Online Identification of Takagi-Sugeno Fuzzy Models Based on Self-Adaptive Hierarchical Particle Swarm Optimization Algorithm

Saeid Rastegar\textsuperscript{a,*}, Rui Araújo\textsuperscript{a}, Jérôme Mendes\textsuperscript{a}

\textsuperscript{a}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

Abstract

This paper presents an approach for online learning of Takagi-Sugeno (T-S) fuzzy models. A novel learning algorithm based on a Hierarchical Particle Swarm Optimization (HPSO) is introduced to automatically extract all fuzzy logic system (FLS)’s parameters of a T-S fuzzy model. During online operation, both the consequent parameters of the T-S fuzzy model and the PSO inertia weight are continually updated when new data becomes available. By applying this concept to the learning algorithm, a new type T-S fuzzy modelling approach is constructed where the proposed HPSO algorithm includes an adaptive procedure and becomes a self-adaptive HPSO (S-AHPSO) algorithm usable in real-time processes. To improve the computational time of the proposed HPSO, particles positions are initialized by using an efficient unsupervised fuzzy clustering algorithm (UFCA). The UFCA combines the K-nearest neighbour and fuzzy C-means methods into a fuzzy modelling method for partitioning of the input-output data and identifying the antecedent parameters of the fuzzy system, enhancing the HPSO’s tuning. The approach is applied to identify the dynamical behaviour of the dissolved oxygen concentration in an activated sludge reactor within a wastewater treatment plant. The results show that the proposed approach can identify nonlinear systems satisfactorily, and reveal superior performance of the proposed methods when compared with other state of the art methods. Moreover, the methodologies proposed in this paper can be involved in wider applications in a number of fields such as model predictive control, direct controller design, unsupervised clustering, motion detection, and robotics.

Keywords: Online learning of Takagi-Sugeno model, Self-adaptive Hierarchical PSO Algorithm, Activated sludge process

1. Introduction

System identification based on experimental data has been considered as a powerful practical engineering tool in many industrial control fields. A common point among most control methodologies is their assumption of the knowledge of an accurate model of the process to be controlled. Mostly a good control performance can be obtained if a good system modeling of a plant can be estimated. This assumption may cause important problems. It happens a lot that in a system identification process due to complexity in plant behavior or large uncertainties, obtaining an explicit mathematical model based on physical laws for the plant meets with difficulties. Different identification approaches to modeling nonlinear plants were employed to be used in control design. Among them, fuzzy models have received particular attention in the area of nonlinear modelling (Škrjanc, 2009), especially the Takagi-Sugeno (T-S) fuzzy models (Takagi and Sugeno, 1985). The main feature of a T-S fuzzy model is that it may express the local dynamics of each fuzzy implication (rule) by a linear system model (Barros and Dexter, 2007; Chang et al., 2010; Hua et al., 2013; Dovzan et al., 2015). The T-S fuzzy model parameters can be estimated in either or both offline or online modes. However, an online learning mode can have superiority because in most cases, the collected data set used in offline methods is limited, and the estimated T-S fuzzy model may not provide adequate accuracy in parts of,
or the whole, operating areas of the plant (Salahshoor et al., 2012; Li and Du, 2012). Many optimization methods were investigated to be used in T-S fuzzy modelling. Among them, evolutionary algorithms such as particle swarm optimization (PSO) algorithms and genetic algorithms (GAs) have shown a good adaptation to search for optimal T-S fuzzy model parameters (Delgado et al., 2001; dos Santos Coelho and Herrera, 2007; Shuzhi et al., 2012). Although a GA may be able to find the global minimum, it consumes too much search time. The PSO algorithm is an alternative. It can result in a lower computational complexity with a better performance of fitness evaluation in general when compared with GAs (Eberhart and Shi, 1998), consequently resulting in a better prediction performance when applied to learn a T-S fuzzy model. Ali and Ramaswamy (2009) presented an optimal fuzzy logic control (FLC) algorithm for vibration mitigation of buildings using magneto-rheological (MR) dampers. A micro-genetic algorithm (m-GA) and a standard particle swarm optimization (SPSO) were used to optimize the FLC parameters. In the method, only the membership function parameters are optimized, while the other components of the fuzzy system are considered to be fixed. Other common limitation of this method, and other identification methods such as (Rastegar et al., 2014, 2016), concerns to the selection of the correct set of input variables. The variable selection process is usually manual and not accompanied with the accurate selection of the right time delays, probably leading to low-accuracy results. A variable with the correct delay may contain more information about the output, than one which does not consider any delay or which considers an incorrect delay (Souza et al., 2010). The method presented in (Rastegar et al., 2014, 2016) uses a PSO algorithm to construct a self-adaptive T-S model identification methodology, but the method lacks an automatic input selection system. An adaptive PID controller based on adaptive PSO was proposed in (Ali and Modares, 2011). In most of the cases, controlling the inertia weight of PSO has been done based on a constant parameter, or on a linearly decreasing inertia weight PSO (LDW-PSO) has also been used (Shi and Eberhart, 1998). But in (Ali and Modares, 2011), the inertia weight was dynamically adapted for every particle by considering a measure called the adjacency index (AI). This idea lead the classical LDW-PSO algorithm to be evolved into a new class of system modelling based on an adaptive PSO (APSO) methodology. The results proved a superiority of APSO performance when compared with the LDW-PSO or GA.

The prediction performance of a Takagi-Sugeno fuzzy model depends on its complexity (e.g. number of fuzzy rules), on the type of membership functions, on the antecedent variables, and on the consequent regressors (Yao et al., 2014). A hierarchical genetic algorithm (HGA) was suggested by Delgado et al. (2001) to find optimal parameters of T-S fuzzy systems through an evolutionary genetic algorithm and a neuro-based technique, and then was improved (Delgado et al., 2009) by proposing an idea for pre-selection of input variables using an auxiliary criterion. However, the variable and delay selection are not jointly performed with the learning of the fuzzy model, which precludes the global optimization of the prediction setting. As an evolution of the work (Delgado et al., 2001, 2009), Mendes et al. (2012) proposed an identification methodology based on the application of a HGA, and also the proposed identification method was applied for the design of a direct control methodology (Mendes et al., 2014b). The main advances in this work are the improvement of the whole hierarchical structure, automatically extracting the fuzzy control rules. Despite of having a successful performance, the HGA did not utilize any significant improvement in its interior structure to be augmented for online applications. A related methodology for online applications was proposed in (Mendes et al., 2013), using a combination with other method, the recursive least squares (RLS) method with adaptive directional forgetting approach (Kulhavý, 1987). For automatic extraction of all FLS parameters, the number of partitions is a key parameter. Approximately all the FLS units of a T-S fuzzy model such as the set of fuzzy rules, the individual rules, the consequent parameters, as well as the unit of the FLS, use this parameter during the FLS construction. However, in both the methods proposed in (Delgado et al., 2001) and in (Mendes et al., 2012), the number of partitions was defined firstly and then considered as a constant value during all the running time of the algorithm.

