IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 4, 2021, accepted April 7, 2021, date of publication April 14, 2021, date of current version April 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3073067

Control of a MIMO Coupled Plant Using a Neuro-Fuzzy Adaptive System Based on Boolean Relations

HELBERT ESPITIA[1], **IVÁN MACHÓN**[2], **AND HILARIO LÓPEZ**[2]

[1]Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas, Bogotá 11021-110231588, Colombia
[2]Departamento de Ingeniería Eléctrica, Electrónica de Computadores y Sistemas, Universidad de Oviedo, Campus de Viesques, 33203 Gijón, Spain

Corresponding author: Helbert Espitia (heespitiac@udistrital.edu.co)

This work was supported in part by the Universidad de Oviedo and in part by the Universidad Distrital Francisco José de Caldas.

ABSTRACT This document describes the implementation of a neuro-fuzzy adaptive system MIMO (Multiple Input Multiple Output), using two neuro-fuzzy MIMO systems: one for control and the other for identifying the plant. Under this approach, the controller is optimized, employing the model obtained during the identification of the plant that utilizes data generated from the controller's operation. In this way, the plant identification and the controller optimization is performed iteratively. The application case consists of controlling a MIMO non-linear hydraulic system fed by a pump and a three-way valve. In order to observe the controller performance various experimental configurations are considered.

INDEX TERMS Adaptive, control, hydraulic, MIMO, neuro-fuzzy.

I. INTRODUCTION

When having uncertainty, inaccuracy, and ambiguity in the phenomenon to model and control, the fuzzy logic systems seem a suitable option given the flexibility when describing this type of behavior [1], [2]. A systematic manipulation of vague and inaccurate concepts is visible when using fuzzy sets, at the same time, such sets can be employed to represent variables in linguistic terms [3]. Thus, the sets become a suitable option for solutions in control systems given the flexibility and capacity of fuzzy logic systems when having uncertainty, vagueness, and ambiguity in the phenomenon to model [1], [2].

Automation applications have developed tools based on Boole's algebra allowing the definition of control rules [4], [5]. Nevertheless, these systems display limited performance due to abrupt transitions in the actions of control. One way to improve the performance of the systems is replacing the Boolean sets for fuzzy sets [6]. A methodology for the design of fuzzy sets under this approach is displayed in [7], [8].

Neuro-fuzzy systems are an option of control when having a plant of high complexity given by non-linearities, parameter variations, among others [2]; thus, the implementation of neuro-fuzzy control strategies require a previous plant identification and the controller's optimization with this model.

Essentially, adaptability consists of the capacity of an organism to survive in an environment [9]. Reference [10] indicates that this principle can be applied to optimization and in adaptive intelligent control systems. Thus, intelligent control is the study to achieve the automation by emulating of intelligent biological systems. There are examples of optimization processes when considering biological or behavioral processes. This focus is useful in numerous practical problem controls where a mathematical model is difficult or non-viable; in such cases, heuristics can be employed for the design of control systems.

According to [11] and [12], the Adaptive Control (AC) includes a set of techniques that provide a systematic approach for the controller's adjustment in real-time, aiming at keeping the desired performance level of the control system when plant's parameters of the dynamic model are known and variable through time. Adaptive control techniques may provide an automatic adjustment procedure in closed-loop for controller's parameters; in such cases, the effect of the adaptation disappears as time passes. Besides, it must be considered that changes in the operating conditions may require a reboot of the adaptation procedure.

For control of uncertain non-linear systems, adaptive control is a suitable technique to handle system uncertainties using a parameter estimator. In this way, adaptive control provides controller's automatic adjustment for maintaining the desired system operation when the plant parameters are

The associate editor coordinating the review of this manuscript and approving it for publication was Dipankar Deb.

not precisely known. For the operation of adaptive control, the system output is measured and compared with the desired values; then, based on the comparison error, the adjustable controller adapts its parameters using adaptation mechanisms [13]. Besides, adaptive control has shown to be applicable in different Multiple Input Multiple Output (MIMO) systems, developing techniques such as Multivariable Model Reference Adaptive Control (MRAC), being this an important area in a theoretical and practical way [14]. Another important aspect is concerning to the sensor and actuator failures, which can cause intermittent fault, system performance deterioration, and even damages. To solve the uncertain actuator failure issues, different control techniques were proposed, where adaptive control based failure compensation designs are a suitable option [15].

According to [16], [17], most of the techniques employed to design control systems are based on the plant's exceptional knowledge and environment. Nevertheless, the plant may be remarkably complex, and the basic physical procedures are not entirely understood. Thus, the control techniques are improved with an identification technique to understand the plant better. If the identification of the system is recursive, that is, if the model of the plant is periodically updated based on previous estimations and new data, the control and identification can be made simultaneously. The adaptive control can be taken as a direct aggregation of a method with a system's form of identification that may determine if the plant is a linear or non-linear, finite, or infinite dimension and if it displays a discreet or continuous dynamic of events. Thus, adaptive control is all about applying a system's identification technique to reach a model of the process and its surroundings from input-output experiments, and then using this model for control design. Controller parameters are adjusted during plant's operation as the data amount for identification increases. In practice, these controllers are applied to unknown plants that slowly vary in time. The application of such systems arises in contexts like advanced systems for flight control for aircraft and spacecraft, robot manipulators, power-system controls, among others.

A fuzzy system allows establishing an initial configuration and structure for identifying the plant and the controller's optimization. At the same time, this addresses the difficulty present in neural networks to establish this structure, and its parameters initialization [2]. In that sense, when employing neural networks, the initial configuration is usually random, while a fuzzy system permits a previous configuration based on the preliminary knowledge of the system.

Plant identification for the design of control systems is an important aspect that has been approached in different ways. Some of these solutions have used fuzzy logic and neural networks. However, there are drawbacks, such as the black box nature of the neural network and the problem of determining suitable membership functions for fuzzy systems. These weaknesses can be avoided by implementing neuro-fuzzy systems since these include hybrid structure that combines both approaches [18]. The rules of a neuro-fuzzy system allow to incorporate previous knowledge of the system, therefore, this can be a useful instrument to deal with the identification of uncertain non-linear systems. Also, fuzzy logic and neural network controllers are generally considered applicable to plants that are mathematically less understood and where the experience of human operators are available for providing qualitative rules. Based on the universal approximation theorem, fuzzy logic controllers are general enough to perform any non-linear control actions. An adaptive fuzzy system is equipped with a training algorithm in which an adaptive controller is synthesized from a collection of fuzzy rules and the parameters of the membership functions change according to some adaptive law in order to control a plant to track a reference trajectory [19].

Regarding training techniques associated with optimization methods, evolutionary algorithms have demonstrated to be useful to approach an optimum global value; however, those require a high number of evaluations of the objective function and various executions. On the other hand, methods based on gradient calculations present rapid convergence but are also susceptible to the initial search point showing convergence toward local minima [20]. As reported by [10], techniques based on gradient calculations offer practical and effective methods to optimize online and, thus, adjust all control system parameters. The basic approach deals with the iterative adjustment of parameters that minimize the approximation error; however, local minima are usually present since the approximation error's objective function is not convex.

As stated in [21], gradient calculations are widely used in algorithms for adjusting neural systems; mainly, the descending gradient method is consistently employed in the Back Propagation (BP) algorithm for training neural networks. Thus, an alternative to improve the adaptive control performance employing gradient-based algorithms consists of a suitable preliminary configuration of the systems used for both plant identification and the controller.

A. APPLICATIONS OF ADAPTIVE NEURO-FUZZY CONTROL

Adaptive control is a robust and suitable alternative to face uncertainties related to parameter prediction in a dynamic system. Applicable structures for the controller and the plant model are necessary to employ these types of systems with training methods to adjust the parameters effectively.

According to [22], neuro-fuzzy systems are a suitable alternative for identification and control implementing adaptive systems where the Takagi-Sugeno (TS) model is mainly used for identification and adaptive control of non-linear systems. Generically, TS is employed to approximately parameterize the characterizable uncertainties existing in the plants. In adaptive control, plant identification can be carried out offline and online. The design of controllers can be carried out considering designs stable parameter-adaptation algorithms for both linearly and non-linearly parameterized TS fuzzy systems. Also, these systems can be used to address adaptive fault compensation problems subject to actuator faults and fault-tolerant control [22].

Regarding the structure, neuro-fuzzy systems allow modeling non-linear processes and obtain information about a set of data using learning algorithms. Also, neural-fuzzy logic systems ease the direct use of experts' knowledge as a start point for optimization [23]. Despite all the advantages of neural networks and fuzzy inference systems, neuro-fuzzy control structures present limitations when increasing the number of fuzzy rules associated with the system's order or the number of fuzzy sets employed in each input [24].

On some developments of neuro-fuzzy adaptive control, in [25] is presented an adaptive backstepping control approach for Single Input Single Output (SISO) non-linear networked control systems with network-induced delay and data loss. The paper [26] proposed a fractional-order integral fuzzy sliding mode control scheme for uncertain fractional-order non-linear systems subjected to external disturbances and uncertainties. Meanwhile, in [27] is presented the design of an adaptive-fuzzy control compensation strategy for direct adaptive control. Finally, in [28] is presented an Online Neuro-Fuzzy Controller (ONFC) that uses a simple structure displaying low computational cost and can control non-linear, time-varying, and uncertain systems. A brief review of different adaptive control applications using neuro-fuzzy systems is presented below, where Fig. 1 shows the main aspects identified.

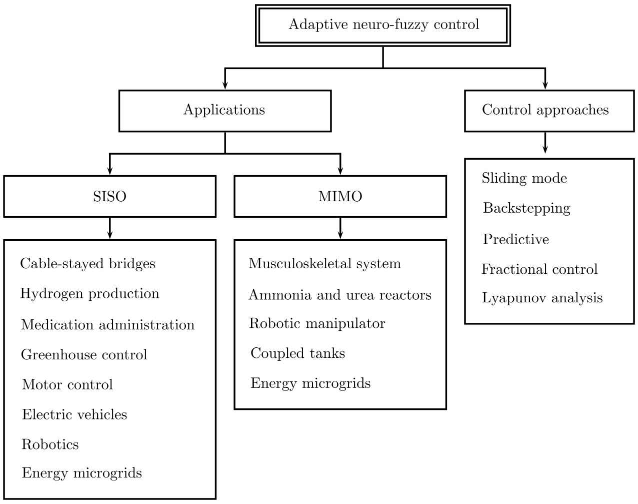

FIGURE 1. Main aspects identified of adaptive neuro-fuzzy control applications.

1) SPECIFIC APPLICATIONS

On applications related to the development of adaptive neuro-fuzzy control systems, reference [29] proposes a scheme to attenuate the seismic responses for cable-stayed bridges considering different parametric scenarios, conditions of the place, and seismic characteristics. The study compares the performance of adaptive methods with passive schemes before and after considering parametric variations. The simple adaptive control scheme offers a general and successful reduction for the bridge parameters variation.

