosen as $c = 8$, and $\eta = 2$, respectively. Numerical results comparing the performance of the proposed method and the works RPLS (Dayal & MacGregor, 1997), ILLSA (Kadlec & Gabrys, 2011), and (Mendes et al., 2012) are presented in Table 5. Applying the proposed methodology, the selected variables were $[y(k-1), y(k-2), u(k-1), u(k-3)]$ and a fuzzy system with 20 rules was generated.

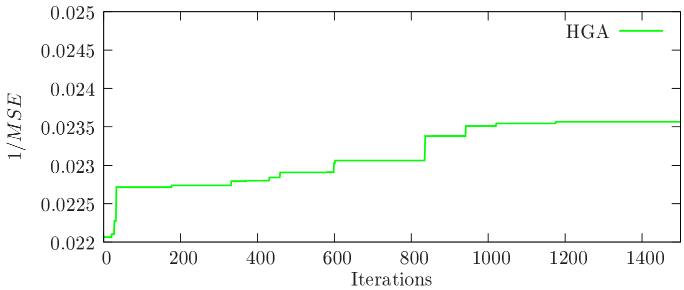
Fig. 9(b) shows the comparison of the velocity values of the motor obtained by the proposed (HGA-FCM) methodology and by the previous work (Mendes et al., 2012) (HGA), and the real/observed velocity values. It can be seen that the modeling of the velocity by the proposed (HGA-FCM) methodology is accurate and better than the one obtained by the HGA method proposed in (Mendes et al., 2012).

Table 5: Comparison results of the dataset test for the DC motor.

Method	$1/MSE$
RPLS (Dayal & MacGregor, 1997)	0.02410
ILLSA (Kadlec & Gabrys, 2011)	0.0197
HGA (Mendes et al., 2012)	0.0158
HGA-FCM (proposed method)	0.06095



(a)



(b)

Figure 10: Evolution of the best fitness functions value on Level 5 for all generations on the DC motors process: (a) HGA-FCM and (b) HGA.

Fig. 10(a) and (b) present the evolution of the fitness functions on Level V for all generations of HGA-FCM and HGA, respectively. As can be seen, the proposed work presents a good initialization performed by the FCM algorithm, and afterwards, the evolution attained by the proposed hierarchical GA in also good. The proposed HGA-FCM method attains faster response and better results when compared to the results obtained by the HGA proposed in (Mendes et al., 2012).

5.3.2. Adaptive predictive fuzzy control

The model learned by HGA-FCM in Section 5.3.1 is used to initialize the prediction model of the adaptive fuzzy GPC controller.

The following controller parameters were chosen by the user: $N_p = 10$, $N_u = 1$, $\lambda = 28$, $d = 0$, $\rho = 0.93$, $\varphi_i = 1$, $\tau_i = 1 \times 10^{-3}$, $\nu_i = 1 \times 10^{-6}$, for all $1 \leqslant i \leqslant c$. The

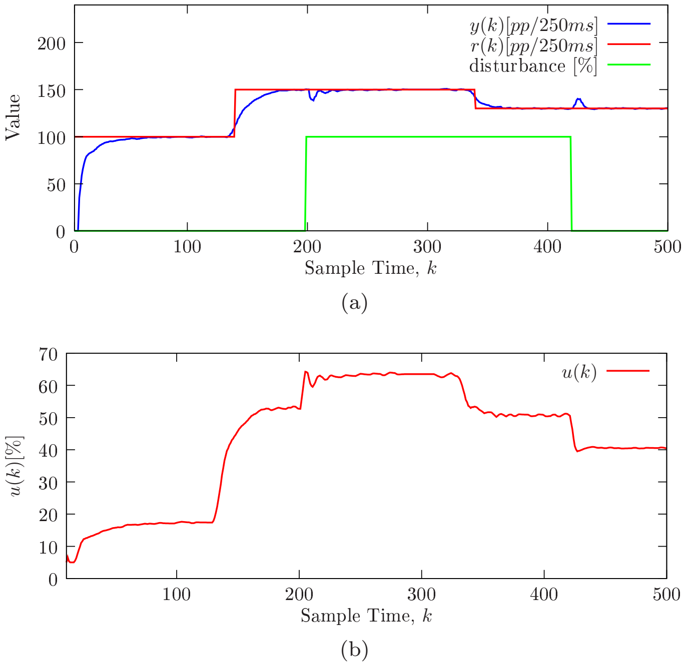


Figure 11: (a) Results of the proposed controller with HGA-FCM in presence of load disturbances in the DC motors process; and (b) respective applied command signal.

reference input is

$$r(k) = \begin{cases} 100, & 0 < k \leqslant 140, \\ 150, & 140 < k \leqslant 340, \\ 130, & 340 < k \leqslant 600, \end{cases} \tag{64}$$

and the load disturbance is applied at $200 \leqslant k \leqslant 420$ (lamps switched on).

From the results presented in Fig. 11(a) and (b), it can be seen that the proposed controller is able to adequately (attain and) control the system output at the desired reference $r(k)$. When the load disturbance is applied at $200 \leqslant k \leqslant 420$, there is an undershoot at $k = 200$ and an overshoot at $k = 420$ in the system response. As can be seen the controller eliminates this disturbance. By the results, it is concluded that the proposed controller methodology can control the process using only a dataset of the process to initialize the T-S fuzzy model.

