

Evolutionary Learning of a Fuzzy Controller for Industrial Processes

Jérôme Mendes*, Rui Araújo*, Tiago Matias*, Ricardo Seco* and Carlos Belchior*

*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

jermendes@isr.uc.pt, rui@isr.uc.pt, tmatias@isr.uc.pt, cabelchior@isr.uc.pt

Abstract—The paper proposes a new framework to learn a Fuzzy Logic Controller (FLC), from data extracted from a process while it is being manually controlled, in order to control nonlinear industrial processes. The learning of the FLC is performed by a hierarchical genetic algorithm (HGA). First, the fuzzy *c*-means (FCM) clustering algorithm is applied to initialize the HGA population, in order to reduce the computational cost and increase the performance of the HGA. The HGA is composed by five hierarchical levels and it is an automatic tool since it does not require any prior knowledge concerning the structure (e.g. the number of rules) and the database (e.g. antecedent and consequent fuzzy sets) of the FLC, and concerning the selection of the adequate input variables and their respective time delays. After the extraction of the FLC by the proposed method, in order to obtain a better control results, if necessary, the learned FLC can be improved manually by using the information transmitted by a human operator, and/or the learned FLC could be easily applied to initialize the required fuzzy knowledge-base of adaptive controllers. In order to improve the results of the learned FLC, a direct adaptive fuzzy controller is applied. Moreover, the proposed method is applied on control of the dissolved oxygen in an activated sludge reactor within a simulated wastewater treatment plant. The results are presented, showing that the proposed method successfully extracted the parameters of the FLC.

I. INTRODUCTION

Fuzzy Logic Control (FLC) has been used for extensive application in a wide variety of industrial systems and consumer products and has attracted the attention of many researchers. A major application of fuzzy logic theory has been in control of nonlinear systems, which are typically difficult to model and control. If the mathematical model of the plant is known, then in many cases conventional control may be used to provide a solution. On the other hand, FLC should be used in situations where a mathematical model is poorly understood or is unknown, and where expert human knowledge (e.g. from experienced operators) is available and can describe the control of the plant. However, there still exist many difficulties in designing FLCs to solve certain complex nonlinear problems.

In general, it is not easy to determine the most suitable fuzzy rules and membership functions to control the output of a plant, when the only available knowledge concerning the process is the empirical information transmitted by a human operator. Thus, a major challenge in current fuzzy control research is translating human empirical knowledge into FLCs. Genetic Algorithms (GAs) are an important optimization tool which have proved to be useful in solving a variety of search and optimization problems and provide a robust search with the ability to find near optimal solutions in complex and large search spaces [1], [2]. Thus, GAs are a useful soft

computing technique to design a FLC from data extracted from a given process while it is being manually controlled. A hierarchical genetic algorithm (HGA) will be used instead of a GA with just one optimization level due to the complexity of the problem (optimization problems become more difficult to solve when the dimensionality increases), and because when more complex design decisions involving a large number of parameters must be made, a global formulation of the problem representing all the parameters in just one optimization level can be inadequate.

GAs have been applied to tune fuzzy systems, which have been previously employed to select adequate sets of membership functions and fuzzy rules, as it is done in [3], [4], [5]. However, they have a common limitation, which is 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 [6].

A HGA approach with five levels used to optimize the parameters of a FLC is proposed in [7]. In the first level, the input variables and respective delays are chosen with the goal of attaining the highest possible learning accuracy of the FLC. The selection of variables and delays is performed jointly with the learning of the FLC, which increases the global optimization performance. The second level represents the population of the antecedent and consequent membership functions which constitute the fuzzy control rules. The individual rules are defined at the third level. The population of the set of rules is defined in the fourth level, and a population of fuzzy systems is treated at the fifth level.

The present paper is an improvement on the previous work [7]. The main advances and differences contemplated in this work are (1) the improvement of the initialization algorithm, (2) the different representation of the membership functions on Level 2, (3) the improvement of the fitness functions of the Levels 1 and 4, and (4) the different representation of Level 5 in order to integrate an adaptive approach. The initialization algorithm proposed in this work is based on [8] and on a fuzzy *c*-means (FCM) clustering algorithm [9], [10], and its advantage is the design of the antecedent membership functions taking into account the data by using a FCM clustering algorithm, thus the antecedent membership functions are not distributed uniformly over the respective universe of discourse as it is done in [7]. On Level 2, a different representation of the antecedent and consequent membership functions was done to

improve the design/learning of the membership functions. This is motivated by the fact that in [7] on Level 2, when an allele of a membership functions is changed, this modification affects all the following membership functions which decreases the performance of the learning. In this paper, Levels 1 and 4 also have different fitness functions that penalize more complex individuals in order to avoid overparameterization, i.e. penalizes the individuals of the HGA with more input variables/delays pairs (Level 1) and with more fuzzy rules (Level 4). The choice of the t -norm, implication, aggregation, and defuzzification operators on Level 5 is not performed by the HGA as in [7] in order to make possible the use of an adaptation law, being the objective the improvement of the results of the learned FLC. The considered operators are the product t -norm, the Mamdani product implication, and the center-average defuzzification.

In order to validate and demonstrate the performance and effectiveness of the proposed algorithm, it is applied in the control of the dissolved oxygen in an activated sludge reactor within a simulated wastewater treatment plant (WWTP).

The paper is organized as follows. Section II introduces the fuzzy logic system and its direct adaptive law. Section III presents an initialization method based on FCM. The proposed HGA is described in Section IV. The application of the HGA and the respective results are presented and analyzed in Section V. Finally, remarks and conclusions are made in Section VI.

