

Self-adaptive Takagi-Sugeno Model Identification Methodology For Industrial Control Processes

Saeid Rastegar*, Rui Araújo*, Jérôme Mendes*, Tiago Matias* and Alireza Emami*

*Institute of Systems and Robotics (ISR-UC), and

Department of Electrical and Computer Engineering (DEEC-UC),

University of Coimbra, Pólo II, PT-3030-290 Coimbra, Portugal

srastegar@isr.uc.pt, rui@isr.uc.pt, jermendes@isr.uc.pt, tmatias@isr.uc.pt, emami@isr.uc.pt

Abstract—A novel adaptive evolving Takagi-Sugeno (T-S) model identification method is investigated and integrated in a control architecture to control of nonlinear processes is investigated. The proposed system identification approach consists of two main steps: antecedent T-S fuzzy model parameters identification and consequent parameters identification. First, a new unsupervised fuzzy clustering algorithm (NUFCA) is introduced to combine the K-nearest neighbor and fuzzy C-means methods into a fuzzy modeling method for partitioning of the input-output data and identifying the antecedent parameters of the fuzzy system. Then, a recursive procedure using a particle swarm optimization (PSO) algorithm is exploited to construct an on-line fuzzy model identification and adaptive control methodology. For better demonstration of the robustness and efficiency of the proposed methodology, it is applied to the identification of a model for the estimation of the flour concentration in the effluent of a real-world wastewater treatment plant (WWTP), and identification and control, using a generalized predictive controller (GPC), of a real experimental setup composed of two coupled DC motors. The results show that the developed evolving T-S fuzzy model methodology can identify nonlinear systems satisfactorily and can be successfully used for a prediction model of the process for the GPC.

I. INTRODUCTION

System identification is considered as a crucial step for both data-driven soft sensors (DDSS) and model predictive control (MPC). A common underlying assumption of methodologies to address these DDSS and MPC problems is their assumption of the knowledge of an accurate model of the process to be identified. This assumption may cause important problems, because many complex plants are difficult to be mathematically modeled based on physical laws or have large uncertainties and strong nonlinearities. Several types approaches to modeling nonlinear plants can be considered. Among them, fuzzy models have received particular attention in the area of nonlinear modeling, especially the Takagi-Sugeno (T-S) fuzzy models [1]. The main feature of a T-S fuzzy model is to express the local dynamics of each fuzzy implication (rule) by a linear system model [2], [3]. The T-S fuzzy model parameters can be estimated through offline and/or online modes. However, online learning mode can be superior because in most cases, the collected dataset used in offline methods is limited, and the estimated T-S fuzzy model may not provide adequate accuracy in parts or the whole operating areas of the plant. Among many optimization methods, evolutionary algorithms such as particle swarm optimization algorithm (PSO) [4] and genetic algorithms (GAs) [5] have shown a good adaptation to search for optimal T-S fuzzy model parameters. In [6], it was proposed a new approach for on-line Takagi-Sugeno fuzzy model identification. It combines a recursive fuzzy C-means algorithm and recursive least squares. The calculation

of the membership degrees requires all the past observations. This affects the performance of the recursive approach. So the algorithm uses an approximate calculation by introducing the exponential weighting of the past membership degrees, which are calculated at each time instant. The method requires a large amount of memory because the fuzzy covariance matrix must be stored. Also, the memory demands are constant because of the fixed number of clusters. In the method proposed in [6] similarly to the concern in the conventional fuzzy C-means (FCM), the requirement of initial estimation of the clusters centroid vector is remains [7]. A hierarchical evolutionary approach for learning T-S fuzzy models was proposed in [8]. This algorithm despite of successful performance with five levels contains a complex design and needs to pass a time-consuming procedure.

The main motivations of this work are: (1) Introduction an unsupervised fuzzy classification algorithm to cluster high dimensional data; (2) Combining the proposed fuzzy clustering algorithm and a particle swarm optimization (PSO) procedure to construct an adaptive approach for online learning/updating of the consequent parameters of the T-S model; And (3) the integration of the T-S fuzzy learned by the online identification methodology with the classical GPC [9] to construct an adaptive fuzzy GPC controller (AFGPC). The resulting proposed online identification and adaptive control methodology can deal with non-linear plants, time-varying processes, disturbances or varying operating regions and parameters of the model. To validate the performance and effectiveness of the proposed identification methodology, it is applied to the modeling and estimation in two applications, a flour concentration in the effluent of a real-world wastewater treatment plant (WWTP), and a real experimental setup composed of two coupled DC motors. The application and the results are analyzed and quantitatively compared with two adaptive approaches: a recursive partial least squares (RPLS) [10], and an incremental local learning soft sensing algorithm (ILLSA) [11], and two regressors: a new fuzzy c-regression model algorithm (NFCRMA) [12], and a hierarchical genetic approach (HGA) [8]. Moreover, the performance and effectiveness of the AFGPC is demonstrated on the real-world experimental setup composed of two coupled DC motors.

