applied sciences

MDPI

Article

Self-Evolving Fuzzy Controller Composed of Univariate Fuzzy Control Rules

Jérôme Mendes [1,*], Ricardo Maia [1,2], Rui Araújo [1] and Francisco A. A. Souza [3]

1 University of Coimbra, Institute of Systems and Robotics, Department of Electrical and Computer Engineering, Pólo II, PT-3030-290 Coimbra, Portugal; rmmaia@isr.uc.pt (R.M.); rui@isr.uc.pt (R.A.)
2 Critical Software S.A., Pq. Ind. de Taveiro, Lt 49, PT-3045-504 Coimbra, Portugal
3 Oncontrol Technologies, Lda, Av. Sá Bandeira, 33, Escritório 519, PT-3000-279 Coimbra, Portugal; francisco.alexandre@oncontrol-tech.com
* Correspondence: jermendes@isr.uc.pt

Received: 11 July 2020; Accepted: 19 August 2020; Published: 23 August 2020

Abstract: The paper proposes a methodology to online self-evolve direct fuzzy logic controllers (FLCs), to deal with unknown and time-varying dynamics. The proposed methodology self-designs the controller, where fuzzy control rules can be added or removed considering a predefined criterion. The proposed methodology aims to reach a control structure easily interpretable by human operators. The FLC is defined by univariate fuzzy control rules, where each input variable is represented by a set of fuzzy control rules, improving the interpretability ability of the learned controller. The proposed self-evolving methodology, when the process is under control (online stage), adds fuzzy control rules on the current FLC using a criterion based on the incremental estimated control error obtained using the system's inverse function and deletes fuzzy control rules using a criterion that defines "less active" and "less informative" control rules. From the results on a nonlinear continuously stirred tank reactor (CSTR) plant, the proposed methodology shows the capability to online self-design the FLC by adding and removing fuzzy control rules in order to successfully control the CSTR plant.

Keywords: evolving design; fuzzy controller; univariate fuzzy rules; CSTR plant

1. Introduction

The globalization of markets, environmental legislation restrictions, the necessity of having an efficient management of energy and sustainable resources, zero defect trends, customer pressure to reduce costs, personalized products, low product lifetime, and several other facts have resulted in a significant increase in industrial process complexity. Industrial processes have increasingly exhibited nonlinear behaviors and other complex characteristics, such as unknown and time-varying dynamics, constraints, and disturbances. Considering the above facts and trends of the "Industry 4.0" strategy, more advanced industrial control solutions with high levels of efficiency, flexibility, reliability, and performance are currently required.

Motivated by these problems, as an advanced industrial control solution, fuzzy logic controllers (FLCs), which are rule-based systems that allow control of complex ill-defined processes using the experience of expert operators, have been applied in a wide variety of industrial processes [1]. However, there still exist many difficulties in designing FLCs for complex nonlinear industrial processes, using only human knowledge to control complex processes, and there is not a standard approach to translate this knowledge into fuzzy control rules [2].

The design of fuzzy logic systems has attracted many researchers, many of them working on identification, classification, control, and decision making problems. Regarding the fuzzy control systems, the design methods can be generally distinguished as indirect or direct methods, where in direct methods, the FLC is designed through operator control knowledge, and in indirect methods, the FLC is designed through operator knowledge about the process [3,4]. However, the number of proposed direct fuzzy control design methodologies are significantly less when comparing to identification, classification, and indirect control applications. Evolutionary algorithms have been used to design fuzzy controllers [5–11]. However, such methods (e.g., GA, PSO, and ABC) are computationally heavy, mainly offline, and do not consider changes in the dynamics of the process, when the process is under control.

In order to deal with systems characterized by changing the characteristics, evolving systems have recently emerged as promising methods that adapt their structure (and parameters, as happens with adaptive systems) to new operating conditions, changing system dynamics, drifting situations, and non-stationary environments [12]. It is important to note that evolving systems self-change or organize their structure (and parameters) according to novelties such as, for example, new regions of operation, anomalies, unknown environment, drifts, and shifts, which differs from adaptive systems, which only adapt their parameters, having the model/control structure fixed. Several relevant evolving methodologies have been proposed to design fuzzy systems. In [13], a review of evolving fuzzy and neuro-fuzzy methodologies for clustering, regression, identification, and classification applications was presented, in which the addition, merging, splitting, and removing evolving mechanisms were presented and discussed. Additionally, in [12], an overview of the following evolving systems was performed: fuzzy rule-based, neuro-fuzzy, Cauchy possibilistic clustering, granular neural network, and fuzzy linear regression tree. However, in these reviews [12,13], control applications were not considered. Focusing on control applications, the work in [14] presented an approach for on-line designing indirect T-S (Takagi-Sugeno)fuzzy controllers with evolving learning in a recursive way. A fuzzy model reference adaptive control was proposed in [15], in which the evolving fuzzy model (eFuMo) method was used to learn the controller's fuzzy model. In addition to [15], an adaptive law was proposed in [16] in order to guarantee global stability. In [17], an evolving neuro-fuzzy controller based on the Taylor series neuro-fuzzy (TaSe-NF) model was proposed, in which the proposed method analyzes the error surface to obtain the fuzzy rule with the worst performance to be split, based on the presented criteria. A model-based evolving granular fuzzy control approach was proposed in [18], where the model's structure and parameters were adapted based on information extracted from uncertain data streams. In [19], an evolving probabilistic fuzzy neural controller using an asymmetric membership function was presented. In [20], an evolving fuzzy model-based controller for a generic hypersonic vehicle (HV) was proposed, where the control law was designed based on the fuzzy model, and the stability analysis was performed using the Lyapunov method. Self-evolving Type-2 control was also studied [21–23]. However, the above control applications focused on indirect control, in which the evolving strategy was proposed for the model. In the evolving fuzzy systems, when compared with identification and classification problems (and even with indirect control applications), just a few works have been proposed for direct control learning and evolving (and not for identification of a system or data model, even to be possibly used in indirect control learning). In order to design the entire fuzzy control structure in an online way, methodologies to design direct FLCs in an evolving way have been proposed. In [2,24], an evolving FLC design methodology was proposed, where the proposed method was initialized with no fuzzy control rules (the online process starts with an empty control structure). However, the proposed solution can lead to a complex controller with a huge number of control rules. Later, based on data clouds, the robust evolving cloud-based controller (RECCo) was proposed [25]. In RECCo, the antecedent part of the fuzzy control rules to be designed is defined by data clouds (instead of the typical membership functions). RECCo was used also in [26–28] and improved in [27,28]. In [29], an evolving data cloud-based PID-like controller was proposed for uncertain nonlinear systems, in which, a stable recursive method (based on the Lyapunov function) was proposed to adapt the parameters aiming at fast convergence performance. However, the control rules' interpretability ability by a human operator was lost since the antecedent part of the control rules was defined by data clouds, which make the control rules very hard to be understood by human operators [26]. In [30], the problem of translating the control knowledge of a human expert operator into fuzzy control rules was addressed, where an approach was proposed to automatically design a Mamdani FLC iteratively. In [31], a methodology was proposed to online evolve a fuzzy control structure defined by univariate fuzzy control rules, which was based on the inverse function of the system under control. However, the works in [30,31] only considered adding control rules during the evolving learning process.

In order to reach an online self-organizing fuzzy control design, which can be easily interpretable by human operators, a new methodology is proposed in this paper, to online self-evolve direct FLCs. The fuzzy control rules' structure is defined by univariate rules. In this way, the interpretability ability of the controller is improved, as well as the complexity (the number of membership functions and fuzzy control rules) of the learned FLC controller is reduced. In an offline stage, the initial fuzzy control structure is designed using only the variables range (minimum and maximum admissible values). Then, the online stage, which occurs when the process is under control, corresponds to the evolving stage where fuzzy control rules can be added or removed from the current knowledge base (fuzzy control rules). To add fuzzy control rules on the current FLC, the proposed self-evolving design method uses two criteria: Criterion 1 is based on the incremental estimated control error obtained using the system's inverse function, and Criterion 2 is based on the minimal distance allowed between the center of two consecutive membership functions. To delete fuzzy control rules, two criteria to delete "less active" or "less informative" fuzzy control rules are used. To evaluate the proposed self-evolving FLC, a simulated nonlinear continuously stirred tank reactor (CSTR) plant is used. The main contributions of the proposed self-evolving design methodology for direct FLCs are: the evolving of a control structure composed of univariate control rules, which together with the defined evolving mechanisms, makes the controller better interpretable by human operators, and having a light control structure (small number of membership functions and control rules); the criteria to add control rules are defined in order to reduce the sensitivity to noise/outliers; to delete fuzzy control rules, two criteria to delete "less active" or "less informative" rules are presented; and the threshold for the criteria to add or remove control rules are intuitively defined (by the percentage of the variables range and $\epsilon \in [0, 1]$).

The paper is organized as follows. Section 2 presents the FLC controller to be online designed by the proposed methodology. Section 3 presents the proposed self-evolving FLC design method. Section 4 analyzes the performance of the proposed approach on a nonlinear continuously stirred tank reactor system. The final conclusions are presented in Section 5.