Another industrial application is observed in [30] that studied a pilot-scale reformer to produce hydrogen (H2) by reforming methane (CH4) with CO2. An Adaptive Neuro Fuzzy Inference System (ANFIS) approach employs fuzzy type 2 systems, which allowed parts of linearization of the non-linear model. The membership functions (generalized bell) are optimized using empirical training data. The model is employed in a Model-Based Predictive Control (MPC) for designing optimal control strategies with no restrictions using a quasi-linear model adopting suitable weights in the control inputs.

A medical application is displayed in [31], developing control strategies in a closed-loop for infusion and medication administration being especially useful in anesthetic during numerous surgeries to provide stability for the necessary state of consciousness. The authors propose a neuro-fuzzy adaptive controller to overcome the current challenges in control closed loops of the anesthetic, like inter and intrapatient variability, complex and non-linear dynamics, measurement noises and surgical alterations, and sub-impulse and overflow in the induction stage.

2) MOTOR CONTROL, ELECTRIC VEHICLES, AND ROBOTICS

Control of electric motors is another application of neuro-fuzzy control; in this sense, reference [32] proposes an operation scheme for an induction motor controlled by a neuro-fuzzy system; such scheme uses the amplitude of the stator flux and the electromagnetic torque errors through an adaptive neuro-fuzzy inference system to act on both the amplitude and the angle of the desired reference voltage.

In regards to electric vehicles is considered the work [33], where the objective is to determine the minimization of the total energy consumption (electric battery and fuel) in hybrid vehicles (hydraulic and electric) with a combination of energy management that includes elements of fuzzy logic, neural networks, and rule-based algorithms. In this work, vehicle's global efficiency is calculated considering the electric motor, hydraulic pump, hydraulic motor, and transmission. An adaptive neuro-fuzzy algorithm is integrated into the vehicle to real-time control.

Another related work is [34] that presents an Adaptive Neuro-Fuzzy Sliding Mode Control (ANFSMC) architecture for diving motion control of an Autonomous Underwater Vehicle (AUV) in the presence of parameter perturbations and wave disturbances. The problem of non-linear uncertain diving behavior is addressed using a neural network designed to approximate a part that presents non-linear unknown dynamics and external disturbances. A corrective control based on an adaptive fuzzy proportional-integral control is applied to eliminate the chattering phenomenon.

A work on robotics is seen in [35], proposing hybrid adaptive algorithms of neuro-fuzzy control of a manipulator with uncertainties. The controller's outputs are used to generate the final actuation signal (action) based on the current position and speed errors. Thus, the industrial robot control includes non-linearities, uncertainties, and external disturbances in control laws design.

In [36], the control of a spherical rolling robot is presented via adaptive neuro-fuzzy control using a learning algorithm based on the theory of Sliding Mode Control (SMC). The proposed control architecture is composed of a neuro-fuzzy network and a conventional controller that is employed to guarantee the asymptotic system stability.

3) GREENHOUSE CONTROL

The regulation of climate in greenhouses is commonplace since this is a complex non-linear system that provides suitable environmental conditions for plant growth. The climatic control issue is creating a favorable crop environment to achieve high performance, quality, and low-cost predetermined results. In this regard, document [37] displays a system control for a greenhouse that utilizes geothermal energy as energy source for a heating system. It contemplates four techniques: Proportional Integral (PI) control, fuzzy control, control with an artificial neural network, and adaptive neuro-fuzzy control employed to adjust the indoor temperature to the value required. Equally, reference [38] presents the application of an ANFIS system to face numerous uncertainties in such systems; also, to improve the performance, a Genetic Algorithm (GA) is employed for adapting control parameters like the number and the form of the membership functions and scale factors.

4) SLIDING MODE CONTROL

The development of adaptive neuro-fuzzy sliding mode controls has been relevant. In this regard, reference [39] researched the issue of adaptive neuro-fuzzy sliding mode control of fractional order (Fractional Order - FO) for a type of fuzzy system perturbed subjected to uncertainties and external disturbances. This work presents a fuzzy sliding mode surface FO employing a Lyapunov function for analyzing the robust stability of the sliding mode; besides, it proposes a Hybrid Neuro-Fuzzy Network System (HNFNS) for estimating uncertainty. The FO adaptive fuzzy sliding mode control is designed to propel the state trajectories of singular disturbed fuzzy systems to the predetermined FO sliding mode's surface in a finite time system.

A sliding mode control called Adaptive Fuzzy Sliding Mode Control (AFSMC) is seen in [40] for issues like chaos synchronization between two different uncertain fractional-order chaotic systems. For the synthesis approach using the definitions of fractional derivatives and integrals, a fuzzy Lyapunov design process is presented to tune online the parameters of the AFSMC considering the output feedback.

Another application is displayed in [41] using an adaptive control method to regulate induction motors' speed. The scheme employed is online learning based on a training algorithm in sliding mode and fuzzy systems type 2. By control parameters adjustment, it is possible to address the issue associated with parametric uncertainties and disturbances. The adaptive mechanism of sliding mode adjusts the parameters of the type 2 membership functions according to inputs of speed error and its derivative. Since the induction motor parameters may vary and the information to build the membership functions and systems' fuzzy logic rules are uncertain, it is then selected the neuro-fuzzy type 2 structure as a controller.

5) NON-LINEAR SYSTEM CONTROL

In general, neuro-fuzzy control systems are used in non-linear plants. In this regard, document [42] suggests a neuro-fuzzy adaptive control for dynamic systems' follow-up with non-linearities unknown. A Takagi-Sugeno is used to describe local sub-models obtained using non-linear input and output dynamic mapping. This scheme is based on approximating specific terms involving the Lyapunov function's derivative with non-linearities of the unknown system. Besides, it is proposed a restart scheme to ensure the control input validation. The stability analysis provides the control law and adaptation rules for the network weights.

Another work is observed in [43] that presents an adaptive predictive control method for non-linear systems through an adaptive neuro-fuzzy inference system. The model proposed employs a non-linear Generalized Predictive Controller (GPC), while the plant's model is achieved using an adaptive system. The dynamic is classified as linear and non-linear. The linear part is approximated using the least-squares estimation technique, while the non-linear employs an ANFIS-based identifier. The controller is updated using the prediction obtained in an adaptive form. The method can be used in real-time with no stage of network's pre training required. A liquid level control system is considered an application case with a Continuous Stirred Tank Reactor (CSTR).

Meanwhile, reference [24] proposes a neuro-fuzzy adaptive controller based on quaternionics to reduce the issue associated with the number of extensive rules using quaternionic Back Propagation. Moreover, they are employed for reinforcement, which is achieved by evaluating the response with a system evaluation. As an application case, it is used to control a chaotic spinning disk.

In addition, paper [44] develops a dynamic Fractional Order Backstepping Dynamic Surface Control (DSC) for facing the problem of stabilization of non-linear systems of fractional order with uncertainties and external disturbances. Each adaptive step employs a neuro-fuzzy network system to approximate the unknown non-linear function existing in the fractional-order sub-system. A modified filter of fractional-order is also designed to avoid the complexity explosion issue caused by the recursive procedure. Based on Lyapunov's theory of fractional order, it is proposed a DSC controller of adaptive backstepping to ensure closed-loop systems stability.

Finally, in [45], a hybrid non-linear controller is displayed for the follow-up of speed and height based on the increase of dynamics and kinematics of spacecraft. It is derived from pseudo-linear formulation to develop the controller, which follows a Modified State-Dependent Riccati Equation (MSDRE) scheme. Here, a neuro-fuzzy controller is designed using ANFIS that also employs out of line of MSDRE solutions. The combined control scheme is applied according to long time intervals of MSDRE solutions to acquire optimal control torques. In contrast, along with each time interval, the controller ANFIS provides the input signal needed. Global asymptotic stability is investigated utilizing Lyapunov's theorem and is verified by Monte Carlo simulations.

6) ENERGY MICROGRIDS

Considering the uncertain nature of Renewable Energy Resources (RERs) and its integration into Micro Grids (MGs), adaptive control is a suitable option for power management. In this regard, in [46], using a neuro-fuzzy controller is proposed a Home Energy Management System (HEMS) to carry out day ahead management and real-time regulation; according to the inputs and outputs, this approach is considered a MIMO application. Another work can be seen in [47], where it is developed an adaptive neuro-fuzzy system based on Power Oscillation Damping (POD) controller to damp Low-Frequency Oscillations (LFOs) in hybrid AC/DC microgrids. Finally, in [48] an adaptive neuro-fuzzy control power is proposed to regulate the voltage in a distribution network when there is a variation in the load. A SISO case is considered for the connection of one generator while a MIMO case is regarded for three generators.

7) MIMO CONTROL SYSTEMS

About MIMO systems, document [49] displays a research made of multiple inputs and multiple outputs musculoskeletal model of the human arm with six muscles. It proposes an optimal adaptive neuro-fuzzy controller to control the endpoint of the arm. The adaptability and optimization of the muscular force are essential features of the neuro-fuzzy controller proposed.

Another MIMO application is found in [50] presenting three control strategies to regulate ammonia and urea reactors. The first is the Adaptive Model Predictive Controller (AMPC); the second is an Adaptive Neural Network Model Predictive Control (ANNMPC). The third is the Adaptive Neuro-Fuzzy Sliding Mode Controller (ANFSMC). The primary purpose of the controllers is to stabilize the output concentration of ammonia and urea, obtaining a stable speed of carbon monoxide (CO) conversion into carbon dioxide (CO2) to reduce the contamination effect, and a rise in ammonia and urea, keeping the relation NH3/CO2 equal to 3 to reduce both CO2 and NH3 unreacted. Lastly, the controller is also used to keep a suitable temperature in both reactors in the correct ranges of operation when an external disturbance occurs or the reactor parameters change.

Meanwhile, reference [51] presents a Terminal Sliding Mode Control (TSMC) using fuzzy logic for a rigid robotic manipulator of two links. The TSMC is developed for faster convergence and higher accuracy than the linear sliding control based on the hyperplane. The proposed controller combines a continuous TSMC with an adaptive learning algorithm and one fuzzy logic system to measure the plant's dynamic. In this way, the purpose is to ensure the stability in closed-loop and the convergence in finite time of tracking errors.

Finally, in [52] is presented an adaptive state-space neuro-fuzzy control scheme that combined a quadratic state feedback controller and eight-layer neuro-fuzzy model used to approximate the dynamics of non-linear plants. Both the neuro-fuzzy model and the controller are updated online. The control scheme is tested in a MIMO plant composed of three coupled tanks.

B. PROPOSAL APPROACH AND DOCUMENT DISTRIBUTION

This document describes the proposal for a MIMO adaptive neuro-fuzzy control system based on Boolean relations used for filling control of a MIMO non-linear hydraulic system. The adaptation process is carried out using the data taken during the control system's operation with which the plant is identified; later, the model is employed to optimize the controller. The plant corresponds to the MIMO hydraulic system presented in [53], consisting of two tanks fed by a pump and a three paths valve.

A compact fuzzy inference system based on Boolean relations is employed for the structures of the neuro-fuzzy systems facilitating the inference process calculations [6]–[8]. Moreover, the compact systems allow analogies with linear and non-linear systems in such a way that it is possible to establish their initial structure and configuration.

The main contributions of this work are described as follows.