II. FUZZY LOGIC CONTROL

This section briefly overviews the main concepts of fuzzy systems, which are knowledge-based systems defined by a set of IF-THEN rules. The following is an example of such a rule:

$$\begin{aligned} &\text{IF the } \textit{speed} \text{ of a car is } \textit{high}, \\ &\text{THEN apply } \textit{less force} \text{ to the accelerator,} \end{aligned} \quad (1)$$

where *speed* and *force* are input and output variables, respectively. These variables are defined by semantic terms, associated to fuzzy sets *high* and *less*, respectively. A fuzzy set A is characterized by a mapping $\mu_A(x) = \mathbf{U} \rightarrow [0, 1]$. The knowledge base consists a rule base composed of a set of N fuzzy IF-THEN rules R_i of the form

$$\begin{aligned} R_i : & \text{ IF } x_1 \text{ is } A_1^i, \text{ and } \dots \text{ and } x_n \text{ is } A_n^i \\ & \text{ THEN } u \text{ is } B_i, \end{aligned} \quad (2)$$

where $i = 1, \dots, N$, x_j ($j = 1, \dots, n$) are input variables of the fuzzy system, u is the output of the fuzzy system. and A_j^i and B_i are linguistic terms characterized by fuzzy membership functions $\mu_{A_j^i}(x_j)$ and $\mu_{B_i}(u)$, respectively.

The fuzzy logic system used on fuzzy controller presented in this paper is composed by: singleton fuzzifier, center-average defuzzifier, and the product inference engine. In these conditions, considering rules (2), the fuzzy system implements the following function:

$$u(\mathbf{x}) = \frac{\sum_{i=1}^N \bar{b}_i \left(\prod_{j=1}^n \mu_{A_j^i}(x_j) \right)}{\sum_{i=1}^N \prod_{j=1}^n \mu_{A_j^i}(x_j)}, \quad (3)$$

where \bar{b}_i is the center of B_i . Equation (3) can be rewritten as:

$$u(\mathbf{x}) = \boldsymbol{\theta}^T \boldsymbol{\xi}(\mathbf{x}), \quad (4)$$

where $\boldsymbol{\theta} = [\bar{b}_1, \dots, \bar{b}_N]^T$ is a parameter vector with adjustable parameters, $\mathbf{x} = [x_1, x_2, \dots, x_n]$, and $\boldsymbol{\xi}(\mathbf{x}) = [\xi_1(\mathbf{x}), \dots, \xi_N(\mathbf{x})]^T$ is a vector defined as:

$$\xi_i(\mathbf{x}) = \frac{\prod_{j=1}^n \mu_{A_j^i}(x_j)}{\sum_{i=1}^N \prod_{j=1}^n \mu_{A_j^i}(x_j)}, \quad i = 1, \dots, N. \quad (5)$$

A. Direct Adaptive Fuzzy Control

In order to obtain a better control performance, the FLC learned by the proposed method (HGA) can be improved manually by using the information transmitted by a human operator, and/or by adaptive methodologies. In this paper the adaptive methodology used was the direct adaptive fuzzy control suggested in [11].

The adaptive law is derived by Lyapunov synthesis. For more details, references [11], [12], are recommended. The following Lyapunov function is used [12]:

$$V = \frac{1}{2} \mathbf{e}^T \mathbf{P} \mathbf{e} + \frac{1}{2\gamma} (\boldsymbol{\theta}^* - \boldsymbol{\theta})^T (\boldsymbol{\theta}^* - \boldsymbol{\theta}), \quad (6)$$

where γ is a positive constant, \mathbf{e} is the closed loop error defined in [12], and $\boldsymbol{\theta}^*$ are the optimal values of the parameters $\boldsymbol{\theta}$ that optimize the min-max approximation error between u and the ideal control u^* . \mathbf{P} is a positive-definite matrix satisfying the following Lyapunov equation:

$$\boldsymbol{\Lambda}^T \mathbf{P} + \mathbf{P} \boldsymbol{\Lambda} = -\mathbf{Q}, \quad (7)$$

where \mathbf{Q} is an arbitrary positive-definite matrix, and $\boldsymbol{\Lambda}$ is a matrix that can be designed to attain the desired error dynamics [12]. The employed adaptation law is [12]:

$$\dot{\boldsymbol{\theta}} = \gamma \mathbf{e}^T \mathbf{p} \boldsymbol{\xi}(\mathbf{x}), \quad (8)$$

where \mathbf{p} is the last column of \mathbf{P} .

III. INITIALIZATION METHOD

GAs are usually initialized with random population elements, an approach leading to a large tuning/search difficulty on the GA, requiring more iterations to attain convergence than other possible more efficient approaches. Therefore, in order to reduce the computational cost and increase the algorithm's performance, an initialization algorithm based on the algorithm proposed in [8] and on a fuzzy c -means (FCM) clustering algorithm [9], [10] is applied.

A. Fuzzy c -Means Clustering

The objective of the fuzzy c -means (FCM) clustering algorithm is the partitioning of the dataset X into a predefined number of clusters, N (typically named as c). Consider n samples which compose an observation l (one sample of each input variable), which are grouped as an n -dimensional vector $\mathbf{x}_l = [x_{l1}, \dots, x_{ln}]^T$, where $\mathbf{x}_l \in \mathbb{R}^n$. A set of L observations is then denoted as

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

The fuzzy partition of the set \mathbf{X} into N clusters, is a family of fuzzy subsets $\{A^i \mid 1 \leq i \leq N\}$. The membership functions of

Algorithm 1 FCM algorithm.