2. Fuzzy Logic Controller

This section briefly presents the main concepts of fuzzy logic control and the control structure to be evolved by the proposed method in Section 3. Fuzzy logic controllers (FLCs) have become an important research area in fuzzy systems [3], being widely applied in the control of complex industrial processes where the control knowledge of human expert operators is available through a set of IF-THEN control rules. A fuzzy control rule describing a simplistic control action to control NO_x emissions in the cement industry is presented in the following example:

$$\textbf{IF} \text{ the } NO_x \text{ is } \textit{high}, \textbf{ THEN} \text{ inject } \textit{more ammonia}, \tag{1}$$

where NO_x and *ammonia* (variation) are the input and output linguistic variables being, respectively, defined by the linguistic terms associated with the fuzzy sets *high* and *more*.

The FLC, which will be evolved by the proposed method in this paper, is composed of a set of univariate fuzzy control rules, allowing improving the controller interpretability ability, since the influence of each input variable on the overall behavior of the controller will be more easily interpretable. In this way, the fuzzy control rules for each input variable are given by [31]:

$$\begin{aligned}
R_j^1 :&\quad \text{IF } x_j(k) \text{ is } A_j^1 \text{ THEN } u_j^1(k) = \theta_j^1, \\
\vdots \\
R_j^{N_j} :&\quad \text{IF } x_j(k) \text{ is } A_j^{N_j} \text{ THEN } u_j^{N_j}(k) = \theta_j^{N_j}, \\
&\quad j = 1, \ldots, n, \\
&\quad i_j = 1, \ldots, N_j,
\end{aligned} \tag{2}$$

where x_j is the input variable j ($j = 1, \ldots, n$), $R_j^{i_j}$ is the i_j-th fuzzy control rule of x_j, N_j is the number of rules for x_j, $\theta_j^{i_j}$ is the consequent parameter, and $A_j^{i_j}$ are linguistic terms characterized by a complementary fuzzy membership function [32]. Using the following fuzzy system configuration: the singleton as the fuzzifier, the center-average as the defuzzifier, and the product inference engine, the FLC is given by [31]:

$$u(\mathbf{x}(k)) = \frac{\sum_{i_1=1}^{N_1} \mu_{A_1^{i_1}}(x_1(k))\theta_1^{i_1} + \ldots + \sum_{i_n=1}^{N_n} \mu_{A_n^{i_n}}(x_n(k))\theta_n^{i_n}}{\sum_{i_1=1}^{N_1} \mu_{A_1^{i_1}}(x_1(k)) + \ldots + \sum_{i_n=1}^{N_n} \mu_{A_n^{i_n}}(x_n(k))},$$

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

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

$$\omega_j^{i_j}[\mathbf{x}(k)] = \frac{\mu_{A_j^{i_j}}(x_j(k))}{\sum_{i=1}^{N} \mu_{A^i}(\mathbf{x}(k))}, \tag{3}$$

$$\boldsymbol{\psi}_j(\mathbf{x}(k)) = \left[\omega_j^1[\mathbf{x}(k)], \ldots, \omega_j^{N_j}[\mathbf{x}(k)]\right], \tag{4}$$

$$\boldsymbol{\Theta}_j = \left[\theta_j^1, \ldots, \theta_j^{N_j}\right]^T, \tag{5}$$

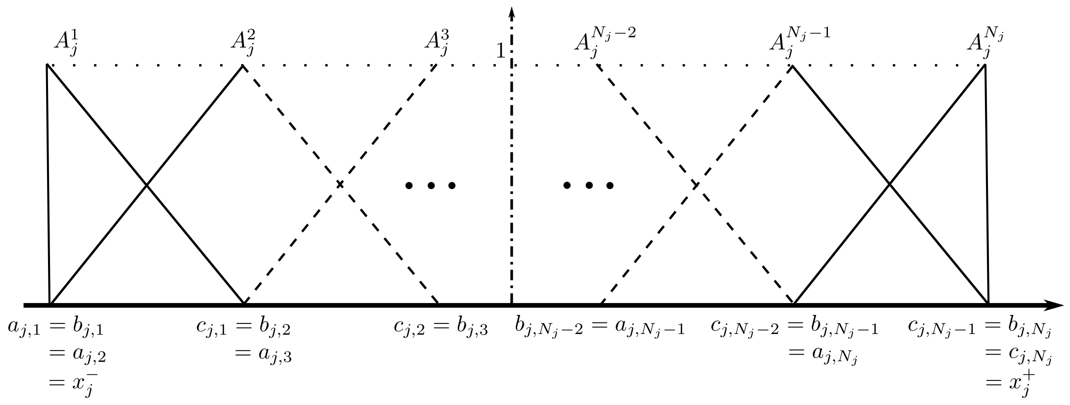
$$\mathbf{x}(k) = [x_1(k), \ldots, x_n(k)], \tag{6}$$

then the FLC (3) is given by:

$$\begin{aligned}
u(\mathbf{x}) &= u_1(\mathbf{x}(k)) + \ldots + u_n(\mathbf{x}(k)), \\
u_j(\mathbf{x}(k)) &= \boldsymbol{\psi}_j^T(\mathbf{x}(k))\boldsymbol{\Theta}_j.
\end{aligned} \tag{7}$$

$\boldsymbol{\psi}_j^T(\mathbf{x}(k))$ (4) in (7) can be seen as a weight of fuzzy control rules of input variable j on the overall controller. By Equation (7), the influence of each input variable on the overall FLC's behavior can be analyzed, taking into account the fuzzy control rules; structure, making the FLC more interpretable.

In order to allow more interpretability for the FLC, the membership functions are characterized by complementary triangular membership functions, as defined in Figure 1.



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

Figure 1. Definition of membership functions for x_j.

a_{j,i_j}, b_{j,i_j}, and c_{j,i_j} are respectively the lower, center, and upper values of the membership function $A_j^{i_j}$, and x_j^- and x_j^+ are the lower and upper limits of x_j.

The next section proposes a new methodology to online design, in an evolving way, the FLC represented by (7).

3. Proposed Self-Evolving FLC Design Method

This section describes the proposed methodology to self-evolve the fuzzy logic controller (FLC) presented in Section 2. This section starts by presenting the main formulation for the online evolving learning process based on the plant's inverse function in Section 3.1; the initialization (offline stage) of the proposed method is described in Section 3.2; and the online stage (self-evolving learning process) is described in Section 3.3.

3.1. Formulation

In this paper, the process to be controlled by the proposed self-evolving methodology is defined by following nonlinear autoregressive exogenous (NARX) model:

$$y(k+1) = f(\mathbf{x}(k), u(k)), \tag{8}$$

where $\mathbf{x}(k) = [y(k), \ldots, y(k-n_y), u(k-1), \ldots, u(k-n_u)]$, $u(k)$ and $y(k)$ are the process input and output, respectively, n_y and n_u are, respectively, the output and input orders, and f is an unknown continuous and differentiable single-input single-output (SISO) function.

Assumption 1. $\frac{\partial f(\mathbf{x},u)}{\partial u} \neq 0$ *for* $\mathbf{x} \in \mathbf{\Omega_x}$*, where* $\mathbf{x} \in \mathbf{\Omega_x}$ *is the controllability region [2,33].*

Assumption 1 is a controllability condition of System (8), which imposes that there is no state at which the system's output does not depend on the control signal u.

In a general way, the controller can be represented by the function g to control the system (8) [2,31]:

$$u(k) = g(\tilde{\mathbf{x}}(k), \mathbf{\Theta}), \tag{9}$$

where $\tilde{\mathbf{x}}(k) = [r(k), y(k), \ldots, y(k-n_y), u(k-1), \ldots, u(k-n_u)]$, which in addition to $\mathbf{x}(k)$ includes the reference signal $r(k)$, and $\mathbf{\Theta}$ are the control parameters. Fuzzy systems are universal approximators [34,35]. Therefore, to approximate (9), the FLC (3) will be designed by the proposed self-evolving learning method.

Sliding Window

A temporal sliding window is created in order to allocate the variables needed to obtain the inverse function of the plant, to be used in the evolving process phase. The sliding window will be used to estimate the control estimation error in order to select the input variable that contributes to the largest estimated control error, and such a variable is selected to receive a new fuzzy control rule (if the defined criteria are met). The concept is, if a command signal $u(k)$ generates $y(k+1)$ at system state $\mathbf{x}(k)$, then if the system returns again to the same state $\mathbf{x}(k)$ and the reference is the same as the generated output, i.e., $r(k) = y(k+1)$, then $u(k)$ can be considered as the optimum control signal [2,31].

The temporal sliding window, with T_M elements, is defined by:

$$\mathbf{M} = [\mathbf{z}(k - T_M)^T, \ldots, \mathbf{z}(k-1)^T], \tag{10}$$

where $\mathbf{z}(m) = [\tilde{\mathbf{x}}(m), u(m)]$ $(m = 1, \ldots, T_M)$ and $\tilde{\mathbf{x}}(m) = [y(m+1), y(m), \ldots, y(m-n_y), u(m-1), \ldots, u(m-n_u)]$.

