Online Evolving Fuzzy Control Design: An Application to a CSTR Plant

Jérôme Mendes, Francisco Souza, and Rui Araújo
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
jermendes@isr.uc.pt, fasouza@isr.uc.pt, rui@isr.uc.pt

Abstract—**The paper proposes a methodology to self-evolve an online fuzzy logic controller (FLC). The proposed methodology does not require any initialization at all, it can start with an empty set of fuzzy control rules or with a simple collection of fuzzy control rules obtained from an expert operator. The FLC design is online, using only the input/output data obtained during the normal operation of the system while it is being controlled. The FLC is composed of a simple structure, where each input variable has its own set of fuzzy control rules, and is evaluated individually by the proposed methodology avoiding the high increase in the number of fuzzy control rules. The FLC structure and their antecedent and consequent parameters are both online modified by the proposed methodology. Only simple information about the system and controller is need, specifically the universe of discourse of the input and output variables, an information that is mandatory to control any process. The performance of the proposed methodology is tested on a simulated continuous-stirred tank reactor (CSTR) system where the results show that the proposed methodology has the capability of designing the FLC in order to successfully controlling the CSTR system by evolving/modifying the FLC structure when unknown regions of operation are reached (unknown for the controller).**

I. Introduction

Fuzzy logic controllers (FLCs) have been successfully used in industrial nonlinear control applications, typically in systems which are difficult to model and control, but where human-expert knowledge about the operation of such systems is available [1]. In such approach, the FLCs (designed by human-expert knowledge) need no knowledge about the dynamics/mathematical model of the system to be controlled, and can be applied when human-expert knowledge (e.g. from experienced operators) about the control actions to control the system is available. However, there is no a standard approach to translate the knowledge of the control actions from a human-expert operator into fuzzy control rules [2], and there are several difficulties to design the appropriate fuzzy control rules using just the human-expert knowledge to control complex nonlinear industrial systems.

The automatic design of fuzzy logic systems, during the past several years, has attracted the attention of many researchers with several proposed methods for identification, classification, and control applications. However, just a few research works (when compared with the proposed works for identification and classification problems) have been proposed to learn a FLC, specifically on the design of an entire "standard" FLC (e.g. the fuzzy rules and the respective membership functions of FLCs with a control structure which allows to have results with good interpretability as for example in Mamdani or T-S (Takagi–Sugeno) fuzzy zero order fuzzy controller types). Moreover, less works have been proposed to automatically design direct FLCs. FLCs can be categorized [3] as direct (which are designed using human control knowledge) and indirect (which are designed using human knowledge about the system to be controlled). The focus of this paper is the design of direct FLCs, i.e. the learning of fuzzy control rules and not fuzzy rules to identify the model of the system to be controlled (as done on indirect FLCs).

The design of FLCs have been mainly performed typically by biologically inspired optimization algorithms (e.g. [1], [4], [5], [6]), such as genetic algorithms (GAs), ant colony optimization (ACO), artificial bee colony algorithm (ABC), and particle swarm optimization (PSO). However, using such methods the learning process has large computational costs, being such methods mainly offline which do not consider the changes in the dynamics of the system on the online stage, and/or they typically try learn an existing manual or automatic controller by using offline data which can be difficult to obtain (or even impossible) or can be poor not "describing" the entire global region of operation, and/or they need a model of the system to be controlled. More recently, online evolving methodologies have been proposed for direct FLCs design which do not require knowledge about the dynamics/mathematical model of the system to be controlled, and use available data collected during the process under control. In [2], [7], online evolving methodologies to design direct FLCs which can start with an empty and/or simple structure of fuzzy control rules are proposed. In [2] the evolving process is comprised on two phases, the adaptation of the consequents and the addition of new membership functions (MFs). In [7] the proposed method presents global learning capabilities. Both [2] and [7] present disadvantages in the process of adding MFs, which typically lead to a large (or even exponential) increase in the number of fuzzy control rules.

This paper proposes a new methodology for online self-evolving direct fuzzy logic controller (SEDFLC). The proposed methodology, SEDFLC, does not require knowledge about the dynamics/mathematical model of the system to be controlled, uses only available data collected during the system under control, and only needs the information about the range of the input and output variables (their universes of discourse limits) that are easily to known and mandatory to control any system. The proposed methodology presents a FLC structure which is composed of a set of univariate fuzzy control rules, which leads to a better understanding of the importance/influence of each input variable on the behaviour of the controller, allows a good interpretability of the learned FLC, and avoids the fast/exponentially growth in the number of fuzzy control rules which is a limitation in [2], [7].

The paper is organized as follows. Section II presents a FLC structure to be evolved by the proposed methodology. The proposed methodology is described in Section III. Section IV presents the results of the proposed methodology on a simulated nonlinear continuous-stirred tank reactor (CSTR) plant, and Section V makes concluding remarks.

II. Fuzzy Logic Controller

This section presents the FLC structure to be evolved/learned in this paper. The controller structure used in this paper is a zero order Takagi-Sugeno (T-S) fuzzy system, however reorganized to a structure similar to a neo-fuzzy neuron system. This section will start to briefly review the zero order T-S fuzzy and neo-fuzzy neuron systems, and then will present the FLC structure to be evolved in this paper.

A. Zero Order T-S Fuzzy and Neo-Fuzzy Neuron Systems

This section briefly reviews the zero order Takagi-Sugeno (T-S) fuzzy system and the neo-fuzzy neuron system. For more details the references [3], [8] are recommended.

Being T-S fuzzy systems universal approximators [9], for one input variable $x(k)$ a zero order T-S fuzzy system can be defined as being composed of the following set of fuzzy rules:

$$R^i: \quad \text{IF } x(k) \text{ is } A^i \text{ THEN } y^i(k) = q^i, \tag{1}$$

where R^i $(i = 1, 2, \ldots, N)$ represents the i-th fuzzy rule of the zero order T-S fuzzy system, N is the total number of rules, x and y are the input and output variables, A^i is a linguistic term characterized by fuzzy membership function $\mu_{A^i}(x)$ for the input variable x, and q^i is scalar value which represents the consequent parameter of $y^i(k)$.