- 1) Obtain a dataset \mathbf{X} (9) and define number of clusters N , degree of fuzziness η , and the stop conditions $\epsilon > 0$ and Max . Initialize the partition matrix \mathbf{U} , randomly; Let the initial vectors of cluster centers be denoted by $\mathbf{v}_i^{(0)}$;
- 2) Find initial cluster centers using (13) with the membership values of initial partition matrix \mathbf{U} ;
- 3) For iteration $t = 1, \dots, Max$ do:
 - a) Using (11), calculate membership values at iteration t , $\mu_i^{(t)}(l)$, of each input data object \mathbf{x}_l in cluster i , using the cluster center vector $\mathbf{v}_i^{(t-1)}(l)$, from iteration $(t-1)$; Let $\mathbf{U} \leftarrow [\mu_i^{(t)}(l)]$;
 - b) Calculate cluster center of each cluster i at iteration t , $\mathbf{v}_i^{(t)}(l)$, by (13), using the membership values (11) at iteration t , $\mu_i^{(t)}(l)$; Let $\mathbf{V} \leftarrow [\mathbf{v}_1^{(t)}, \dots, \mathbf{v}_N^{(t)}]$;
 - c) Exit the 'For' cycle if a termination condition satisfied, e.g., $|\mathbf{v}_i^{(t)} - \mathbf{v}_i^{(t-1)}| \leq \epsilon$, and save the last iteration of the matrices \mathbf{U} and \mathbf{V} . Otherwise let $t \leftarrow t + 1$ and go to step 3a;
- 4) Compute the parameters σ_i using (14).

these fuzzy subsets are defined as $\mu_i(l) = \mu_{A^i}(\mathbf{x}_l)$, and form the fuzzy partition matrix $\mathbf{U} = [u_{il}] = [\mu_i(l)] \in \mathbb{R}^{N \times L}$. The i -th row of the matrix \mathbf{U} contains the values of the membership function of the i -th fuzzy subset A^i for all the observations belonging to the data matrix \mathbf{X} .

In order to find the fuzzy clusters in the dataset \mathbf{X} , the following equation is minimized:

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

where \mathbf{V} is a matrix of cluster centroid vectors $\mathbf{v}_i = [v_{i1}, \dots, v_{in}]^T$, $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_N]^T \in \mathbb{R}^{N \times n}$, d_{il} is the Euclidean distance (l^2 -norm) between the observation \mathbf{x}_l and the cluster centroid \mathbf{v}_i , and the overlapping factor is denoted as η .

Minimizing the objective function (10), the optimum membership values are calculated as follows [10]:

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

where

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

and $\mathbf{v}_i = [v_{i1}, \dots, v_{in}]^T$,

$$v_{ij} = \frac{\sum_{l=1}^L \mu_i^\eta(l) x_{lj}}{\sum_{l=1}^L \mu_i^\eta(l)}, \quad j = 1, \dots, n. \quad (13)$$

To finalize the identification of the premise parameters, the $\sigma_i = [\sigma_{i1}, \dots, \sigma_{in}]^T$, $i = 1, \dots, N$, can be calculated using $\mathbf{U} = [\mu_i(l)]$, as follows:

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

The FCM algorithm is presented in Algorithm 1, and the initialization algorithm of the antecedent and consequent membership functions is presented in Algorithm 2.

Algorithm 2 INICONTROL-FCM - Initialization algorithm of the antecedent and consequent membership functions to be used on the AHGA-Control methodology.

- 1) For the sole role of specification of this Algorithm 2, let m be the total number of input variables used in the training and operation of the HGA. In the rest of the paper, this number is denoted by n ;
- 2) For all sampling instances $l = 1, \dots, L$, apply the input vector $\mathbf{x}_l = [x_{l1}, \dots, x_{lm}]^T$ to the system to be modeled (in this case the controller) and obtain the data sample $(\mathbf{x}_l, u(l))$, where $u(l)$ is the output scalar;
- 3) Construct a data set containing all the data samples $(\mathbf{x}_l, u(l))$;
- 4) Using the collected data samples $(\mathbf{x}_l, u(l))$, select a set of variables $X_s = \{x_{v_1}(t), \dots, x_{v_n}(t)\}$ which will be used to initialize the HGA; $v_j \in \{1, \dots, m\}$, for $j = 1, \dots, n \leq m$; For the sole role of specification of this Algorithm 2, renumber the input variables such that X_s becomes $X_s = \{x_1(t), x_2(t), \dots, x_n(t)\}$;
- 5) Obtain the antecedent parameters of each rule i , namely the vector of centers $\mathbf{v}_i = [v_{i1}, \dots, v_{in}]^T$ (13) and the vector of dispersions $\sigma_i = [\sigma_{i1}, \dots, \sigma_{in}]^T$ (14) of the antecedent membership functions, for $i = 1, \dots, N$, using Algorithm 1, where N is the number of clusters; Using \mathbf{v}_i and σ_i , each Gaussian antecedent membership function $\mu_{A_j^i}$ becomes defined;
- 6) For $i = 1, \dots, N$ do:
 - a) Obtain the centers, b_i , of the consequent membership functions of the i -th fuzzy rule by using the heuristic method,
$$b_i = \frac{\left(\sum_{l=1}^L u(l) \prod_{j=1}^n \mu_{A_j^i}(x_{lj}) \right)}{\left(\sum_{l=1}^L \prod_{j=1}^n \mu_{A_j^i}(x_{lj}) \right)}, \quad (15)$$
where $u(l)$ is the observation l of the output variable u , and $\mu_{A_j^i}(x_{lj})$ is the antecedent fuzzy membership function used for variable x_j on the i -th fuzzy rule;
- 7) Build the output membership functions assuming (for example) a triangular shape where the centers are given by b_i , and the triangular aperture can be represented by $(u^+ - u^-) / (N.g)$, where u^+ and u^- are the limits of the output universe of discourse, and is g a scaling factor (e.g. $g = 2, 4, 8, \dots$).