3.2. Offline Stage

Since in this paper, the assumption is made that there is no knowledge about the dynamics of the process under control, in the offline design, the FLC is designed using the range values of the process variables.

Taking into account that the input variables are described by complementary triangular membership functions (MFs), as presented in Figure 1, then each input variable x_j is initially represented by two MFs, $N_j = 2$ for $j = 1, \ldots, n$; covered in this way by all the universe of discourse of the respective variable. The consequent parameters of all fuzzy control rules are initially set to the minimum admissible control value, u^-.

In general terms, two initial fuzzy control rules $R_j^{i_j}$ $(j = 1, \ldots, n$ and $i_j = 1, 2)$ for each input variable x_j $(\mathbf{x} = [x_1, \ldots, x_n])$ are, at the beginning offline stage, defined as follows:

$$\begin{aligned} R_j^1 : &\quad \text{IF } x_j(k) \text{ is } A_j^1 \text{ THEN } u_j^1(k) = \theta_j^1 = u^-, \\ R_j^2 : &\quad \text{IF } x_j(k) \text{ is } A_j^2 \text{ THEN } u_j^2(k) = \theta_j^2 = u^-, \end{aligned} \tag{11}$$

where the initial membership functions are defined as presented in Figure 2.

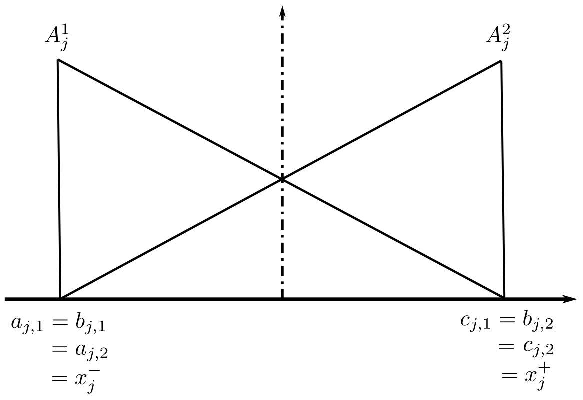

A_j^1
A_j^2
$a_{j,1} = b_{j,1} = a_{j,2} = x_j^-$
$c_{j,1} = b_{j,2} = c_{j,2} = x_j^+$

Figure 2. Initial membership functions for x_j.

3.3. Online Stage

This section presents the main steps of the proposed self-evolving methodology for online designing of the FLC presented in Section 2. The main steps are (1) adaptation of the rules' consequents (Section 3.3.1), (2) variable selection (Section 3.3.2), (3) definition of the fuzzy control rule candidate to be added (Section 3.3.3), (4) criteria to add the candidate control rule (Section 3.3.4), and (5) criteria to delete a fuzzy control rule (Section 3.3.5).

3.3.1. Consequent Adaptation

In order to reach a better control performance, the consequent parameters $\theta_j^{i_j}(k)$ are updated based on the tracking error $e(k) = r(k) - y(k)$ at all instants of time by:

$$\theta_j^{i_j}(k) = \theta_j^{i_j}(k-1) + \gamma \omega_j^{i_j}[x_j(k)]e(k), \tag{12}$$

where γ is a positive adaptation gain.

3.3.2. Variable Selection

The first step of the proposed self-evolving controller design is the selection of the candidate input variable x_{j^*} where a new fuzzy control rule should be added.

The criterion to select the candidate input variable x_{j^*} is based on the estimated controller error obtained by the function approximation of the plant's inverse function (Section 3.1). The estimation control error $e_c(k)$ is obtained by:

$$e_c(k) = \sum_{m=1}^{T_M} (u(m) - \hat{u}(m))^2, \tag{13}$$

where $u(m)$ represents the applied control signal (real controller), which is stored in the sliding window $\mathbf{M}$ (10) at the window's sample time m, and $\hat{u}(m)$ is the control signal for sample time m given by the current FLC designed at time instant k, $(u(\tilde{\mathbf{x}}(m)))$, using the vector $\tilde{\mathbf{x}}(m)$ instead of $\mathbf{x}(m)$. In this way, the estimation control error of x_j at instant k is obtained by:

$$e_{c_j}(k) = \sum_{m=1}^{T_M} (\boldsymbol{\psi}_j(\boldsymbol{x}(m))\mathbf{u}_j^T(m) - \boldsymbol{\psi}_j(\tilde{\mathbf{x}}(m))\boldsymbol{\Theta}_j)^2, \tag{14}$$

where $\boldsymbol{\psi}_j(\boldsymbol{x}(m))$ is defined in (4), and each element of $\mathbf{u}_j(m) = u(m)[1, \ldots, 1]_{1 \times N_j}$ is the real controller signal $u(m)$ stored in the sliding window $\mathbf{M}$ at sample m of the window.

The candidate input variable x_{j^*} to which can be added a fuzzy control rule is given by the one that has the largest estimation control error:

$$j^* = \arg\max_{j \in J}(e_{c_j}(k)), \tag{15}$$

where $J = \{1, \ldots, n\}$ is the set of indices of the input variables.

3.3.3. New Fuzzy Control Rule

When the criteria, defined later in Section 3.3.4, to add fuzzy control rules are met, a fuzzy control rule will be added to the current knowledge base (i.e., fuzzy control rules defined by (2)). In this way, for the candidate input variable x_{j^*} will be added a new fuzzy control rule $R_{j^*}^{new}$, defined by the membership function $A_{j^*}^{new}$ and the consequent $\theta_{j^*}^{new}$.

To design the new fuzzy control rule $R_{j^*}^{new}$, the first step is to create the new membership function $A_{j^*}^{new}$. The membership functions being characterized by complementary triangular MFs (Figure 1), it is necessary to define the center of $A_{j^*}^{new}$. In order to reduce the sensitivity to noise/outliers, the center of the new membership function will be obtained based on (as a function of) the distribution of the control estimation error e_c. For that purpose, first, a variable m_c representative of such distribution is defined as follows:

$$m_c = \frac{\sum_{m=1}^{T_M} m(u(m) - \hat{u}(m))^2}{\sum_{m=1}^{T_M} (u(m) - \hat{u}(m))^2}, \tag{16}$$

where, similarly to (13), $u(m)$ represents the real controller, which is stored in the sliding window $\mathbf{M}$ (10) at the window's sample time m, and $\hat{u}(m)$ is the control signal (13) given by the current FLC designed at time instant k ($u(\tilde{\mathbf{x}}(m))$) using the vector $\tilde{\mathbf{x}}(m)$. The center of the new membership function $A_{j^*}^{new}$ is given by $\tilde{x}_{j^*}(\lceil m_c \rfloor)$, where $\tilde{x}_{j^*}$ is the j^*-th variable (15) (component) of $\tilde{\mathbf{x}}$, $\lceil m_c \rfloor$ is the integer nearest to m_c (16), and $\tilde{\mathbf{x}}(\lceil m_c \rfloor)$ is the value of $\tilde{\mathbf{x}}$ on the $\lceil m_c \rfloor$-th sample time of the sliding window $\mathbf{M}$ (10).

The new membership function $A_{j^*}^{new}$ and the nearest left ($A_{j^*}^{left}$) and right ($A_{j^*}^{right}$) membership functions (which should be updated due to the introduction of a new MF on the respective variable) are given by:

- $A_{j^*}^{left}$: $c_{j^*,left} = b_{j^*,new}$;
- $A_{j^*}^{new}$: $a_{j^*,new} = b_{j^*,left}$; $b_{j^*,new} = \tilde{x}_{j^*}(\lceil m_c \rfloor)$; and $c_{j^*,new} = b_{j^*,right}$;
- $A_{j^*}^{right}$: $a_{j^*,right} = b_{j^*,new}$.

Since the antecedent membership functions are defined as complementary triangular membership functions (Figure 1), then in $A_{j^*}^{left}$ and $A_{j^*}^{right}$, only one of their parameters is updated due to the introduction of the membership function $A_{j^*}^{new}$.

The consequent parameter of $R_{j^*}^{new}$ is obtained as follows:

$$\theta_{j^*}^{new}(k) = \frac{\sum_{i_j=left}^{right} \mu_{A_{j^*}^{i_j}}(x_{j^*}(k))\,\theta_j^{i_j}}{\sum_{i_j=left}^{right} \mu_{A_{j^*}^{i_j}}(x_{j^*}(k))}, \tag{17}$$

where $left = i_j^* - 1$ and $right = i_j^* + 1$ are respectively the nearest left $A_{j^*}^{left}$ and right $A_{j^*}^{right}$ membership functions (MFs). With (17), the impact of the introduction of a new control rule, while the process is under control, is reduced.

3.3.4. Criteria to Add Control Rules

Two criteria are used to create a new fuzzy control rule.