- For the non-linear MIMO plant, the structure and initial configuration of the neuro-fuzzy systems used for identification and control are determined. For this, compact fuzzy systems based on Boolean relations are used, which allow the analogy with linear systems as a starting point for the design of neuro-fuzzy systems.
- Different MIMO structures are presented that use neuro-fuzzy subsystems based on Boolean relations. Considering the functional parts of the MIMO plant, the configuration used for identification is chosen. In this way, the structure of the neuro-fuzzy MIMO system is selected based on preliminary knowledge of the plant.
- Considering the structure of a zero, pole, and gain controller, the structure of SISO neuro-fuzzy subsystems based on Boolean relations is determined, which are used to implement the MIMO controller architecture.
- Since the employed neuro-fuzzy MIMO systems have feedbacks, it is shown the deduction of the recurrent equations used for the parameters adaptation. Considering these equations, the steps of the algorithms used

for plant identification and controller optimization are described.

Regarding the challenges addressed for the control of the non-linear MIMO plant, it is worth mentioning that in this application there is no prior offline training of the neuro-fuzzy systems, this is why it was of importance to determine an adequate structure of the MIMO neuro-fuzzy systems for the plant and controller. In this way, using the data taken during the plant operation, the adjustment of the plant model and the controller is carried out in a way that the adaptive control system achieves the references tracking as observed in the results.

The document is distributed as follows: Section II shows the structure of the hydraulic MIMO plant to know the parts of this system; Section III describes the general architecture of the adaptive neuro-fuzzy control system employed, presenting the adaptive strategy used to control the plant (some key aspects are described); then, a detailed design of a compact neuro-fuzzy system based on Boolean relations is shown in Section IV, where first the configuration of the compact system used is described, and how it can be configured to achieve the analogy with dynamic systems in discrete time obtaining neuro-fuzzy subsystems. Using these subsystems as a basis, different neuro-fuzzy MIMO configurations are presented in Section V. With the MIMO structures defined for the controller and the plant identification, the equations for controller training are deduced in Section VI, and for plant identification in Section VII, where, the algorithms used for parameters adaptation of the MIMO neuro-fuzzy systems are also described. Later, the application for the MIMO non-linear hydraulic system considering different controller configurations is presented in Section VIII, where the operation of the adaptive process can be appreciated considering different references, in a way that after several identification and optimization processes of the controller, the system outputs reach the desired values. Finally, the conclusions are displayed in Section IX.

II. MIMO HYDRAULIC SYSTEM

This section focuses on showing the characteristics of the plant to be controlled, where the model of the system is presented in a block diagram used as a reference to build the MIMO neuro-fuzzy system employed for plant identification.

The MIMO system consists of two tanks fed by a three-way valve connected to a hydraulic pump for distributing the flow in both tanks [53]. The design is aimed at the simulation of liquids' transport and storage and handled by a Programmable Logic Controller (PLC) connected to an OPC server (OLE for processes control). The plant is located at the Department of Systems Engineering of the Universidad de Oviedo. The MIMO system includes two lower tanks ($D1$ and $D2$) and two upper tanks ($D3$ and $D4$) that operate using a cascade drainage system. The lower tanks release the flow into a collector tank that works as a source for pumping the flow to each tank employing pumps ($P1$ and $P2$), and two three-way valves ($V1$ and $V2$) [53].

This work employs the operation method called 2×2 where pump $P1$ proportionally supplies flow to tanks $D1$ and $D4$ through the three-way valve $V1$. In this configuration, if the set point of $V1$ is equal to 0% all the liquid is sent to tank $D4$, but if the set point is equal to 100%, it implies that all the liquid is sent to tank $D4$. Both pumps present maximum and minimum flows. The tank capacity goes between lower and upper levels, continuously oscillating given the uninterrupted liquid supply and the three liquid outlets in each tank: lower outlet, upper outlet (defined as a relief), and controllable outlet [53].

The three-way valve's opening percentage $u_2 = s(t)$ and the power of the pump $u_1 = p(t)$ are considered as the inputs of the system, while the liquid levels in both tanks $y_1(t) = D1$ and $y_2(t) = D4$ are outputs; all the variables oscillate between 0% and 100%. As observed, the plant constitutes a MIMO system. Fig. 2 shows the parts of the system's physical disposition, while Fig. 3 displays the schematic representation.

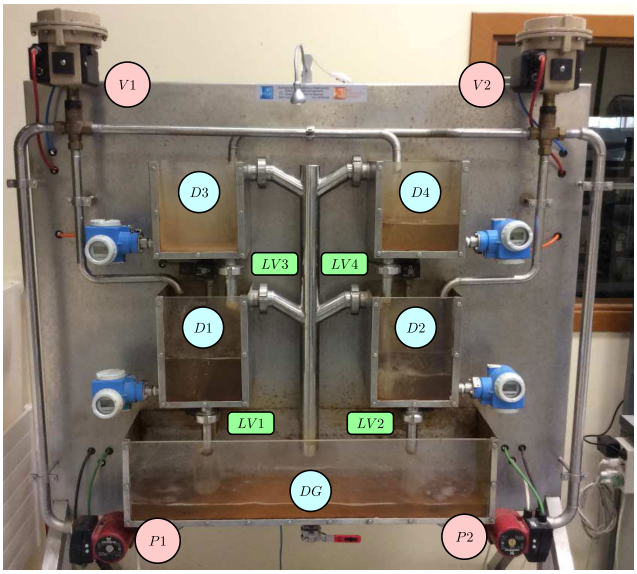

FIGURE 2. MIMO plant [53].

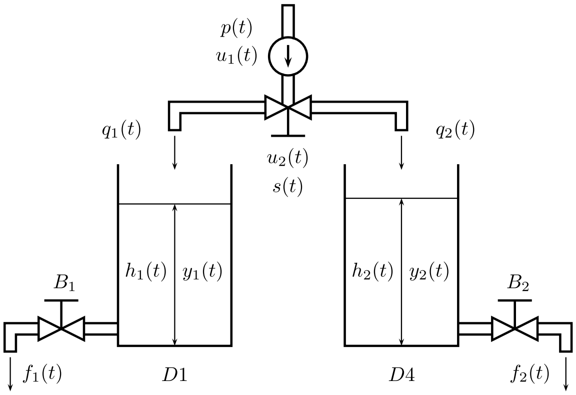

FIGURE 3. Representation of the MIMO hydraulic system.

The differential equations associated with the plant model are obtained performing mass balances [54], [55]. In a simplified way, the equations that represent the hydraulic MIMO system are:

$$\frac{dh_1}{dt} = -B_1\sqrt{h_1} + C_1 q_1(t) \tag{1}$$

$$\frac{dh_2}{dt} = -B_2\sqrt{h_2} + C_2 q_2(t) \tag{2}$$

$$q_1(t) = u_1(100 - u_2) = p(t)(100 - s(t)) \tag{3}$$

$$q_2(t) = u_1 u_2 = p(t)s(t) \tag{4}$$

Considering the reported in [54] and [55], the plant parameters are: $B_1 = a_1\sqrt{2g}/A_1$, $B_2 = a_2\sqrt{2g}/A_2$, $C_1 = k_1/A_1$, $C_2 = k_2/A_2$, where, A_i and a_i are cross-section areas of the i-th tank and its outlet pipeline, respectively; g is the gravitational acceleration. Also h_1 and h_2 are water levels of tanks D_1 and D_4, finally, q_1 and q_2 represent the input ratios of mass flows; where k_1 and k_2 are the respective gains associated to the input flows.

Fig. 4 shows the block diagram of the MIMO hydraulic system composed of two tanks.

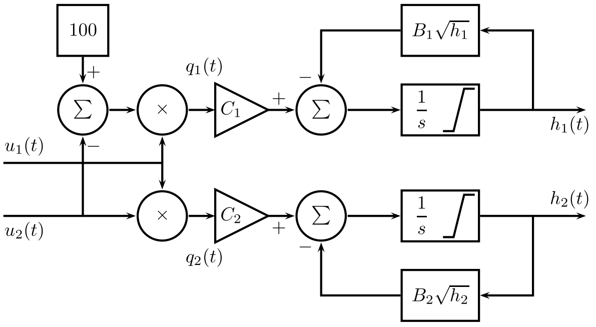

FIGURE 4. MIMO hydraulic system schematic.

Fig. 5 and 6 show examples of the behavior of the plant in open loop using different values of the pump and the position of the three-way valve. Fig. 6 shows that the settling time of the plant is 100 seconds.

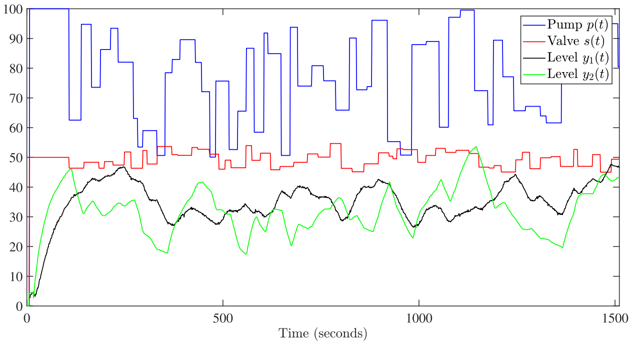

FIGURE 5. Plant response in open loop.

Using the data of Fig. 5 and the MATLAB® optimization function "*fminunc*" [56], taking $X = [C_1, C_2, B_1, B_2]$ as the variables of the fitness function (5) where N_T is the total amount of data, $y_{r,1}$, $y_{r,2}$ real data, and $y_{s,1}$, $y_{s,2}$ as simulated data, then it is established that $C_1 = 2.95 \times 10^{-4}$, $B_1 = 0.2$, $C_2 = 2.95 \times 10^{-4}$ and $B_2 = 0.2$. The simulation results can be observed in Fig. 7.

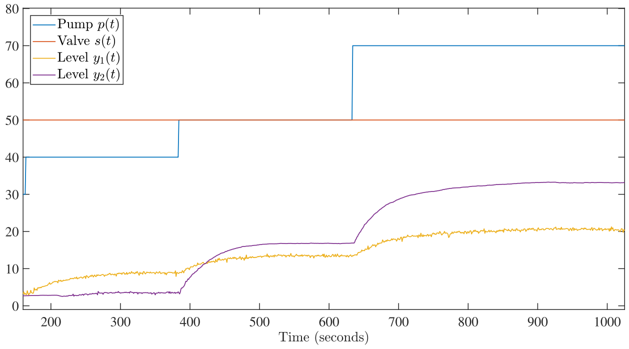

FIGURE 6. Behavior of the plant in open loop.

