Evoking Fuzzy Controller, and Application to a Distributed Two-Tank Process

Jérôme Mendes¹*, Ricardo Maia¹, Rui Araújo¹, and Gonçalo Gouveia²

¹Institute of Systems and Robotics (ISR-UC), and Department of Electrical and Computer Engineering (DEEC-UC), University of Coimbra, Pólo II, PT-3030-290 Coimbra, Portugal.
²Critical Software S.A., Pq. Ind. de Taveiro, Lt 49, PT-3045-504 Coimbra, Portugal.
jermendes@isr.uc.pt, rmmaia@isr.uc.pt, rui@isr.uc.pt, gpgouveia@criticalsoftware.com

Abstract—An evolving design approach to fuzzy logic controllers (FLCs) is proposed. The FLC to be evolved is composed of a set of univariate fuzzy control rules, allowing to have a better understanding of the influence of each input variable on the controller’s behavior. The criterion to add new control rules is based on a novelty detection criterion allowing to detect data which are considered new, unknown, or badly representative, with respect to the learned FLC. The proposed approach is tested on a Two-Tank process controlled in a networked environment under a proof-of-concept platform, named KhronoSim, for testing cyberphysical systems in closed-loop. The proposed approach has successfully evolved/designed the FLC, controlling the process on unknown (for the controller) regions of operation.

Index Terms—Fuzzy logic controller, Evolving design, Cyber-physical system (CPS), V&V platform

I. INTRODUCTION

Several research works have shown that a fuzzy logic controller (FLC) can be an appropriate solution to control nonlinear complex industrial processes [1]. This holds, typically, for example, on industrial processes which are difficult to model/control, when the model of the process is unknown, when classic methods are not possible or difficult to be used to design a controller, or when the expert operator control actions knowledge is available [2], [3]. However, there is no standard approach, and it is very difficult, to design an FLC using only the expert operator control actions to control complex nonlinear processes [4].

The design of fuzzy systems has been the focus on several research works on identification, classification, and (indirect and direct) control problems. FLCs are typically distinguished as direct, designed through operator control knowledge, and indirect, designed through operator knowledge about the process [5], [6]. However, only a few works have been focused on direct FLC design. Swarm intelligence and evolutionary algorithms have been used to design FLCs generally in an offline way. However, such methods are computationally heavy. They do not consider the future online system dynamics’ changes and the unknown regions of operation which are not contemplated on the training data. Recently, the interest on online evolving design of direct FLCs has been increasing.

In [4], [7], the design of direct FLCs is performed online, and can start with an empty or a simple set of fuzzy control rules. However, the evolving learning process of [4], [7] can reach a complex control structure, i.e. the learned FLC can be composed of an large number of membership functions (MFs), and control rules. In [8], a new Robust Evolving Cloud-based Controller (RECCo) was proposed, being used on [9]–[11]. The RECCo’s structure differs from standard FLC on the antecedent part which is formed by data clouds instead of MFs. In [8], [9], the evolving process adds new clouds based on the global density of the data, while in [10] the local density is used, and the consequent part is adapted by a stable gradient-based learning method. In [11], a new normalization of the cloud space and consequents adaptation law is proposed. However, since the membership functions belonging to an FLC are replaced by data clouds which do not have a specific shape [9], the interpretability capability of the cloud based controllers is lost. In [12], the method to online evolve the FLC is based on the fact that the FLC, as a universal approximator, can approximate the true inverse function of the process to be controlled. However, because of being based on the plant’s inverse function, the input variables to such function are limited to the reference and the plant variables; and using data from a temporal window the detection of changes in the plant’s dynamics, or of changes related to the transition to an unknown region of operation, are slow.