Criterion 1. *$\|\Delta e_c(k)\| > \delta$, with $\Delta e_c(k) = e_c(k) - e_c(k-1)$, where δ is a threshold defined by the user.*

Criterion 1 is based on the variation of the estimated control error, which if it is larger than a threshold (δ), gives an indication to add a control rule. In this paper, the threshold (δ) is defined as a percentage of the range of the universe of discourse of the control variable $\Delta u = u^+ - u^-$, for example $\delta = 5\% \times \Delta u$. If Criterion 1 is met, the candidate input variable x_{j^*} is selected by (15) (Section 3.3.2) to which a new fuzzy control rule can be added. Then, a candidate fuzzy control rule $R_{j^*}^{new}$ is defined, using the candidate membership function $A_{j^*}^{new}$ as explained in Section 3.3.3. Then, Criterion 2 is used to decide if the candidate fuzzy control rule $R_{j^*}^{new}$ should be added to the candidate input variable x_{j^*}, ensuring a minimal distance between membership functions.

Criterion 2. *$\left| b_{j,i_j^*}(k) - b_{j,i_j^{\pm}}(k) \right| > \eta_j$, where b_{j,i_j^*} and $b_{j,i_j^{\pm}}$ are the centers of $A_{j^*}^{new}$ and of its nearest membership function ($A_{j^{\pm}}$) respectively, and η_j is a threshold defined by the user to define the minimal distance between two closest membership functions for input variable x_j.*

Criterion 2 is defined in order to avoid the learning of a complex controller structure, namely limiting an excessively fine partitioning of the input variables spaces (also avoiding overfitting cases and the creation of a large number of fuzzy control rules) and improving the controller interpretability. In this paper, the threshold (η_j) is defined based on the range of the universe of discourses of the input variables, for example, for input variable x_j, $\eta_j = \left| x_j^+ - x_j^- \right| / K_j$, where K_j can be represented by the maximal number of membership functions for x_j.

3.3.5. Delete Fuzzy Control Rule

Taking into account the complex characteristics of industrial processes, such as unknown and time-varying dynamics, and disturbances, some control rules created earlier may become obsolete, and the presence of outliers (besides the criteria to avoid that) can create the wrong rules. However, in order to not remove a fuzzy control rule with significant information, the deletion of a control rule must be done carefully. In this way, two criteria to delete "less active" or "less informative" fuzzy control rules are used [36].

Criterion 3. $\frac{N_{i_j}^{total}(k)}{k-k_{i_j}^{ini}} < \epsilon$*, for* $i_j = 1, \ldots, N_j$ *and* $j = 1, \ldots, n$*, where* $N_{i_j}^{total}(k) = N_{i_j}^{total}(k-1) + \omega_j^{i_j}[\mathbf{x}(k)]$ *is the indicator of the total activation degrees (antecedent values) associated with the* i_j*-th fuzzy rule, being* $N_{i_j(0)}^{total} = 0$ *[37,38].* k *is the current instant of time, and* $k_{i_j}^{ini}$ *is the instant of time at which the fuzzy control rule* $R_j^{i_j}$ *was created.* $\epsilon \in [0,1]$ *is a positive constant.*

A usual choice has been $\epsilon = 0.1$ [36,39].

Criterion 4. $1 - \frac{N_{i_j}^{min}(k)}{k-k_{i_j}^{ini}} < \epsilon/2$*, for* $i_j = 1, \ldots, N_j$ *and* $j = 1, \ldots, n$*, where* $N_{i_j}^{min}(k)$ *is given by (18), which is the total number of times (during the interval of time* $[k_{i_j}^{ini}, k]$*) in which the antecedent value of the* i_j*-th fuzzy rule,* $\omega_j^{i_j}[\mathbf{x}(k)]$*(3), is the minimum among all the antecedent values of all fuzzy rules, and* $k_{i_j}^{ini}$ *is the instant of time at which fuzzy control rule* $R_j^{i_j}$ *was created.* $\epsilon \in [0,1]$ *is the same positive constant as in Criterion (3).*

$$N_{i_j}^{min}(k) = \sum_{k_{i_j}^{ini}}^{k} \left[\!\!\left[\omega_j^{i_j}[\mathbf{x}(k)] = \min_{\substack{i_j=1,\ldots,N_j \\ j=1,\ldots,n}} (\omega_j^{i_j}[\mathbf{x}(k)]) \right]\!\!\right], \tag{18}$$

where $[\![\ldots]\!]$ are the Iverson brackets and $[\![P]\!]$ is defined to be one if P is true, and otherwise zero.

Criterion 3 gives an indication of the use of a given fuzzy rule, by its antecedent value, $\omega_j^{i_j}[\mathbf{x}(k)]$(3), during an interval of time ($[k_{i_j}^{ini}, k]$). In this way, Criterion 3 is defined in order to remove control rules that are less active. On the other hand, Criterion (4) gives an indication of how long a given rule had the lowest antecedent value $\omega_j^{i_j}[\mathbf{x}(k)]$ (3), during an interval of time ($[k_{i_j}^{ini}, k]$). In this way, Criterion (4) is defined in order to remove control rules that are non-informative.

For all instants of time k, Criteria 3 and 4 are verified for all control rules, and when simultaneously they are met, then fuzzy control rule $R_{j^-}^{del}$ will be deleted from the current knowledge base, as well as the associated membership function $A_{j^-}^{del}$. After removing membership function $A_{j^-}^{del}$, the nearest left ($A_{j^-}^{left}$) and right ($A_{j^-}^{right}$) membership functions (which are updated due the elimination of $A_{j^-}^{del}$) are given by:

- For $A_{j^-}^{left}$: 1) $c_{j^-,left} = b_{j^-,right}$;
- For $A_{j^-}^{right}$: 1) $a_{j^-,right} = b_{j^-,left}$.

Since complementary triangular membership functions are used (Figure 1), only one of the parameters is updated for each of $A_{j^-}^{left}$ and $A_{j^-}^{right}$.

3.3.6. Algorithm

Algorithm 1 presents the steps of the proposed self-evolving method to design the FLC controller, and Figure 3 presents the respective flowchart.