Motivated by the issues discussed above, this work introduces a novel identification methodology which, differently from (Rastegar et al., 2014, 2016), is based on a hierarchical PSO (HPSO) to learn the FLS’s parameters of a T-S fuzzy model from a set of input/output data. The use of a hierarchical structure has the advantage of decreasing the complexity of the learning method, also leading to the improvement of computational time and prediction performance. The search engine design of the HPSO was inspired by the method proposed in (Mendes et al., 2012), but has the following three main advantages in comparison to (Mendes et al., 2012): less running time, larger prediction accuracy, and possibility to be used in online model learning. Unlike the works in (Rastegar et al., 2014, 2016) which use a simple sequential structure, and where no selection of input variables and delays is performed, the algorithm proposed in this paper is composed of a six level hierarchy, where the first level is responsible for the selection of
an adequate set of input variables and delays. The second level considers the antecedent membership functions. The consequent parameters are defined on the third Level. Level four is devoted to particles of individual rules. The set of rules are obtained on the fifth level, and finally, the sixth level is appointed as the control unit of FLS, specifying the choice of several configurations in the lower levels. For improving the convergence time, an initialization algorithm based on an efficient unsupervised fuzzy clustering algorithm (Rastegar et al., 2016) is employed to initialize particles’ positions inside the HPSO, as well as the number of partitions. Finally, an online adaptive approach based on the HPSO is proposed which results in a self-adaptive HPSO (S-AHPSO) approach to make an online learning/updating of the fuzzy T-S model.

To validate the performance and effectiveness of the proposed algorithm, prediction of the dynamic behavior of the dissolved oxygen concentration in an activated sludge reactor within a wastewater treatment plant (WWTP) is investigated. WWTPs are large and complex non-linear systems subject to large disturbances in influent flow rate and pollutant load, together with uncertainties concerning the composition of the incoming wastewater (Belchior et al., 2012). In (Belchior et al., 2012), a basic fuzzy controller and an adaptive fuzzy controller (AFC) were developed and applied to control the dissolved oxygen in an activated sludge reactor within a WWTP. Using the BFC controller, a learning data set was obtained for the proposed HPSO algorithm to learn the T-S fuzzy model of, and predict, the dissolved oxygen concentration in an activated sludge reactor. The extracted parameters of the T-S fuzzy model which are continually updated by the S-AHPSO can be used to design an adaptive and/or predictive fuzzy controller such as the AFC in (Belchior et al., 2012) or the adaptive fuzzy generalized predictive controller (AFGPC) which recently has shown a successful performance in non-linear industrial applications (Mendes et al., 2013). The proposed method is quantitatively compared with two recursive approaches: a recursive partial least squares (RPLS) (Dayal and MacGregor, 1997), and an RLS with a directional forgetting approach (Kulhavý, 1987) which uses fuzzy C-means (FCM) (Windham, 1982) for initialization. Furthermore, the results are compared with the results of four other methods: using a multi-layer perceptron neural network (MLPNN) for nonlinear system identification (Nelles, 2001), a new fuzzy c-regression model algorithm (NFCRMA) proposed by Li et al. (2009), an extreme learning machine for regression (ELM) (Huang et al., 2012), and a genetic fuzzy system for data-driven soft-sensor design (Mendes et al., 2012). Moreover, the performance of an identification method based on adaptive PSO (APSO) (Alfi and Modares, 2011) is also evaluated and compared. The system identification methodology proposed in this work can deal with non-linear plants, time-varying processes, varying operating regions and parameters of the model.

The paper is organized as follows. Section 2 presents a description about the nonlinear system modelling methodology based on the T-S fuzzy model, and on the fuzzy C-means clustering algorithm. Section 3 briefly presents the PSO algorithm. Section 4 presents the proposed HPSO algorithm which will be accompanied initially by the introduction of the UFCA as an initialization algorithm to enhance the HPSO’s performance, and subsequently by the presentation of the S-AHPSO algorithm which proposes an online learning/updating of the T-S fuzzy model. In Section 5, results in identification of a plant are presented and analysed. Finally, Section 6 makes concluding remarks.

2. T-S Fuzzy Models Based on Fuzzy C-Means Clustering

In this section it is briefly presented a methodology for modeling nonlinear processes. The structure is based on the T-S fuzzy model, and the initialization of its antecedent part is performed by learning from data using the Fuzzy C-Means Clustering method. The antecedent part will then be improved using the proposed methodology presented on Section 4.

2.1. Modelling Using T-S Fuzzy Models

This section describes the T-S fuzzy model that with their simplified linear rule consequents are universal approximators capable of approximating any continuous nonlinear system (Ying, 1997). In this way, a nonlinear system with continuous constituent functions can be described a T-S fuzzy model composed by the following fuzzy rules:

\[
R_i: \text{IF } x_1(k) \text{ is } A_{i1}, \text{ and } \ldots \text{ and } x_N(k) \text{ is } A_{iN} \text{ THEN } y(k) = y_i(k) = \theta_{i1} x_1(k) + \cdots + \theta_{iN} x_N(k),
\]  

(1)
where $R_i$ ($i = 1, 2, \ldots, c$) represents the $i$-th fuzzy rule, $c$ is the number of rules, and $x_1(k), \ldots, x_N(k)$ are the input variables of the T-S fuzzy model. $A_i^j$ ($j = 1, 2, \ldots, N$) are linguistic terms characterized by fuzzy membership functions $\mu_{A_i^j}(x_j)$ which describe the local operating regions of the plant. $\theta_1, \ldots, \theta_N$ are model parameters of $y_i(k)$. From (1), the T-S fuzzy model $y(k)$ can be rewritten as

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

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

$$\mathbf{x}(k) = [x_1(k), \ldots, x_N(k)],$$

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

$$\tilde{\omega}^i[k] = \frac{\prod_{j=1}^{N} \mu_{A_i^j}(x_j)}{\sum_{i=1}^{c} \prod_{j=1}^{N} \mu_{A_i^j}(x_j)},$$

$$\mathbf{\theta}_i = [\theta_{i1}, \ldots, \theta_{iN}]^T,$$

$$\Theta = [\mathbf{\theta}_1^T, \ldots, \mathbf{\theta}_c^T]^T,$$

$$\Psi(k) = \left[\left(\tilde{\omega}^1[k] \mathbf{x}(k)\right), \ldots, \left(\tilde{\omega}^c[k] \mathbf{x}(k)\right)\right].$$

where $v_{ij}$ and $\sigma_{ij}$ are the center and width of the antecedent membership functions, respectively, to be learned/designated. In an initial stage, $v_{ij}$ and $\sigma_{ij}$ will be learned by the Fuzzy C-means method (Section 2.2), and then will be improved using the proposed methodology presented on Section 4.