The paper is organized as follows. Section II presents a description about nonlinear system modeling based on the T-S fuzzy model, and on the FCM clustering algorithm. Section III presents the PSO algorithm. Section IV proposes the NUFCA clustering algorithm and online T-S fuzzy model identification methodology. In Section V, results in identification and control of a plant are presented and analyzed. Finally, Section VI makes concluding remarks.

II. T-S FUZZY MODELS BASED ON FUZZY C -MEANS CLUSTERING

This Section presents a nonlinear system modeling methodology based on the T-S fuzzy model, and on FCM Clustering. Specifically, as an initialization method, the FCM clustering algorithm [6] is employed to learn the antecedent parameters of the T-S fuzzy model from data on the T-S fuzzy model learning methodology.

A. Modeling Using T-S Fuzzy Models

Takagi-Sugeno fuzzy models with simplified linear rule consequents are universal approximators capable of approximating any continuous nonlinear system with continuous constituent functions [13]. With a T-S fuzzy model, the global operation of a 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:

$$\begin{aligned} R_i : & \text{IF } x_1(k) \text{ is } A_1^i, \text{ and } \dots \text{ and } x_N(k) \text{ is } A_N^i \\ & \text{THEN } y(k) = y_i(k) = \theta_{i1}x_1(k) + \dots + \theta_{iN}x_N(k), \end{aligned} \quad (1)$$

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. $\theta_{i1}, \dots, \theta_{iN}$ are model parameters of $y_i(k)$. From (1), the model output $y(k)$ can be rewritten as

$$y(k) = \sum_{i=1}^c \bar{\omega}^i[\mathbf{x}(k)] \mathbf{x}^T(k) \boldsymbol{\theta}_i = \boldsymbol{\Psi}(k) \boldsymbol{\Theta}, \quad (2)$$

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

$$\mathbf{x}(k) = [x_1(k), \dots, x_N(k)], \quad (3)$$

$$\mu_{A_j^i}(x_j) = \exp\left(-\frac{(x_j - v_{ij})^2}{\sigma_{ij}}\right), \quad j = 1, \dots, N, \quad (4)$$

$$\bar{\omega}^i[\mathbf{x}(k)] = \frac{\prod_{j=1}^N \mu_{A_j^i}(x_j)}{\sum_{i=1}^c \prod_{j=1}^N \mu_{A_j^i}(x_j)}, \quad (5)$$

$$\boldsymbol{\theta}_i = [\theta_{i1}, \dots, \theta_{iN}]^T, \quad (6)$$

$$\boldsymbol{\Theta} = [\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, \dots, \boldsymbol{\theta}_c^T]^T, \quad (7)$$

$$\boldsymbol{\Psi}(k) = [(\bar{\omega}^1[\mathbf{x}(k)]) \mathbf{x}(k), \dots, (\bar{\omega}^c[\mathbf{x}(k)]) \mathbf{x}(k)], \quad (8)$$

where v_{ij} and σ_{ij} are the antecedent parameters, which represent the center and width of the antecedent membership functions, respectively, and which need to be defined/learned. Parameters v_{ij} and σ_{ij} will be learned from data using the Fuzzy C -means method presented in Section II-B below.

B. Fuzzy C -Means

The objective of the fuzzy c -means (FCM) clustering algorithm is the partitioning of a dataset \mathbf{X} into a pre-defined number of clusters, c . In 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), which are grouped into an N -dimensional observation/sample vector

$\mathbf{x}_l = [x_{l1}, \dots, x_{lN}]^T \in \mathbb{R}^N$. A set of L observations/objects is then denoted as

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

A fuzzy partition of the set \mathbf{X} into c clusters, is a family of fuzzy subsets $\{A^i \mid 1 \leq i \leq c\}$. 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 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 [6]: The membership degrees are real numbers in the interval $\mu_i(l) \in [0, 1]$, for $l = 1, \dots, L$; 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 contains all the data $0 < \sum_{l=1}^L \mu_i(l) < L$, for $i = 1, \dots, c$. FCM clustering tries to minimize the following objective function, which has a pre-defined number of clusters, c , and includes a fuzziness parameter, η :

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

where $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_c]^T \in \mathbb{R}^{c \times n}$ is a matrix of cluster centroid vectors $\mathbf{v}_i = [v_{i1}, \dots, v_{iN}]^T$, $d_{il}(\mathbf{x}_l, \mathbf{v}_i)$ 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 η influences the fuzziness of the resulting partition. The partition can range from a hard partition ($\eta = 1$) to a completely fuzzy partition ($\eta \rightarrow \infty$). In order to find the fuzzy clusters in the dataset \mathbf{X} , equation (10) must be minimized. If the derivative of 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 [6]:

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

where

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

and

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

The v_{ij} parameters of (4) are obtained from the center-vectors $\mathbf{v}_i = [v_{i1}, \dots, v_{iN}]^T$ of (13). To finalize the identification of the premise parameters in (4) of the T-S model (1)-(2), the $\boldsymbol{\sigma}_i = [\sigma_{i1}, \dots, \sigma_{iN}]^T$, $i = 1, \dots, c$, can be calculated from $\mathbf{U} = [\mu_i(l)]$, as follows:

$$\sigma_{ij} = \sqrt{\frac{2 \sum_{l=1}^L \mu_i(l) (x_{lj} - v_{ij})^2}{\sum_{l=1}^L \mu_i(l)}}, \quad j = 1, \dots, N. \quad (14)$$

C. Fuzzy Validity Indices

Although the FCM algorithm has received much attention, and since FCM is an unsupervised clustering algorithm, some cluster validity index is required to evaluate the quality of the clustering that results from the algorithm. Each index is categorized based on specified criterion. Generally, there are two main optimal clustering criteria, namely, compactness and separation. Compactness criteria have been proposed based on the idea that the members in one same cluster should be as close to each other as possible. For the compactness criteria the variance, which should be minimized, is one common measure. Some conventional fuzzy validity index for this class of scheme are for example the partition coefficient (PC) [14], and the proportion exponent [15]. Separation criteria are organized based on the distance between the closest members of the clusters, the distance between the most distant members, or the distance between the centers of the clusters. A conventional performance index in this class of scheme is the Xie-Beni index [16]. Although the Xie-Beni index has proved that it can provide reliable response over a wide range of choices for the number of clusters and fuzziness weighting exponent, the Xie-Beni index has two intrinsic drawbacks: (1) the validation index monotonically decreases when the number of clusters gets very large and close to the number of data points, and (2) strong interaction between the cluster validity index and fuzziness weighting exponent η imposes unpredictable behavior in results when fuzziness weighting exponent approaches infinity. Tang-Sun in [17], imposed a punishing function to eliminate the decreasing tendency, to overcome the above two problems. This work uses the Tang-Sun's validation index which is defined as follows:

$$V_T(\mathbf{U}, \mathbf{V}; \mathbf{X}) = \frac{\sum_{i=1}^c \sum_{l=1}^L \mu_{il}^2 \|\mathbf{x}_l - \mathbf{v}_i\|^2}{\min_{i \neq k} \|\mathbf{v}_i - \mathbf{v}_k\|^2 + 1/c} + \frac{\frac{1}{c(c-1)} \sum_{i=1}^c \sum_{k=1, k \neq i}^c \|\mathbf{v}_i - \mathbf{v}_k\|^2}{\min_{i \neq k} \|\mathbf{v}_i - \mathbf{v}_k\|^2 + 1/c}, \quad (15)$$

where c is number of clusters, and \mathbf{v}_i is the center of cluster i , $i = 1, \dots, c$. The numerator of the second term in (15) is an *ad hoc* punishing function (average distance between cluster centers) which is applied to eliminate the decreasing tendency of $V_T(\cdot)$ as $c \rightarrow L$.

III. PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM

Particle Swarm Optimization (PSO) algorithm first was proposed by Kennedy [18] and is motivated by the notion of collective intelligence in biological populations. PSO unlike the genetic algorithm (GA) is motivated by the simulation of social behavior and each candidate solution is associated with a velocity. The candidate which called 'particles' try to fly through the design space. The basic PSO algorithm consists of three steps, namely, generating particles positions and velocities, velocity update, and finally, position update. Here, a particle refers to a point in the design space that changes its position from one move (iteration) to another based on velocity updates. At each iteration, the velocity of every particle will be calculated as follows:

$$v_r^{t+1} = w_r^t v_r^t + c_1 r_1 (pbest_r^t - x_r^t) + c_2 r_2 (gbest^t - x_r^t), \quad (16)$$

where x_r^t is the position of the particle r in t -th iteration, $pbest_r^t$ is the best previous position of this particle (memorized by each individual particle), $gbest^t$ is the best previous position among all the particles in t th iteration (memorized in a common repository), w is the inertia weight, c_1 and c_2 are acceleration coefficients and are known as the cognitive and social parameters, respectively. Finally, r_1 and r_2 are two random numbers in the range $[0, 1]$. After calculating the velocity, the new position of every particle can be worked obtained as follows:

$$x_r^{t+1} = x_r^t + v_r^{t+1}. \quad (17)$$

The PSO algorithm performs repeated applications of the above update equations until the pre-specified number of generations T is reached. Although conventional PSO has shown some important advances by providing high speed of convergence in specific problems, it does exhibit some shortages. It was found that PSO has a poor ability to search at a fine grain because it lacks a velocity control mechanism. Many approaches are attempted to improve the performance of conventional PSO by variable inertia weight [19]. The inertia weight is critical for the performance of PSO, which balances global exploration and local exploitation abilities of the swarm. A big inertia weight facilitates exploration, but it makes the particle long time to converge. Conversely, a small inertia weight makes the particle fast converge, but it sometimes leads to local optimal. Motivated by the aforementioned, in this paper, for each individual particle r , the inertia weight is dynamically adapted as a function of the iteration number t as follows:

$$w_r^t = w_r^{Max} - ((w_r^{Max} - w_r^{Min}) * t) / T, \quad (18)$$

where, w_r^t is inertia weight in t th iteration, w_r^{Max} is the final inertia weight, w_r^{Min} is the minimum inertia weight, and T is maximum number of iterations. On the case study in this work the x_i in (16) is considered to be the vector Θ (7). Then the (16) is reformulated as:

$$v_r^{t+1} = w_r^t v_r^t + c_1 r_1 (pbest_r^t - \Theta_r^t) + c_2 r_2 (gbest^t - \Theta_r^t), \quad (19)$$

and the new position of every particle is

$$\Theta_r^{t+1} = \Theta_r^t + v_r^{t+1}. \quad (20)$$

The fitness function here applied in the PSO is $J_i = 1/MSE(i)$, where $MSE(i) = \frac{1}{L} \sum_{k=1}^L (y_k - \hat{y}_k)^2$, is the mean square error of the i -th fuzzy system, \hat{y}_k is the predicted value is calculated by (2), and y_k is the target output value at instant k , respectively.

IV. PROPOSED T-S FUZZY MODEL IDENTIFICATION ALGORITHM

In this section a new T-S fuzzy model identification algorithm is proposed. To construct a T-S fuzzy system of the form (2) it is necessary to obtain the number of rules, the antecedent membership functions, the set of rules, and also to update the consequent parameters (θ_i). The antecedent part is given by a new unsupervised fuzzy clustering algorithm (NUFCA) and the consequent parameters are estimated by the particle swarm optimization algorithm (Sec. III). The complete NUFCA algorithm is presented in Algorithm 1. The FCM algorithm with pre-defined initial values such as the number of clusters, initial cluster centers and fuzziness weighting

Algorithm 1 Proposed T-S fuzzy model identification Algorithm.

- 1) Construct the matrix $\mathbf{X} = [x_{lj}]_{L \times N}$, $1 \leq l \leq L$ and $1 \leq j \leq N$, in (9) using L observations;
 - 2) Choose the degree of fuzziness $\eta > 1$; And let \mathbf{g}_0 be the center of data \mathbf{X} , and $v_t^* \leftarrow 0$;
 - 3) Repeat the procedure below for $c = 1, 2, \dots, c_{max} = \sqrt{L}$:
 - a) Initialization for iteration c :
 - i) Let $K = \lfloor \frac{L}{c} - 1 \rfloor$, and $I = \{1, 2, \dots, L\}$, where $\lfloor \cdot \rfloor$ is the floor function;
 - b) **For** $i = 1, \dots, c$ **construct** E_i **using** K **nearest neighbourhood**:
 - i) In I find the index i of the unknown sample \mathbf{x}^i which is farthest from \mathbf{g}_{i-1} ;
 - ii) $E_i = \{\mathbf{x}^i\} \cup \text{KNN}(K-1, \mathbf{x}^i)$, where $\text{KNN}(K-1, \mathbf{x}^i)$ is the set of $K-1$ nearest-neighbour samples of \mathbf{x}^i that do not belong to any other already existing E_i .
 - iii) Let $\mathbf{g}_i = \frac{\sum_{\mathbf{x}_k \in E_i} \mathbf{x}_k}{K}$, and $E_i \leftarrow E_i \cup \{\mathbf{g}_i\}$;
 - iv) $I \leftarrow I \setminus \{i\} \setminus \text{IKNN}(K-1, \mathbf{x}^i)$, where $\text{IKNN}(K-1, \mathbf{x}^i)$ is the set of all indices n such that $\mathbf{x}^n \in \text{KNN}(K-1, \mathbf{x}^i)$;
 - c) While $I \neq \emptyset$, do:
 - i) Select $r \in I$, let $I \leftarrow I \setminus \{r\}$, and calculate the distances from the still unclustered sample \mathbf{x}^r to the center \mathbf{g}_i of all E_i by $d(\mathbf{x}^r, \mathbf{g}_i), \forall i = 1 \dots c$;
 - ii) Assign \mathbf{x}^r to the E_i with the nearest \mathbf{g}_i , so that $E_i \leftarrow E_i \cup \{\mathbf{x}^r\}$;
 - iii) Perform the update of $\mathbf{g}_i = \frac{\sum_{\mathbf{x}_k \in E_i} \mathbf{x}_k}{K+1}$;
 - d) **Perform one iteration of FCM**:
 - i) Calculate the fuzzy clustering matrix $\mathbf{U} = [\mu_{il}]_{c \times L}$ using (11)-(12) with $\mathbf{v}_i = \mathbf{g}_i$ in (12);
 - ii) Calculate clustering validity index by (15) and assign it to v_t .
 - e) If $v_t > v_t^*$, then
 - i) Let the optimal number of clusters be $c^* \leftarrow c$;
 - ii) Let $E_i^* \leftarrow E_i$, for $i = 1, \dots, c^*$, be the optimal clustering sets;
 - iii) Update the optimal clustering validity index: $v_t^* \leftarrow v_t$.
 - 4) Using $\mathbf{U} = [\mu_{il}]_{c^* \times L}$ calculate \mathbf{v}_i and σ_{ij} by (13)-(14).
-