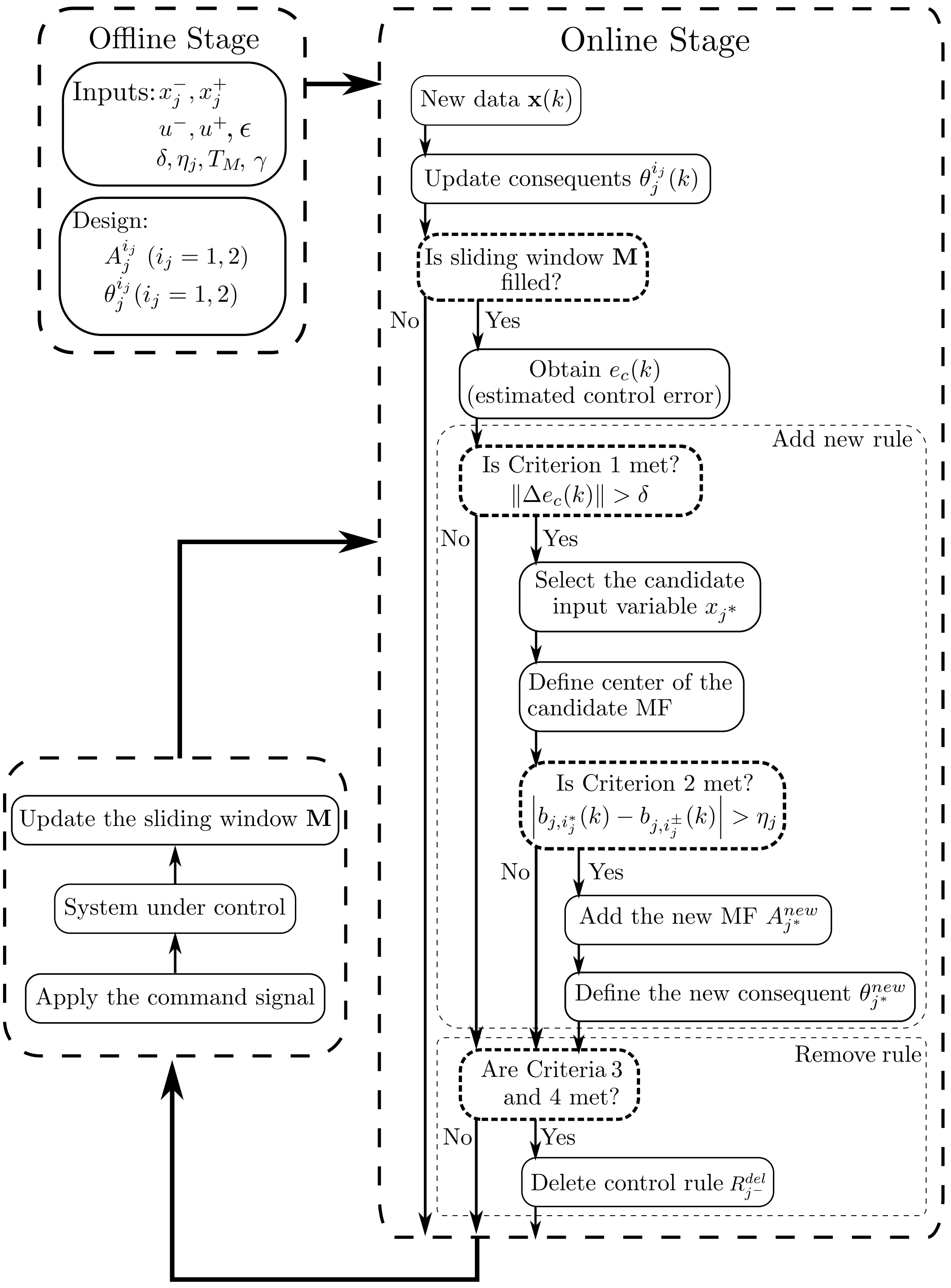

Offline Stage
Inputs: x_j^-, x_j^+ u^-, u^+, ϵ $\delta, \eta_j, T_M, \gamma$
Design: $A_j^{i_j}$ $(i_j = 1, 2)$ $\theta_j^{i_j}$ $(i_j = 1, 2)$
Online Stage
New data $\mathbf{x}(k)$
Update consequents $\theta_j^{i_j}(k)$
Is sliding window $\mathbf{M}$ filled?
No
Yes
Obtain $e_c(k)$ (estimated control error)
Add new rule
Is Criterion 1 met? $\|\Delta e_c(k)\| > \delta$
No
Yes
Select the candidate input variable x_{j^*}
Define center of the candidate MF
Is Criterion 2 met? $\left|b_{j,i_j^*}(k) - b_{j,i_j^\pm}(k)\right| > \eta_j$
No
Yes
Add the new MF $A_{j^*}^{new}$
Define the new consequent $\theta_{j^*}^{new}$
Remove rule
Are Criteria 3 and 4 met?
No
Yes
Delete control rule $R_{j^-}^{del}$
Update the sliding window $\mathbf{M}$
System under control
Apply the command signal

Figure 3. Flowchart of the proposed self-evolving FLC design method.

Algorithm 1 Proposed self-evolving FLC design method.

```
Input:
 1: Range, minimum, and maximum values, of the variables $x_j^-$ and $x_j^+$ ($j = 1, \ldots, n$) and $u^-$ and $u^+$;
 2: Thresholds: $\delta$ (Criterion 1), $\eta_j$ for $j = 1, \ldots, n$ (Criterion 2); $\epsilon$ (Criteria 3 and 4), sliding window's
    size $T_M$, and $\gamma$;
Offline Stage: Design the initial fuzzy controller (Section 3.2);
 3: Antecedent part: design the membership functions $A_j^{i_j}$ ($i_j = 1, 2$):
 4: for all all input variables $j = 1, \ldots, n$ do
 5:     parameters of $A_j^1$: $a_{j,1} = b_{j,1} = x_j^-$ and $c_{j,1} = x_j^+$;
 6:     parameters of $A_j^2$: $a_{j,2} = x_j^-$ and $b_{j,2} = c_{j,2} = x_j^+$;
 7: end for
 8: Consequent part: define all consequent parameters as the minimum control value $\theta_j^{i_j} = u^-$
    ($i_j = 1, 2$);
Online Stage:
 9: while the controller is turned on, $k = 1, 2, \ldots$ do
10:     Update the consequent parameters by (12) (Section 3.3.1);
11:     if sliding window M is filled then
12:         Obtain the estimated control error $e_c(k)$ using (13);
13:         if Criterion 1 is met then
14:             Select, using (15), the candidate input variable $x_{j^*}$ in which a new control rule can be
    added (Section 3.3.2);
15:             Obtain the center of the candidate MF ($A_{j^*}^{new}$) by (16);
16:             if Criterion 2 is met then
17:                 Add $A_{j^*}^{new}$ (new MF) to the selected input variable $x_{j^*}$ (Section 3.3.3);
18:                 Update the nearest left ($A_{j^*}^{left}$) and right ($A_{j^*}^{right}$) membership functions (Section 3.3.3);
19:                 Define the consequent parameter $\theta_{j^*}^{new}$ of the new fuzzy control rule using (17);
20:             end if
21:         end if
22:         if Criteria 3 and 4 are met then
23:             Delete the fuzzy control rule $R_{j^-}^{del}$ (Section 3.3.5);
24:             Delete membership function $A_{j^-}^{del}$ (Section 3.3.5);
25:             Update the nearest left ($A_{j^-}^{left}$) and right ($A_{j^-}^{right}$) membership functions (Section 3.3.5);
26:         end if
27:     end if
28:     Apply the command signal of the current designed fuzzy logic controller, and read the output
    variable $y(k)$.
29:     Update M (sliding window);
30: end while
```

4. Results

This section presents the results' analysis of the proposed evolving methodology, named as uSelf-FLC (Available at https://home.isr.uc.pt/~jermendes/uSelf-FLC.html). For that purpose, a nonlinear continuously stirred tank reactor is used.

4.1. Description of the CSTR Plant

The nonlinear CSTR plant is given by [31,40]:

$$\frac{\partial C_A(t+d_c)}{\partial t} = \frac{q(t)}{V}\left(C_{A0}(t) - C_A(t+d_c)\right) - k_0 C_A(t+d_c)\exp\left(-\frac{E}{RT(t)}\right), \tag{19}$$

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

$$y(t) = C_A(t), \quad u(t) = q_c(t), \tag{20}$$

where Table 1 presents the description of the variables and the respective nominal values. In this plant, the goal is to control the $C_A(t)$ by manipulating $q_c(t)$.

Table 1. CSTR variables [40,41].

Variable-Description [31]	**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)

4.2. Initialization and Offline Stage

For the results' analysis presented here, the following input variables for the FLC were defined: $\mathbf{x}(k) = [r(k), y(k)]$, where $r(k)$ is the reference signal. The parameters of the proposed evolving methodology, which were defined manually, are: variables' range $x_1^- = x_2^- = 0.05$, $x_1^+ = x_2^+ = 0.12$; $u^- = 90$ and $u^+ = 110$ ($\Delta u = 15$); thresholds $\eta_j = \left|x_j^+ - x_j^-\right|/15$ (for $j = 1,2$), $\delta = 0.05 \times \Delta u$, and $\epsilon = 0.1$; $M = 1000$; and $\gamma = 0.35$. The FLC was initially offline designed with the assumption that there was no control knowledge about the process under control; see Section 3.2. The parameters of the SEDFLC (Available at https://home.isr.uc.pt/~jermendes/SEDFLC.html) evolving method proposed in [31] were defined the same as the proposed method, in order to compare both evolving design methods in the same conditions.

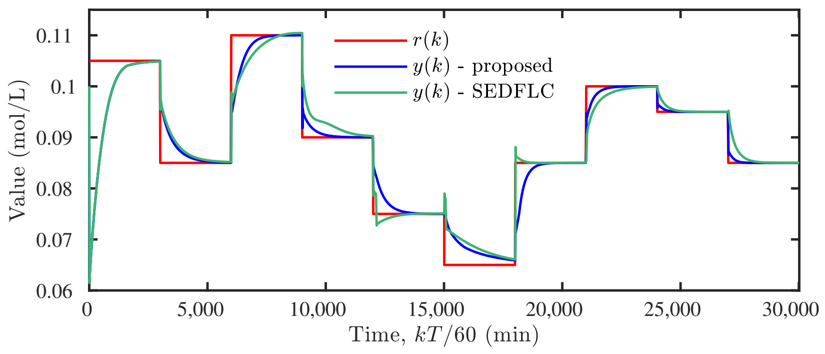
4.3. Regions of Operation

In order to test the evolving capability of the proposed design methodology, the reference $r(t)$ was designed so as to expose the controller to unknown regions of operation; some regions of operation were close to the ones previously learned; and other regions of operation that the controller had previously learned, but never reached again, in order to make some control rules obsolete.

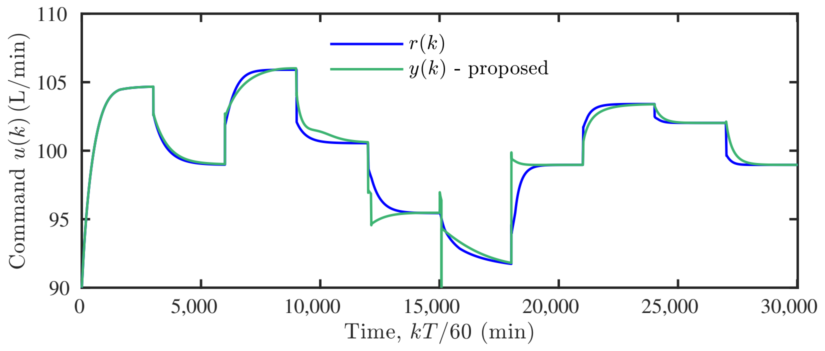
4.4. Results' Analysis

The results, along time, are presented in Figures 4–6, where:

- Figure 4 presents the global results of the direct FLC controller online designed by the proposed self-evolving methodology, in which Figure 4a presents the evolution of the tracking performance for the unknown regions of operation, Figure 4b presents the evolution of control signal, $u(k)$, Figure 4c shows the evolution of the number of fuzzy control rules for each input variable, x_1 and x_2, and Figure 4d shows the evolution of $\Delta e_c(k)$, which is associated with Criterion 1 to add new fuzzy control rules. Figure 4a–b presents also the results of the SEDFLC evolving method proposed in [31].
- Figures 5 and 6 present the evolution of the antecedent and consequent parameters, i.e., the structural changes along time and the consequent adaptation, and the final membership functions for $x_1(k) = r(k)$ and $x_2(k) = y(k)$, respectively.