### 2.2. Fuzzy C-Means

This section describes the fuzzy c-means (FCM) method. In the fuzzy clustering methods the objects can belong to a predefined number of clusters, $c$. Defining $\mathbf{x}_l = [x_{l1}, \ldots, x_{LN}]^T \in \mathbb{R}^N$ as an observation $l$ composed by a set of $N$ samples, one sample for each input variable $j$. A set of $L$ observations is then defined as

$$X = \begin{bmatrix}
    x_{11} & x_{12} & \ldots & x_{1N} \\
    x_{21} & x_{22} & \ldots & x_{2N} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{L1} & x_{L2} & \ldots & x_{LN}
\end{bmatrix}.$$  

(9)

The fuzzy partition of the set $X$ into predefined number of clusters $c$, is a family of fuzzy subsets $\{A_i^l | 1 \leq i \leq c\}$, in which their membership functions are defined as $\mu_i(x_l) = \mu_{A_i^l}(x_l)$, and form the fuzzy partition matrix $\mathbf{U} = [u_{ij}] = [\mu_i(l)] \in \mathbb{R}^{c \times L}$. Each row $i$ of matrix $\mathbf{U}$ contains the values of the membership function of the $i$-th fuzzy subset $A_i^l$ for all the observations stored in the data matrix $X$. Similarly to (Dovžan and Škrjanc, 2011), the partition matrix should satisfy the following conditions: (i) The membership degrees are real numbers in the interval $\mu_i(l) \in [0, 1]$, for $l = 1, \ldots, L$; The total membership of each sample in all the clusters must be equal to one $\sum_{l=1}^{L} \mu_i(l) = 1$; (ii) And none of the fuzzy clusters is empty, neither do any contain all the data $0 < \sum_{l=1}^{L} \mu_i(l) < L$, for $i = 1, \ldots, c$. FCM method will obtain the antecedents parameters $v_{ij}$ and $\sigma_{ij}$ by minimizing the following objective function:

$$J(X, U, V) = \sum_{i=1}^{c} \sum_{l=1}^{L} [\mu_i(l)]^2 d_{ij}^2(x_l, v_i),$$

(10)
where $V = [v_1, \ldots, v_c]^T \in \mathbb{R}^{c \times N}$ is a matrix of cluster centroid vectors $v_i = [v_{i1}, \ldots, v_{iN}]^T$, $d_{ij}(x_i, v_i)$ is the Euclidean distance ($l^2$-norm) between the observation $x_i$ and the cluster centroid $v_i$, and the overlapping factor or the fuzziness parameter $\eta$ influences the fuzziness of the resulting partition. The partition can range from a hard partition ($\eta = 1$) to a completely fuzzy partition ($\eta \to \infty$). If the derivative of the objective function (10) is taken with respect to the cluster centers $V$ and to the membership values $U$, then optimum membership values are calculated as follows (Dovžan and Škrjanc, 2011):

$$\mu_i(l) = \left(\frac{d_{ij}^2}{\sum_{q=1}^{c} (d_{ij}^2)^{1/(\eta-1)}}\right)^{-1},$$  \hspace{1cm} (11)

where

$$d_{ij}^2 = (x_i - v_j)^T (x_i - v_j),$$  \hspace{1cm} (12)

and

$$v_i = \frac{\sum_{l=1}^{L} \mu_i(l)x_i}{\sum_{l=1}^{L} \mu_i(l)},$$  \hspace{1cm} (13)

The centers $v_j$ of the antecedent membership functions are obtained by (13) where $v_i = [v_{i1}, \ldots, v_{iN}]^T$, and the respective widths, $\sigma_i = [\sigma_{i1}, \ldots, \sigma_{in}]^T$, $i = 1, \ldots, c$, are obtained using $U = [\mu_i(l)]$, as follows:

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

FCM is an unsupervised clustering algorithm. For unsupervised clustering algorithm, a cluster validity index can be employed to evaluate the fitness of partitions produced by clustering algorithms. This work uses the Tang and Sun (2005) validity index which is based on the evaluation two criteria, compactness and separation, and is defined as follows:

$$V_T(U, V; X) = \frac{\sum_{i=1}^{c} \sum_{l=1}^{L} \mu_i(l)^2 \| x_i - v_i \|^2}{\min_{i,k} \| v_i - v_k \|^2 + 1/c} + \frac{1}{c(c-1)} \sum_{i=1}^{c} \sum_{k=1, k \neq i}^{c} \frac{\| v_i - v_k \|^2}{\min_{i,k} \| v_i - v_k \|^2 + 1/c},$$  \hspace{1cm} (15)

where $c$ is the number of clusters, and $v_i$ is the center of cluster $i$, $i = 1, \ldots, c$. 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 $c \to L$.

3. Particle Swarm Optimization (PSO) Algorithm

The particle swarm optimization (PSO) concept is based on a stochastic optimization technique which was first proposed by Kennedy and Eberhart (1995). The algorithm was inspired by the social behavior of birds or fishies. The basic PSO concept is trying to simulate the choreographed motion of swarms of birds as part of a socio-cognitive science. Like genetic algorithms (GA), PSO is initialized with a, typically random, population of solutions. However, unlike GA, PSO has no evolution operators such as cross over, or mutation. Compared to GA, PSO is easier to be implemented. With PSO, there are just a few parameters that need to be adjusted before PSO is put to run. In PSO, the potential candidate solutions are called ‘particles’ which try to fly through the problem space. Performance of each particle is evaluated through a fitness function which needs to be optimized. In every iteration, each particle is updated by following two best fitness function values. The first one is the best solution which a particle has obtained so far. In this paper this value is called “pbest”. Another value is the best solution that was obtained by any particle
in the swarm. In this paper this value is called “gbest”. At each iteration, $t$, the velocity of every particle $v^t_r$ will be iteratively calculated as follows:

$$v^{t+1}_r = w^t v^t_r + c_1 r_1 (p_{best}^t_r - x^t_r) + c_2 r_2 (gbest^t - x^t_r),$$

(16)

where $x^t_r$ is the position of the particle $r$ in iteration $t$, $p_{best}^t_r$ 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 iteration $t$ (memorized in a common repository), $w$ is the inertia weight, and $c_1$ and $c_2$ are positive 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 obtained as

$$x^{t+1}_r = x^t_r + v^{t+1}_r.$$  

(17)

To better address the notation of the particles inside each level of the HPSO algorithm proposed in this work, and also based on the case study in this work, equation (16) is rearranged as follows:

$$v^p_r(t + 1) = w^p_r(t)v^p_r(t) + c_1(p)r_1(p)(p_{best}^p_r(t) - x^p_r(t)) + c_2(p)r_2(p)(gbest^p_r(t) - x^p_r(t)), $$

(18)

where, $w^p_r(t)$ and $p_{best}^p_r(t)$ are the inertia weight and the best personal experience of particle $r$ in iteration $t$ inside Level $p$, respectively. $gbest^p_r(t)$ is the best previous experience among all the particles inside Level $p$ in iteration $t$. $c_1(p)$ and $c_2(p)$ are acceleration coefficients of the $r$-th particle in Level $p$. Then, (17) is rearranged as:

$$x^p_r(t + 1) = x^p_r(t) + v^p_r(t + 1).$$

(19)

The learning process of the HPSO algorithm utilizes the conventional linearly decreasing inertia weight PSO (LDW-PSO) concept (Shi and Eberhart, 1998), which means that for each individual particle $r$, and iteration $t$:

$$w^p_r(t) = w^p_{r,max} - (w^p_{r,max} - w^p_{r,min}) * t/T_1,$$

(20)

where $w^p_r(t)$ is the inertia weight in iteration $t$ in Level $p$. $w^p_{r,max}$, $w^p_{r,min}$ are the initial and the final (minimum) inertia weight of particle $r$ in Level $p$, respectively, and $T_1$ is the number of PSO iterations.