In this paper a new approach to online evolve a direct FLC is proposed. The rules’ structure of the FLC controller is defined by a set of univariate control rules for each input variable, improving the interpretability of the controller, and reducing the complexity of the learned controller. The proposed approach is composed of two phases, the offline and the online phases. The offline phase, corresponding to the initialization phase, is the initial design of the FLC, which is performed using only the information of the limits of the universe of discourse of the variables, and occurs before the controller is turned on. The online phase, corresponding to the evolving phase, occurs when the system is under control, and it is when new control rules are added based on the novelty criterion used on Recursive Gath–Geva clustering [13], which allows to recursively measure the novelty of new data by a measurement.

*Corresponding author.
between zero and one, allowing an easier definition of the threshold to add new rules. The target system, for which the proposed approach was designed, are multiple-inputs-single-output (MISO) systems with online varying dynamics.

Industrial manufacturing systems are increasingly incorporating cyberphysical systems (CPS) as a result of developments in the industry [14]. In the planning and production context, the several involved information systems must be up-to-date to maintain integration of all areas of the manufacturing process, as in smart factories [15], [16]. The proposed control approach on a Two-Tank process operated under the KhronoSim real-time platform for testing cyberphysical systems in closed-loop are presented on Section IV. Section V presents the conclusions.

The paper is organized as follows. The FLC structure is described in Section II, and the proposed approach to design the FLC is presented in Section III. The results of the proposed approach on a Two-Tank process operated under the KhronoSim real-time platform for testing cyberphysical systems in closed-loop are presented on Section IV. Section V presents the conclusions.

II. FUZZY LOGIC CONTROLLER

This section presents the FLC to be evolved by the proposed approach on Section III. The FLC is defined by a set of control rules for each input variable $x_j$ [12]:

$$R^i_j: \quad \text{IF } x_j(k) \text{ is } A^i_j \text{ THEN } u^i_j(k) = q^i_j,$$

$$R^N_i: \quad \text{IF } x_j(k) \text{ is } A^N_i \text{ THEN } u^N_i(k) = q^N_i,$$

$$R^i_n: \quad \text{IF } x_n(k) \text{ is } A^i_n \text{ THEN } u^i_n(k) = q^i_n,$$

$$R^N_n: \quad \text{IF } x_n(k) \text{ is } A^N_n \text{ THEN } u^N_n(k) = q^N_n,$$

where $R^i_j$ (j = 1, 2, ..., n, and i = 1, 2, ..., N_j) represents the $i$-th control rule of the input variable $x_j$ (x = [x_1, ..., x_n]), N_j is the number of rules for $x_j$, $q^i_j$ is the consequent parameter of $R^i_j$, and $A^i_j$ are linguistic terms.

III. PROPOSED EVOLVING FLC DESIGN METHOD

This section presents the proposed evolving FLC design approach, which is defined by the following main steps:

- Offline initialization, using only the knowledge of the universe of discourse limits of each variable.
- Online process, where new control rule(s) are added on the selected input variable(s), defining the new consequent(s), and membership function(s).

A. Initialization

The initial FLC is designed using only the knowledge of the universe of discourse limits of each variable. Thus, each input variable $x_j$ will be initially described by two MFs (membership functions), $N_j = 2$, corresponding also to

$$\psi_j(x_j(k)) = [\mu^{x_j}(x_j(k)) + \mu^{x_j^+}(x_j(k))] / 2.$$
their initial number of control rules. Since, it is assumed that there is no knowledge about the process to be controlled, the initial consequent parameters \( q_{ij} \) are defined as the minimum admissible control value.

B. Antecedents and Consequents Adaptation

Excluding the centers of the first and last MFs of each input variable \( x_j \), which are fixed to the limits of the universe of discourse of the variable, in order to do fine adjustments, the MFs’ centers are updated by:

\[
b_{j,i}(k) = b_{j,i}(k-1) + \beta \omega_{ij} [x_j(k)][x_j(k) - b_{j,i}(k-1)],
\]

where \( \beta \) is a small positive gain, e.g., \( \beta = 1 \times 10^{-7} \).

The consequent parameters are updated by:

\[
q_{ij}^n(k) = q_{ij}^n(k-1) + C \omega_{ij} [x_j(k)]e(k),
\]

where \( e(k) = r(k) - y(k) \) is the tracking error, and \( C \) is an adaptation gain with the sign of the monotonicity of the output signal of the process with respect to the control signal [7].