IV. HIERARCHICAL GENETIC ALGORITHM

This section describes the proposed algorithm, which is an automatic method based on a HGA constituted by five hierarchical levels (Figure 1), for the extraction of all fuzzy parameters of a FLC from a dataset obtained from an existing controller (human or automatic). The detailed description of each level is given below:

Level 1: it is formed by the population of the set of input variables, and respective delays, which will be used to design the fuzzy controller. Its chromosome is represented by binary encoding, where each allele (element of the chromosome located in a specific position) corresponds to each variable/delay that is considered as possible candidate to be used as input for the fuzzy system (see Figure 1). The fitness function of this level is given by

$$J_1^m = \frac{\max(J_5^{d_1}, \dots, J_5^{d_s})}{1 + \frac{nVar}{nTotalVar}}, \quad (16)$$

where $\{d_1, \dots, d_s\} \subseteq \{1, \dots, i_{max}\}$ is the subset of all chromosomes of Level 5 that contain the m -th selection of inputs and delays on allele 3 in Level 5, i_{max} is the maximum

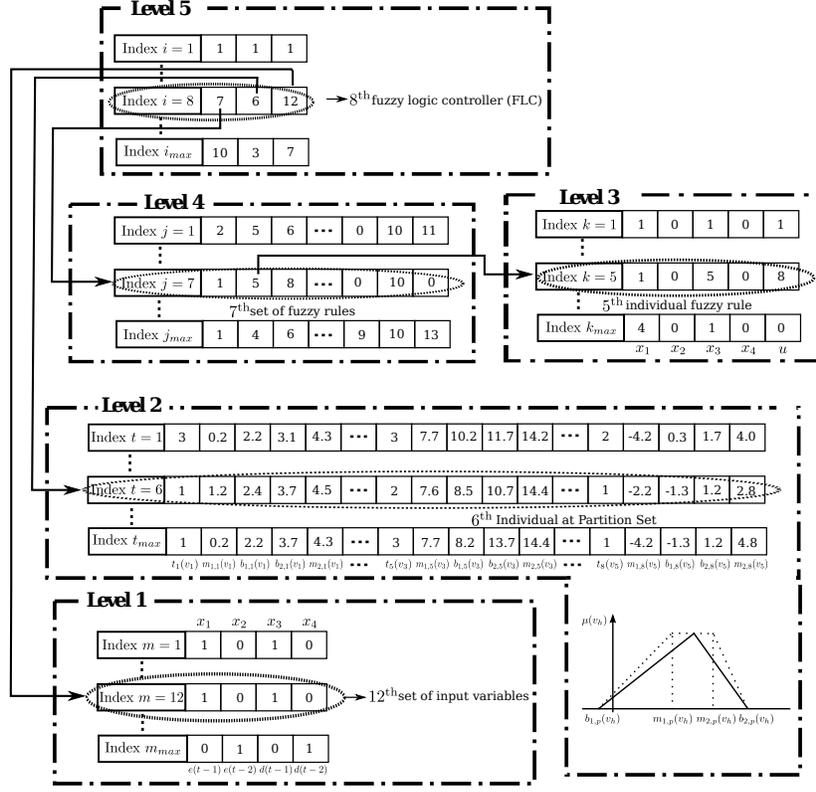


Figure 1. Encoding and hierarchical relations among the individuals of different levels of the genetic hierarchy.

number of chromosomes of Level 5, $nVar$ is the number of variables of the m -th chromosome of Level 1, and $nTotalVar$ is the total number of candidate variables.

Level 2: it is formed by the aggregations of all partition sets (collection of fuzzy sets) associated with each input and output variable. An example of its structure can be seen in Figure 1, where the first allele uses integer encoding to represent the type of membership function: trapezoidal ($t_p(v_h) = 1$), triangular ($t_p(v_h) = 2$) and Gaussian ($t_p(v_h) = 3$); and the following four alleles ($m_{1,p}(v_h)$, $b_{1,p}(v_h)$, $b_{2,p}(v_h)$ and $m_{2,p}(v_h)$) use real encoding to represent the parameters of the membership function p ($p = 1, \dots, K_h$) of an input or output variable v_h , ($h = 1, \dots, n+1$), where v_h , for $h = 1, \dots, n$, correspond to input variables, and v_{n+1} corresponds to the output variable, and K_h is the number of individual membership functions defined for variable v_h . The parameters of trapezoidal are represented in Figure 1. For triangular and Gaussian membership functions, the center is found by the average between $m_{1,p}(v_h)$ and $m_{2,p}(v_h)$, i.e., $m_p(v_h) = (m_{1,p}(v_h) + m_{2,p}(v_h))/2$, and the dispersion of Gaussian functions is given by $\sigma_p(v_h) = (b_{2,p}(v_h) - b_{1,p}(v_h))/6$. The fitness function of this level is

$$J_2^t = \max(J_5^{c_1}, \dots, J_5^{c_r}), \quad (17)$$

where $\{c_1, \dots, c_r\} \subseteq \{1, \dots, i_{max}\}$ is the subset of all chromosomes of Level 5 that contain partition set t on allele 2 of Level 5.