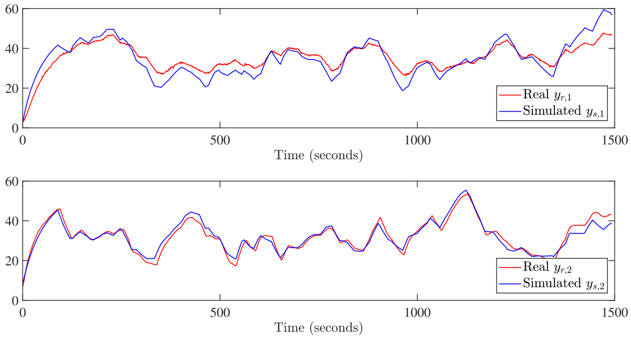

FIGURE 7. Plant simulation in open loop.

$$J_s(X) = \frac{1}{N_T}\sum_{n=1}^{N_T}\left[(y_{r,1} - y_{s,1})^2 + (y_{r,2} - y_{s,2})^2\right] \tag{5}$$

III. ADAPTIVE NEURO-FUZZY CONTROL SYSTEM ARCHITECTURE

This section describes the general architecture of the neuro-fuzzy adaptive control system utilized, presenting the adaptive strategy employed to control the non-linear MIMO plant.

Adaptive control consists of techniques that provides a systematic approach for automatic adjustment of control settings in real-time to achieve or maintain the required performance when the system parameters change. In this way, the adaptive control technique through the adaptation law can cope with disturbances, uncertainties in the system dynamics, as well as variations in operating conditions [57].

According to [57], regarding the techniques for adaptive control: direct methods for adaptive control combine the control objective and the parameter estimation in one step, whereas the indirect methods used here separate the parameter estimation and control into two steps. Direct methods generally rely on developing techniques for the parameter estimation, such that the estimation error and the trajectory error are driven to zero simultaneously. In this way, the parameter estimation and control method are separated, and the control is based on the Certainty Equivalence Principle (CEP).

Besides, for the strategy used to control the MIMO plant, Model Reference Adaptive Control (MRAC) is one of the most important adaptive control design methods that provides feedback controller structures and adaptive laws for plant controlled to guarantee output tracking for a given reference model system and closed-loop signal boundedness, in the presence of system uncertainties [58].

The architecture uses two neuro-fuzzy systems, one as a controller and another for the plant model. The first aspect is the identification of the plant, then the training of the controller is performed. Fig. 8 shows the systems employed.

The identification of the plant can be done in two ways: one offline by collecting the data in an open loop, including various input signals to characterize the plant's behavior. The second option is online and is represented by Fig. 8, where the training data are collected, during the control system's operation in closed loop. The block of the reference model corresponds to the expected system's behavior.

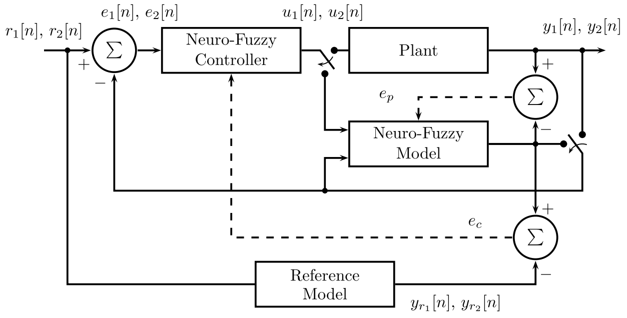

FIGURE 8. Neuro-fuzzy adaptive control system.

In Fig. 8, after identifying the plant, the control loop is incorporated to train the controller using the Real-Time Recurrent Learning (RTRL) algorithm [59]. In an adaptive scheme, the identification and training processes are executed iteratively, aiming at the control system's adaptability.

A. DESCRIPTION OF THE ADAPTIVE CONTROL

Identifying the plant and later the training of the neuro-fuzzy controller are performed to carry out the adaptive control system. This approach integrates the loop control to train the controller. The adaptive neuro-fuzzy system control process is seen in Fig. 9.

The first step defines initial configurations for the controller and the neuro-fuzzy model (plant); these can be obtained from previous knowledge or by offline training. In the second step plant's input-output data are acquired during the operation in a close loop; then, new plant identification is made with the data collected. Next, the training of the controller is made employing plant's updated model. The next step is to operate the optimized controller to correct the variation in the system. This process is repeated from step 2 for the next time interval until it meets a stopping criterion.

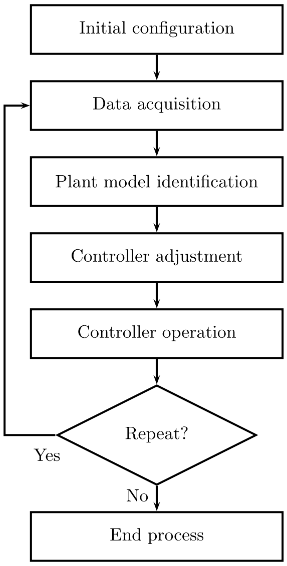

FIGURE 9. Adaptive control process algorithm.

Considering the limited amount of data produced during controller operation, the plant identification and controller training process occurs iteratively. The relevance of establishing an initial search point to identify the plant and optimize the controller is noticeable, which is achieved with the neuro-fuzzy systems designated in the first process.

IV. FUZZY INFERENCE SYSTEMS BASED ON BOOLEAN RELATIONS

This section shows the design of a SISO compact neuro-fuzzy system based on Boolean relations, first providing a general description of the fuzzy inference process, followed by the design of the compact fuzzy system. Finally, it is shown how this compact system can be configured to achieve the analogy with discrete time dynamic systems.

In automation applications, tools based on Boolean algebra can be used since it is easy to represent a set of rules in a true table [4], [5]. However, this type of system presents limited performance due to abrupt transitions of control actions. Therefore, a way to improve the performance of these systems is to replace the Boolean sets with fuzzy ones [6], [7]. In [7], a methodology for the design of fuzzy systems based on Boolean relations is proposed using Kleene algebra to achieve a transition between a system that uses Boolean sets into another with fuzzy sets. The essential aspects are described below.

A. DESIGN OF FUZZY INFERENCE SYSTEMS BASED ON BOOLEAN RELATIONS

The structure of the inference process in a fuzzy inference system based on Boolean relations is given by the truth table associated with the Boolean encoding of the rules. In order to implement the FIS-BBR (Fuzzy Inference System Based on Boolean Relations) the two-element Boolean encoding is extended to a three-element Kleene encoding. Table 1 presents the coding of the sets for each input A_j and the activation outputs $Y_m \in [0, 1]$. In this table, each row represents an inference rule; on the right side (output section), each column represents an activation function. By multiplying (weighting) the virtual actuators $v_m \in \mathbb{R}$ using the activation functions Y_m are obtained the virtual outputs $y_m = v_m Y_m$; the inference system output is obtained by adding all these products [7], [8].

TABLE 1. Truth table with activation rules and functions.

Input sets						Activation functions					
A_1	A_2	$\cdots$	A_j	$\cdots$	A_P	Y_1	Y_2	$\cdots$	Y_m	$\cdots$	Y_M
$a_{1,1}$	$a_{1,2}$	$\cdots$	$a_{1,j}$	$\cdots$	$a_{1,P}$	$f_{1,1}$	$f_{1,2}$	$\cdots$	$f_{1,m}$	$\cdots$	$f_{1,M}$
$a_{2,1}$	$a_{2,2}$	$\cdots$	$a_{2,j}$	$\cdots$	$a_{2,P}$	$f_{2,1}$	$f_{2,2}$	$\cdots$	$f_{2,m}$	$\cdots$	$f_{2,M}$
$a_{3,1}$	$a_{3,2}$	$\cdots$	$a_{3,j}$	$\cdots$	$a_{3,P}$	$f_{3,1}$	$f_{3,2}$	$\cdots$	$f_{3,m}$	$\cdots$	$f_{3,M}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
$a_{k,1}$	$a_{k,2}$	$\cdots$	$a_{k,j}$	$\cdots$	$a_{k,P}$	$f_{k,1}$	$f_{k,2}$	$\cdots$	$f_{k,m}$	$\cdots$	$f_{k,M}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
$a_{Q,1}$	$a_{Q,2}$	$\cdots$	$a_{Q,j}$	$\cdots$	$a_{Q,P}$	$f_{Q,1}$	$f_{Q,2}$	$\cdots$	$f_{Q,m}$	$\cdots$	$f_{Q,M}$

In Table 1, the variables $a_{k,j}$ and $f_{k,m}$ allow representing relationships between A_j and Y_m. For the Boolean case, this variable has values $\{0, 1\}$, while Kleene case has values $\{0, u, 1\}$. In addition, A_j can be a Boolean or fuzzy set associated with the input, and Y_m is an activation function associated with the output.

For the system implementation the m-th activation output Y_m, in Disjunctive Normal Form (DNF) can be expressed as:

$$Y_m = \bigvee_{k=1}^{Q} \bigwedge_{j=1}^{P} \widehat{A}(a_{k,j}, f_{k,m}) \tag{6}$$

Considering the activation outputs of Table 1, the total system output can be calculated as:

$$y = \sum_{m=1}^{M} Y_m v_m \tag{7}$$

where v_m corresponds to the m-th virtual actuator, obtaining the m-th virtual output:

$$y_m = Y_m v_m \tag{8}$$

Function $\widehat{A}$ depends on $a_{k,j}$ and $f_{k,m}$ [60], Equation (9) shows the values of $\widehat{A}(a_{k,j}, f_{k,m})$.

$$\begin{cases} \widehat{A}(0,0) = 0 \\ \widehat{A}(u,0) = 0 \\ \widehat{A}(1,0) = 0 \\ \widehat{A}(0,u) = \bar{A}_j \\ \widehat{A}(u,u) = A_j \wedge \bar{A}_j \\ \widehat{A}(1,u) = A_j \\ \widehat{A}(0,1) = \bar{A}_j \\ \widehat{A}(u,1) = 1 \\ \widehat{A}(1,1) = A_j \end{cases} \tag{9}$$

B. COMPACT FUZZY SYSTEM BASED ON BOOLEAN RELATIONS

For the compact fuzzy system design used for identification and control, it is considered that a set of the input allows the action of an output activation function, an example of the respective Boolean encoding of this case can be seen in Table 2.

TABLE 2. Boolean encoding for the compact system.

A_1	A_2	$\cdots$	A_m	$\cdots$	A_M	Y_1	Y_2	$\cdots$	Y_m	$\cdots$	Y_M
1	0	$\cdots$	0	$\cdots$	0	1	0	$\cdots$	0	$\cdots$	0
0	1	$\cdots$	0	$\cdots$	0	0	1	$\cdots$	0	$\cdots$	0
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
0	0	$\cdots$	1	$\cdots$	0	0	0	$\cdots$	1	$\cdots$	0
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
0	0	$\cdots$	0	$\cdots$	1	0	0	$\cdots$	0	$\cdots$	1

In this way, Table 3 shows the extent of the table from Boole to Kleene where X can be $\{1, u, 0\}$.

TABLE 3. Kleene extension of Boolean encoding.

A_1	A_2	$\cdots$	A_m	$\cdots$	A_M	Y_m
X	X	$\cdots$	1	$\cdots$	X	1
X	X	$\cdots$	u	$\cdots$	X	u
X	X	$\cdots$	0	$\cdots$	X	0

According to [7], to have a regular table and maintain a monotonous transition, Y_m is 1 for any case that $A_m = 1$, also is u where $A_m = u$ and 0 for $A_m = 0$. As observed, the cases where there is a transition to Y_m between 0 and 1 is assigned u.