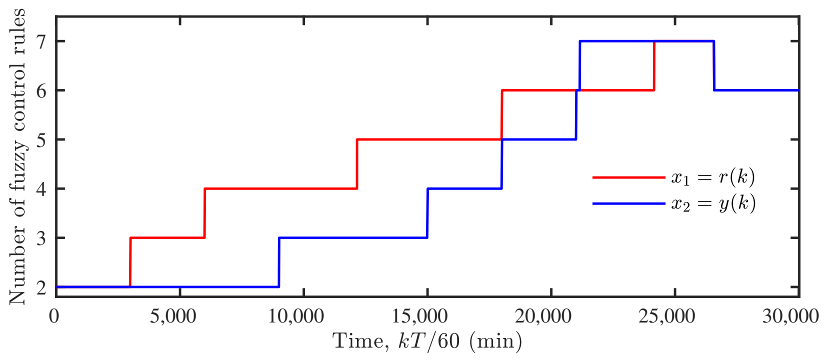


(**a**)

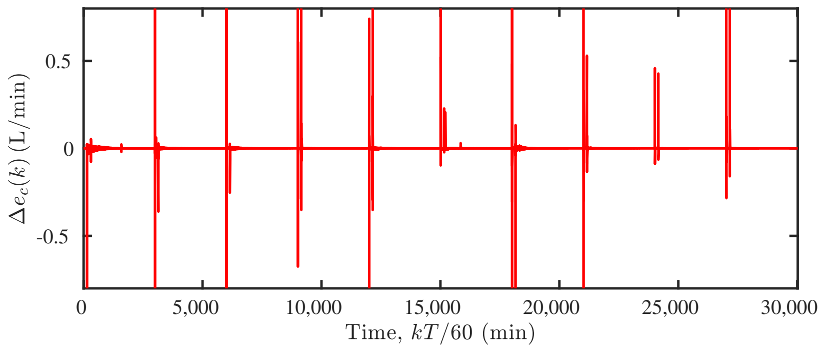


(**b**)

Figure 4. *Cont.*

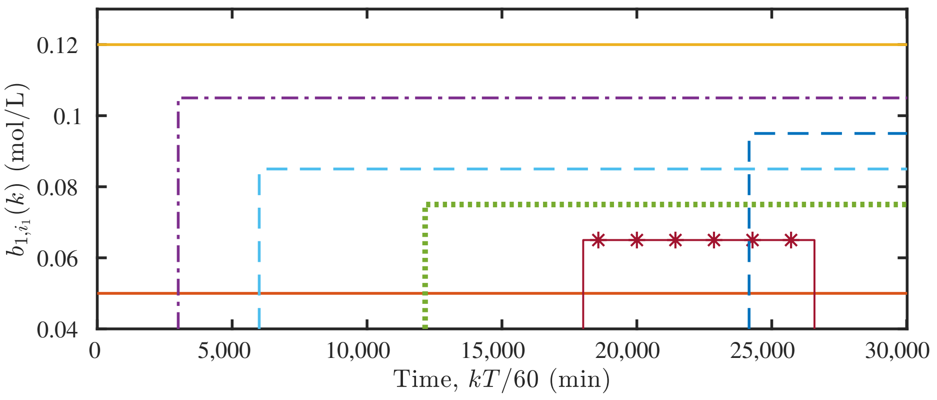


(**c**)



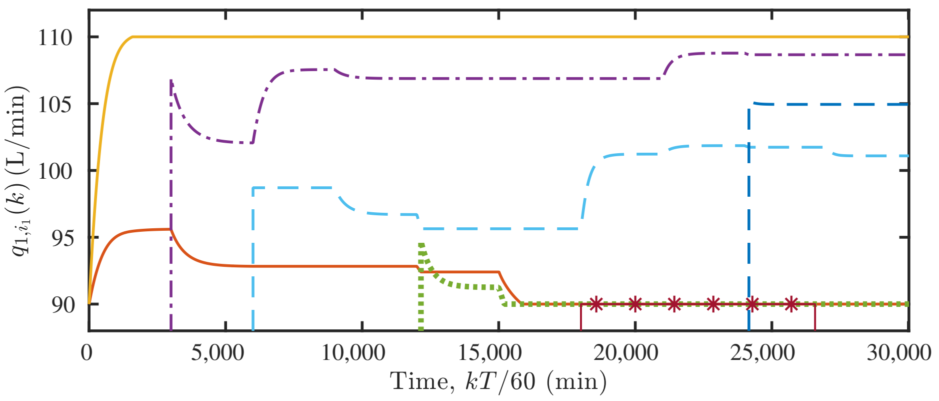
(**d**)

Figure 4. Results of the proposed self-evolving FLC design methodology on the CSTR plant. (**a**) Results of the proposed methodology, and of the SEDFLC method [31]. (**b**) Command signal of the proposed method and of the SEDFLC method [31]. (**c**) Evolution of the number of fuzzy control rules. (**d**) Evolution of $\Delta e_c(k)$.

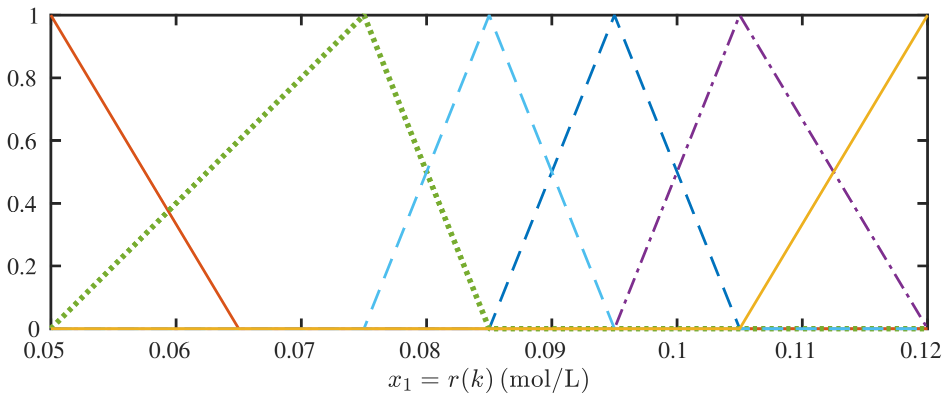


(**a**)

Figure 5. *Cont.*

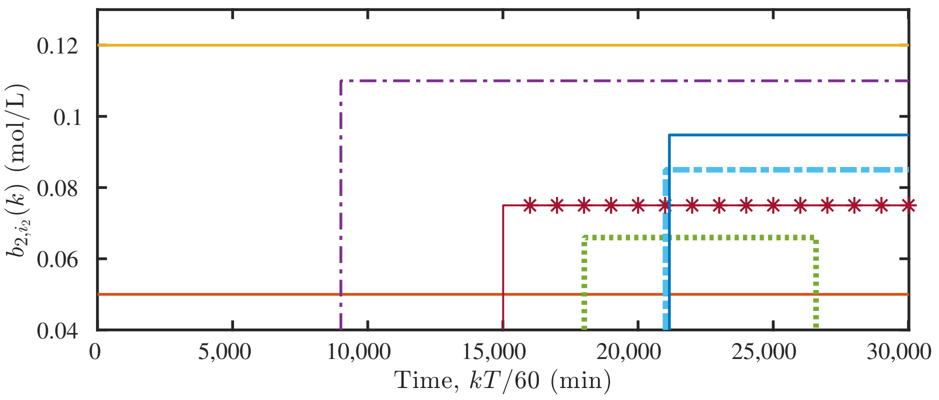


(**b**)



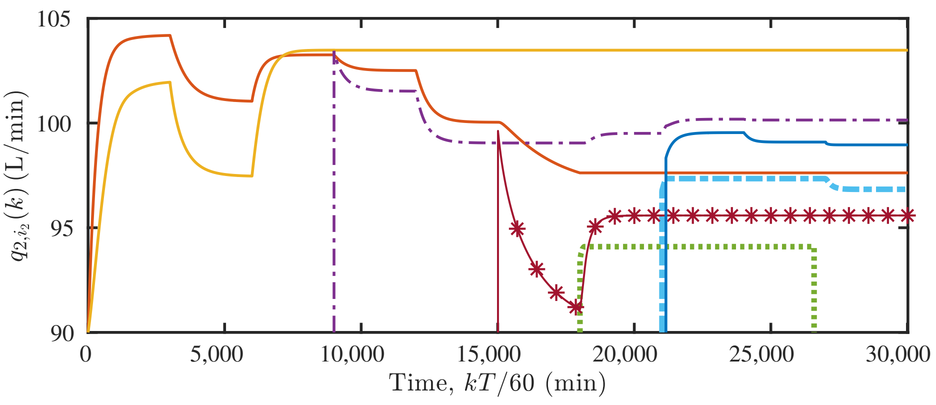
(**c**)

Figure 5. Evolution of the antecedent and consequent parameters of $x_1 = r(t)$. (**a**) Evolution of the antecedent parameters of $x_1 = r(t)$. (**b**) Evolution of the consequent parameters of $x_1 = r(t)$. (**c**) Final membership functions of $x_1 = r(t)$.