Level 3: it is constituted by a population of individual rules. The length of the chromosome is determined by the number

of input variables selected by Level 1, plus an additional allele that characterizes the output variable. The chromosome is represented by integer encoding, where each allele contains the index of the corresponding antecedent and consequent membership function. Null values indicate the absence of membership function. The fitness function of this level is given by

$$J_3^k = \max(J_4^{b_1}, \dots, J_4^{b_q}), \quad (18)$$

where $\{b_1, \dots, b_q\} \subseteq \{1, \dots, j_{max}\}$ is the subset of all chromosomes of Level 4 that contain individual rule k , and j_{max} is the maximum number of chromosomes of Level 4.

Level 4: it is constituted by a set of fuzzy rules, where each allele contains the index of the corresponding individual rule of Level 3 that has been included in the set. The chromosome 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 the chromosome is determined by the maximum number of fuzzy rules ($nTotalRules$). The fitness function of this level is given by

$$J_4^j = \frac{\max(J_5^{a_1}, \dots, J_5^{a_p})}{1 + \frac{nRules}{nTotalRules}}, \quad (19)$$

where $\{a_1, \dots, a_p\} \subseteq \{1, \dots, i_{max}\}$ is the subset of all chromosomes of Level 5 that contain rule-base j on allele 1 of Level 5, and $nRules$ is the number of fuzzy rules of j -th chromosome of Level 4.

Level 5: it represents a fuzzy system, i.e., all the information required to develop the fuzzy controller is contemplated on

this level. The chromosome is represented by integer encoding and is constituted by three alleles. The first allele indicates the index, j , of the set of rules specified on Level 4. The second allele contains the t -th partition set given by Level 2. The third allele represents the index, m , of the set of input variables selected on Level 1. The fitness function of this level is given by

$$J_5^i = \frac{1}{\left(1 + \frac{nRules+nVar}{N+nTotalVar}\right)MSE(\mathbf{u}, \hat{\mathbf{u}}^i)}, \quad (20)$$

where $MSE(\mathbf{u}, \hat{\mathbf{u}}^i) = \frac{1}{L} \sum_{l=1}^L (u(l) - \hat{u}^i(l))^2$ is the mean square error between the target control output \mathbf{u} and estimated control output $\hat{\mathbf{u}}^i$ obtained with individual i , and $\left(1 + \frac{nRules+nVar}{N+nTotalVar}\right)$ penalizes the more complex individuals to avoid overparameterization.

An example of the encoding and the hierarchical relations is given in Figure 1. In this example the i -th set of Level 5 indicates that the fuzzy system uses the 7th ($j = 7$) set of fuzzy rules of Level 4 (allele 1), the 6th ($t = 6$) partition set of Level 2 (allele 2), and the 12th ($m = 12$) set of selected input variables and delays in Level 1 (allele 3). The 7th set of fuzzy rules (Level 4) contains the 1st ($k = 1$), 5th ($k = 5$), 8th ($k = 8$), and 10th ($k = 10$) individual rules, where the 5th individual rule (Level 3) is composed by two input variables with linguistic terms 1 and 5, respectively, and one output variable corresponding to linguistic term 8, i.e.:

$$\begin{aligned} R_{15} : \quad & \text{IF } x_1(t) \text{ is "1" and } x_3(t) \text{ is "5"} \\ & \text{THEN } u(t) \text{ is "8"}. \end{aligned} \quad (21)$$

The linguistic terms "1", "5" and "8" are defined in the 6th chromosome of Level 2, and the input variables x_1 and x_3 are determined on Level 1.

The main steps of the GA algorithm used to learn/improve the fuzzy controller parameters are presented in Algorithm 3. The evolutionary operators used are the Roulette Wheel selection method, the Single Point crossover, and Flip Bit mutation or Uniform mutation (according to if the chromosome is binary or not). For more details about the evolutionary operators see [13].

V. RESULTS

This section addresses the application of the HGA for automatic extraction of the fuzzy parameters of a fuzzy controller in order to control the dissolved oxygen (DO) in an activated sludge reactor within a simulated wastewater treatment plant (WWTP) in the Benchmark Simulation Model n.1 (BSM1). BSM1 is a platform-independent simulation environment dedicated to the optimization of performance and cost-effectiveness of wastewater management systems [14]. The proposed method is applied with the aim of determining a controller with a response similar to the one that is being replicated. Note that the aim is to learn an automatic controller output, but it could be to learn the control knowledge of a human operator.

Wastewater treatment plants 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 [15]. A general overview of the BSM1 plant is presented in Figure 2. The

Algorithm 3 Proposed AHGA-Control methodology.