Using the rules to establish the disjunctive normal form, it can be seen that A_m is common for the terms when Y_m is 1 or u, therefore, Equation (10) describes Y_m [8].

$$Y_m = (A_m \wedge F_B) \vee (A_m \wedge F_E) \vee A_m = A_m \tag{10}$$

where:

- F_B: It is the disjunction of all the conjunctions obtained from the truth table for which Y_m is 1 eliminating from these A_m.
- F_E: It is the disjunction of all the conjunctions obtained from the truth table for which Y_m is u eliminating from these A_m.

As observed for all conjunctions, when Y_m is 1 always appear A_m and when Y_m is u always appear the conjunction $A_m \wedge \bar{A}_m$. In this way, when the variable A_m is presented in all terms, an absorption process occurs [8].

Considering an input variable x_1 an activation output can be encoded using Table 4, in this way, $Y_{1,1} = A_{1,1}(x_1)$. In general, by incrementing the columns, the activation outputs can be written as:

$$Y_{1,l} = A_{1,l}(x_1) \tag{11}$$

Having for each set $A_{1,l}$ a membership function $\mu_{1,l}$, also, using virtual actuators $v_{1,l}$, it has an output f_1 as presented in Fig. 10.

TABLE 4. Coding for one input and one activation function.

$A_{1,1}(x_1)$	$A_{1,2}(x_1)$	$\cdots$	$A_{1,l}(x_1)$	$Y_{1,1}$
1	X	$\cdots$	X	1
0	X	$\cdots$	X	0

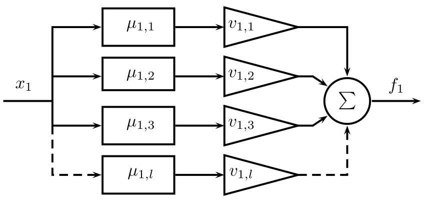

FIGURE 10. Configuration example for one output.

Extending Table 4 for more input variables $x_1, x_2, \ldots, x_j$ is obtained the coding shown in Table 5.

TABLE 5. Coding for the compact system for an input x_j.

$A_{j,1}(x_j)$	$A_{j,2}(x_j)$	$\cdots$	$A_{j,l}(x_j)$	$Y_{j,1}$	$Y_{j,2}$	$\cdots$	$Y_{j,l}$
1	X	$\cdots$	X	1	0	$\cdots$	0
X	1	$\cdots$	X	0	1	$\cdots$	0
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
X	X	$\cdots$	1	0	0	$\cdots$	1

In this way, each activation function $Y_{j,l}$ directly depends on a set $A_{j,l}$ obtaining:

$$Y_{j,l} = A_{j,l}(x_j) \tag{12}$$

Using the respective membership functions $\mu_{j,l}$ associated with sets $A_{j,l}$ it is obtained the general scheme shown in Fig. 11.

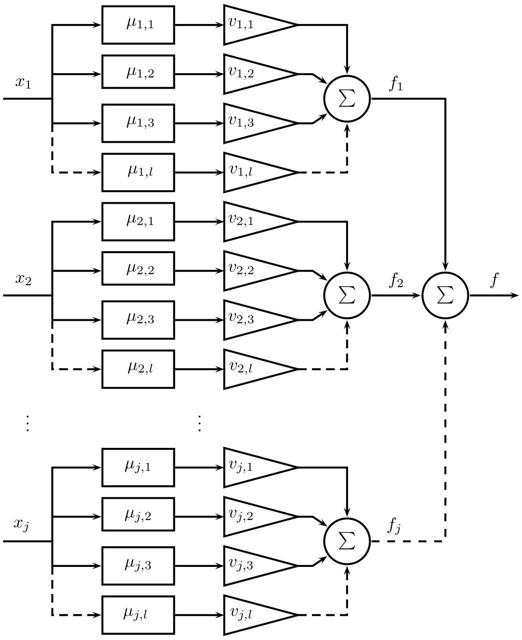

FIGURE 11. General diagram of the compact system.

The output of the inference process can be calculated as:

$$f = \sum_{j=1}^{M} \sum_{l=1}^{N} v_{j,l}\mu_{j,l}(x_j) \tag{13}$$

Fig. 12 presents the Boolean sets considered (base) and the respective membership functions used to implement compact systems. In this way, the sigmoidal fuzzy sets $\mu_{j,1}$ and $\mu_{j,2}$ are used to represent negative and positive values of the universe of discourse x_j.

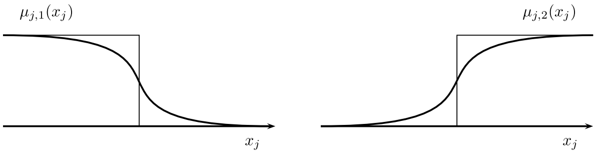

FIGURE 12. Boolean sets and used membership functions.

Employing the fuzzy sets of Fig. 12 is found the diagram of Fig. 13, which shows the proposed fuzzy system. This is considered as a basic block to build the respective MIMO models for the plant model and the controller.

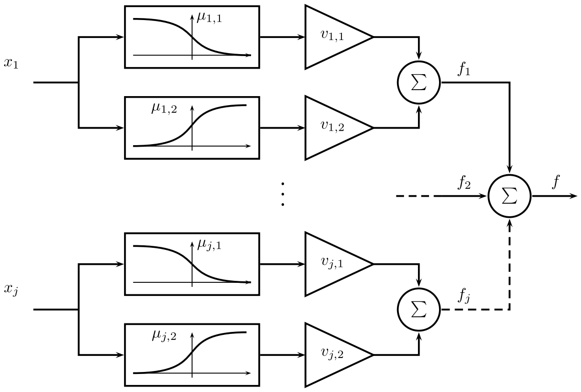

FIGURE 13. Neuro-fuzzy control system scheme.

The membership functions of Fig. 12 are given by the Equation (14).

$$\mu_{j,l}(x_j) = \left(1 + e^{-\sigma_{j,l}(x_j - \gamma_{j,l})}\right)^{-1} \tag{14}$$

The set of parameters in Fig. 13 are $h \in \{v_{j,l}, \sigma_{j,l}, \gamma_{j,l}\}$. Finally, the inference process for this type of system can be calculated as:

$$f = \sum_{j=1}^{M} f_j(x_j) = \sum_{j=1}^{M} \sum_{l=1}^{2} v_{j,l}\mu_{j,l}(x_j) \tag{15}$$

V. MIMO CONFIGURATIONS USING SYSTEMS BASED ON BOOLEAN RELATIONS

In order to implement the MIMO adaptive control system, for the plant model and controller different configurations of first-order BBR subsystems are considered. Seeing the described in [53], first-order BBR subsystems are used for the identification of the MIMO hydraulic system. Meanwhile, considering the structure of a zero, pole, and gain controller, the structure of a neuro-fuzzy subsystem based on Boolean relations is determined, which are used to implement the architecture of the MIMO controller.

For plant identification the subsystems are established considering the analogy with a first-order linear system (discrete time) with a transfer function as:

$$G(z) = \frac{X(z)}{W(z)} = \frac{b_0}{1 - a_0 z^{-1}} \tag{16}$$

The respective discrete time equation is:

$$x[n] = b_0 w[n] + a_0 x[n-1] \tag{17}$$

In the case of the BBR subsystem, the previous equation can be represented as:

$$x[n] = f_1(w[n]) + f_2(x[n-1]) \tag{18}$$

Thus, the scheme of a BBR subsystem used for the plant identification can be seen in Fig. 14.

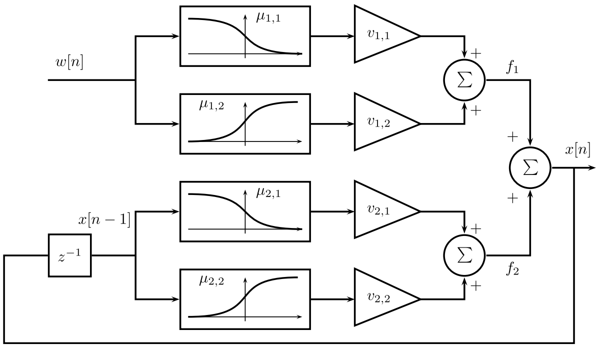

FIGURE 14. Equivalent first-order BBR-SISO system.

Meanwhile, for the controller is used the BBR configuration shown in Fig. 15 to have an analogy with a zero pole and gain compensator that has a transfer function:

$$C(z) = \frac{W(z)}{E(z)} = \frac{b_1 + b_0 z^{-1}}{1 - a_0 z^{-1}} \tag{19}$$

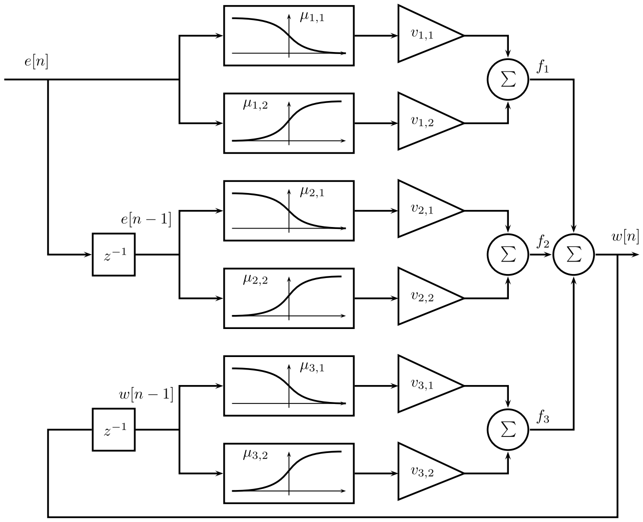

FIGURE 15. BBR-SISO system used for the controller.

This controller in discrete time can be described as:

$$w[n] = b_1 e[n] + b_0 e[n-1] + a_0 w[n-1] \tag{20}$$

For the BBR subsystem, it can be seen as:

$$w[n] = f_1(e[n]) + f_2(e[n-1]) + f_3(w[n-1]) \tag{21}$$

For the MIMO system, two possible configurations are mainly considered depending on the relationship that can be had between the outputs of the BBR subsystems. These configurations correspond to:

- Sum of outputs.
- Product of the outputs.

Fig. 16 shows the configuration that adds the outputs of the BBR subsystems. On the other hand, the diagram where is used the product of the outputs of the BBR subsystems is presented in Fig. 17.

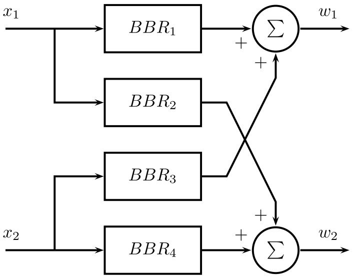

FIGURE 16. BBR-MIMO system using sum of outputs.

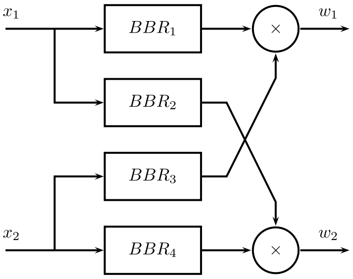

FIGURE 17. BBR-MIMO system using output product.