C. Creation of a Fuzzy Control Rule

For the proposed approach, it was defined: \( x_j^* \) is/are the input variable(s) selected to receive a new control rule, and \( j^* \) the respective index; \( i_j^* \) is the index of the MF which will be added on \( x_j^* \); \( A_{ij}^{new} \) the new MF added on \( x_j^* \); \( A_{ij}^{left} \) and \( A_{ij}^{right} \) the nearest left and right MFs of \( A_{ij}^{new} \) respectively, and \( left = i_j^* - 1 \) and \( right = i_j^* + 1 \) are the respective indexes.

1) Variables Selection: The variable(s) to be candidate(s) for receiving new control rules are selected based on the novelty criterion used on the Recursive Gath-Geva clustering method proposed in [13]. The goal of this criterion is to detect if an incoming data is new, unknown, or badly represented with respect to the learned FLC. The novelty, for the \( i_j \)-th fuzzy control rule of \( x_j \), is given by the following measurement between zero and one:

\[
M_{ij}(k) = \exp \left( -\frac{1}{2} \left( x_j(k) - \mu_{ij}(k) \right) \Sigma_{ij}^{-1}(k) \left( x_j(k) - \mu_{ij}(k) \right) \right),
\]

where \( \Sigma_{ij}(k) \) is the covariance obtained recursively by:

\[
\Sigma_{ij}(k) = \frac{N_{ij}(k-1)}{N_{ij}(k)} \left( \Sigma_{ij}^{-1}(k) + \frac{\nu_{ij}(k) m}{N_{ij}(k)} \left( x_j(k) - \mu_{ij}(k-1) \right)^2 \right),
\]

where \( \mu_{ij}(k) = \mu_{ij} \left( x_j(k) \right) \), \( m \) is a positive integer greater than one which defines the fuzzification degree of the clusters, usually defined in the literature as \( m = 2 \),

\[
\mu_{ij}(k) = \mu_{ij}(k-1) + \frac{x_j(k) - \mu_{ij}(k-1)}{N_{ij}(k)} \nu_{ij}^m(k),
\]

and

\[
N_{ij}(k) = N_{ij}(k-1) + \nu_{ij}^m(k),
\]

is the sum, for all data points \( x_j(l) \) \( (l = 1, \ldots, k) \), of the membership value of \( x_j(l) \) on the \( i_j \)-th MF of \( x_j \) [13], and \( N_{ij}(0) = 0 \).

In this way, the input variables which satisfy the following criterion, \( x_j^* \), are chosen to receive a new MF,

\[
M_{j,\text{max}}(k) = \frac{N_{ij}}{\sum_{i=1}^{N_{ij}} M_{ij}(k)} \leq M_{th},
\]

where \( M_{th} \) is a threshold between zero and one for \( x_j \), allowing an easier control rule creation threshold definition.

An additional criterion was defined to avoid the learning of a complex controller structure, improving its interpretability, and avoiding overfitting cases. To this end, a minimal distance between the center of two MFs is defined by

\[
\left| b_{j^*,i_j^*} - b_{j_j^*,i_j^*} \right| > \eta_j,
\]

where \( b_{j^*,i_j^*} \) and \( b_{j_j^*,i_j^*} \) are the centers of the MF to be added and of its nearest MF, respectively, and \( \eta_j \) is a threshold to define the minimal distance between the center of two closest MFs for \( x_j \). If the distance is not above the threshold then the new rule is not added.

2) New Consequent Parameter: When criteria (13)–(14) are met, then a new rule is added on \( x_j^* \). In order to reduce the impact of the introduction of the new control rule, the consequent of the new rule \( \beta_{ij}^{new} \) is defined by

\[
q_{ij}^{new}(k) = \frac{\sum_{i=i_j^*-left}^{i_j^*-right} \mu_{A_{ij}^{new}} \left( x_j^*(k) \right) q_{ij}^n(k)}{\sum_{i=i_j^*-left}^{i_j^*-right} \mu_{A_{ij}^{new}} \left( x_j^*(k) \right)}. \tag{15}
\]

In this way the new consequent is defined by the influence of the nearest (previous and next) rules, reducing the impact of the new on the controller’s behavior on the transient moment.