- 1) Compute the antecedent and consequent membership functions parameters by the Algorithm 2 - INICONTROL-FCM;
- 2) Initialize populations of all levels:
 - a) Level 1: initialize with ones the alleles of the first individual that represent the input variables used to initialize the HGA in Algorithm 2 - INICONTROL-FCM (these are the variables that belong to the set X_s). Initialize with zeros all the remaining alleles of the first individual;
 - b) Level 2: initialize all individuals with the antecedent and consequent membership functions computed in step 1;
 - c) Level 3: initialize the first N individuals with the antecedent part of the fuzzy rules which is learned by Algorithm 2. The way this is done is by initializing the first individual with ones, the second individual with twos, until the N -th individual with N 's;
 - d) Level 4: initialize the first individual with the indexes of the first N individuals of Level 3. Initialize with zeros all the remaining alleles;
 - e) Level 5: initialize the first individual of Level 5 with ones;
 - f) The remaining individuals of Levels 1, 3, 4, and 5 are randomly initialized;
- 3) Set Generation $\leftarrow 1$;
- 4) Let i_{max} , j_{max} , k_{max} , t_{max} and m_{max} , be the maximum number of chromosomes at levels 5, 4, 3, 2, and 1, respectively. Compute the fitness of each individual, from Level 5 to Level 1:
 - a) For $i = 1, \dots, i_{max}$, evaluate J_5^i using Eq. (20);
 - b) For $j = 1, \dots, j_{max}$, evaluate J_4^j using Eq. (19);
 - c) For $k = 1, \dots, k_{max}$, evaluate J_3^k using Eq. (18);
 - d) For $t = 1, \dots, t_{max}$, evaluate J_2^t using Eq. (17);
 - e) For $m = 1, \dots, m_{max}$, evaluate J_1^m using Eq. (16);
- 5) If the stop condition does not hold, do for each level:
 - a) Generation \leftarrow Generation + 1;
 - b) Apply the evolutionary operators, on each level, to form a new population: selection, crossover and mutation;
 - c) Replace the current population, on each level, with the new evolved population;
 - d) Return to step 4.
- 6) Online operation: for/using each newly arriving online-sample, do:
 - a) Compute control signal $u(k)$ with (4);
 - b) If it is desirable to improve the results of the FLC learned in Steps 3 to 5, then:
 - i) Adapt the consequent parameters θ using the adaptation law (8) defined in Section II-A.

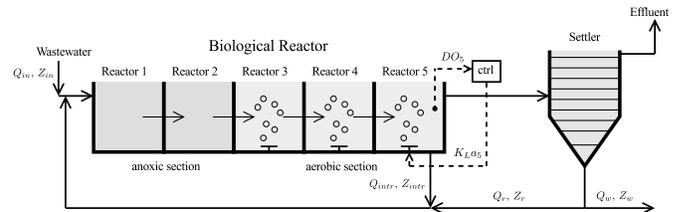


Figure 2. General overview of the BSM1 plant [15].

biological reactor is distributed over five reactors connected in cascade. Reactors 1 and 2 are non-aerated compartments, and reactors 3, 4, and 5 are aerated. Reactors 3 and 4 have a fixed oxygen transfer coefficient, and the DO of reactor 5 should be controlled by manipulation of the oxygen transfer rate, KLa_5 , from an aerator to the activated sludge inside the biological reactor. The DO concentration is measured on

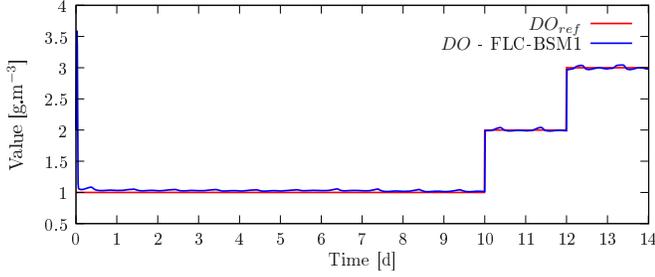


Figure 3. FLC-BSM1 performance used to compile the learning dataset of the BSM1 plant.

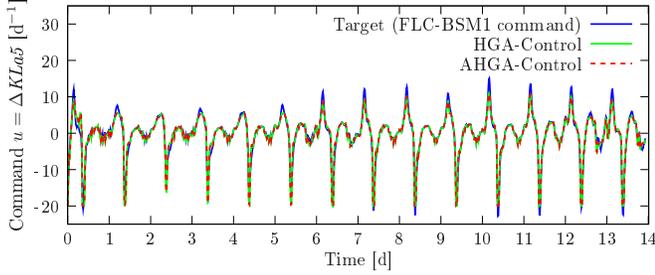


Figure 4. FLC target $\Delta KLa5$ and command signals learned by the proposed AHGA-Control methodology and by the HGA-Control proposed in [7].

reactor 5 and is controlled by manipulation of the $KLa5$ on the same reactor. The sampling period is 15 [min], and the simulations have a maximum of 14 [days].

The results were obtained by considering that the crossover and mutation probabilities are 80% and 10%, respectively, the number of generations is $Gen_{max} = 1000$, and the numbers of chromosomes for each level of the architecture are: $i_{max} = 100$, $j_{max} = 80$, $k_{max} = 200$, $t_{max} = 15$, and $m_{max} = 200$. The methodology proposed in this paper, named as AHGA-Control, is compared with the methodology proposed in [7], named as HGA-Control.