4. Proposed T-S Fuzzy Model Identification Algorithm

This section proposes a novel learning algorithm based on a hierarchical particle swarm optimization (HPSO) algorithm to automatically extract all the parameters of the T-S fuzzy model from a set of input/output data. First, before presenting the HPSO, and in order to avoid random initialization of the particles in the HPSO and to improve computational time, an initialization algorithm based on an efficient unsupervised fuzzy clustering algorithm (UFCA) is employed to initialize the particles positions of the HPSO. Finally, after presenting the central structure of the HPSO, a self-adaptive algorithm based on the HPSO will be presented to perform an online learning/updating of the T-S fuzzy model from data.

4.1. Initialization Algorithm

In PSO algorithms, random initialization of the particles can result in an exhausting optimality search. In this condition more iterations are required to attain convergence. Finding a solution to reduce the computational time, which at the same time can increase the algorithm’s performance is desired. To achieve these desired goals, an efficient unsupervised fuzzy clustering algorithm (UFCA) (Rastegar et al., 2016) is applied to initialize the particles of the proposed HPSO algorithm. The complete UFCA algorithm is presented in Algorithm 1. The FCM is a well-known fuzzy clustering method which allows one object in a group to belong to two or more clusters. The FCM uses some pre-defined initial values such as the number of clusters, initial cluster centers and fuzziness weighting exponent $\eta$. By minimization of the objective function $J$ in (10), converge of FCM performance is attained. The FCM has been frequently used in pattern recognition. However, in many conditions it was seen that random initialization of the FCM causes the algorithm converge to a local optimal solution. For example, the number of clusters for
the FCM initialization is considered as an important parameter which plays an important role in convergence of the FCM performance. Finding a solution to find an optimal number of clusters by using real input/output dataset instead of using a random initialization, can be an efficient strategy. To overcome random initialization concern for clusters number, this paper employs the UFCA which has been recently proposed in (Rastegar et al., 2016) as a hybrid clustering algorithm based on two layers. UFCA iteratively tests several values for the number of clusters in the range of \( c = 1, 2, \ldots, c_{max} = \sqrt{L} \), in order to find an optimal number of clusters which is denoted as \( c^* \) in this paper. The first layer of UFCA uses a technique based on the K-Nearest Neighbor (KNN) approach (Cover and Hart, 1967) (Step 3b) to obtain the initial centers of the clusters, for each \( c \). The \( K \) value in the K-Nearest Neighbor (KNN) is given by \( K = \left\lfloor \frac{L}{2} - 1 \right\rfloor \), where \( L \) is total number of samples of a data set. The basic purpose in using of the KNN method is to try to identify, from \( L \) samples of a data set, the \( K \) samples which have the largest levels of homogeneity to a specified feature vector. Specifically, in the first layer of UFCA, data set \( X \) of (9) is divided into \( c \) clusters, in which members of each cluster have homogeneity 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 \( i \). Once all \( E_i \) sets are constructed for a certain \( c \), the algorithm arrives to the second level of UFCA where (only) one iteration of FCM is performed (Step 3d). The final step of UFCA consists on determining the best \( c \), and the corresponding collection of the best \( E_i (i = 1, \ldots, c), \) which are termed as \( c^* \), and \( E_i^* (i = 1, \ldots, c^*) \), respectively (Step 3e). In the process of finding the best \( c \), after the calculation of the FCM fuzzy clustering matrix \( U = [\mu_{ij}] \), and clusters centers \( v_i, (i = 1, \ldots, c) \), the clustering validity index (15) is employed to evaluate the quality of the clustering results for each \( c \).

The results extracted by Algorithm 1 will be used to initialize the first individual of Level 1 of the HPSO algorithm (Algorithm 2, Sec. 4.2) and all the individuals of Levels 2, 4, and 5 of HPSO. Furthermore, the initial solution is also used to initialize alleles of the first individual in Level 6 of HPSO. The other individuals of Levels 1 and 6, and all individuals of Level 3 are randomly initialized.

Algorithm 1 Unsupervised fuzzy clustering algorithm (UFCA).

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

This section presents HPSO, an automatic evolutionary algorithm to extract from data all the FLS parameters of the T-S fuzzy model (1)-(8), not requiring prior explicit expert knowledge. The data from which the FLS is extracted is composed of a set of input observations \(X\) (9), and a corresponding set of output observations \(y = [y_1, \ldots, y_L]^T \in \mathbb{R}^L\), where \(y_l\) is the output corresponding to \(x_l\), for \(l = 1, \ldots, L\).

The HPSO is constructed based on six hierarchical levels (Figure 1). The first level represents the particles of the set of input variables and their respective time delays. The particles of the second level represent all the antecedent fuzzy membership functions which constitute the fuzzy rules of the fuzzy system. Particles at the third level represent parameters sets for the consequent parts of the fuzzy rules. The individual rule particles are defined on the fourth level, and the particles that represent sets of rules are obtained on the fifth level. Finally, the sixth level represents T-S fuzzy models, where the particles include the indices of the selected elements of the previous levels. The detailed description of each level is given below.

**Level 1:** it is composed of particles that represent the set of input variables, and respective delays. The particles of this level are represented by binary encoding, where each allele (element of the particle located in a specific position) corresponds to each input variable/delay pair (see Figure 1). The length of the particle 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 fuzzy system.

**Level 2**: contains Gaussian membership functions defined in the universes of the variables involved. The particle is erected by the aggregations of all Gaussian partition sets associated with the input variables. A partition set of a variable is a collection of fuzzy sets associated to the variable.

**Level 3**: it is constructed based on particles, where each particle represents the consequent parameters that can be used for a fuzzy rule. The length of each particle in this level is determined by the maximum number of input variables. The particle is represented by real number encoding. On each particle, the specific alleles that are taken into account for a rule are the ones that correspond to the collection of input variables selected by Level 1. Null values indicate the absence of consequent parameter value for the corresponding variable.

**Level 4**: it is formed by particles of individual rules. The length of the particle is given by the maximum number of antecedent variables plus one. The particle is represented by non-negative integer encoding, where each allele contains the index of the corresponding antecedent membership function (defined at Level 2), except the last allele in each particle which is appointed to select the indices of Level 3. Null values indicate the absence of membership function for the corresponding variable (i.e. the absence of the variable) and are only considered for antecedent indices.

**Level 5**: it is constituted by a set of fuzzy rules, where each allele contains the index of the corresponding individual rule that has been included in the set. The particle is represented by integer encoding, where once again, null values indicate that the corresponding allele does not contribute to the inclusion of any rule in the set of fuzzy rules. The length of each particle is determined by the maximum number of fuzzy rules.

**Level 6**: it represents a control and specification unit for the T-S fuzzy model (TSFM). Key information required to develop and describe a TSFM design is contemplated on this level. The particle is represented by integer encoding and is constituted by three alleles. The first allele indicates the index, \( b \), of the set of rules specified on Level 5. The second allele contains the \( s \)-th individual of Level 2 which represents the \( s \)-th cooling of partition sets given by Level 2. The third allele represents the index, \( m \), of the set of input variables selected on Level 1 (see Figure 1).

**Level 6** is a flexible unit. Other alleles can be added to this unit in order to design a new HPSO usable with different characteristics. For example, a new extended HPSO with more complexity can be constructed for a direct fuzzy controller (DFC) design, where other alleles can be added to correspond to other important fuzzy elements such as t-norm operators, inference engine, and defuzzifier methods, and where, consequently, such additional elements of the fuzzy system can have a chance to be identified in a competition by the PSO methodology. However, as can be concluded from section 2.1, the design of the T-S fuzzy model (as the case study of this work) is composed by: singleton fuzzifier (direct representation of a number by a singleton fuzzy set), center average defuzzifier, and the product inference engine (Wang, 1996).