Using a zero order T-S fuzzy system composed of the set of rules (1), and the singleton fuzzifier, center-average defuzzifier, and product inference engine [3], $y(k)$ can be rewritten as

$$y(k) = \sum_{i=1}^{N} \omega^i[x(k)]q^i = \boldsymbol{\psi}^T(x(k))\mathbf{q}, \tag{2}$$

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

$$\omega^i[x(k)] = \frac{\mu_{A^i}(x(k))}{\sum_{i=1}^{N} \mu_{A^i}(x(k))}, \tag{3}$$

$$\mathbf{q} = \left[q^1, \ldots, q^N\right]^T, \tag{4}$$

$$\boldsymbol{\psi}(x(k)) = \left[\omega^1[x(k)], \ldots, \omega^N[x(k)]\right]^T. \tag{5}$$

A neo-fuzzy neuron system can be seen as the addition of a set of zero order T-S fuzzy systems, each one representing a mapping between one input variable x_j (for $j = 1, 2, \ldots, n$) and the output. In this way, the neo-fuzzy neuron system can be represented in the following form

$$\begin{aligned} y(\mathbf{x}) &= f_1(x_1) + \ldots + f_n(x_n), \\ &= \sum_{j=1}^{n} f_j(x_j) = \sum_{j=1}^{n} y_j(x_j), \end{aligned} \tag{6}$$

where x_j $(j = 1, \ldots, n)$ are the input variables, and each individual zero order T-S fuzzy system $y_j(x_j) = f_j(x_j)$ $(j = 1, 2, \ldots, n)$ is composed of N_j zero order T-S fuzzy rules of the form (1).

B. Fuzzy Controller Structure

Similarly to a neo-fuzzy neuron system which can be seen as the addition of a set of zero order T-S fuzzy systems (Section II-A), the controller structure used in this paper can also be seen as the addition of a set of zero order T-S fuzzy systems, but instead of a "simple" addition is used a weighted addition, where the weight is given by the importance of a given input variable in a set of fuzzy rules that includes all input variables.

To allow a better understanding of the importance/influence of each input variable x_j on the behaviour of the controller, each fuzzy control rule will only describe the control action as a function of one input variable x_j. In this way, the FLC to be evolved in this paper will be composed of a set of univariate fuzzy control rules in the following format:

$$\begin{aligned} R_1^1: &\quad \text{IF } x_1(k) \text{ is } A_1^1 \text{ THEN } u_1^1(k) = q_1^1, \\ &\vdots \\ R_1^{N_1}: &\quad \text{IF } x_1(k) \text{ is } A_1^{N_1} \text{ THEN } u_1^{N_1}(k) = q_1^{N_1}, \\ &\vdots \\ R_j^1: &\quad \text{IF } x_j(k) \text{ is } A_j^1 \text{ THEN } u_j^1(k) = q_j^1, \\ &\vdots \\ R_j^{N_j}: &\quad \text{IF } x_j(k) \text{ is } A_j^{N_j} \text{ THEN } u_j^{N_j}(k) = q_j^{N_j}, \\ &\vdots \\ R_n^1: &\quad \text{IF } x_n(k) \text{ is } A_n^1 \text{ THEN } u_n^1(k) = q_n^1, \\ &\vdots \\ R_n^{N_n}: &\quad \text{IF } x_n(k) \text{ is } A_n^{N_n} \text{ THEN } u_n^{N_n}(k) = q_n^{N_n}, \\ &\quad i = 1, 2, \ldots, N_j, \text{ and } j = 1, 2, \ldots, n, \end{aligned} \tag{7}$$

where $R_j^{i_j}$ $(j = 1, 2, \ldots, n$, and $i_j = 1, 2, \ldots, N_j)$ represents the i_j-th fuzzy control rule of the variable j, N_j is the total number of fuzzy control rules for the variable j, $A_j^{i_j}$ are linguistic terms characterized by fuzzy membership functions $\mu_{A_j^{i_j}}(x_j)$ which describe the local operating regions for the input variable x_j, and $q_j^{i_j}$ is a scalar value which represents the controller (consequent) parameter of $u_j^{i_j}(k)$. The total number of fuzzy rules is given by $N = \sum_{j=1}^{n} N_j$.

Similarly to (2), using a singleton fuzzifier, center-average defuzzifier, and product inference engine [3], and using the fuzzy control rules in form of (7), the fuzzy controller output $u(\mathbf{x}(k))$ can be written as

$$u(\mathbf{x}(k)) = \frac{\sum_{i_1=1}^{N_1} \mu_{A_1^{i_1}}(x_1(k))q_1^{i_1} + \ldots + \sum_{i_n=1}^{N_n} \mu_{A_n^{i_n}}(x_n(k))q_n^{i_n}}{\sum_{i_1=1}^{N_1} \mu_{A_1^{i_1}}(x_1(k)) + \ldots + \sum_{i_n=1}^{N_n} \mu_{A_n^{i_n}}(x_n(k))}, \tag{8}$$

and defining $A^1, \ldots, A^i, \ldots, A^N = A_1^1, \ldots, A_1^{N_1}, \ldots, A_j^1, \ldots, A_j^{N_j}, \ldots, A_n^1, \ldots, A_n^{N_n}$,

$$\sum_{i=1}^{N} \mu_{A^i}(\mathbf{x}(k)) = \sum_{i_1=1}^{N_1} \mu_{A_1^{i_1}}(x_1(k)) + \ldots + \sum_{i_n=1}^{N_n} \mu_{A_n^{i_n}}(x_n(k)), \tag{9}$$

and

$$\omega_j^{i_j}[x_j(k)] = \frac{\mu_{A_j^{i_j}}(x_j(k))}{\sum_{i=1}^{N} \mu_{A^i}(\mathbf{x}(k))}, \tag{10}$$