exponent η converges to a solution at which the objective function J in (10) is minimized. In practice, in many cases, randomly choosing initial FCM parameters may cause the FCM to just obtain results which are only locally optimal. Furthermore, while the FCM is a semi-supervised method that requires the knowledge of the number of clusters. To overcome these problems, this paper proposes the NUFCA which uses a hybrid clustering algorithm based on two layers (Algorithm 1). NUFCA iteratively tests several values for the number of clusters c , in order to find an optimal value which is denoted as c^* . In the first layer of NUFCA, for each c , the initial centers of the clusters are obtained by using the KNN approach [Step 3b]. The basic idea in the KNN method is to try to find, from L samples of a dataset, the k samples which have the highest levels of similarity to a specified feature vector. Specifically, in the first layer of NUFCA, dataset \mathbf{X} of (9) is partitioned into c clusters, in which samples of each cluster have similarity in the Euclidean distance sense, and will belong to one set E_i [Steps 3a-3c]. E_i is an auxiliary set of samples to gather the members of tentative cluster i . After all E_i sets are constructed for a certain c , then one iteration of FCM is performed [Step 3d]. The final step of NUFCA consists on determining the best c , and the corresponding collection

Algorithm 2 Adaptive fuzzy system identification algorithm.

- 1) Design PSO parameters: Initial population of Θ and v , w_r^{Max} , w_r^{Min} , c_1 , c_2 , and T . Design the identification parameters (ρ , φ_i , τ_i , ν_i , for $i = 1, \dots, c$) with the same values as the ones defined in Algorithm 1;
 - 2) For initialization, use the fuzzy rule base (input variables, respective membership functions, the fuzzy rules and the final learned model parameters) learned in Algorithm 1;
 - 3) Find the initial $pbest_r^t$ and $gbest^t$ by putting of initial populations in fitness function;
 - 4) For/using each newly arriving online sample, do:
 - a) For $t = 1, \dots, T$:
 - i) Compute w_r^t and v_r^{t+1} using (18) and (19), respectively;
 - ii) Adapt the T-S fuzzy model parameters (Θ of (7)) by performing one iteration of recursion (20);
 - iii) Update $pbest_r^t$ and $gbest^t$ using the new Θ ;
-

TABLE I
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.

of the best E_i ($i = 1, \dots, c$), which are termed as c^* , and E_i^* ($i = 1, \dots, c^*$), respectively [Step 3e]. The results of this proposed hybrid clustering algorithm are used to set the antecedent parameters of the T-S model (1)-(2). The results extracted by Algorithm 1 will be used to initialize the proposed recursive PSO methodology for online identification of T-S a fuzzy model. The complete description of the proposed online system identification methodology is presented in Algorithm 2.

V. EXPERIMENTS AND RESULTS

A. Application to Wastewater Treatment System

In this section, the performance of the proposed identification methodology specifically on 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, \dots, u_{11} , and one target output 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 I. To construct the dataset, the first three delayed versions of each variable were chosen as inputs of the

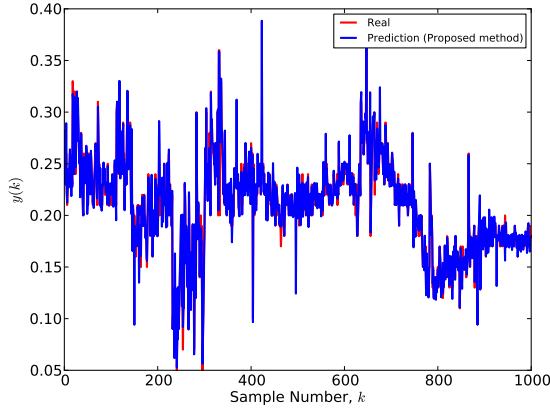


Fig. 1. Modeling performance of the proposed adaptive methodology for wastewater treatment system data set.

TABLE II

COMPARISON RESULTS ON THE TEST DATASET FOR THE WASTEWATER TREATMENT PLANT (WWTP).

Method	$1/MSE$
RPLS [10]	840.9
NFCRMA [12]	206.13
ILLSA [11]	1197.6
HGA [8]	901.2
Proposed method (Algorithm 2)	2166.4

T-S model. Specifically, the following combinations of process variables and delays are used as inputs of the T-S model to predict $y(t)$: $[u_1(t-1), u_1(t-2), u_1(t-3), \dots, u_{11}(t-1), u_{11}(t-2), u_{11}(t-3)]$. The selected degree of fuzziness was set to $\eta = 2$, and the optimal number of clusters that resulted from Algorithm 1 was $c = 13$. Figure 1 shows the predicted and desired (real) values of the target variable to be estimated, for the WWTP experiment. As can be seen, the accuracy of the modeling is acceptable. Numerical results comparing the performance of the proposed method and the works RPLS [10], NFCRMA [12], ILLSA [11], and HGA [8] are presented in Table II.