A. MIMO CONFIGURATIONS USED FOR THE ADAPTIVE CONTROL SYSTEM

The configuration shown in Fig. 18 (sum of the outputs) is employed for implementing the controller. Meanwhile, for plant identification it is employed the scheme presented in Fig. 19, which consists of a cascade connection of the configurations in Figs. 16 and 17. Using this structure, an adequate plant identification is achieved since the structure allows representing the connections of the block diagram of Fig. 4 that describes the real plant.

VI. EQUATIONS AND ALGORITHM FOR CONTROLLER TRAINING

This section displays the process to establish the equations for the controller training and the steps involved in the training

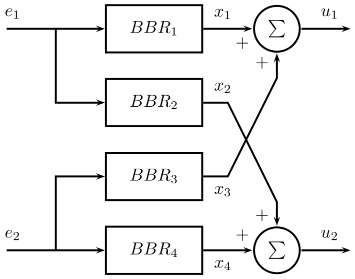

FIGURE 18. BBR-MIMO system used for the controller.

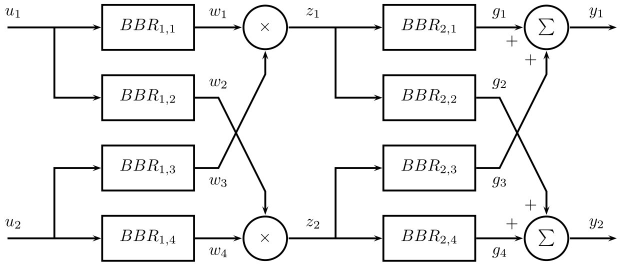

FIGURE 19. BBR-MIMO system used for plant identification.

algorithm implementation using these equations. To carry out the controller training, the plant model is included in the feedback loop, therefore, a set of recurrence equations are used, which are described in this section.

In order to determine the training equations, both the architecture of the MIMO controller and the neuro-fuzzy plant model are employed. Integrating the diagrams of Figs. 18 and 19 the diagram of the Fig. 20 is obtained. In this way, the following equations associated with the controller training are presented:

$$\begin{aligned} y_1 &= g_1 + g_3 \quad & u_1 &= x_1 + x_3 \\ y_2 &= g_2 + g_4 \quad & u_2 &= x_2 + x_4 \\ z_1 &= w_1 \cdot w_3 \quad & e_1 &= r_1 - y_1 \\ z_2 &= w_2 \cdot w_4 \quad & e_2 &= r_2 - y_2 \end{aligned} \tag{22}$$

Taking into account the configuration shown in Fig. 14, the output of each BBR subsystem is:

$$\begin{aligned} g_1 &= f_{g,1,1}(z_1) + f_{g,1,2}(g_1) \\ g_2 &= f_{g,2,1}(z_1) + f_{g,2,2}(g_2) \\ g_3 &= f_{g,3,1}(z_2) + f_{g,3,2}(g_3) \\ g_4 &= f_{g,4,1}(z_2) + f_{g,4,2}(g_4) \end{aligned} \tag{23}$$

$$\begin{aligned} w_1 &= f_{w,1,1}(u_1) + f_{w,1,2}(w_1) \\ w_2 &= f_{w,2,1}(u_1) + f_{w,2,2}(w_2) \\ w_3 &= f_{w,3,1}(u_2) + f_{w,3,2}(w_3) \\ w_4 &= f_{w,4,1}(u_2) + f_{w,4,2}(w_4) \end{aligned} \tag{24}$$

$$\begin{aligned} x_1[n+1] &= f_{x,1,1}(e_1[n+1]) + f_{x,1,2}(e_1[n]) + f_{x,1,3}(x_1[n]) \\ x_2[n+1] &= f_{x,2,1}(e_1[n+1]) + f_{x,2,2}(e_1[n]) + f_{x,2,3}(x_2[n]) \\ x_3[n+1] &= f_{x,3,1}(e_2[n+1]) + f_{x,3,2}(e_2[n]) + f_{x,3,3}(x_3[n]) \\ x_4[n+1] &= f_{x,4,1}(e_2[n+1]) + f_{x,4,2}(e_2[n]) + f_{x,4,3}(x_4[n]) \end{aligned} \tag{25}$$

Taking the subscripts for each block $s \in \{g, w, x\}$, for the subsystem $i = 1, 2, 3, 4$, for the inner connection $j = 1, 2$ and for partial output $l = 1, 2$, then, a function $f_{s,i,j}$ corresponds to:

$$\begin{aligned} f_{s,i,j}(\chi) &= \sum_{l=1}^{2} f_{s,i,j,l}(\chi) \\ &= \sum_{l=1}^{2} v_{sijl}\left(1 + e^{-\sigma_{sijl}(\chi - \gamma_{sijl})}\right)^{-1} \end{aligned} \tag{26}$$

where:

$$f_{s,i,j,l}(\chi) = v_{sijl}\left(1 + e^{-\sigma_{sijl}(\chi - \gamma_{sijl})}\right)^{-1} \tag{27}$$

namely:

$$\begin{aligned} f_{s,i,j}(\chi) = v_{sij1}&\left(1 + e^{-\sigma_{sij1}(\chi - \gamma_{sij1})}\right)^{-1} \\ &+ v_{sij2}\left(1 + e^{-\sigma_{sij2}(\chi - \gamma_{sij2})}\right)^{-1} \end{aligned} \tag{28}$$

The adaptation of the parameters is done using the equation:

$$h_c(k+1) = h_c(k) - \eta \frac{dJ_c(k)}{dh_c(k)} \tag{29}$$

where η is the learning rate and J_c is the fitness function defined as:

$$J_c = \frac{1}{2}\left[(y_{r_1} - y_1)^2 + (y_{r_2} - y_2)^2\right] \tag{30}$$

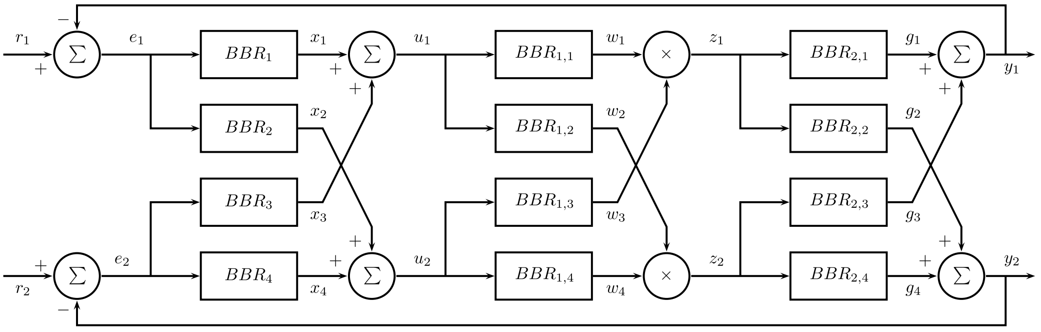

FIGURE 20. Integration of BBR-MIMO systems.

where y_{r_1} and y_{r_2} are the desired outputs (obtained from the reference model) and y_1, y_2 the outputs obtained from the controller using the neuro-fuzzy plant model.

Considering an adjustment parameter h_c, for controller training, the derivative of J_c depending on the adjustment parameters is:

$$\frac{dJ_c}{dh_c} = -e_1\frac{dy_1}{dh_c} - e_2\frac{dy_2}{dh_c} \tag{31}$$

A. DERIVATIVES FOR THE PLANT MODULE

In this section, the respective derivatives of the plant components are established with respect to the controller parameters. For the first part (output) of the plant is obtained:

$$\begin{aligned}\frac{dy_1}{dh_c} &= \frac{dg_1}{dh_c} + \frac{dg_3}{dh_c}\\ \frac{dy_2}{dh_c} &= \frac{dg_2}{dh_c} + \frac{dg_4}{dh_c}\end{aligned} \tag{32}$$

Considering that plant parameters are different from those used in the controller, the function $f_{g,i,j}$ does not depend directly on h_c; however, the variables z_i and g_i depend implicitly on h_c, then, the respective derivatives are:

$$\begin{aligned}\frac{dg_1}{dh_c}[n+1] &= \frac{df_{g,1,1}}{dz_1}\frac{dz_1}{dh_c}[n] + \frac{df_{g,1,2}}{dg_1}\frac{dg_1}{dh_c}[n]\\ \frac{dg_2}{dh_c}[n+1] &= \frac{df_{g,2,1}}{dz_1}\frac{dz_1}{dh_c}[n] + \frac{df_{g,2,2}}{dg_2}\frac{dg_2}{dh_c}[n]\\ \frac{dg_3}{dh_c}[n+1] &= \frac{df_{g,3,1}}{dz_2}\frac{dz_2}{dh_c}[n] + \frac{df_{g,3,2}}{dg_3}\frac{dg_3}{dh_c}[n]\\ \frac{dg_4}{dh_c}[n+1] &= \frac{df_{g,4,1}}{dz_2}\frac{dz_2}{dh_c}[n] + \frac{df_{g,4,2}}{dg_4}\frac{dg_4}{dh_c}[n]\end{aligned} \tag{33}$$

where is stated that:

$$\frac{df_{g,i,j}}{d\chi} = \sum_{l=1}^{2} v_{gijl}\left(1+e^{-\sigma_{gijl}(\chi-\gamma_{gijl})}\right)^{-2} e^{-\sigma_{gijl}(\chi-\gamma_{gijl})}\sigma_{gijl} \tag{34}$$

For the second part (input) of the plant model, the equations are:

$$\begin{aligned}\frac{dz_1}{dh_c} &= w_3\frac{dw_1}{dh_c} + w_1\frac{dw_3}{dh_c}\\ \frac{dz_2}{dh_c} &= w_4\frac{dw_2}{dh_c} + w_2\frac{dw_4}{dh_c}\end{aligned} \tag{35}$$

It is noteworthy that function $f_{w,i,j}$ does not depend directly on h_c; however, the variables u_i and w_i depend implicitly on h_c, then:

$$\begin{aligned}\frac{dw_1}{dh_c}[n+1] &= \frac{df_{w,1,1}}{du_1}\frac{du_1}{dh_c}[n] + \frac{df_{w,1,2}}{dw_1}\frac{dw_1}{dh_c}[n]\\ \frac{dw_2}{dh_c}[n+1] &= \frac{df_{w,2,1}}{du_1}\frac{du_1}{dh_c}[n] + \frac{df_{w,2,2}}{dw_2}\frac{dw_2}{dh_c}[n]\\ \frac{dw_3}{dh_c}[n+1] &= \frac{df_{w,3,1}}{du_2}\frac{du_2}{dh_c}[n] + \frac{df_{w,3,2}}{dw_3}\frac{dw_3}{dh_c}[n]\\ \frac{dw_4}{dh_c}[n+1] &= \frac{df_{w,4,1}}{du_2}\frac{du_2}{dh_c}[n] + \frac{df_{w,4,2}}{dw_4}\frac{dw_4}{dh_c}[n]\end{aligned} \tag{36}$$

In a general way, the respective derivatives used in the above equations can be represented as:

$$\frac{df_{w,i,j}}{d\chi} = \sum_{l=1}^{2} v_{wijl}\left(1+e^{-\sigma_{wijl}(\chi-\gamma_{wijl})}\right)^{-2} e^{-\sigma_{wijl}(\chi-\gamma_{wijl})}\sigma_{wijl} \tag{37}$$

As seen, a set of recurrence equations is presented which are associated with the neuro-fuzzy architecture employed for plant identification. In this regard, Fig. 20 displays the integration of the neuro-fuzzy controller and the plant model.