3) New Membership Function: Since complementary triangular MFs are used, then the introduction of the new MF \( A_{ij}^{new} \) will modify the parameters of other MFs, namely \( A_{ij}^{left} \) and \( A_{ij}^{right} \). The new parameters that must be updated are:

- for \( A_{ij}^{left} \): 1) \( c_{ij}^{*,left} = b_{j^*,new} \);
- for \( A_{ij}^{new} \): 1) \( a_{ij}^{*,new} = b_{j^*,left} \), 2) \( b_{j^*,new} = x_j(k) \), and
- for \( A_{ij}^{right} \): 1) \( a_{ij}^{*,right} = b_{j^*,right} \).

4) Proposed Approach - SDeFLC: Algorithm 1 presents the steps of the proposed approach to Simple Design of an Evolving FLC (SDeFLC).

IV. RESULTS

The results and performance of the proposed approach on the control of a Two-Tank process to demonstrate the proof-of-concept of the KhronoSim platform, for testing cyberphysical systems in real-time and closed-loop, are presented in this section.

A. KhronoSim Framework and True Time

The KhronoSim framework is a distributed, modular, and extensible System Verification Facility (SVF) developed for testing cyberphysical systems (CPS) in real-time and closed-loop. The KhronoSim’s architecture is based on a scalable and distributed structure, so additional modules can be added as needed to increase the system capabilities in scenarios where
Algorithm 1 Proposed approach, SDeFLC:

1) **Inputs:** Define for each \( x_j \) \((j = 1, \ldots, n)\), the thresholds \( M_{th}^j \) and \( n_j \), and the initial covariance value \( \Sigma_i (0) \); the universe of discourse limits of the inputs and output, \( x_j^0 \) and \( u^0 \) and \( u^+ \), respectively; and the learning gains \( \beta \) (antecedents) and \( C \) (consequents).

2) **Initialization.** Design the initial FLC structure.
   a) Antecedent part: design the initial membership functions \( A_j^i \) (Figure 1) for each input variable \( x_j \), for \( j = 1, \ldots, n \), and \( i = 1, 2 \):
      i) \( A_1^j \): \( a_{j,1} = b_{j,1} = x_j^0 \), and \( c_{j,1} = b_{j,2} \);
      ii) \( A_2^j \): \( a_{j,2} = b_{j,1} \), and \( b_{j,2} = c_{j,2} = x_j^+ \).
   b) Consequent part: define the initial value of all consequent parameters (\( q \)) as \( u^- \).

3) **Online Evolving learning.**
   a) Antecedent adaptation by (7); b) Consequent adaptation by (8);
   c) Measure the novelty, \( M_{j,\text{max}} \), of \( x(k) \) by (9).
      i) If Criteria (13) and (14) are met, then create the new control rule for each selected the input variable \( x_j \) (Section III-C1):
         A) Obtain the new consequent parameter of the new fuzzy rule using (15) (see Section III-C2).
         B) Add the new MF \( A_j^{\text{new}} \), and update the nearest MFS \( A_j^{\text{left}} \) and \( A_j^{\text{right}} \) (see Section III-C3).
   d) Apply the learned FLC to the process under control.
   e) Go to step (3a) (until the controller is turned off).

Algorithm 1 proposed approach, SDeFLC.

---

A single module does not fulfill all the interface requirements. The modular architecture allows reuse, reconfiguration, and simultaneous testing of multiple independent equipment of a complex system, since there is no guarantee that a system as a whole will work successfully even if each component has been successfully tested.