The main steps of the HPSO algorithm used to learn/improve the FLS parameters are presented in Algorithm 2. Each level utilizes its own PSO parameters and fitness function to estimate the new position of the particles in that level. The fitness evaluation of each individual of the hierarchical particles in each level is defined as follows:

- **Control unit of the fuzzy logic system (Level 6):**
  The value of the fitness function of the \( a \)-th particle in Level 6 is defined by
  \[
  J_6^a = \frac{1}{\text{MSE}(\hat{y}, y)}, \tag{21}
  \]
  where \( \text{MSE}(y, \hat{y}) = \frac{1}{L} \sum_{l=1}^{L} (y_l - \hat{y}_l)^2 \), is the mean square error of the \( a \)-th fuzzy system \((a = 1, \ldots, a_{\text{max}})\), \( y \) is the \( l \)-th output observation (target), and \( \hat{y}_l \) is the corresponding output value predicted by the FLS;

- **Units of Levels 1, 2, and 5:**
  The value of the fitness function of the \( u \)-th particle in Level \( e \), for \( e = 1, 2, 5 \), is evaluated by
  \[
  J_e^u = \max\left(J_{6,1}^{e,u}, \ldots, J_{6,a_{\text{max}}}^{e,u}\right), \tag{22}
  \]
  where \( \{f_{e,1}, \ldots, f_{e,a_{\text{max}}}\} \subseteq \{1, \ldots, a_{\text{max}}\} \) is the subset of all the particles of Level 6 that contain the \( u \)-th individual of Level \( e \) (see Figure 1);

- **Individual rule (Level 4):**
  The value of the fitness function of the \( d \)-th particle in Level 4 is defined by
  \[
  J_4^d = \max(J_1^d, \ldots, J_6^d), \tag{23}
  \]
Algorithm 2 Hierarchical Particle Swarm Optimization (HPSO) Algorithm.

**Inputs:** Set of input observations \( X \) (9), and a corresponding set of output observations \( y = [y_1, \ldots, y_L]^T \in \mathbb{R}^L \), where \( y_l \) is the output corresponding to \( x_l \), for \( l = 1, \ldots, L \); Maximum number of particles for each level, \( a_{max}, b_{max}, d_{max}, \sigma_{max}, s_{max}, m_{max} \); Velocity limits for each level, \( v_{min}^p = [v_{min1}^p], v_{max}^p = [v_{max1}^p], p = 1, \ldots, 6 \); Maximum number of iterations, \( T \); \( w_{min}, w_{max}, c_1, \) and \( c_2 \), PSO parameters of the \( r \)-th particle for each level; Maximum number of particles in each level, \( r_{max} \)

**Output:** Optimized T-S fuzzy model (TSFM);

```
procedure
  for all levels \( p = 1, 2, \ldots, 6 \) do // For all levels of the HPSO
    for all particles \( r = 1, 2, \ldots, r_{max} \), of Level \( p \) do // For all particles of the level
      if \( p \in \{2, 4, 5\} \) or \( (p = 1) \) or \( (p = 6 \text{ and } r = 1) \)
        then Initialize \( x_0^p \) of \( p \)-th Level using the results extracted with UFCA Algorithm 1;
        else Randomly initialize \( x_0^p \) endif
        Initialize \( v_0^p \) of \( p \)-th Level randomly within the velocity range of \( (v_{min}^p, v_{max}^p) \);
        \( \text{pbest}^p = x_0^p \);
      end for
    end for
  Evaluate all particles using one of the equations (21)-(24), according to the level of each particle;
  Identify the \( \text{gbest}^p \) of the swarm \( p \), for \( p = 1, \ldots, 6 \);
  \( t \leftarrow 1 \);
  while \( t \leq T \) do
    for all levels \( p = 1, 2, \ldots, 6 \) do
      for all particles \( r = 1, 2, \ldots, r_{max} \), of Level \( p \) do
        Compute \( w_r^p(t) \) and \( v_r^p(t) \) using (20) and (18), respectively;
        Update \( x_r^p(t) \) using (19);
      end for
    end for
    Evaluate \( J_r^p \), for each particle \( r = 1, 2, \ldots, r_{max} \), of each level \( p = 1, 2, \ldots, 6 \), using (21)-(24);
    Update \( \text{pbest}_r^p(t) \) and \( \text{gbest}_r^p(t) \) based on the \( J_r^p \) results from (21)-(24), for \( p = 1, 2, \ldots, 6, r = 1, 2, \ldots, r_{max} \);
    \( t \leftarrow t + 1 \);
  end while
end procedure
```

where \( \{r_1, \ldots, r_6\} \subseteq \{1, \ldots, a_{max}\} \) is the subset of all the particles of Level 6 that involve the \( d \)-th individual rule of Level 4 (indirect involvement through Level 5, see Figure 1);

- Consequent sets (Level 3):
  The value of the fitness function of the \( z \)-th particle in Level 3 is defined by
  \[
  J_3^z = \max(J_6^{g_1}, \ldots, J_6^{g_p}),
  \]
  (24)
  where \( \{g_1, \ldots, g_p\} \subseteq \{1, \ldots, a_{max}\} \) is the subset of all the particles of Level 6 that involve the \( z \)-th consequent set of Level 3 (indirect involvement through Levels 5 and 4, see Figure 1).

Each level of the particles’ hierarchy is evolved separately as an independent PSO algorithm using its own particles, own fitness function, and own PSO parameters. However, since the values of the fitness evaluations of the particles in each level also depend on the particles in other levels, then the evolution of each level is also influenced by the evolution of all the other levels.

4.3. Adaptive Online the T-S Fuzzy Modelling

The results extracted by the HPSO methodology, Sec. 4.2, and Algorithm 2 will be used to initialize the T-S fuzzy model for the operation in online mode. For the online process the consequent parameters vector \( \Theta \) (7) is denoted by
In PSO, the inertia weight is an important parameter which significantly affects the speed of PSO convergence. With many researches, it was proved that the performance of PSO can be improved if an appropriate adaptive strategy for the inertia weight (Ratnaweera et al., 2004) replaces typical inertia weight strategies like the conventional linearly decreasing inertia weight PSO (LDW-PSO) (Shi and Eberhart, 1998). For example, a variable inertia weight can be formulated to depend on the particle’s position or velocity. In fact, applying an adaptive inertia weight can balance global exploration and local exploitation abilities of the swarm.