6. Conclusion

A methodology was proposed to identify a T-S fuzzy model from input/output data to approximate unknown nonlinear processes. A coevolutionary hierarchical GAs method has been proposed to identify the structure and parameters of the model: the input variables and delay selection, the fuzzy rules, and the location of membership functions are automatically learned from system data. The identification methodology was proposed for two application problems: (1) the design of data-driven soft sensors, and (2) the learning of a model for the Generalized predictive control (GPC) algorithm. A recursive least squares method with adaptive directional forgetting is used for on-line adaptation of the T-S fuzzy model. The learned model was integrated with a fuzzy GPC controller. The integration of the proposed adaptive identification method with the GPC results in an effective adaptive predictive fuzzy control methodology. To validate and demonstrate the performance and effectiveness of the proposed algorithms, they were tested on the identification problem estimation of the flour concentration in the effluent of a real-world wastewater treatment system; and on the control of a simulated continuous stirred tank reactor (CSTR), and on a real-world experimental setup composed of two coupled DC motors. On identification, the results have shown that the proposed methodology has a faster response and better results when compared with the results obtained by the HGA proposed in (Mendes et al., 2012). The results have also shown that the proposed controller methodology can control the process using only a dataset of the process.

Acknowledgment

This work was supported by Project SCIAD/2011/21531 co-financed by QREN, in the framework of the “Mais Centro - Regional Operational Program of the Centro”, and by the European Union through the European Regional Development Fund (ERDF).

Jérôme Mendes and Francisco Souza were supported by Fundação para a Ciência e a Tecnologia (FCT) under grants SFRH/BD/63383/2009, and SFRH/BD/63454/2009, respectively.

References

Bobál, V., Böhm, J., Fessl, J., & Macháček, J. (2005). *Digital Self-tuning Controllers: Algorithms, Implementation and Applications*. London, UK: Springer.

Camacho, E. F., & Bordons, C. (1998). *Model Predictive Control*. Springer-Verlag.

Cazarez-Castro, N. R., Aguilar, L. T., & Castillo, O. (2010). Fuzzy logic control with genetic membership function parameters optimization for the output regulation of a servomechanism with nonlinear backlash. *Expert Systems with Applications*, *37*, 4368–4378.

Celikyilmaz, A., & Trksen, I. B. (2009). *Modeling Uncertainty with Fuzzy Logic: With Recent Theory and Applications*. (1st ed.). Springer Publishing Company, Incorporated.

Cordón, O., Herrera, F., Hoffmann, F., & Magdalena, L. (2001). *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases* volume 19 of *Advances in Fuzzy Systems–Applications and Theory*. (1st ed.). World Scientific Publishing Co. Pte. Ltd.

Dayal, B. S., & MacGregor, J. F. (1997). Recursive exponentially weighted pls and its applications to adaptive control and prediction. *Journal of Process Control*, *7*, 169–179.

Dovžan, D., & Škrjanc, I. (2011). Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes. *ISA Transactions*, *50*, 159–169.

Fortuna, L., Graziani, S., & Rizzo, A. (2006). *Soft Sensors for Monitoring and Control of Industrial Processes (Advances in Industrial Control)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Han, H.-G., Qiao, J.-F., & Chen, Q.-L. (2012). Model predictive control of dissolved oxygen concentration based on a self-organizing rbf neural network. *Control Engineering Practice*, *20*, 465–476.

Herrera, F. (2008). Genetic fuzzy systems: Taxonomy, current research trends and prospects. *Evolutionary Intelligence*, *1*, 27–46.

Hung, P.-C., & Lin, S.-F. (2012). The partial solutions consideration based self-adaptive evolutionary algorithm: A learning structure of neuro-fuzzy networks. *Expert Systems with Applications*, *39*, 10749–10763.

Kadlec, P., & Gabrys, B. (2011). Local learning-based adaptive soft sensor for catalyst activation prediction. *AIChE Journal*, *57*, 1288–1301.

Kasabov, N. K. (1996). *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*. MIT Press, Cambridge.

Kayadelen, C. (2011). Soil liquefaction modeling by genetic expression programming and neuro-fuzzy. *Expert Systems with Applications*, *38*, 4080–4087.

Kosko, B. (1994). Fuzzy systems as universal approximators. *IEEE Transactions on Computers*, *43*, 1329–1333.

Kulhavý, R. (1987). Restricted exponential forgetting in real-time identification. *Automatica*, *23*, 589–600.

Li, C., Zhou, J., Li, Q., An, X., & Xiang, X. (2010). A new t–s fuzzy-modeling approach to identify a boiler–turbine system. *Expert Systems with Applications*, *37*, 2214–2221.

Mendes, J., Souza, F., Araújo, R., & Gonçalves, N. (2012). Genetic fuzzy system for data-driven soft sensors design. *Applied Soft Computing*, *12*, 3237–3245.

Morningred, J. D., Paden, B. E., Seborg, D. E., & Mellichamp, D. A. (1992). An adaptive nonlinear predictive controller. *Chemical Engineering Science*, *47*, 755–762.

Sivanandam, S. N., & Deepa, S. N. (2007). *Introduction to Genetic Algorithms*. Springer Publishing Company.

Souza, F., Santos, P., & Araújo, R. (2010). Variable and delay selection using neural networks and mutual information for data-driven soft sensors. In *Proc. IEEE Conference on Emerging Technologies and Factory Automation (ETFA), 2010* (pp. 1–8).

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, *15*, 116–132.

Wang, L.-X. (1997). *A Course in Fuzzy Systems and Control*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Wang, L.-X., & Mendel, J. M. (1992). Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Transactions on Neural Networks*, *3*, 807–814.

Wu, M., Wang, C., Cao, W., Lai, X., & Chen, X. (2012). Design and application of generalized predictive control strategy with closed-loop identification for burn-through point in dintering process. *Control Engineering Practice*, *20*, 1065–1074.

Ying, H. (1997). General miso takagi-sugeno fuzzy systems with simplified linear rule consequent as universal approximators for control and modeling applications. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1335–1340). volume 2.