During the development of the CPS (as well as in the KhronoSim framework), the modules construction can occur in the following stages: simulation stage, where models are used to simulate the behavior of the required system components; prototyping stage, where the modules, previously constructed, are incrementally replaced by the real components of the CPS; and the pre-production stage, where the CPS is tested, in its final form, in the real environment.

The proposed simulation approach was purely virtual (simulation stage) to demonstrate the proof-of-concept of the KhronoSim platform. For this purpose, the main components (sensors, controller, actuators, and plant) were arranged in nodes of an Ethernet network. Figure 2, shows the components separated by their modules. The control system itself consists of modules #1, #3, and #4. Module #2 simulates the reduction of network performance caused by multi-node network traffic. Please notice that the data flow among modules may be delayed due to network traffic. The network environment is supplied by the TrueTime [17] toolkit - a Matlab/Simulink-based simulator for real-time control systems and wireless/wired networking.

B. Two-Tank Process Control

The Two-Tank process is composed of two tanks in cascade, as can be seen in Figure 3. The upper tank (Tank 1) is fed by two water flows, one hot (\( f_{hc} \)) at temperature \( t_{hc} \) and one cold (\( f_{cc} \)) at temperature \( t_{cc} \). Two actuators, \( f_{hc} \) and \( f_{cc} \), control the hot and cold flows respectively. An exit at the bottom of Tank 1 feeds the lower tank (Tank 2) with the mixed flow \( f_1 \) at temperature \( t_1 \). Tank 2 is also fed by a fixed cold water flow \( f_b \) at temperature \( t_b \). The variables of the plant and their designations are described as follows:

\[
\begin{align*}
    f_{hc}, f_{cc} : & \text{Commands to flow actuators (hot, and cold)}, \\
    f_b, f_c : & \text{Water flows (hot, and cold) into tank 1}, \\
    t_{hc}, t_{cc} : & \text{Water supply temperatures (hot, and cold)}, \\
    A_1, A_2 : & \text{Cross-sectional areas of tank 1 and tank 2}, \\
    t_1, t_2 : & \text{Temperatures of tank 1 and tank 2}, \\
    f_1 : & \text{Total flow out of tank 1}, \\
    f_b : & \text{Flow rate of the tank 2 bias stream}, \\
    t_b : & \text{Temperature of the tank 2 bias stream}.
\end{align*}
\]

The Two-Tank process control is described in detail in [18], [19]. Temperature and flow units were normalized to the interval \([0, 1]\), called \( t_{unit} \) and \( f_{unit} \), respectively. Thus, \( 0[t_{unit}] \) corresponds to cold water \( t_{cc} \) and \( 1[t_{unit}] \) corresponds to hot water \( t_{hc} \). Analogously, \( 0[f_{unit}] \) corresponds to no flow and \( 1[f_{unit}] \) corresponds to maximum flow.

The control target is to adjust the temperature in Tank 2 by acting on the Tank 1 input flows. The networked control system (Figure 2) was implemented in Simulink, having the network infrastructure simulated by the TrueTime toolkit.
C. Result Analysis

The tracking error \( x_1 = e(k) = r(k) - y(k) \), and \( x_2 = \Delta e(k) = e(k) - e(k-1) \) were defined as input variables of the fuzzy controller. The parameters of the proposed approach defined by the user were: \( x_1^\prime = -1 \) and \( x_1^\prime = 1 \); \( x_2^\prime = -0.5 \) and \( x_2^\prime = 0.5 \), \( \eta_j = |x_j^\prime - x_j^\prime|/30 \) (for \( j = 1, 2 \)), \( u^\prime = 0 \) and \( u^+ = 1 \), \( C = 1 \), \( \beta = 1 \times 10^{-7} \), the initial covariance values \( \Sigma_j(0) = \Sigma_{ij}(0) = 1 \times 3 \), and \( M_{1h} = 0.6 \), and \( M_{2h} = 0.2 \).