$$\boldsymbol{\psi}_j(x_j(k)) = \left[\omega_j^1[x_j(k)], \ldots, \omega_j^{N_j}[x_j(k)]\right], \tag{11}$$

$$\mathbf{q}_j = \left[q_j^1, \ldots, q_j^{N_j}\right]^T, \tag{12}$$

then the fuzzy logic controller (8) can be rewritten as

$$u(\mathbf{x}) = \boldsymbol{\psi}_1^T(x_1(k))\mathbf{q}_1 + \ldots + \boldsymbol{\psi}_n^T(x_n(k))\mathbf{q}_n, \tag{13}$$

$$u(\mathbf{x}) = u(x_1(k)) + \ldots + u(x_n(k)), \tag{14}$$

where $u(x_j(k)) = \boldsymbol{\psi}_j^T(x_j(k))\mathbf{q}_j$, for $j = 1, \ldots, n$.

The FLC to be evolved by the proposed methodology (Section III) is defined by (14) where each term of the equation is represented by a set of univariate fuzzy control rules allowing a better understanding of the importance/influence of each input variable on the behaviour of the controller. Note that (14) is not a "simple" univariate addition, since each term of (14) is also dependent of the other input variables, namely of $\sum_{i=1}^{N} \mu_{A^i}(\mathbf{x}(k))$ (9) which can be seen as a weight.

III. Proposed Evolving Fuzzy Control Design

This section presents the proposed online evolving process to design the FLC defined in (14) which uses the fuzzy control rules defined in (7). The main steps of the proposed evolving methodology are the 1) initialization (Section III-A), 2) adaptations of the fuzzy control rules parameters, namely, the antecedents and consequents parameters (Section III-B), and 3) the modification of the FLC structure (Section III-C).

Let us consider the following single-input single-output (SISO) system

$$y(k+1) = f(\mathbf{x}(k), u(k)), \tag{15}$$

where $\mathbf{x}(k) = [y(k), y(k-1), \ldots, y(k-n_y), u(k-1), u(k-2), \ldots, u(k-n_u)]$ represents the state of the plant, $u(k)$ is the control signal, n_y and n_u are the orders of the output and input, respectively, and f is an unknown continuous and differentiable function.

Similarly to [7], for the proposed evolving methodology, to control the system (15), it is imposed the condition that, for every value of the desired reference, $r(k)$, for the output $y(k+1)$, there is always a control signal capable of translating the output of the system, $y(k)$ (and consequently the state $\mathbf{x}(k)$), to the command signal, $u(k)$, required for the output $y(k+1)$ to reach the desired reference, $r(k)$. In this way, for all the state, $y(k)$, where the output may be, the output variable depends on the control signal $u(k)$ [10]. Therefore, the system should be monotonic as a function of the control signal, and the derivative of plant output with respect to the command signal $u(k)$ should have a constant sign.

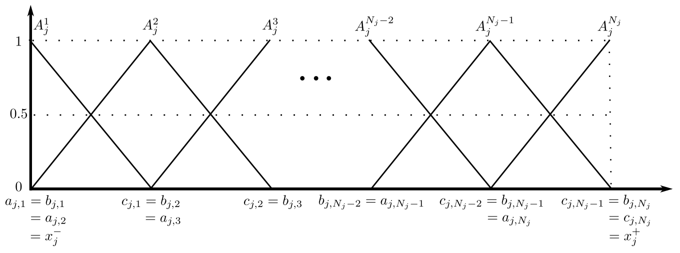

A_j^1
A_j^2
A_j^3
$A_j^{N_j-2}$
$A_j^{N_j-1}$
$A_j^{N_j}$
1
0.5
0
$a_{j,1} = b_{j,1} = a_{j,2} = x_j^-$
$c_{j,1} = b_{j,2} = a_{j,3}$
$c_{j,2} = b_{j,3}$
$b_{j,N_j-2} = a_{j,N_j-1}$
$c_{j,N_j-2} = b_{j,N_j-1} = a_{j,N_j}$
$c_{j,N_j-1} = b_{j,N_j} = c_{j,N_j} = x_j^+$

Figure 1: Complementary triangular membership functions example for a input variable x_j.

The controller to control the system (15) can be expressed by the following function g

$$u(k) = g(\tilde{\mathbf{x}}(k), \boldsymbol{\Phi}), \tag{16}$$

where $\tilde{\mathbf{x}}(k) = [r(k), y(k), y(k-1), \ldots, y(k-n_y), u(k-1), u(k-2), \ldots, u(k-n_u)]$, and $\boldsymbol{\Phi}$ represents the control parameters. In this way, by (16), the control problem can be considered as a problem of function approximation of the plant's inverse function [7], [11]. In this paper, it is used (8) (rewritten in (14)) to approximate (16).

A. Initialization

For the proposed evolving methodology it is necessary an initial fuzzy controller structure composed of only two fuzzy control rules which are defined using only the information of the universe of discourse of the input and output variables, information which is mandatory to control any process. The fuzzy controller in this paper is composed of a set of fuzzy control rules of the form of (7). In this way it is necessary define the input variables x_j ($j = 1, \ldots, n$) and the respective membership functions.

In this paper all membership functions $A_j^{i_j}$ ($j = 1, 2, \ldots, n$, $i_j = 1, 2, \ldots, N_j$) are defined as complementary triangular membership functions which implies that for each input variable x_j a maximum of two rules are activated at instant k. Figure 1 represents an example of complementary triangular membership functions, where x_j^- and x_j^+ are the minimum and maximum values of input variable x_j (i.e. the universe of discourse limits), respectively. The parameters a_{j,i_j}, b_{j,i_j}, and c_{j,i_j} are the lower limit, center value, and upper limit of $A_j^{i_j}$, i.e., the parameters of the i_j-th triangular membership function of the input variable x_j, respectively.