The dataset was obtained, while controlling the BSM1 plant with the FLC described in [15], named as FLC-BSM1, by extracting the incremental command signal, $\Delta KLa5(t)$, the tracking error of the DO concentration, $E(t) = DO_{ref}(t) - DO(t)$, and the derivative of $E(t)$, $\Delta E(t)$, where $DO_{ref}(t)$ is the desired reference for $DO(t)$. The first four delays of $E(t)$ and $\Delta E(t)$, are also included in the learning dataset, i.e. $E(t-1), \dots, E(t-4)$ and $\Delta E(t-1), \dots, \Delta E(t-4)$, allowing a better selection of the FLC's input variables. The input variables were divided into two groups: one group for variables $[E(t), E(t-1), E(t-2), E(t-3), E(t-4)]$ and the other for $[\Delta E(t), \Delta E(t-1), \Delta E(t-2), \Delta E(t-3), \Delta E(t-4)]$. For the AHGA-Control methodology, the number of clusters and the degree of fuzziness were chosen as $N = 20$, and $\eta = 2$, respectively, and for the HGA-Control methodology 5 membership functions were considered for each of the input variables and for the output variable. Also, for both methodologies $X_s = \{E(t), \Delta E(t)\}$ and $g = 5$ were used.

Figure 3 shows the system response obtained with the FLC-BSM1 that was used to construct the dataset. Figure 4 shows the target response (FLC-BSM1) and the responses of

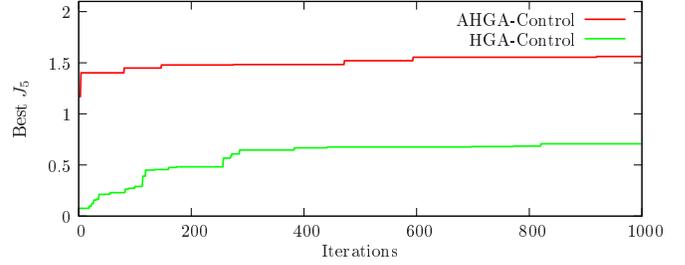


Figure 5. Evolution of the best individual fitness function value along the generations for the HGA-Control methodology and for the proposed AHGA-Control methodology.

Table I
COMPARISON OF RESULTS ON TRAIN DATASET. MAE IS THE MEAN ABSOLUTE ERROR.

Methodology	No. rules	Inputs	1/MSE	1/MAE
HGA-Control	25	$E(t), E(t-1), E(t-2), \Delta E(t-1), \Delta E(t-2)$	0.68	1.43
AHGA-Control	25	$E(t-4), \Delta E(t-3), \Delta E(t-4)$	0.87	1.91

Table II
COMPARISON OF RESULTS FOR THE BSM1 TEST.

Methodology	1/MSE	1/MAE
INICONTROL [7]	10.459	5.13
INICONTROL-FCM	26.341	12.34
HGA-Control [7]	503.61	34.70
AHGA-Control-OFF	740.33	77.09
AHGA-Control	1335.1	947.80

the proposed AHGA-Control methodology and of the HGA-Control method proposed in [7]. In Figure 4 it can be seen that the performed identification was still sufficient to obtain a controller with a response resembling the one that was intended to be achieved. Figure 5 shows the time evolution of the fitness function of the proposed AHGA-Control methodology and of the HGA-Control proposed in [7]. As can be seen in Figure 5 and Table I, show that the AHGA-Control methodology attains faster response and better results when compared to the results obtained by the HGA-Control methodology, and that proposed AHGA-Control has the best initialization that outperforms the initialization proposed in [7], showing the importance of the using a good initialization method.

Figure 6 and Table II show the results, including the time responses, obtained by the FLCs learned by the HGA-Control [7] and by the proposed AHGA-Control methodology, by the corresponding initialization methods INICONTROL [7] and INICONTROL-FCM, and by the AHGA-Control method with the adaptation law turned off until $t = 10$ [days], a method named as AHGA-Control-OFF, for a reference signal different from the one used to generate the data set that was employed for the training, and the respective applied command signals. In Table II, $1/MSE = 1/(\frac{1}{T} \sum_{t=1}^T (DO_{ref}(t) - DO(t))^2)$ and $1/MAE = 1/(\frac{1}{T} \sum_{t=1}^T |DO_{ref}(t) - DO(t)|)$, where T

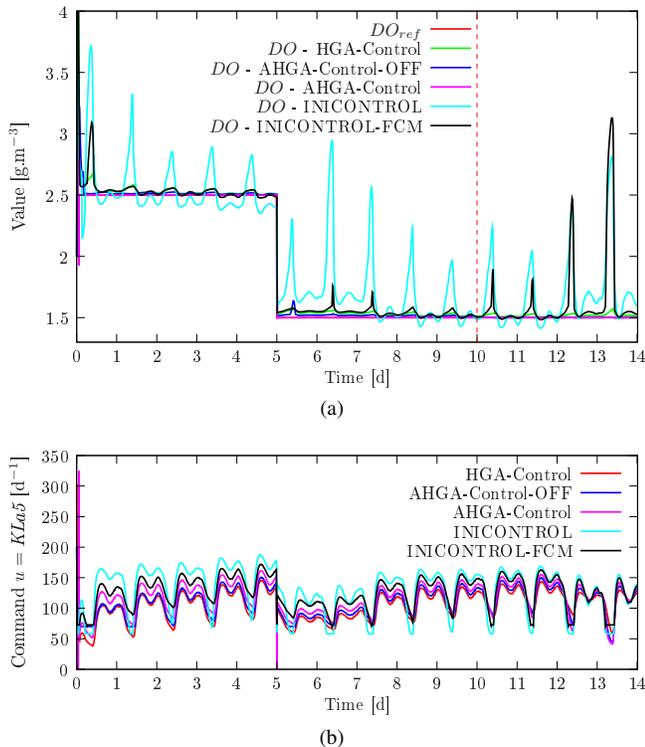


Figure 6. (a) DO results of the HGA-Control [7] and the proposed AHGA-Control, of the respective initialization methods INICONTROL [7] and INICONTROL-FCM, and of the AHGA-Control method with the adaptation law turned off until $t = 10$ [days] (red dashed line), a method named as AHGA-Control-OFF; and (b) the respective applied $KLa5$ command signals.