B. DERIVATIVES FOR THE CONTROLLER MODULE

In this section, the derivatives of the controller components are calculated with respect to their parameters obtaining:

$$\begin{aligned}\frac{du_1}{dh_c} &= \frac{dx_1}{dh_c} + \frac{dx_3}{dh_c}\\ \frac{du_2}{dh_c} &= \frac{dx_2}{dh_c} + \frac{dx_4}{dh_c}\end{aligned} \tag{38}$$

In this case, the function $f_{x,i,j}$ directly depends on h_c and implicitly of e_i and x_i; therefore, the respective derivatives are:

$$\begin{aligned}&\frac{dx_1}{dh_c}[n+1]\\ &= \frac{df_{x,1,1}(e_1)}{dh_c}[n+1] + \frac{df_{x,1,2}(e_1)}{dh_c}[n] + \frac{df_{x,1,3}(x_1)}{dh_c}[n]\\ &\frac{dx_2}{dh_c}[n+1]\\ &= \frac{df_{x,2,1}(e_1)}{dh_c}[n+1] + \frac{df_{x,2,2}(e_1)}{dh_c}[n] + \frac{df_{x,2,3}(x_2)}{dh_c}[n]\\ &\frac{dx_3}{dh_c}[n+1]\\ &= \frac{df_{x,3,1}(e_2)}{dh_c}[n+1] + \frac{df_{x,3,2}(e_2)}{dh_c}[n] + \frac{df_{x,3,3}(x_3)}{dh_c}[n]\\ &\frac{dx_4}{dh_c}[n+1]\\ &= \frac{df_{x,4,1}(e_2)}{dh_c}[n+1] + \frac{df_{x,4,2}(e_2)}{dh_c}[n] + \frac{df_{x,4,3}(x_4)}{dh_c}[n]\end{aligned} \tag{39}$$

In order to establish the respective derivatives, it is noteworthy that the set of controller parameters is $h_c = h_{xmnp} \in \{v_{xmnp}, \sigma_{xmnp}, \gamma_{xmnp}\}$. There are different cases depending on the variable with respect to which the derivative calculation is made; therefore, firstly if $m \neq i$ and $n \neq j$ then:

$$\frac{df_{xij}}{dh_{xmnp}} = \frac{df_{xij}}{d\chi}\frac{d\chi}{dh_{xmnp}} \tag{40}$$

where:

$$\frac{df_{xij}}{d\chi} = \sum_{l=1}^{2} v_{xijl}\left(1+e^{-\sigma_{xijl}(\chi-\gamma_{xijl})}\right)^{-2} e^{-\sigma_{xijl}(\chi-\gamma_{xijl})}\sigma_{xijl} \tag{41}$$

Second, when $m = i$ and $n = j$, for parameter $h_{xijp} = v_{xijp}$ is obtained:

$$\frac{df_{xij}}{dv_{xijp}} = \frac{df_{xij1}}{dv_{xijp}} + \frac{df_{xij2}}{dv_{xijp}} \tag{42}$$

In the case when $l = p$, the respective derivative corresponds to:

$$\begin{aligned}\frac{df_{xijp}}{dv_{xijp}} &= \left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-1} \\ &\quad - v_{xijp}\left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-2} \\ &\quad e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}(-\sigma_{xijp})\frac{d\chi}{dv_{xijp}}\end{aligned} \tag{43}$$

$$\begin{aligned}\frac{df_{xijp}}{dv_{xijp}} &= \left[\left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-1}\right] \\ &\quad + v_{xijp}\left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-2} \\ &\quad e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\sigma_{xijp}\frac{d\chi}{dv_{xijp}}\end{aligned} \tag{44}$$

$$\frac{df_{xijp}}{dv_{xijp}} = Fv_{xijp} + Kv_{xijp}\frac{d\chi}{dv_{xijp}} \tag{45}$$

where:

$$Fv_{xijp} = \left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-1} \tag{46}$$

$$\begin{aligned}Kv_{xijp} &= v_{xijp}\left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-2} \\ &\quad e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\sigma_{xijp}\end{aligned} \tag{47}$$

Meanwhile, when $l \neq p$ then:

$$\begin{aligned}\frac{df_{xijl}}{dv_{xijp}} &= v_{xijl}\left(1 + e^{-\sigma_{xijl}(\chi - \gamma_{xijl})}\right)^{-2} \\ &\quad e^{-\sigma_{xijl}(\chi - \gamma_{xijl})}\sigma_{xijl}\frac{d\chi}{dv_{xijp}}\end{aligned} \tag{48}$$

$$\frac{df_{xijl}}{dv_{xijp}} = Rv_{xijl}\frac{d\chi}{dv_{xijp}} \tag{49}$$

where:

$$Rv_{xijl} = v_{xijl}\left(1 + e^{-\sigma_{xijl}(\chi - \gamma_{xijl})}\right)^{-2} e^{-\sigma_{xijl}(\chi - \gamma_{xijl})}\sigma_{xijl} \tag{50}$$

In general, considering $l \neq p$ is obtained:

$$\frac{df_{xij}}{dv_{xijp}} = Fv_{xijp} + \left(Rv_{xijl} + Kv_{xijp}\right)\frac{d\chi}{dv_{xijp}} \tag{51}$$

In the same way, for the parameter $h_{xijp} = \sigma_{xijp}$ is stated that:

$$\frac{df_{xij}}{d\sigma_{xijp}} = \frac{df_{xij1}}{d\sigma_{xijp}} + \frac{df_{xij2}}{d\sigma_{xijp}} \tag{52}$$

For the case when $l = p$ then:

$$\begin{aligned}\frac{df_{xijp}}{d\sigma_{xijp}} &= -v_{xijp}\left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-2} \\ &\quad e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\left(\gamma_{xijp} - \chi - \sigma_{xijp}\frac{d\chi}{d\sigma_{xijp}}\right)\end{aligned} \tag{53}$$

$$\begin{aligned}\frac{df_{xijp}}{d\sigma_{xijp}} &= v_{xijp}\left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-2} \\ &\quad e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\left(\chi - \gamma_{xijp}\right) \\ &\quad + v_{xijp}\left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-2} \\ &\quad e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\sigma_{xijp}\frac{d\chi}{d\sigma_{xijp}}\end{aligned} \tag{54}$$

$$\frac{df_{xijp}}{d\sigma_{xijp}} = F\sigma_{xijp} + K\sigma_{xijp}\frac{d\chi}{d\sigma_{xijp}} \tag{55}$$

where:

$$\begin{aligned}F\sigma_{xijp} &= v_{xijp}\left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-2} \\ &\quad e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\left(\chi - \gamma_{xijp}\right)\end{aligned} \tag{56}$$

$$\begin{aligned}K\sigma_{xijp} &= v_{xijp}\left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-2} \\ &\quad e^{-\sigma_{xijp}(x_i - \gamma_{xijp})}\sigma_{xijp}\end{aligned} \tag{57}$$

In the case when $l \neq p$, then is obtained:

$$\begin{aligned}\frac{df_{xijl}}{d\sigma_{xijp}} &= v_{xijl}\left(1 + e^{-\sigma_{xijl}(\chi - \gamma_{xijl})}\right)^{-2} \\ &\quad e^{-\sigma_{xijl}(\chi - \gamma_{xijl})}\sigma_{xijl}\frac{d\chi}{d\sigma_{xijp}}\end{aligned} \tag{58}$$

$$\frac{df_{xijl}}{d\sigma_{xijp}} = R\sigma_{xijl}\frac{d\chi}{d\sigma_{xijp}} \tag{59}$$

where:

$$R\sigma_{xijl} = v_{xijl}\left(1 + e^{-\sigma_{xijl}(\chi - \gamma_{xijl})}\right)^{-2} e^{-\sigma_{xijl}(\chi - \gamma_{xijl})}\sigma_{xijl} \tag{60}$$

In general, considering $l \neq p$ is stated that:

$$\frac{df_{xij}}{d\sigma_{xijp}} = F\sigma_{xijp} + \left(R\sigma_{xijl} + K\sigma_{xijp}\right)\frac{d\chi}{d\sigma_{xijp}} \tag{61}$$

Finally, for the parameter $h_{xijp} = \gamma_{xijp}$ is obtained:

$$\frac{df_{xij}}{d\gamma_{xijp}} = \frac{df_{xij1}}{d\gamma_{xijp}} + \frac{df_{xij2}}{d\gamma_{xijp}} \tag{62}$$

Considering the case $l = p$, then:

$$\begin{aligned}\frac{df_{xijp}}{d\gamma_{xijp}} &= -v_{xijp}\left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-2} \\ &\quad e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\left(\sigma_{xijp} - \sigma_{xijp}\frac{d\chi}{d\gamma_{xijp}}\right)\end{aligned} \tag{63}$$

$$\begin{aligned}\frac{df_{xijp}}{d\gamma_{xijp}} &= -v_{xijp}\left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-2} \\ &\quad e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\sigma_{xijp} \\ &\quad + v_{xijp}\left(1 + e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\right)^{-2} \\ &\quad e^{-\sigma_{xijp}(\chi - \gamma_{xijp})}\sigma_{xijp}\frac{d\chi}{d\gamma_{xijp}}\end{aligned} \tag{64}$$

$$\frac{df_{xijp}}{d\gamma_{xijp}} = F\gamma_{xijp} + K\gamma_{xijp}\frac{d\chi}{d\gamma_{xijp}} \tag{65}$$

where:

$$F\gamma_{xijp} = -v_{xijp}\left(1+e^{-\sigma_{xijp}(\chi-\gamma_{xijp})}\right)^{-2} e^{-\sigma_{xijp}(\chi-\gamma_{xijp})}\sigma_{xijp} \quad (66)$$

$$K\gamma_{xijp} = v_{xijp}\left(1+e^{-\sigma_{xijp}(\chi-\gamma_{xijp})}\right)^{-2} e^{-\sigma_{xijp}(\chi-\gamma_{xijp})}\sigma_{xijp} \quad (67)$$

Meanwhile if $l \neq p$ then:

$$\frac{df_{xijl}}{d\gamma_{xijp}} = v_{xijl}\left(1+e^{-\sigma_{xijl}(\chi-\gamma_{xijl})}\right)^{-2} e^{-\sigma_{xijl}(\chi-\gamma_{xijl})}\sigma_{xijl}\frac{d\chi}{d\gamma_{xijp}} \quad (68)$$

$$\frac{df_{xijl}}{d\gamma_{xijp}} = R\gamma_{xijl}\frac{d\chi}{d\gamma_{xijp}} \quad (69)$$

where:

$$R\gamma_{xijl} = v_{xijl}\left(1+e^{-\sigma_{xijl}(\chi-\gamma_{xijl})}\right)^{-2} e^{-\sigma_{xijl}(\chi-\gamma_{xijl})}\sigma_{xijl} \quad (70)$$