As can be seen comparing with other methods, the largest value of the fitness function in the test dataset is obtained with the method proposed in this paper. For the HGA, the number of individual rules and the number of inputs are 20 and 13, respectively. While with the proposed method, results were obtained with 13 individual rules and 33 input variables. Comparing with the HGA, the proposed method uses a lower number of fuzzy rules, but shows a better performance.

B. Real-World Control of Two Coupled DC Motors

The experimental system consists of two similar DC motors coupled by a shaft (Figure 2), 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 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 com-

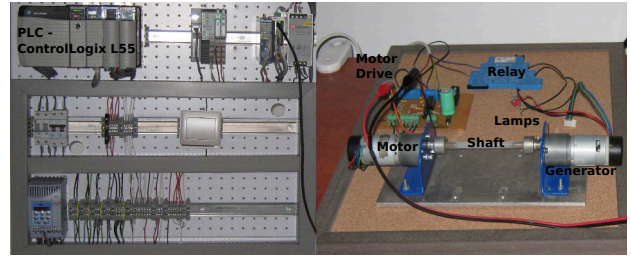


Fig. 2. The experimental scheme of the DC motors.

TABLE III

COMPARISON RESULTS ON THE TEST DATASET FOR THE DC MOTOR.

Method	$1/MSE$
RLS [20]	0.0523
RPLS [10]	0.02410
ILLSA [11]	0.0197
HGA [8]	0.0158
Proposed method (Algorithm 2)	0.1254

municates by OPC¹ to a PLC² (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 milliseconds). 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 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 control of the motor velocity where the load of the DC motor can be changed.

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 Figure 3a. 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. Applying the proposed methodology, the optimal number of clusters was calculated as $c = 12$. The degree of fuzziness was chosen as $\eta = 2$. Numerical results comparing the performance of the proposed method and the works RLS [20], RPLS [10], ILLSA [11], and HGA [8] are presented in Table III. Figure 3b shows the comparison of the velocity values of the motor obtained by the proposed methodology in Algorithm 2, and the real/observed velocity values. Figures 4a and 4b present the evolution of the fitness functions for all generations of HGA and proposed adaptive method in Algorithm 2, respectively. As can be seen, the proposed work presents a good initialization and attains faster response and better results when compared to the results obtained by the HGA proposed in [8].

2) *Adaptive Predictive Fuzzy Control*: The model learned by Algorithm 2 is used to convert the classical GPC [9] into

¹OLE (Object Linking and Embedding) for Process Control.

²Programmable Logic Controller.

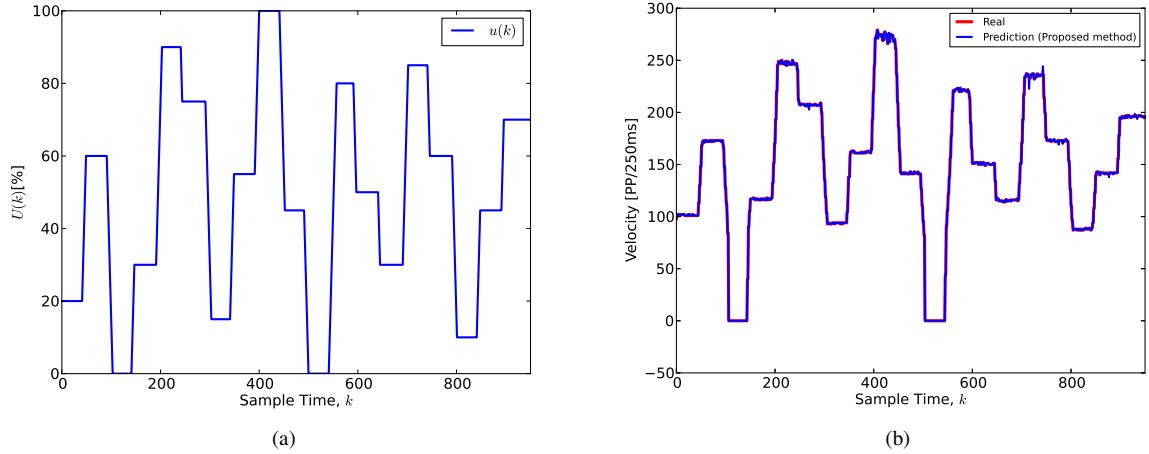


Fig. 3. Motor dataset: (a) control signal used to compile the dataset on the DC motors process and (b) modeling performance obtained by the proposed adaptive methodology.