is the total number of sample on the test. It can be seen that the proposed controller is able to adequately (attain and) control the system output at the desired reference $DO_{ref}(t)$. It can also be seen in Figure 6 and in Table II, on the AHGA-Control-OFF results, that when the AHGA-Control methodology turns on the adaptation law at $t = 10$ [days] (red dashed line in Figure 6) the results are improved, and that the performance of AHGA-Control methodology with its adaptation law always turned on is the best. Thus, the proposed AHGA-Control method outperforms the HGA-Control proposed in [7], and that the performance of the learned FLC can be improved using the adaption law.

VI. CONCLUSION

This paper proposed a new framework to automatically extract a FLC in order to control nonlinear processes. The learning of the FLC is performed by a HGA composed by five levels, using a set of input/output data, previously extracted from a process under control (e.g. by manual control).

The main purpose of the HGA is to develop a FLC with a response similar to the one used to compile the dataset, or in less successful attempts, to develop a controller which constitutes a starting point for further adjustments. In order to obtain a better control error, the proposed algorithm was applied to initialize the required fuzzy knowledge-base of adaptive controllers. Additionally, the method may also be used to understand a process for which there is little or no information available, since it automatically extracts all fuzzy

parameters, and it is able to gather a knowledge-base about the process control.

The control of the dissolved oxygen in an activated sludge reactor within a simulated WWTP plant is studied. The results showed that the proposed method extracted all the parameters of the FLC controlling the process with success.

ACKNOWLEDGMENT

This work was supported by Mais Centro Operational Program, financed by European Regional Development Fund (ERDF), and Agência de Inovação (AdI) under Project SCIAD/2011/21531.



Jérôme Mendes has been supported by Fundação para a Ciência e a Tecnologia (FCT) under Grant SFRH/BD/63383/2009. The authors acknowledge the support of FCT project PEST-C/EEI/UI0048/2014.

REFERENCES

- [1] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, 1st ed., ser. Advances in Fuzzy Systems—Applications and Theory. World Scientific Publishing Co. Pte. Ltd., 2001, vol. 19.
- [2] F. Herrera, “Genetic fuzzy systems: Taxonomy, current research trends and prospects,” *Evolutionary Intelligence*, vol. 1, no. 1, pp. 27–46, 2008.
- [3] N. R. Cazarez-Castro, L. T. Aguilar, and O. Castillo, “Fuzzy logic control with genetic membership function parameters optimization for the output regulation of a servomechanism with nonlinear backlash,” *Expert Systems with Applications*, vol. 37, no. 6, pp. 4368–4378, June 2010.
- [4] R. Coban and B. Can, “A trajectory tracking genetic fuzzy logic controller for nuclear research reactors,” *Energy Conversion and Management*, vol. 51, no. 3, pp. 587–593, March 2010.
- [5] P. G. Kumar, T. A. A. Victoire, P. Renukadevi, and D. Devaraj, “Design of fuzzy expert system for microarray data classification using a novel genetic swarm algorithm,” *Expert Systems with Applications*, vol. 39, no. 2, pp. 1811 – 1821, 2012.
- [6] F. A. Souza, R. Araújo, T. Matias, and J. Mendes, “A multilayer-perceptron based method for variable selection in soft sensor design,” *Journal of Process Control*, vol. 23, no. 10, pp. 1371 – 1378, 2013.
- [7] J. Mendes, R. Araújo, T. Matias, R. Seco, and C. Belchior, “Automatic extraction of the fuzzy control system by a hierarchical genetic algorithm,” *Engineering Applications of Artificial Intelligence (Accepted)*, 2014.
- [8] H. Andersen, A. Lotfi., and A. Tsoi, “A new approach to adaptive fuzzy control: The controller output error method,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 27, no. 4, pp. 686–691, August 1997.
- [9] A. Celikyilmaz and I. B. Trksen, *Modeling Uncertainty with Fuzzy Logic: With Recent Theory and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2009.
- [10] D. Dovžan and I. Škrjanc, “Predictive functional control based on an adaptive fuzzy model of a hybrid semi-batch reactor,” *Control Engineering Practice*, vol. 18, no. 8, pp. 979 – 989, 2010.
- [11] J. Mendes, R. Araújo, P. Sousa, F. Apóstolo, and L. Alves, “An architecture for adaptive fuzzy control in industrial environments,” *Computers in Industry*, vol. 62, no. 3, pp. 364–373, April 2011.
- [12] L.-X. Wang, “Stable adaptive fuzzy controllers with application to inverted pendulum tracking,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 5, pp. 677–691, October 1996.
- [13] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*. Springer, 2007.
- [14] U. Jeppsson and M. Pons, “The COST benchmark simulation model—current state and future perspective,” *Control Engineering Practice*, vol. 12, no. 3, pp. 299 – 304, 2004.
- [15] C. A. C. Belchior, R. A. M. Araújo, and J. A. C. Landeck, “Dissolved oxygen control of the activated sludge wastewater treatment process using stable adaptive fuzzy control,” *Computers & Chemical Engineering*, vol. 37, pp. 152 – 162, 2012.