An initial number of fuzzy control rules associated to each input variable, x_j, in rule base (7), is defined to be $N_j^{ini} \geqslant 2$, which in the case of this paper will be defined as $N_j^{ini} = 2$. In addition to defining the initial membership functions, $A_j^{i_j}$, the initial consequents $q_j^{i_j}$ must be defined. As in this paper it is assumed that there is no knowledge about the system's dynamics, the initial value of all consequent parameters ($\mathbf{q}_j$) is the minimum control value admissible.

B. Antecedents and Consequents Adaptation

In the proposed methodology both antecedents and consequents parameters are adapted for each instant k.

For the antecedents, as the membership functions are complementary, just the center of the membership functions of each input variable x_j is updated by:

$$b_{j,i_j}(k) = b_{j,i_j}(k-1) + \beta\omega_j^{i_j}[x_j(k)](x_j(k) - b_{j,i_j}(k-1)), \tag{17}$$

where $\omega_j^{i_j}[x_j(k)]$ is given by (10), and β is the adaptation gain. Note that, the centers of the first and last membership functions of each input variable j must be kept constant since they correspond to limits of the universe of discourse, i.e. the minimum and maximum values, x_j^- and x_j^+, respectively.

For the consequents, the consequent parameters are updated taking into account the information (signal) presented on the tracking error since it indicates the direction in which the rule consequents have to be updated [7]:

$$q_j^{i_j}(k) = q_j^{i_j}(k-1) + C\omega_j^{i_j}[x_j(k)]e(k), \tag{18}$$

where $e(k) = r(k-1) - y(k)$, $\omega_j^{i_j}[x_j(k)]$ is given by (10), and C is the adaptation gain with the same sign as the monotonicity of the plant with respect to the control signal.

C. Creation of a Fuzzy Control Rule

The main steps of the proposed evolving methodology are:

- Select the input variable into which a new membership function will be added. In order to avoid that the number of rules grows fast/exponentially, only one variable will receive a new membership function at each instant k. The variable to receive the new membership function is selected as the one with the largest estimated control error.
- Add a new membership function to the selected input variable: the position of the new membership function is given by the distribution of the control estimation error. In this way the sensitivity to noise/outliers is reduced when comparing to other methods (e.g. [1], [4], [5], [6]) that locate the new membership function on the location of largest error.
- After adding a new membership function, to finish the design of the new fuzzy control rule, the consequent parameter of the new fuzzy rule must be defined. The consequent parameter must be defined in such a way that the controller performance will be minimally affected by the introduction of a new fuzzy control rule.

1) Formulation: Because of the assumption that the dynamics of the system to be controlled are unknown, the partial derivative $\partial y/\partial u$ cannot be obtained and therefore it is not possible use techniques based on gradient to minimize the tracking error, i.e. minimize $J = (\mathbf{y} - \hat{\mathbf{y}})^2$, where $\mathbf{y}$ is the real output and $\hat{\mathbf{y}}$ is the output of the estimated (controller) model. However, considering that the fuzzy controller can be an approximator of the plant's true inverse function [7], [10], the control estimation error can be obtained, and therefore the tracking error is minimized by minimizing the control estimation error, avoiding the use of a model of the plant to be controlled.

The procedure for adding a new fuzzy control rule is based on the fact that the very operation of the plant provides a set input/output data about the true inverse function of the system to be controlled [11]. In this way, if the control signal $u(k)$ applied at instant k generates the output $y(k+1)$, then if the system returns again to the same state, and the reference is $r(k) = y(k+1)$, then the applied control signal $u(k)$ can be considered as the optimum control signal [7]. In this way, it is stored in a temporal sliding window $\mathbf{M} = [\mathbf{z}(k - T_M)^T, \ldots, \mathbf{z}(k-1)^T]$ with the size of T_M, where each data of the window is composed of $\mathbf{z}(m) = [\tilde{\mathbf{x}}(m), u(m)]$ $(m = 1, \ldots, T_M)$, where $\tilde{\mathbf{x}}(m) = [y(m+1), x_2(m), \ldots, x_n(m)]$. This temporal sliding window provides information about the inverse function of the plant, allowing to: obtain the control estimation error to select the input variable in which a new membership function will be added, define the position of the new membership function, and define the consequent parameter of the new fuzzy control rule.

2) Input Variable Selection: As mentioned before, the first step for the evolving method is to select the input variable in which a new membership function can be added. In order to avoid a high growing of the number of rules just one input variable can receive a new membership function at each instant k. The variable to be selected is the input variable with the biggest estimated controller error. Using the temporal sliding window, the control estimation error at sample time k is given by:

$$e_u(k) = \sum_{m=1}^{T_M} (u(m) - \hat{u}(m))^2, \tag{19}$$

where $u(m)$ is the control signal stored in the temporal sliding window (real controller) at sample m of the window, and $\hat{u}(m)$ is control signal produced by the current fuzzy controller using the input vector $\tilde{\mathbf{x}}(m)$ instead of $\mathbf{x}(m)$, i.e. the FLC controller given by $u(\tilde{\mathbf{x}}(m))$.

Being $e_{u_j}(k)$ the control estimation error for a specific variable x_j at instant k, $e_{u_j}(k)$ is given by

$$e_{u_j}(k) = \sum_{m=1}^{T_M} (\boldsymbol{\psi}_j(x_j(m))\mathbf{u}_j^T(m) - \boldsymbol{\psi}_j(\tilde{x}_j(m))\mathbf{q}_j)^2, \tag{20}$$

where $\boldsymbol{\psi}_j(x_j(m))$ is given by (11), $\mathbf{u}_j(m) = u(m)[1, \ldots, 1]_{1 \times N_j}$ is a vector where each element is the control signal stored in the temporal sliding window (real controller) at sample m of the window, $u(m)$, and $\tilde{x}_j(m)$ is the j-th component-variable of $\tilde{\mathbf{x}}$. Note that the term $\boldsymbol{\psi}_j(x_j(m))\mathbf{u}_j^T(m)$ has the weight given by $\boldsymbol{\psi}_j(x_j(m))$ which is the importance of the variable x_j on all fuzzy control rules.