Figures 4, 5, and 6 present the results of the online evolved FLC applied to the Two-Tank process. Figure 4a shows the performance of the learned FLC designed by the proposed approach. The evolution of the number of MFs (corresponding also to the number of control rules) for each input variable is presented on Figure 4b, and Figure 4c presents the evolution of criterion (13), i.e. the value of \( M_{j,max} \), for each input variable. Figure 5 presents the evolution of the antecedent parameters, being the final learned MFs presented in Figure 6.

The reference signal (Figure 4a) was defined with different reference values in order to reach several regions of operation to test the controller’s behavior on unknown regions of operation. From the overall results, it can be seen that the performance of the FLC increases with the time of operation, being able to adequately control \( y(k) = t_2(k) \). Analyzing the effects of the proposed approach on the input variable \( x_1 = e(k) \), it can be seen that:

- At the initial instants (\( k < 25 \) [s]) two control rules were added on \( x_1 \) (see Figure 4b), corresponding to when the criterion (13) is met, i.e. \( M_{1,max} < M_{1h} = 0.6 \) since that regions of operation were unknown to the controller;
- For \( 25 \leq k < 225 \) [s] more control rules are added on \( x_1 \) corresponding also to when \( M_{1,max} < M_{1h} = 0.6 \), being the centers of the added membership functions presented in Figure 5a which cover the values (regions) that were unknown for the controller;
- For \( k \geq 225 \) [s], although the criterion (13) is met at some instants of time (\( M_{1,max} < M_{1h} = 0.6 \)), no more control rules were added. This happens because the other criterion (14) is not met, thus limiting an excessively fine

Figure 4: Results on the Two-Tank process.

Figure 5: Evolution of the antecedent parameters.

Figure 6: Final membership functions.
granulation of the domain region of $x_1$.

Analyzing the proposed approach on the results of $x_2 = \Delta e(k)$, it can be seen that:

- The additional process of adding control rules on $x_2$ is very influenced by the reference signal changes, since the addition of rules has occurred when the variation of the tracking error $e(k)$ is larger. This can be seen by the evolution of $M_{2,\text{max}}$ in Figure 4b:
- When $M_{2,\text{max}} < M_{2,\text{th}} = 0.2$ (excluding one situation because of criterion (14) not being met) control rules were added on $x_2$.
- It also can be seen that when the variations of the reference signal is $-0.05$ (at $k = 575$ [s] and $k = 650$ [s]) and $0.1$ ($k = 800$ [s]), no control rules were added. This is because, for these reference variations $M_{2,\text{max}} > M_{2,\text{th}} = 0.2$, thus the proposed approach considered that these values were from a known region of operation since they are close to the centers of the membership functions already defined, as can be seen by analyzing Figure 5b.

V. CONCLUSION

In this paper an approach to online evolve direct FLC is proposed. The FLC’s knowledge base is composed of a simple set of univariate fuzzy control rules, in order to improve the interpretability of the learned FLC and the understanding of the influence of each input variable on the controller’s behavior, since each fuzzy control rule only describes the control action of one input variable. The proposed approach is composed by two phases: 1) the offline phase, consisting on the definition of the initial control rules, which were defined using only the knowledge of the universe of discourse limits of each variable; and 2) the online phase, which is the evolving phase, where new control rules are added based on a novelty criterion, and where controller parameters are recursively adjusted.

The proposed approach was tested on a Two-Tank process under the KhronoSim proof-of-concept platform for testing cyberphysical systems in real-time and closed-loop using the TrueTime tool. The proposed approach has successfully evolved (online designed) the FLC, controlling the plant on unknown regions of operation.

ACKNOWLEDGMENT

This work was supported by Project KhronoSim “System for Simulation and Test of Complex Systems” (reference: KhronoSim/2016/17611), co-financed by the “Competitiveness and Internationalization Operational Programme” (COMPETE 2020), Portugal 2020 (PT2020), and by the European Union through the European Regional Development Fund (ERDF), and by Critical Software S.A., Jérôme Mendes has been supported by Fundação para a Ciência e a Tecnologia (FCT) under grant SFRH/BPD/99708/2014.

REFERENCES