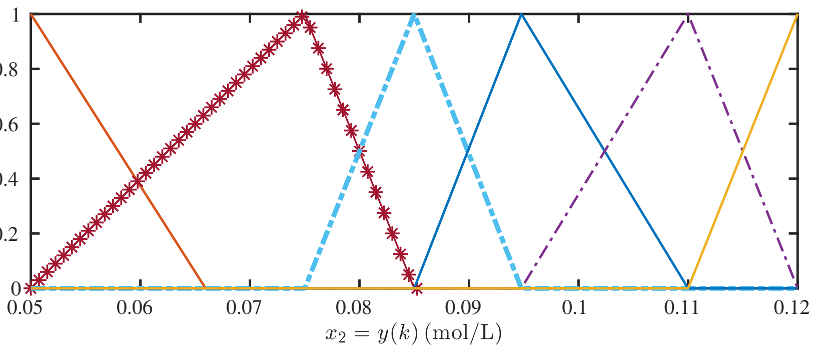


(**a**)

Figure 6. *Cont.*



(**b**)



(**c**)

Figure 6. Evolution of the antecedent and consequent parameters of $x_2 = y(t)$. (**a**) Evolution of the antecedent parameters of $x_2 = y(t)$. (**b**) Evolution of the consequent parameters of $x_2 = y(t)$. (**c**) Final membership functions of $x_2 = y(t)$.

From the results, analyzing the performance of the direct FLC, online designed by the proposed self-evolving methodology, it can be seen that:

- Since, initially, the controller is offline designed using only the variables range values, using two fuzzy control rules per input variable, where the membership functions were defined as presented in Figure 2, thus, the controller is initialized without any control knowledge or previous data of the process under control, i.e., without knowledge of any region of operation. It can be seen from the results that the tracking performance increases during the time of operation and that the proposed evolving methodology adds new fuzzy control rules (see Figure 4c) when unknown regions of operation are reached.
- When the reference has the value $r(k) = 0.065$ ($1500 \leq k < 1800$), that $r(k)$ region of operation has not been learned (reached) previously, and new control rules were added in both input variables. Afterwards, as that region of operation was never reached again, the control rules that were previously added for that region were deleted, due to the fact that Criteria 3 and 4 considered these rules obsolete ("less active" and "less informative").
- Additionally, it can be seen that for $k < 1800$, there are large changes in the control structures, namely in the antecedent (Figures 5a and 6a) and consequent parts (Figures 5b and 6b) because, until then ($k < 1800$), most of the regions of operation were unknown to the controller, and the proposed design methodology online evolved the control structure (i.e., the fuzzy control rules); afterwards, for $k \geqslant 1800$, with the exception of the process of deleting the control rules, only small changes were made in the antecedent and the consequent parts of the rules, since in that time interval, the regions of operation were similar to the ones already learned before.

- Figure 4a–b shows that the proposed method has outperformed the SEDFLC evolving method [31], where both methods have used the same initial parameters, and that the SEDFLC method did not react as well as the self-evolving FLC to the new $r(k) = 0.065$ ($1500 \leq k < 1800$) region of operation, which is distant from the previous operating region.
- It can be seen that the proposed self-evolving direct FLC controller design methodology successfully online designed the FLC controller, reaching a simple control structure, where each of the input variables ($r(k)$ and $y(k)$) was described by six fuzzy control rules, whose membership functions are described in Figures 5c and 6c, and the final fuzzy control rules for $x_1(k) = r(k)$ are described by:

$$\begin{aligned}
\text{Rule 1}: &\quad \text{IF } r(k) \text{ is } A_1^1 \text{ THEN } q_c(k) \text{ is } 90,\\
\text{Rule 2}: &\quad \text{IF } r(k) \text{ is } A_1^2 \text{ THEN } q_c(k) \text{ is } 90,\\
\text{Rule 3}: &\quad \text{IF } r(k) \text{ is } A_1^3 \text{ THEN } q_c(k) \text{ is } 101.1,\\
\text{Rule 4}: &\quad \text{IF } r(k) \text{ is } A_1^4 \text{ THEN } q_c(k) \text{ is } 104.9,\\
\text{Rule 5}: &\quad \text{IF } r(k) \text{ is } A_1^5 \text{ THEN } q_c(k) \text{ is } 108.7,\\
\text{Rule 6}: &\quad \text{IF } r(k) \text{ is } A_1^6 \text{ THEN } q_c(k) \text{ is } 110,
\end{aligned}$$

and the final fuzzy control rules for $x_2(k) = C_A(k)$ are described by:

$$\begin{aligned}
\text{Rule 1}: &\quad \text{IF } C_A(k) \text{ is } A_2^1 \text{ THEN } q_c(k) \text{ is } 97.6,\\
\text{Rule 2}: &\quad \text{IF } C_A(k) \text{ is } A_2^2 \text{ THEN } q_c(k) \text{ is } 95.6,\\
\text{Rule 3}: &\quad \text{IF } C_A(k) \text{ is } A_2^3 \text{ THEN } q_c(k) \text{ is } 96.8,\\
\text{Rule 4}: &\quad \text{IF } C_A(k) \text{ is } A_2^4 \text{ THEN } q_c(k) \text{ is } 99,\\
\text{Rule 5}: &\quad \text{IF } C_A(k) \text{ is } A_2^5 \text{ THEN } q_c(k) \text{ is } 100.1,\\
\text{Rule 6}: &\quad \text{IF } C_A(k) \text{ is } A_2^6 \text{ THEN } q_c(k) \text{ is } 103.5.
\end{aligned} \tag{21}$$

Regarding the application of the proposed self-evolving method to other SISO processes, defined by (8), it is expected that the initialization and offline stage will be intuitively defined, similarly to the CSTR plant. The offline design is performed using only the range limits of the process's variables (input and output), which is mandatory information to control any process; and the thresholds used on the criteria to add or remove control rules are also intuitively defined, being given by a percentage of the variables range, while $\epsilon \in [0, 1]$ has also an intuitive meaning. The sliding window's size T_M and the consequent adaptation gain γ are parameters that must have some know-how about the process to be controlled; T_M must be defined in order to allocate relevant information to obtain the estimated control error $e_c(k)$ and γ defined in order to have an acceptable performance in the beginning (until several control rules are created in order to cover several regions of operation). In terms of performance on other SISO processes, it is expected that the proposed method will have a better performance in industrial process with slow variations, which has happened in several industrial processes with, for example, sample times of $0.5, 1, 2,$ and 5 s. In terms of the interpretation of the designed controller in other SISO processes, which is an important goal, due to the defined univariate control structure (2) and the defined criteria adding/removing control rules (mainly Criterion 2 that limits excessive partitioning of the input variables spaces), it is expected that the designed FLC will have a simple (interpretable) structure.

5. Conclusions

This paper proposed an online evolving methodology for the design, in an evolving way, of a direct fuzzy logic controller (FLC). The proposed methodology, in an initial offline stage, initializes the control rules using only the range information of the process' variables; then, in an online stage (when the process is under control), the proposed evolving methodology, in an evolving way, can add new fuzzy control rules or remove them, based on the respective defined criterion. The controller structure is formed by univariate fuzzy control rules, in order to improve the understandability by human operators.

Experimental tests were performed on a simulated CSTR plant showing that the proposed methodology successfully online designed the FLC, adding new control rules when new (for the controller) regions of operations were reached and deleting control rules when some previously added control rules became obsolete.

Author Contributions: Conceptualization, methodology, formal analysis, and writing, original draft preparation: J.M.; software, J.M. and R.M.; validation and editing, J.M., R.A., and F.A.A.S. All authors read and agreed to the published version of the manuscript.