Defining x_{j^-} as the variable to be selected in which a new membership function can be added, its index j^- is given by

$$j^- = \arg\max_{j=1,\ldots,n}(e_{u_j}(k)). \tag{21}$$

3) Addition of the New Membership Function: The second step is to choose a candidate membership function to add on the selected input variable x_{j^-}. i_j^+ is here defined as the index of the candidate membership function to be added. In order to reduce the sensibility to noise and/or outliers, instead of the other methods (e.g. [1], [4], [5], [6]) that locate the new membership function on the local of highest error, in the proposed methodology the position/center of the new membership function is given as a function of distribution of control estimation error:

$$m_c = \frac{\sum_{m=1}^{T_M} m(u(m)-\hat{u}(m))^2}{\sum_{m=1}^{T_M}(u(m)-\hat{u}(m))^2}. \tag{22}$$

The new center of the candidate membership function i_j^+ for the selected input variable x_{j^-} is obtained from the temporal sliding window $\mathbf{M}$, and is given by $x_{j^-}([m_c])$, where $[m_c]$ is the integer which is nearest to m_c. The criteria to define if the candidate membership function will be added on the selected input variable will be explained in Section III-C5.

4) Consequent Parameter of the New Fuzzy Control Rule: The third step is to define the consequent parameter of the new fuzzy control rule, which must be defined in a way such that the controller performance is minimally affected by the introduction of the new fuzzy control rule. The consequent parameter of the new fuzzy control rule, $q_{j^-}^{i_j^+}$, is obtained by using the heuristic method [12]:

$$q_{j^-}^{i_j^+} = \frac{\sum_{m=1}^{T_M} u(m)\mu_{A_{j^-}^{i_j^+}}\left(x_{j^-}(m)\right)}{\sum_{m=1}^{T_M}\mu_{A_{j^-}^{i_j^+}}\left(x_{j^-}(m)\right)}, \tag{23}$$

where $u(m)$ is the control signal stored in the temporal sliding window (real controller) at the sample m of the window.

5) Criteria to Add a New Fuzzy Rule: The proposed evolving methodology uses a criterion (Criterion 1) to decide if it can be considered the possibility of adding a new fuzzy control rule, and then, after defining a candidate membership function (Section III-C3) a second criterion (Criterion 2) is used to decide if the candidate membership function can be added on the respective input variable.

The Criterion 1 is obtained by the difference of the estimated control error (19) at instants k and $k-1$, i.e. $\Delta e_u(k) = e_u(k) - e_u(k-1)$. To start the process of add a candidate fuzzy rule, the following criterion must be met

$$|\Delta e_u(k)| > \delta, \tag{24}$$

where δ is a threshold defined by the user.

If Criterion 1 is met, a candidate membership function is defined (Section III-C3), and then Criterion 2 is used to decide if the candidate membership function should be added to the respective variable. The goal of Criterion 2 is to limit an excessively fine granulation of the domain region of the input variable where the candidate membership function will be added, and to avoid that the structure of the learned FLC becomes complex to interpret, and also to avoid overfitting situations. In this way, Criterion 2 controls the minimal distance between the center of two membership functions. Another important advantage of Criterion 2 is that, as the proposed methodology does not use an a priori maximum number of membership functions per input variable x_j, Criterion 2 also limits the creation of a large number of control rules. For the input variable x_j, Criterion 2 is given by

$$\left|b_{j,i_j^+}(k) - b_{j,i_j^-}(k)\right| > \eta_j, \tag{25}$$

where b_{j,i_j^+} and b_{j,i_j^-} are the centers of the candidate membership function and of its nearest membership function, η_j is a threshold defined by the user for the input variable j.

6) Proposed Algorithm - SEDFLC: The proposed methodology to evolve the fuzzy controller is summarized on Algorithm 1.

IV. Results

The results and performance of the proposed methodology SEDFLC to online evolve a FLC on a simulated nonlinear continuous-stirred tank reactor (CSTR) are presented in this section.

A. Control of a Simulated Continuous-Stirred Tank Reactor (CSTR) Plant

The simulated nonlinear Continuous Stirred Tank Reactor (CSTR) plant used in this paper is described by [13], [14]:

$$\frac{\partial C_A(t+d_c)}{\partial t} = \frac{q(t)}{V}\left(C_{A0}(t) - C_A(t+d_c)\right) - k_0 C_A(t+d_c)\exp\left(-\frac{E}{RT(t)}\right), \tag{26}$$

$$\begin{aligned}\frac{\partial T}{\partial t} = {} & \frac{q(t)}{V}\left(T_0(t) - T(t)\right) - \\ & \frac{(-\Delta H)k_0 C_A(t+d_c)}{\rho_{c1}C_p}\exp\left(-\frac{E}{RT(t)}\right) \\ & + \frac{\rho_{c2}C_{pc}}{\rho_{c1}C_p V}q_c(t)\left[1-\exp\left(\frac{-hA}{q_c(t)\rho_{c2}C_{pc}}\right)\right] \\ & \left(T_{c0}(t) - T(t)\right), \\ y(t) = {} & C_A(t), \quad u(t) = q_c(t),\end{aligned} \tag{27}$$

The variables and of the CSTR plant their nominal values are presented in Table I.