Motivated by these facts, this paper proposes an adaptive process for obtaining the inertia weight. For each newly available data sample at time instant $k$, S-AHPSO runs PSO iterations $t = 1, \ldots, T_2$. For each iteration $t$ of S-AHPSO, the inertia weight $w_r$ for each particle $r$ is obtained from feedback taken from the best memories of the individual particle and of the swarm as follows:

$$w_r(t) = w_{r,\text{min}} + (w_{r,\text{max}} - w_{r,\text{min}}) \exp \left( -\gamma \left| y_{(\text{pbest})}(t) - y_{(\text{gbest})}(t) \right| + \beta \right) \times \frac{t - 1}{T_2 - t + \delta(t - T_2)} \left[ 1 - \delta(t - T_2) \right],$$

where $\gamma$ and $\beta$ are positive constants, $\delta(k)$ is the Kronecker Delta function, where $\delta(k) = 0$, for all $k \in \mathbb{N}$, except $\delta(0) = 1$, and $T_2$ is the number of S-AHPSO iterations in $t$ at each time instant $k$. Also, $y_{(\text{pbest})}(t)$ is the result obtained by the best previous position of the $r$-th particle for all non-future values of $t$ for the current $k$, and $y_{(\text{gbest})}(t)$ is the best output estimation attained among all particles of the swarm, until the current iteration $t$.

If the effect of $\left| y_{(\text{pbest})}(t) - y_{(\text{gbest})}(t) \right|$ is not considered in (26), then, as can be seen, the inertia weight in both (20) and (26) follows a decreasing tendency. On the other hand, if a particle $r$ is far away from the swarm, then the term $\left| y_{(\text{pbest})}(t) - y_{(\text{gbest})}(t) \right|$ in the denominator in (26) will reduce the decreasing tendency of the particles’ inertia weights, and thus increase the degree of exploration in order to give better possibilities for particle $r$ to join the swarm.

For updating of consequent rules in order to perform the online construction of the T-S fuzzy model, Eq. (16) is reformulated as:

$$\Theta_r(t + 1) = \Theta_r(t) + c_1 r_1(\text{pbest}(t) - \Delta \Theta_r(t)) + c_2 r_2(\text{gbest}(t) - \Delta \Theta_r(t)),$$

and the new position of every particle can be obtained as

$$\Delta \Theta_r(t + 1) = \Delta \Theta_r(t) + \gamma_r(t + 1).$$

The complete proposed S-AHPSO online adaptive system identification algorithm is presented in Algorithm 3. For each new data sample that becomes available at time instant $k$, when the S-AHPSO algorithm terminates the PSO iterations $t = 1, \ldots, T_2$ for that sample $k$, the final $\Delta \Theta_r(t)$ value is used as $\Delta \Theta_{\text{on}}(k)$ in (25).

The numbers of iterations $T_1$ and $T_2$ of Algorithms 2, and 3, respectively, could be seen as the maximum numbers of iterations if early termination conditions (e.g. as a function of fitness) would be introduced in these algorithms.

The fitness function applied in Algorithm 3 is

$$J^{(r)} = 1/\text{MSE}(r) = \left[ (y_{re}(k) - \hat{y}^{(r)}(k))^2 \right]^{-1},$$

where $\text{MSE}(r)$ is the mean square error of the $r$-th particle, $y_{re}(k)$ is the target output value at instant $k$, and $\hat{y}^{(r)}(k)$ is a prediction of $y_{re}(k)$ calculated by the $r$-th particle using (25) and (2).
Algorithm 3 Self-adaptive Hierarchical Particle Swarm Optimization (S-AHPSO) Algorithm.

**Input:** The T-S fuzzy system learned by the HPSO Algorithm 2 (input variables, antecedent parameters, consequent parameters, the fuzzy rules, and the final learned model parameters).

**Output:** $\Delta \Theta_m(k)$, for each new data sample that becomes available during the online operation at time instant $k$;

1. Initialization:
   a. Select $r_{\text{max}}$, the maximum number of particles for the swarm $\Delta \Theta_r$; as well as the maximum number of iterations $T_2$; Select the velocity limits $v_{\text{min}} = [v_{\text{min},1}, v_{\text{min},1}]$, $v_{\text{max}} = [v_{\text{max},1}, v_{\text{max},1}]$;
   b. Design the PSO parameters $w_{r,\max}$, $w_{r,\min}$, $c_1, c_2$ for each particle $r$ of the swarm $\Delta \Theta_r$;
   c. Initialize $[\Theta_m]_{c,N \times 1}$ of the T-S fuzzy model using the best result obtained from Level 3 of the Algorithm 2, where $[\Theta_m]_{c,N \times 1}$ is the consequent parameters vector and $c$ and $N$ are the optimal number of individual fuzzy rules and the maximum number of selected input variables, respectively, which all have resulted from Algorithm 2;

2. For/using each newly arriving online sample $k$, do:
   
   for all particles of the swarm, $r = 1, 2, \ldots, r_{\text{max}}$ do
   
   Randomly initialize $[\Delta \Theta_r]_{c,N \times 1}$;
   
   Initialize $v_r$ randomly within the velocity range $(v_{\text{min}}, v_{\text{max}})$;
   
   Let $p_{\text{best}} \leftarrow [\Delta \Theta_r]_{c,N \times 1}$;
   
   end for

   Evaluate each particle of the swarm $\Delta \Theta_r$ using the fitness function $J^{(r)} (29)$;
   
   Identify the best particle $g_{\text{best}}$;
   
   for all $t = 1, \ldots, T_2$ do
   
   Compute $w_r(t)$ and $v_r(t + 1)$ using (26) and (27), respectively;
   
   Update $\Delta \Theta_r(t)$ using (28);
   
   Calculate the fitness function of each particle of the swarm $\Delta \Theta_r$ using the fitness function $J^{(r)} (29)$;
   
   Update $p_{\text{best}}$, and $g_{\text{best}}$;
   
   end for

   Let $\Delta \Theta_m(k) \leftarrow [\Delta \Theta_r(t)]_{c,N \times 1}$;

   Adapt the T-S fuzzy model parameters ($\Theta$ of Eq. (7)) using (25);

5. Experiments and Results

In this section, the performance of the proposed identification methodology is studied. The study is performed on a soft sensor application. The objective of the experiment is to predict the dynamic behaviour of the dissolved oxygen concentration in the final compartment of the biological reactor within a wastewater treatment plant (WWTP). Many control strategies have been proposed in the literature of this application but their evaluation and comparison, either practical or based on simulation is difficult. This is due to a number of reasons, including: the variability of the influent, the complexity of the biological and biochemical phenomena, the large range of time constants varying from a few minutes to several days, the lack of standard evaluation criteria among other things, due to region specific effluent requirements and cost levels. The methodology proposed in this paper can be employed to obtain a model that can be used to address these concerns. The T-S fuzzy model that results from Algorithms 2 and 3 can be used to initialize and update an adaptive fuzzy predictive controller design such as the adaptive fuzzy generalized predictive controller (AFGPC) which has recently shown successful performance in nonlinear industrial applications (Mendes et al., 2013). A data set of the plant is obtained with the aim of providing a set of input/output data necessary for the HPSO to learn the FLS’s parameters of the T-S fuzzy model. The data set is obtained by applying to the process the command signal $K_{\text{lab}}$, that manipulates the oxygen transfer process from an aerator to the activated sludge inside the biological reactor, where $K_{\text{lab}}$ is obtained from the controller proposed in (Belchior et al., 2012), and recording along time the values of the input and output variables that constitute the data set.
The total data set includes 1344 samples. The aforementioned parameters in this paragraph were tuned by the designer. On the other hand, the number of iterations of the HPSO algorithm is minimum and maximum velocities are $v_{\text{min}}$ and $v_{\text{max}}$ among all particles in both Algorithm 2 and Algorithm 3. In Algorithms 2 and 3, all components of the algorithm and the remaining 864 samples are used as the test data set for S-AHPSO. The selected degree of fuzziness where the first 480 samples are used as the training data set to learn the Takagi-Sugeno fuzzy model parameters by the HPSO algorithm and consists of 10 main variables ($u_1$, $u_2$, $u_3$, $u_4$, $u_5$, $u_6$, $u_7$, $u_8$, $u_9$, $u_{10}$). For the learning process of the HPSO algorithm, a data set has been gathered from the plant operation and consists of two anoxic tanks followed by three aerobic tanks. Each compartment is characterized by flow rate $Q_a$, concentration $C_a$, volume $V_a$, and reaction rate $r_a$. Volumes $V_1$ for non-aerated compartments are $V_1 = V_2 = 1.000 [m^3]$, and for aerated compartments are $V_3 = V_4 = V_5 = 1.333 [m^3]$. Compartments 3-4 have a fixed oxygen transfer coefficient ($K_{La} = 10 [h^{-1}] = 240 \text{[days^{-1}]}$) while in compartment 5, the dissolved oxygen (DO) concentration, $D_{OC}$, is controlled by manipulation of the $K_{La5}$. For more details about the BMS1 plant, references (Jeppsson and Pons, 2004; Belchior et al., 2012; Mendes et al., 2014a) are recommended. The sampling period is 15 [min], and the simulations have a maximum of 14 [days].