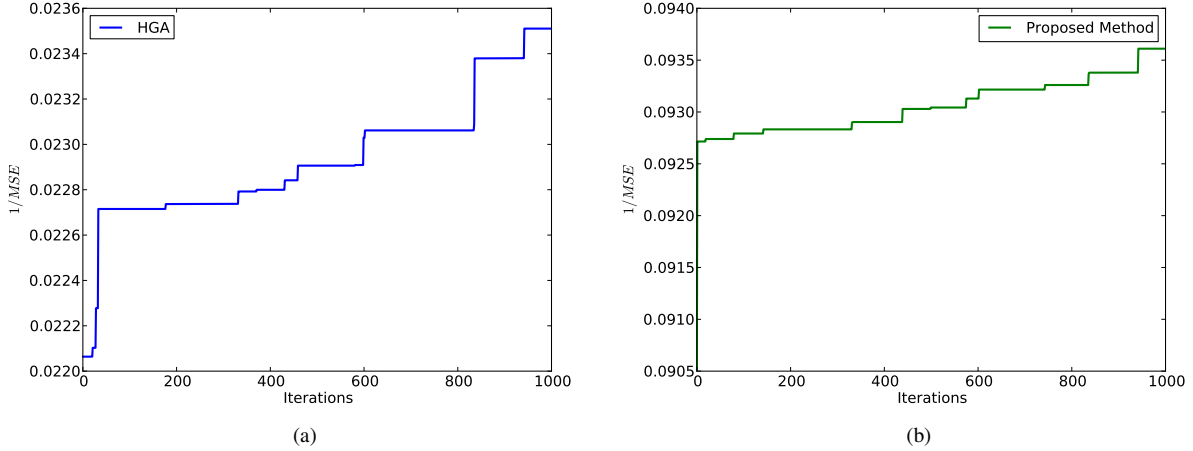


Fig. 4. Evolution of the best fitness functions value for all generations on the DC motors process: (a) HGA and (b) Proposed adaptive methodology.

an adaptive fuzzy GPC (AGPC) controller. The following controller parameters were chosen by the user: $N_p = 8$, $N_u = 1$, $\lambda = 25$, $d = 0$, $\rho = 0.93$, $\varphi_i = 1$, $\tau_i = 1 \times 10^{-3}$, $\nu_i = 1 \times 10^{-6}$, for $i = 1, \dots, c$. More information about the AFGPC controller parameters can be found in [21]. The reference input $r(t)$ [pp/(0.25 seg)] is

$$r(k) = \begin{cases} 100, & 0 < k \leq 120, \\ 150, & 120 < k \leq 200, \\ 130, & 200 < k \leq 360, \\ 120, & 360 < k \leq 560, \\ 100, & 560 < k \leq 640, \end{cases} \quad (21)$$

and the load disturbance (lamps switched on) is applied at $260 \leq k \leq 410$. Performance of the proposed AFGPC controller is presented in Figures 5a and 5b. From the results presented in Figures 5a and 5b, 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 $260 \leq k \leq 410$, there is an undershoot at $k = 260$ and an overshoot at $k = 410$ in the system response. As can be seen the controller shows a robust performance against this disturbance.

VI. CONCLUSION

An adaptive system identification/control methodology based on KNN-FCM and PSO was proposed which uses input/output data. The proposed identification method identifies the structure and parameters of the nonlinear model: the set of fuzzy rules, the number of rules and the location of the membership functions are automatically learned from system data. To validate and demonstrate the performance and effectiveness of the proposed algorithms, they were tested on the problem of identification/estimation of a real-world wastewater treatment plant (WWTP) plant, and identification/control of a real-world experimental setup composed of two coupled DC motors. In general, comparing with other identification methods, bigger fitness values with a lower number of fuzzy rules were attained. Also, results have shown that the proposed

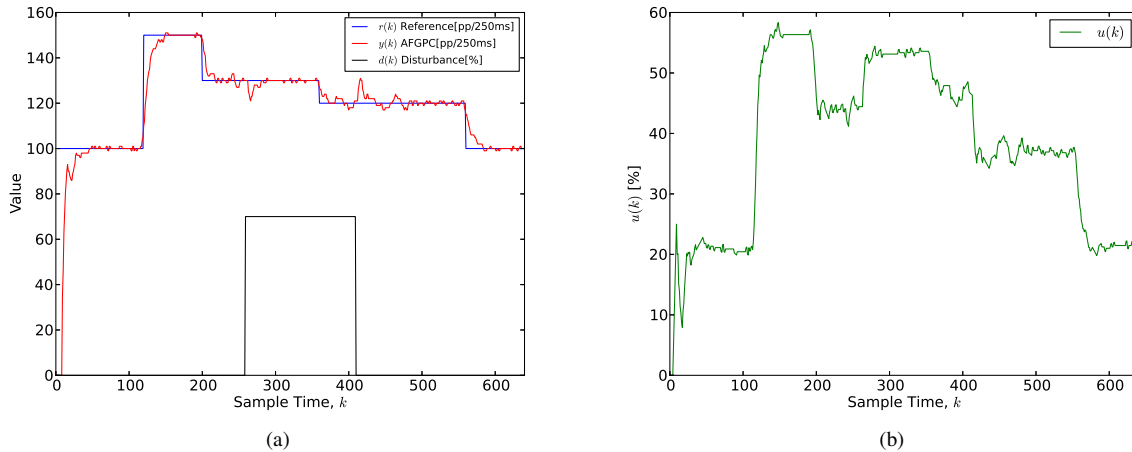


Fig. 5. (a) The performance of the AFGPC controller in presence of load disturbances in the real DC motors process; and (b) respective applied command signal.

controller methodology can control the process using only a dataset of the process to initialize the adaptive T-S fuzzy model.