The goal of the proposed methodology is to control the output variable $y(t)$ (i.e. $C_A(t)$) of the CSTR plant by manipulating the control variable $u(t)$ (i.e. $q_c(t)$). The input variables of the fuzzy controller are $x_1 = r(t)$ (reference signal) and $x_2 = y(t)$ (output variable of the CSTR plant). The parameters, defined by the user, of the proposed methodology

Algorithm 1 Proposed methodology, SEDFLC.

```
1) Inputs:
   a) For all input variables: $x_j$ ($j = 1, \dots, n$): number of initial membership function $x_j$ ($j = 1, \dots, n$), $N_j^{ini}$, where $N_j^{ini} \geqslant 2$; the universe of discourse limits, i.e. the minimum and maximum values, $x_j^-$ and $x_j^+$; and $\eta_j$;
   b) About the process to be controlled: the limits of the universe of discourse of the control variable $u$, i.e. the minimum and maximum values, $u^-$ and $u^+$;
   c) The window's size $T_M$, the learning gains $\beta$ (antecedents) and $C$ (consequents); and the threshold $\delta$.
2) Initialization. Design the initial fuzzy controller structure.
   a) Antecedent part: For all input variables $x_j$ ($j = 1, \dots, n$) design the initial membership functions where the parameters of each complementary triangular membership function $A_j^{i_j}$ (see Figure 1) are, for $j = 1, \dots, n$, and $i_j = 1, \dots, N_j$:
      i) parameters of $A_j^1$: $a_{j,1} = b_{j,1} = x_j^-$, and $c_{j,1} = b_{j,2}$;
      ii) parameters of $A_j^{i_j}$: $a_{j,i_j} = b_{j,i_j-1}$, $b_{j,i_j} = c_{j,i_j-1}$, and $c_{j,i_j} = b_{j,i_j+1}$;
      iii) parameters of $A_j^{N_j}$: $a_{j,N_j} = b_{j,N_j-1}$, and $b_{j,N_j} = c_{j,N_j} = x_j^+$.
   b) Consequent part: define the initial value of all consequent parameters ($\mathbf{Q}_j$) as $u^-$, the minimum admissible control value.
3) Online Evolving learning. For $k = 1, 2, \dots$ (until the controller is turned off)
   a) Update the antecedent parameters using (17) (see Section III-B).
   b) Update the consequent parameters using (18) (see Section III-B).
   c) If the sliding window is filled:
      i) Obtain the estimated control error $e_u(k)$ using (19) (see Section III-C).
      ii) If Criterion 1 (24) is met, i.e. $|\Delta e_u(k)| > \delta$, then:
         A) Select the input variable in which a new membership function can be added (which is the one with a largest controller estimation error) using (21) (see Section III-C).
         B) Locate the center of the candidate membership function using (22). The center is given by $x_{j^-}([m_c])$ (see Section III-C).
         C) If Criterion 2 (25) is met, then:
            • Add the new membership function.
            • Define the consequent parameters of the new fuzzy rule using (23) (see Section III-C).
   d) Apply the learned fuzzy logic controller to the system under control (calculate and send command $u(k)$ to the system).
   e) Read the output variable $y(k)$ of the system under control.
   f) Update the sliding window. Go to step (3a) (until the controller is turned off).
```

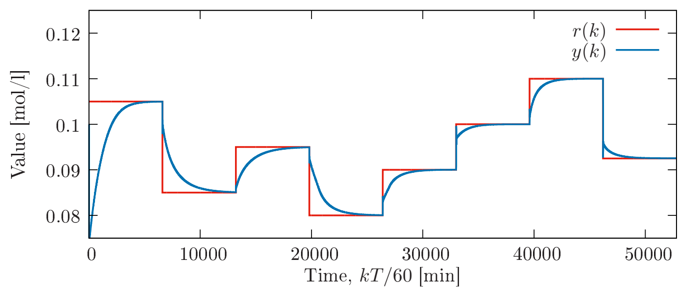
Table I: Description of CSTR variables and their nominal values [13], [14].

Variables-Description	Value
C_A - Product concentration	0.1 [mol/l]
T - Reactor temperature	438.54 [K]
q_c - Coolant flow rate	103.41 [l/min]
q - Process flow rate	100 [l/min]
C_{A0} - Feed concentration	1 [mol/l]
T_o - Feed temperature	350 [K]
T_{c0} - Inlet coolant temperature	350 [K]
V - CSTR volume	100 [l]
hA - Heat transfer term	7×10^5 [cal/min/K]
k_0 - Reaction rate constant	7.2×10^{10} [min^{-1}]
E/R - Activation energy term	1×10^4 [K]
$-\Delta H$ - Heat of reaction	-2×10^5 [cal/mol]
ρ_{c1}, ρ_{c2} - Liquid densities	1×10^3 [g/l]
C_p, C_{pc} - Specific heats	1 [cal/g/K]
T - Sampling period	0.1 [min]
d_c - Time delay	$5T = 0.5$ [min]

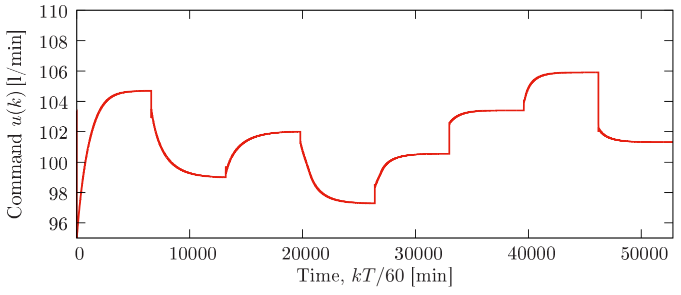
to evolve the fuzzy controller are: $N_1^{ini} = N_2^{ini} = 2$, $x_1^- = x_2^- = 0.06$ and $x_1^+ = x_2^+ = 0.12$; $u^- = 95$ and $u^+ = 110$ ($\Delta u = 15$), $\eta_j = \left|x_j^+ - x_j^-\right|/15$ (for $j = 1, 2$), $C = 1$, $M = 200$, $\beta = 0.00005$, and $\delta = 0.05 \times \Delta u$.

Figure 2 presents the results of the online evolving fuzzy controller, being tested with different reference signals to reach several regions of operation of the systems, and it also presents the evolution of the main parameters of the proposed methodology. From the results, it can be seen that the FLC is able to adequately control the CSTR plant where its performance increases with the time of operation (see Figure 2a).