5.1. General Characteristics

Waste water treatment plants (WWTPs) are industrial structures designed to remove biological or chemical waste products from water. They are complex nonlinear systems subject to large disturbances in the influent flow rate and pollutant load, together with uncertainties concerning the composition of the incoming wastewater (Belchior et al., 2012). BSM1 is a platform-independent simulation environment of WWTPs which has been undertaken in Europe by Working groups of COST Action 682 and 624 that is dedicated to the optimization of performance and cost-effectiveness of wastewater management systems (Jeppsson and Pons, 2004). This development work is now continued under the umbrella of the IWA Task group on Benchmarking of Control Strategies for WWTPs. The architecture proposed in this paper is tested in the Benchmark Simulation Model no.1 (BSM1). A general overview of the BMS1 plant is presented in Figure 2. The benchmark plant is composed of a five compartment activated sludge reactor consisting of two anoxic tanks followed by three aerobic tanks. Each compartment is characterized by concentration $C_a$, volume $V_a$, and reaction rate $r_a$. Volumes $V_1$ for non-aerated compartments are $V_1 = V_2 = 1.000 [m^3]$, and for aerated compartments are $V_3 = V_4 = V_5 = 1.333 [m^3]$. Compartments 3-4 have a fixed oxygen transfer coefficient ($K_{La} = 10 [h^{-1}] = 240 \text{[days^{-1}]}$) while in compartment 5, the dissolved oxygen (DO) concentration, $D_{OC}$, is controlled by manipulation of the $K_{La5}$. For more details about the BMS1 plant, references (Jeppsson and Pons, 2004; Belchior et al., 2012; Mendes et al., 2014a) are recommended. The sampling period is 15 [min], and the simulations have a maximum of 14 [days].

5.2. Application to Wastewater Treatment System

For the learning process of the HPSO algorithm, a data set has been gathered from the plant operation and consists of 10 main variables ($u_1, \ldots, u_{10}$), where 9 of them are input variables ($u_1, \ldots, u_9$) = ($S_S, X_S, X_I, S_{NO}, S_{NH}, X_{NO}, S_{ND}, X_{BH}, K_{La5}$), and the remaining variable is the target output variable to be estimated, $y = u_{10} = D_{OC}$. Figure 3 illustrates the input variables of the data set. The plant variables are described in Table 1.

The available data set is constructed by using of the first three delayed versions of each of the variables $u_1, \ldots, u_9$, as candidates for the inputs of the Takagi-Sugeno (T-S) model. Therefore, the following combinations of process variables and delays were specified to be used as the main candidates for inputs of the Takagi-Sugeno fuzzy model to predict $y(t)$: $[u_1(t-1), u_1(t-2), u_1(t-3), \ldots, u_9(t-1), u_9(t-2), u_9(t-3)]$. The total data set includes 1344 samples where the first 480 samples are used as the training data set to learn the T-S fuzzy model parameters by the HPSO algorithm and the remaining 864 samples are used as the test data set for S-AHPSO. The selected degree of fuzziness was set to $\eta = 2$ in Algorithm 1. The PSO parameters $w_{c,\text{max}} = 1$, $w_{c,\text{min}} = 0.3$, $c_1 = 1.5$, and $c_2 = 2$ are used in common among all particles in both Algorithm 2 and Algorithm 3. In Algorithms 2 and 3, all components of the minimum and maximum velocities are $v_{\text{min},i} = v_{\text{max},i} = 2$, and the maximum number of iterations of the HPSO algorithm is $T_1 = 200$. For the S-AHPSO algorithm, $y = 1$, $b = 0.1$, and maximum number of iterations $T_2 = 50$ were considered. The numbers of particles for each level of the HPSO architecture are: $a_{\text{max}} = b_{\text{max}} = 30$, $d_{\text{max}} = z_{\text{max}} = 50$, $s_{\text{max}} = 40$, and $m_{\text{max}} = 20$. Also, $r_{\text{max}} = 25$ is considered for the S-AHPSO architecture. The aforementioned parameters in this paragraph were tuned by the designer. On the other hand, the
optimal number of clusters $c^*$ is self-adjusted by Algorithm 1 that obtained a value of $c^* = 10$. Also, the consequent parameters are self-adjusted, being calculated by Algorithm 3.
Figure 4: (a) Modeling performance of the proposed system identification methodology for the $y = D_{OC}$ variable of the wastewater treatment system data set; (b) magnified view comparing the real output and predicted output variables corresponding to the rectangle “R” depicted inside Figure 4(a).

Table 2: Comparison results on the test data set for the wastewater treatment plant (WWTP). The MSE values have been multiplied by 10$^3$.