ACKNOWLEDGMENT

This work was supported by Project SCIAD “Self-Learning Industrial Control Systems Through Process Data” (reference: 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).



The authors acknowledge the support of FCT project PEst-C/EEI/UI0048/2014. Saeid Rastegar, and Jérôme Mendes have been supported by FCT under grants SFRH/BD/89186/2012, and SFRH/BD/63383/2009, respectively.

REFERENCES

- [1] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modelling and control,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, February 1985.
- [2] J.-C. de Barros and A. J. Dexter, “On-line identification of computationally undemanding evolving fuzzy models,” *Fuzzy Sets and Systems*, vol. 158, no. 18, pp. 1997–2012, 2007.
- [3] Y. Yao, J. Mi, and Z. Li, “A novel variable precision(θ, σ)-fuzzy rough set model based on fuzzy granules,” *Fuzzy Sets and Systems*, vol. 236, pp. 58–72, 2014.
- [4] A. Alfí and H. Modares, “System identification and control using adaptive particle swarm optimization,” *Applied Mathematical Modelling*, vol. 35, no. 3, p. 1210–1221, 2011.
- [5] M. R. Delgado, E. Y. Nagai, and L. V. R. de Arruda, “A neuro-coevolutionary genetic fuzzy system to design soft sensors,” *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 5, pp. 481–495, 2009.
- [6] D. Dovžan and I. Škrjanc, “Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes,” *ISA Transactions*, vol. 50, no. 2, pp. 159–169, January 2011.
- [7] J. Bezdek, C. Leckie, and L. Hall, “Fuzzy c-means algorithms for very large data,” *Fuzzy Systems, IEEE Transactions on*, vol. 20, no. 6, pp. 1130–1146, 2012.
- [8] J. Mendes, F. Souza, R. Araújo, and N. Gonçalves, “Genetic fuzzy system for data-driven soft sensors design,” *Applied Soft Computing*, vol. 12, no. 10, pp. 3237–3245, 2012.
- [9] E. F. Camacho and C. Bordons, *Model Predictive Control*. Springer-Verlag, February 1998.
- [10] B. S. Dayal and J. F. MacGregor, “Recursive exponentially weighted pls and its applications to adaptive control and prediction,” *Journal of Process Control*, vol. 7, no. 3, pp. 169–179, 1997.
- [11] P. Kadlec and B. Gabrys, “Local learning-based adaptive soft sensor for catalyst activation prediction,” *AIChE Journal*, vol. 57, no. 5, pp. 1288–1301, 2011.
- [12] C. Li, J. Zhou, X. Xiang, Q. Li, and X. An, “T-s fuzzy model identification based on a novel fuzzy c-regression model clustering algorithm,” *Engineering Applications of Artificial Intelligence*, vol. 22, no. 4-5, pp. 646–653, 2009.
- [13] H. Ying, “General miso takagi-sugeno fuzzy systems with simplified linear rule consequent as universal approximators for control and modelling applications,” in *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, 1997, pp. 1335–1340.
- [14] J. C. Bezdek, “Cluster validity with fuzzy sets,” *Cybernet*, vol. 3, no. 3, pp. 58–73, Feb 1974.
- [15] M. P. Windham, “Cluster validity for the fuzzy c-means clustering algorithm,” *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, vol. PAMI-4, no. 4, pp. 357–363, July 1982.
- [16] X. Xie and G. Beni, “A validity measure for fuzzy clustering,” *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, vol. 13, no. 8, pp. 841–847, 1991.
- [17] Y. Tang and F. Sun, “Improved validation index for fuzzy clustering,” in *American Control Conference*. IEEE, June 8-10 2005, pp. 1120–1125.
- [18] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Conference on Neural Networks*, December 1995, pp. 1942–1945.
- [19] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, “Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients,” *IEEE Trans. Evolut. Comput.*, vol. 8, no. 3, p. 240–255, 2004.
- [20] R. Kulhavý, “Restricted exponential forgetting in real-time identification,” *Automatica*, vol. 23, no. 5, pp. 589–600, 1987.
- [21] J. Mendes, R. Araújo, and F. Souza, “Adaptive fuzzy identification and predictive control for industrial processes,” *Expert Systems with Applications*, vol. 40, no. 17, pp. 6964–6975, 2013.