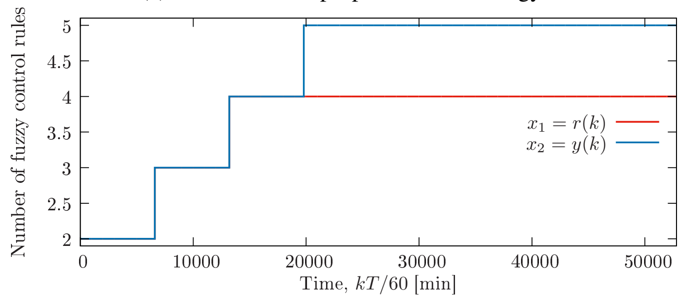
In Figure 2c is can be seen the evolution of the number of fuzzy control rules of each input variable. As can be seen, when some unknown region of operation is reached, one new fuzzy control rule is added. Note that the reference signal starts with four different unknown regions of operation for the FLC (the first four steps/values of the reference signal), and then the reference signal changes, and there are also 4 different values of the reference signal on the last four steps, however they can be considered in the same regions of operation of the ones already learned on the first four values of the reference signal



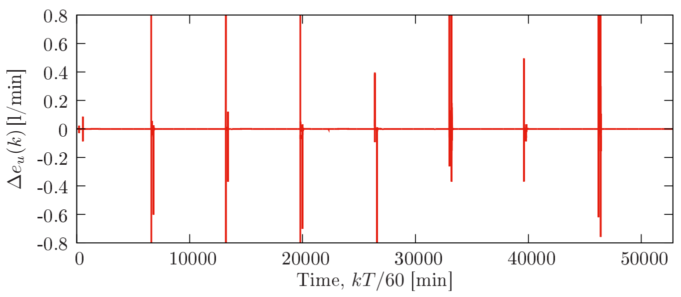
(a) Results of the proposed methodology.



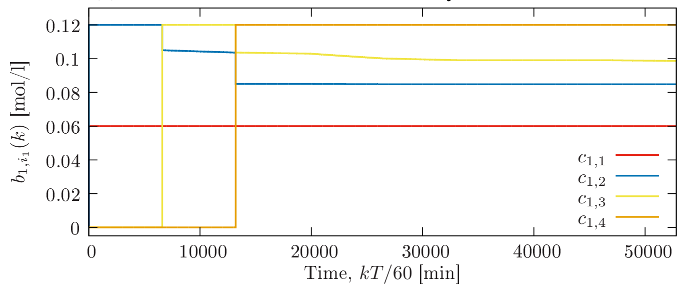
(b) Command signal $u(k)$ applied.



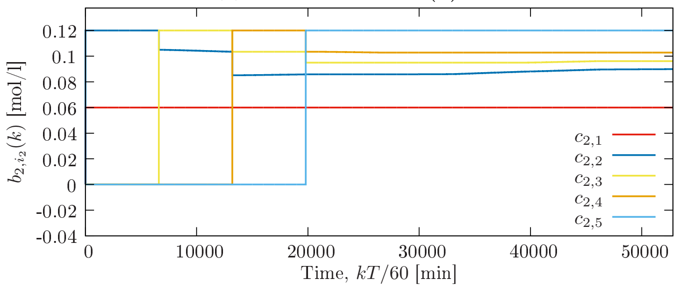
(c) Evolution of the number of fuzzy control rules.



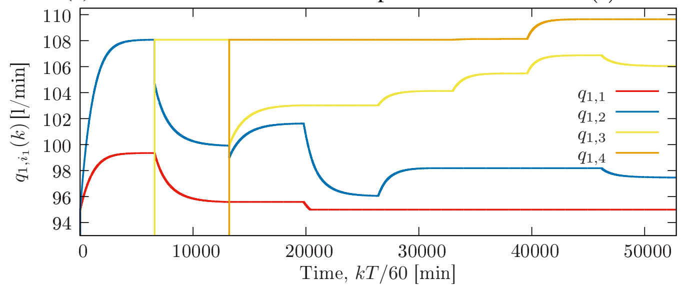
(d) Evolution of $\Delta e_u(k)$.



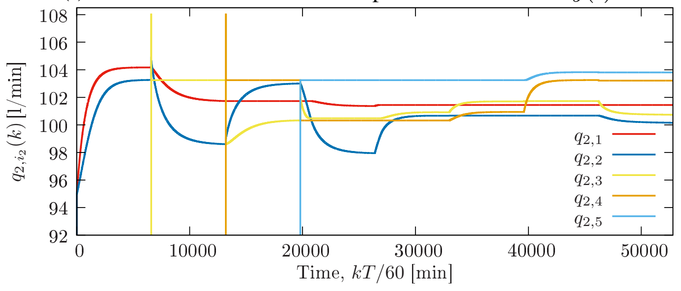
(e) Evolution of the antecedent parameters of $x_1 = r(t)$.



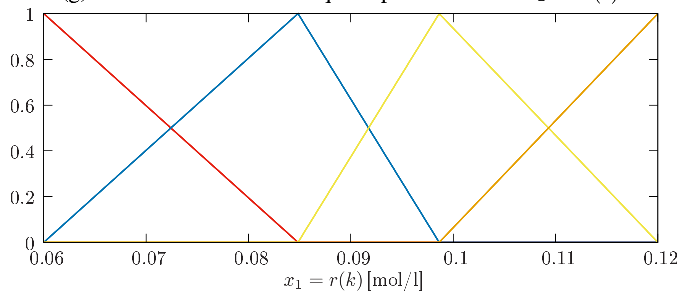
(f) Evolution of the antecedent parameters of $x_2 = y(t)$.