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of rules</th>
<th>Number of inputs</th>
<th>Inputs</th>
<th>MSE</th>
<th>CT [minutes]</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCM-FRLS (Kulhavý, 1987)</td>
<td>-</td>
<td>-</td>
<td>all candidate variables</td>
<td>67.4</td>
<td>2.803</td>
</tr>
<tr>
<td>UFCA-FRLS (Kulhavý, 1987)</td>
<td>-</td>
<td>-</td>
<td>all candidate variables</td>
<td>63.5</td>
<td>3.655</td>
</tr>
<tr>
<td>RPLS (Dayal and MacGregor, 1997)</td>
<td>-</td>
<td>-</td>
<td>all candidate variables</td>
<td>8.78</td>
<td>4.375</td>
</tr>
<tr>
<td>MLPNN (Nelles, 2001)</td>
<td>-</td>
<td>-</td>
<td>all candidate variables</td>
<td>81.5</td>
<td>16.24</td>
</tr>
<tr>
<td>NFCRMA (Random initialization) (Li et al., 2009)</td>
<td>-</td>
<td>-</td>
<td>all candidate variables</td>
<td>1689.2</td>
<td>10.36</td>
</tr>
<tr>
<td>UFCA-NFCRMA (Li et al., 2009)</td>
<td>-</td>
<td>-</td>
<td>all candidate variables</td>
<td>267.6</td>
<td>10.55</td>
</tr>
<tr>
<td>APSSO (FCM initialization) (Ali and Modares, 2011)</td>
<td>-</td>
<td>-</td>
<td>all candidate variables</td>
<td>135.9</td>
<td>13.75</td>
</tr>
<tr>
<td>UFCA-APSO (Ali and Modares, 2011)</td>
<td>-</td>
<td>-</td>
<td>all candidate variables</td>
<td>97.6</td>
<td>14.88</td>
</tr>
<tr>
<td>ELM (Huang et al., 2012)</td>
<td>-</td>
<td>-</td>
<td>all candidate variables</td>
<td>2142.2</td>
<td>9.953</td>
</tr>
<tr>
<td>HGA (Mendes et al., 2012)</td>
<td>32</td>
<td>13</td>
<td>$u_1(t-2), u_2(t-3), u_1(t-1), u_2(t-2), u_1(t-1)$</td>
<td>74.1</td>
<td>24.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$u_3(t-3), u_1(t-3), u_2(t-1), u_1(t-1)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$u_2(t-1), u_1(t-1), u_2(t-1), u_1(t-1), u_2(t-2)$</td>
<td>44.5</td>
<td>13.07</td>
</tr>
</tbody>
</table>

Figure 4 illustrates the predicted values obtained by the proposed method and desired (real) values of the target variable $y = D_{OC}$ to be estimated, for the data set of the WWTP experiment. Figure 4b presents a magnified view comparing the real output and predicted output variables corresponding to the rectangle “R” depicted inside Figure 4a. Numerical results, with the best fitness function, comparing the performance of the proposed method and the works FCM-RLS (Kulhavý, 1987), RPLS (Dayal and MacGregor, 1997), MLPNN (Nelles, 2001), NFCRMA (Li et al., 2009), APSSO (Ali and Modares, 2011), ELM (Huang et al., 2012), and HGA (Mendes et al., 2012) are presented in Table 2. For a test dataset with $L$ samples, define $J_{best} = 1/MSE$ as the fitness function, and $MSE = \frac{1}{L} \sum_{k=1}^{L} (y(k) - y_{re}(k))^2$, where $y_{re}$ is the real plant output value. The main objective here is to obtain a large value of $J_{best}$ which corresponds to the minimum of the MSE. Based on the results in Table 2, the largest value of the fitness function in the test data set is obtained with the method proposed in this paper. Also, comparing with the HGA, the proposed method uses a lower number of variables and fuzzy rules, but shows a better prediction performance and faster learning performance. The computational time (CT) of each of the used methods are given in Table 2. With a Intel(R) core (TM) i7-2600 and CPU 3.4GHz, the simulation time of the proposed algorithm was 53% of the simulation time of the HGA. Also, from the results in Table 2 it is seen that 8 input variables $u_2(t-1), u_3(t-1), u_3(t-3), u_4(t-1), u_3(t-2), u_2(t-1), w(t-2)$, and $w(t-3)$, were selected by the proposed methodology (among all 27 the input (variable,delay) pair candidates that were considered) for the T-S fuzzy model of the BSMM1 process that was learned. The membership functions extracted by Level 2 for variables selected by Level 1 for the whole T-S fuzzy model obtained by the HPSO.
MLPNN method was run based on two different types of iterations. The method has an external iterative cycle, with a maximum number of 1000 iterations for this example which is considered as a stop criterion for the MLPNN. Also the MLPNN learning is by the Levenberg-Marquardt method which has an inner additive iteration to update neural network weights. Updated neural network weights which result from the inner iteration will initialize the weights of the next external iteration. Implementation of the HGA was also accompanied with more iterations than was considered firstly. The method was initially programmed for 1000 iterations first. However, due to the random initialization, the result by the HGA method just depicted a fitness evaluation obtained by the Algorithm 3, and shown in Table 2, is simply obtained with just one test of the proposed UFCA method. A final point should be mentioned regarding to the efficiency of the UFCA method: based on results in Table 2, the efficient role of the UFCA to enhance performance of other regressors is also remarkable.

In another test, it is investigated the robustness of Algorithm 3 to changes in the system being modeled. Normally, the oxygen transfer coefficient in compartment 4 has been constant $K_{la,4} = 10 \text{ [h}^{-1}] = 240 \text{ [days}^{-1}]$ for $0 \leq d \leq 14 \text{ [days]}$. However, now a disturbance $d_{K_{la,4}}(k) = -240 \text{ [days}^{-1}]$ is defined as a change in the oxygen transfer coefficient in compartment 4 to $K_{la,4} = 0 \text{ [days}^{-1}]$, where the change is in effect during $7 \leq d \leq 9 \text{ [days]}$. Also the external carbon (EC) source to reactor 5 which was constant equal to $0 \text{ [kg COD.days}^{-1}]$ (COD is the abbreviation for chemical oxygen demand), for $0 \leq d \leq 14 \text{ [days]}$, now is changed to a disturbance value of $d_{Carbon5}(k) = 20 \text{ [kg COD.days}^{-1}]$, where the change is in effect during $11 \leq d \leq 13 \text{ [days]}$. Figure 6, shows the effect of the disturbances on the BSM1 during the application of the Algorithm 3. In Figure 6, values of both $d_{K_{la,4}}(k)$ and $d_{Carbon5}(k)$ are presented after multiplying them by $-0.01$ and $0.1$ factors, respectively. The results reveal that using Algorithm 3, the T-S fuzzy model constructed by S-AHPSO can be adapted well in response to changes in the plant and can track the output accurately.

![Figure 5: Membership functions of the partition set of the selected variables obtained by the HPSO method with random initialization and by using the UFCA as the initialization algorithm.](image-url)
6. Conclusion

An adaptive methodology for online learning of Takagi-Sugeno (T-S) fuzzy models based on a hierarchical particle swarm optimization (HPSO) algorithm was proposed. To improve the convergence time of the proposed HPSO, particles positions are initialized by using an efficient unsupervised fuzzy clustering algorithm (UFCA). While the HPSO was proposed to automatically extract all the parameters of the T-S fuzzy model from a set of input/output data, as another contribution of this paper, a self-adaptive HPSO (S-AHPSO) algorithm was proposed for online identification of T-S fuzzy models. To validate and demonstrate the performance and effectiveness of the proposed algorithms, identification of the dissolved oxygen in an activated sludge reactor within a wastewater treatment plant (WWTP) was studied. The results show that the proposed method successfully extracted all parameters of the T-S fuzzy model. The proposed framework can extract a model that represents the dynamics of a plant, using only a data set of the process, where the model can be further used to estimate the output of the plant. Moreover, the results reveal superior performance of the proposed methods when compared with other state of the art methods. As a future work, the fuzzy modeling methodology proposed in this paper will be used to design an adaptive fuzzy generalized predictive controller as a proposed solution to address the current concerns in control design inside of BSM1 WWTP environment.

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).

Saeid Rastegar was supported by Fundação para a Ciência e a Tecnologia (FCT) under grant SFRH/BD/89186/2012, and Jérôme Mendes was supported by FCT under grants SFRH/BD/63383/2009 and SFRH/BPD/99708/2014.

References