Funding: Ricardo Maia was supported by Fundação para a Ciência e a Tecnologia (FCT) under the grant SFRH/BD/44644/2008.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Precup, R.E.; Hellendoorn, H. A survey on industrial applications of fuzzy control. *Comput. Ind.* **2011**, *62*, 213–226. [CrossRef]
2. Cara, A.B.; Pomares, H.; Rojas, I.; Lendek, Z.; Babuška, R. Online Self-Evolving Fuzzy Controller With Global Learning Capabilities. *Evol. Syst.* **2010**, *1*, 225–239. [CrossRef]
3. Wang, L.X. *A Course in Fuzzy Systems and Control*; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1997.
4. Mendes, J.; Araújo, R.; Sousa, P.; Apóstolo, F.; Alves, L. An Architecture for Adaptive Fuzzy Control in Industrial Environments. *Comput. Ind.* **2011**, *62*, 364–373. [CrossRef]
5. Mendes, J.; Seco, R.; Araújo, R. Automatic Extraction of the Fuzzy Control System for Industrial Processes. In Proceedings of the 16th IEEE International Conference on Emerging Technologies and Factory Automation, Toulouse, France, 5–9 September 2011; pp. 1–8.
6. Mendes, J.; Araújo, R.; Matias, T.; Seco, R.; Belchior, C. Automatic Extraction of the Fuzzy Control System by a Hierarchical Genetic Algorithm. *Eng. Appl. Artif. Intell.* **2014**, *29*, 70–78. [CrossRef]
7. Mendes, J.; Araújo, R.; Matias, T.; Seco, R.; Belchior, C. Evolutionary Learning of a Fuzzy Controller for Industrial Processes. In Proceedings of the 40th Annual Conference of the IEEE Industrial Electronics Society (IECON 2014), Dallas, TX, USA, 29 October–1 November 2014; pp. 139–145.
8. Lotfy, M.E.; Senjyu, T.; Farahat, M.A.F.; Abdel-Gawad, A.F.; Lei, L.; Datta, M. Hybrid Genetic Algorithm Fuzzy-Based Control Schemes for Small Power System with High-Penetration Wind Farms. *Appl. Sci.* **2018**, *8*, 373. [CrossRef]
9. Castillo, O.; Lizárraga, E.; Soria, J.; Melin, P.; Valdez, F. New Approach Using Ant Colony Optimization With Ant Set Partition for Fuzzy Control Design Applied to the Ball and Beam System. *Inf. Sci.* **2015**, *294*, 203–215. [CrossRef]
10. Caraveo, C.; Valdez, F.; Castillo, O. Optimization of Fuzzy Controller Design Using a New Bee Colony Algorithm With Fuzzy Dynamic Parameter Adaptation. *Appl. Soft Comput.* **2016**, *43*, 131–142. [CrossRef]
11. Rubaai, A.; Young, P. Hardware/Software Implementation of Fuzzy-Neural-Network Self-Learning Control Methods for Brushless DC Motor Drives. *IEEE Trans. Ind. Appl.* **2016**, *52*, 414–424. [CrossRef]
12. Leite, D.; Škrjanc, I.; Gomide, F. An overview on evolving systems and learning from stream data. *Evol. Syst.* **2020**, *11*, 181–198. [CrossRef]
13. Škrjanc, I.; Iglesias, J.A.; Sanchis, A.; Leite, D.; Lughofer, E.; Gomide, F. Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A Survey. *Inf. Sci.* **2019**, *490*, 344–368. [CrossRef]
14. Angelov, P. A fuzzy controller with evolving structure. *Inf. Sci.* **2004**, *161*, 21–35. [CrossRef]
15. Blazic, S.; Dovzan, D.; Skrjanc, I. Robust Evolving Fuzzy Adaptive Control With Input-domain Clustering. *IFAC Proc. Vol.* **2014**, *47*, 5387–5392. [CrossRef]
16. Blažič, S.; Škrjanc, I.; Matko, D. A robust fuzzy adaptive law for evolving control systems. *Evol. Syst.* **2014**, *5*, 3–10. [CrossRef]

17. Cara, A.; Herrera, L.; Pomares, H.; Rojas, I. New Online Self-Evolving Neuro Fuzzy controller based on the TaSe-NF model. *Inf. Sci.* **2013**, *220*, 226–243. [CrossRef]
18. Leite, D.; Palhares, R.M.; Campos, V.C.S.; Gomide, F. Evolving Granular Fuzzy Model-Based Control of Nonlinear Dynamic Systems. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 923–938. [CrossRef]
19. Chen, S.Y.; Liu, T.S. Intelligent tracking control of a PMLSM using self-evolving probabilistic fuzzy neural network. *IET Electr. Power Appl.* **2017**, *11*, 1043–1054. [CrossRef]
20. Rong, H.J.; Yang, Z.X.; Wong, P.K.; Vong, C.M.; Zhao, G.S. Self-evolving fuzzy model-based controller with online structure and parameter learning for hypersonic vehicle. *Aerosp. Sci. Technol.* **2017**, *64*, 1–15. [CrossRef]
21. Le, T.L.; Lin, C.M.; Huynh, T.T. Self-evolving type-2 fuzzy brain emotional learning control design for chaotic systems using PSO. *Appl. Soft Comput.* **2018**, *73*, 418–433. [CrossRef]
22. Le, T.L.; Huynh, T.T.; Lin, C.M. Self-Evolving Interval Type-2 Wavelet Cerebellar Model Articulation Control Design for Uncertain Nonlinear Systems Using PSO. *Int. J. Fuzzy Syst.* **2019**, *21*, 2524–2541. [CrossRef]
23. Le, T.L.; Huynh, T.T.; Nguyen, V.Q.; Lin, C.M.; Hong, S.K. Chaotic Synchronization Using a Self-Evolving Recurrent Interval Type-2 Petri Cerebellar Model Articulation Controller. *Mathematics* **2020**, *8*, 219. [CrossRef]
24. Cara, A.B.; Pomares, H.; Rojas, I. A New Methodology for the Online Adaptation of Fuzzy Self-Structuring Controllers. *IEEE Trans. Fuzzy Syst.* **2011**, *19*, 449–464. [CrossRef]
25. Angelov, P.; Škrjanc, I.; Blažič, S. Robust Evolving Cloud-Based Controller for a Hydraulic Plant. In Proceedings of the 2013 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Singapore, 16–19 April 2013; pp. 1–8.
26. Costa, B.; Skrjanc, I.; Blazic, S.; Angelov, P. A practical Implementation of Self-Evolving Cloud-Based Control of a Pilot Plant. In Proceedings of the IEEE International Conference on Cybernetics (CYBCO), Lausanne, Switzerland, 13–15 June 2013; pp. 7–12.
27. Škrjanc, I.; Blažič, S.; Angelov, P. Robust Evolving Cloud-Based PID Control Adjusted by Gradient Learning Method. In Proceedings of the 2014 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Linz, Austria, 2–4 June 2014; pp. 1–8.
28. Andonovski, G.; Angelov, P.; Blažič, S.; Škrjanc, I. A practical implementation of Robust Evolving Cloud-based Controller with normalized data space for heat-exchanger plant. *Appl. Soft Comput.* **2016**, *48*, 29–38. [CrossRef]
29. Yang, Z.X.; Rong, H.J.; Wong, P.K.; Angelov, P.; Yang, Z.X.; Wang, H. Self-Evolving Data Cloud-based PID-like Controller for Nonlinear Uncertain Systems. *IEEE Trans. Ind. Electron.* **2020**. [CrossRef]
30. Mendes, J.; Craveiro, A.; Araújo, R. Iterative Design of a Mamdani Fuzzy Controller. In Proceedings of the 13th APCA/IEEE International Conference on Automatic Control and Soft Computing (CONTROLO 2018), Ponta Delgada, Portugal, 4–6 June 2018; pp. 85–90.
31. Mendes, J.; Souza, F.; Araújo, R. Online Evolving Fuzzy Control Design: An Application to a CSTR Plant. In Proceedings of the IEEE 15th International Conference on Industrial Informatics (INDIN 2017), Emden, Germany, 24–26 July 2017; pp. 218–225.
32. Silva, A.M.; Caminhas, W.; Lemos, A.; Gomide, F. A Fast Learning Algorithm for Evolving Neo-Fuzzy Neuron. *Appl. Soft Comput.* **2014**, *14 Pt B*, 194–209. [CrossRef]
33. Phan, P.A.; Gale, T.J. Direct adaptive fuzzy control with a self-structuring algorithm. *Fuzzy Sets Syst.* **2008**, *159*, 871–899. [CrossRef]
34. Kosko, B. Fuzzy Systems as Universal Approximators. *IEEE Trans. Comput.* **1994**, *43*, 1329–1333. [CrossRef]
35. Ying, H. General MISO Takagi-Sugeno Fuzzy Systems with Simplified Linear Rule Consequent as Universal Approximators for Control and Modeling Applications. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; Volume 2, pp. 1335–1340.
36. Andonovski, G.; Mušič, G.; Blažič, S.; Škrjanc, I. Evolving model identification for process monitoring and prediction of non-linear systems. *Eng. Appl. Artif. Intell.* **2018**, *68*, 214–221. [CrossRef]
37. Mendes, J.; Maia, R.; Araújo, R.; Gouveia, G. Intelligent Controller for Industrial Processes Applied to a Distributed Two-Tank System. In Proceedings of the 1st IEEE International Conference on Artificial Intelligence for Industries (AI4I 2018), Laguna Hills, CA, USA, 26–28 September 2018; pp. 39–43.
38. Soleimani-B., H.; Lucas, C.; Araabi, B.N. Recursive Gath–Geva Clustering as a Basis for Evolving Neuro-Fuzzy Modeling. *Evol. Syst.* **2010**, *1*, 59–71. [CrossRef]

39. Dovžan, D.; Logar, V.; Škrjanc, I. Implementation of an Evolving Fuzzy Model (eFuMo) in a Monitoring System for a Waste-Water Treatment Process. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 1761–1776. [CrossRef]
40. Morningred, J.D.; Paden, B.E.; Seborg, D.E.; Mellichamp, D.A. An Adaptive Nonlinear Predictive Controller. *Chem. Eng. Sci.* **1992**, *47*, 755–762. [CrossRef]
41. Rastegar, S.; Araújo, R.; Sadati, J.; Mendes, J. A Novel Robust Control Scheme for LTV Systems Using Output Integral Discrete-Time Synergetic Control Theory. *Eur. J. Control* **2017**, *34*, 39–48. [CrossRef]