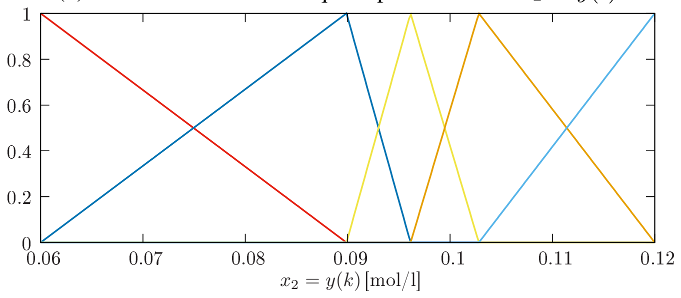
(g) Evolution of the consequent parameters of $x_1 = r(t)$.



(h) Evolution of the consequent parameters of $x_2 = y(t)$.



(i) Final membership functions of $x_1 = r(t)$.



(j) Final membership functions of $x_2 = y(t)$.

Figure 2: Results of the proposed methodology SEDFLC on the CSTR plant.

by the proposed methodology. The analysis of the addition of new fuzzy control rules can be complemented with Figure 2d which represents the evolution of $\Delta e_u(k)$ (one of the criteria (Criterion 1) that must be reached to add a new fuzzy control rule). In Figure 2d, it can be observed that Criterion 1 (24) ($|\Delta e_u(k)| > \delta = 0.05 \times \Delta u = 0.75$) is reached 6 times, however only on the first three times are new fuzzy control rules added. This happens because on the remaining three times Criterion 2 (25) is not met because the proposed methodology considers that the respective regions of operation are similar to regions where the controller was already learned.

Figures 2e and 2f present the evolution of the antecedent parameters of the input variables x_1 and x_2, respectively, and Figures 2g and 2h present the evolution of the consequent parameters of the input variables x_1 and x_2, respectively. In the Figures 2e-2h can be seen that fine adjustments on the antecedent and consequent parameters are performed in order to reach a better controller performance, and that during the first four steps/values of the reference signal new antecedent and consequent parameters are defined (start to exist) since new fuzzy rules were created by the proposed methodology. Figures 2i and 2j present the final membership functions of x_1 and x_2, respectively.

V. Conclusion

In this paper it was proposed a new methodology (SED-FLC) for online self-evolving direct fuzzy logic controllers (FLCs). The FLC is designed online where their parameters (antecedent and consequent parameters) are updated online, and the FLC structure is modified/evolved online using only input/output data obtained during the normal operation of the system while it is being controlled. The proposed methodology can start without fuzzy control rules (empty FLC structure) or with an a priori set of fuzzy control rules. A simple FLC structure is presented which allows a better understanding of the importance/influence of each input variable on the behaviour of the controller, allows a good interpretability of the learned FLC, and avoids a fast/exponentially growing of the number of fuzzy control rules.

The proposed methodology was tested on a simulated continuous-stirred tank reactor (CSTR) plant where the results show that the proposed methodology 1) has the capability of designing a FLC in order to successfully control the CSTR plant; 2) does fine adjustments on the antecedent and consequent parameters in order to have a better accuracy; 3) has the capability to modify/evolve the FLC structure when some unknown region of operation is reached.

Acknowledgment

Jérôme Mendes, and Francisco Souza have been supported by Fundação para a Ciência e a Tecnologia (FCT) under grants SFRH/BPD/99708/2014, and SFRH/BPD/112774/2015, respectively. The authors acknowledge the support of FCT project UID-EEA-00048-2013.

References

[1] J. Mendes, R. Araújo, T. Matias, R. Seco, and C. Belchior, "Evolutionary learning of a fuzzy controller for industrial processes," in *Proc. of the The 40th Annual Conference of the IEEE Industrial Electronics Society (IECON 2014).* Dallas, TX, USA: IEEE, October 29 - November 1 2014, pp. 139–145.

[2] A. B. Cara, H. Pomares, and I. Rojas, "A new methodology for the online adaptation of fuzzy self-structuring controllers," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 3, pp. 449–464, June 2011.

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

[4] O. Castillo, E. Lizárraga, J. Soria, P. Melin, and F. Valdez, "New approach using ant colony optimization with ant set partition for fuzzy control design applied to the ball and beam system," *Information Sciences*, vol. 294, pp. 203–215, February 2015.

[5] C. Caraveo, F. Valdez, and O. Castillo, "Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation," *Applied Soft Computing*, vol. 43, pp. 131–142, June 2016.

[6] A. Rubaai and P. Young, "Hardware/software implementation of fuzzy-neural-network self-learning control methods for brushless dc motor drives," *IEEE Transactions on Industry Applications*, vol. 52, no. 1, pp. 414–424, January/February 2016.

[7] A. B. Cara, H. Pomares, I. Rojas, Z. Lendek, and R. Babuška, "Online self-evolving fuzzy controller with global learning capabilities," *Evolving Systems*, vol. 1, no. 4, pp. 225–239, December 2010.

[8] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, February 1985.

[9] H. Ying, "General miso takagi-sugeno fuzzy systems with simplified linear rule consequent as universal approximators for control and modeling applications," in *IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 2, October 1997, pp. 1335–1340.

[10] H. Pomares, I. Rojas, J. González, F. Rojas, M. Damas, and F. Fernández, "A two-stage approach to self-learning direct fuzzy controllers," *International Journal of Approximate Reasoning*, vol. 29, no. 3, pp. 267–289, March 2002.

[11] H. Pomares, I. Rojas, J. Gonzalez, M. Damas, B. Pino, and A. Prieto, "Online global learning in direct fuzzy controllers," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 2, pp. 218–229, April 2004.

[12] 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*, vol. 29, no. 1, pp. 70–78, March 2014.

[13] J. Mendes, R. Araújo, and F. Souza, "Adaptive fuzzy identification and predictive control for industrial processes," *Expert Systems with Applications*, vol. 40, no. 17, pp. 6964–6975, December 2013.

[14] J. D. Morningred, B. E. Paden, D. E. Seborg, and D. A. Mellichamp, "An adaptive nonlinear predictive controller," *Chemical Engineering Science*, vol. 47, no. 4, pp. 755–762, 1992.