In general, for the case $l \neq p$ is obtained:

$$\frac{df_{xij}}{d\gamma_{xijp}} = F\gamma_{xijp} + \left(R\gamma_{xijl} + K\gamma_{xijp}\right)\frac{d\chi}{d\gamma_{xijp}} \quad (71)$$

In this way, the equations used for the training controller are:

$$\frac{dx_1}{dh_c}[n+1] = Th_{x11}\frac{de_1}{dh_c}[n+1] + Th_{x12}\frac{de_1}{dh_c}[n] + Th_{x13}\frac{dx_1}{dh_c}[n] + Fh_{x11}(e_1[n+1]) + Fh_{x12}(e_1[n]) + Fh_{x13}(x_1[n]) \quad (72)$$

$$\frac{dx_2}{dh_c}[n+1] = Th_{x21}\frac{de_1}{dh_c}[n+1] + Th_{x22}\frac{de_1}{dh_c}[n] + Th_{x23}\frac{dx_2}{dh_c}[n] + Fh_{x21}(e_1[n+1]) + Fh_{x22}(e_1[n]) + Fh_{x23}(x_2[n]) \quad (73)$$

$$\frac{dx_3}{dh_c}[n+1] = Th_{x31}\frac{de_2}{dh_c}[n+1] + Th_{x32}\frac{de_2}{dh_c}[n] + Th_{x33}\frac{dx_3}{dh_c}[n] + Fh_{x31}(e_2[n+1]) + Fh_{x32}(e_2[n]) + Fh_{x33}(x_3[n]) \quad (74)$$

$$\frac{dx_4}{dh_c}[n+1] = Th_{x41}\frac{de_2}{dh_c}[n+1] + Th_{x42}\frac{de_2}{dh_c}[n] + Th_{x43}\frac{dx_4}{dh_c}[n] + Fh_{x41}(e_2[n+1]) + Fh_{x42}(e_2[n]) + Fh_{x43}(x_4[n]) \quad (75)$$

where:

$$Th_{xij}(\chi)=\sum_{l=1}^{2} v_{xijl}\left(1+e^{-\sigma_{xijl}(\chi-\gamma_{xijl})}\right)^{-2} e^{-\sigma_{xijl}(\chi-\gamma_{xijl})}\sigma_{xijl} \quad (76)$$

In addition, $Fh_{xij} \in \{F\gamma_{xijp}, F\sigma_{xijp}, Fv_{xijp}\}$.

C. TRAINING ALGORITHM

In order to show the training process, the following are the algorithm steps to adjust the neuro-fuzzy controller:

1) Establish the plant model (neuro-fuzzy) and choose the initial configuration for the controller parameters.
2) Calculate the reference models response obtaining (y_{r_1}, y_{r_2}).
3) In the respective training iteration k, for the current simulation step n, the close-loop control system response (y_1, y_2) is calculated using the plant neuro-fuzzy model and the reference values (r_1, r_2).
4) Adjust the neuro-fuzzy controller parameters using the equations (29), (31), and subsequent. It is relevant to point out that the adjusted parameters are stored into temporary variables since the neuro-fuzzy system does not employ such values in this step.
5) Return to step 3 for the next simulation step $n = n + 1$ (where the control system output is calculated) until n is equal to a defined value (simulation time).
6) When the simulation time is completed, the optimized parameters are updated, then return to step 3 to a new iteration $k = k + 1$ until error $J_c(k)$ is less than a defined ε, or until k is equal to a defined number.

VII. EQUATIONS AND ALGORITHM FOR PLANT IDENTIFICATION

Considering the structure defined for the plant identification, this section shows the deduction of the equations to carry out the parameters adaptation. As the MIMO structure has internal feedback in the neuro-fuzzy subsystems, a set of recurrent equations is presented which are used in the training algorithm. The steps of this algorithm are shown in the last part of this section.

According to [61], [62], an approach to the system model consists of estimating a neuronal structure that can perform the same function of the plant. For the identification of the plant with the neuro-fuzzy system, samples of the input and output are taken in a way that the response of the neuro-fuzzy system can be seen as a non-linear function of these signals. The adaptation or training of the parameters is carried out as follows:

$$h_p(k+1) = h_p(k) - \eta\frac{dJ_p(k)}{dh_p(k)} \quad (77)$$

where η is the learning rate and J_p corresponds to the adjustment function defined as:

$$J_p = \frac{1}{2}(y_{d_1} - y_1)^2 + \frac{1}{2}(y_{d_2} - y_2)^2 \quad (78)$$

In this equation y_{d_1} and y_{d_2} are the measured data of the plant and y_1, y_2 the data obtained from the neuro-fuzzy system. Considering an adjustment parameter h_p in order to implement the Equation (77), the derivative of J_p depending on the adjustment parameters is:

$$\frac{dJ_p}{dh_p} = -(y_{d_1} - y_1)\frac{dy_1}{dh_p} - (y_{d_2} - y_2)\frac{dy_2}{dh_p} \quad (79)$$

A. TRAINING EQUATIONS FOR THE OUTPUT MODULE

For the first part of the plant (output) where the parameters of this module are h_g is stated that:

$$\begin{aligned} \frac{dy_1}{dh_g} &= \frac{dg_1}{dh_g} + \frac{dg_3}{dh_g} \\ \frac{dy_2}{dh_g} &= \frac{dg_2}{dh_g} + \frac{dg_4}{dh_g} \end{aligned} \tag{80}$$

$$\begin{aligned} \frac{dg_1}{dh_g}[n+1] &= \frac{df_{g,1,1}(z_1)}{dh_g}[n] + \frac{df_{g,1,2}(g_1)}{dh_g}[n] \\ \frac{dg_2}{dh_g}[n+1] &= \frac{df_{g,2,1}(z_1)}{dh_g}[n] + \frac{df_{g,2,2}(g_2)}{dh_g}[n] \\ \frac{dg_3}{dh_g}[n+1] &= \frac{df_{g,3,1}(z_2)}{dh_g}[n] + \frac{df_{g,3,2}(g_3)}{dh_g}[n] \\ \frac{dg_4}{dh_g}[n+1] &= \frac{df_{g,4,1}(z_2)}{dh_g}[n] + \frac{df_{g,4,2}(g_4)}{dh_g}[n] \end{aligned} \tag{81}$$

For the function f_{gij} considering the case where $i \neq m$ and $j \neq n$, this function does not depend on h_{gmnp}, therefore:

$$\frac{df_{gij}}{dh_{gmnp}} = 0 \tag{82}$$

Meanwhile, in the case $i = m$ and $j = n$ is obtained:

$$\frac{df_{gij}}{dh_{gijp}} = \frac{df_{gij1}}{dh_{gijp}} + \frac{df_{gij2}}{dh_{gijp}} \tag{83}$$

For the respective calculation it must be borne in mind that the inputs z_1 and z_2 do not depend on h_g that is f_{gi1p} depends only on h_{gi1p}, therefore, for each parameter is obtained:

$$\frac{df_{gi1p}}{dv_{gi1p}} = \left(1 + e^{-\sigma_{gi1p}(\chi - \gamma_{gi1p})}\right)^{-1} \tag{84}$$

$$\begin{aligned} \frac{df_{gi1p}}{d\sigma_{gi1p}} &= v_{gi1p}\left(1 + e^{-\sigma_{gi1p}(\chi - \gamma_{gi1p})}\right)^{-2} \\ &\quad e^{-\sigma_{gi1p}(\chi - \gamma_{gi1p})}(\chi - \gamma_{gi1p}) \end{aligned} \tag{85}$$

$$\begin{aligned} \frac{df_{gi1p}}{d\gamma_{gi1p}} &= v_{gi1p}\left(1 + e^{-\sigma_{gi1p}(\chi - \gamma_{gi1p})}\right)^{-2} \\ &\quad e^{-\sigma_{gi1p}(\chi - \gamma_{gi1p})}\sigma_{gi1p} \end{aligned} \tag{86}$$

For the case $p \neq k$, is stated that:

$$\frac{df_{gi1k}}{dh_{gi1p}} = 0 \tag{87}$$

Considering the internal signals $g_1, \ldots, g_4$, the function f_{gi2p} depends directly and implicitly on h_{gi2p}, therefore:

$$\begin{aligned} \frac{df_{gi2p}}{dv_{gi2p}} &= \left(1 + e^{-\sigma_{gi2p}(\chi - \gamma_{gi2p})}\right)^{-1} \\ &\quad - v_{gi2p}\left(1 + e^{-\sigma_{gi2p}(\chi - \gamma_{gi2p})}\right)^{-2} \\ &\quad e^{-\sigma_{gi2p}(\chi - \gamma_{gi2p})}(-\sigma_{gi2p})\frac{d\chi}{dv_{gi2p}} \end{aligned} \tag{88}$$

$$\begin{aligned} \frac{df_{gi2p}}{d\sigma_{gi2p}} &= -v_{gi2p}\left(1 + e^{-\sigma_{gi2p}(\chi - \gamma_{gi2p})}\right)^{-2} \\ &\quad e^{-\sigma_{gi2p}(\chi - \gamma_{gi2p})}\left(\gamma_{gi2p} - \chi - \sigma_{gi2p}\frac{d\chi}{d\sigma_{gi2p}}\right) \end{aligned} \tag{89}$$

$$\begin{aligned} \frac{df_{gi2p}}{d\gamma_{gi2p}} &= -v_{gi2p}\left(1 + e^{-\sigma_{gi2p}(\chi - \gamma_{gi2p})}\right)^{-2} \\ &\quad e^{-\sigma_{gi2p}(\chi - \gamma_{gi2p})}\left(\sigma_{gi2p} - \sigma_{gi2p}\frac{d\chi}{d\gamma_{gi2p}}\right) \end{aligned} \tag{90}$$

In addition, if $p \neq k$, the function f_{gi2k} depends implicitly on h_{gi2p} then:

$$\frac{df_{gi2k}}{dh_{gi2p}} = \left(1 + e^{-\sigma_{gi2k}(\chi - \gamma_{gi2k})}\right)^{-2} e^{-\sigma_{gi2k}(\chi - \gamma_{gi2k})}\sigma_{gi2k}\frac{d\chi}{dh_{gi2p}} \tag{91}$$

In this way, in general terms, is stated that:

$$\begin{aligned} \frac{dg_1}{dh_g}[n+1] &= \frac{df_{g,1,1}}{dh_g} + \frac{df_{g,1,2}}{dg_1}\frac{dg_1}{dh_g}[n] + \frac{df_{g,1,2}}{dh_g} \\ \frac{dg_2}{dh_g}[n+1] &= \frac{df_{g,2,1}}{dh_g} + \frac{df_{g,2,2}}{dg_2}\frac{dg_2}{dh_g}[n] + \frac{df_{g,2,2}}{dh_g} \\ \frac{dg_3}{dh_g}[n+1] &= \frac{df_{g,3,1}}{dh_g} + \frac{df_{g,3,2}}{dg_3}\frac{dg_3}{dh_g}[n] + \frac{df_{g,3,2}}{dh_g} \\ \frac{dg_4}{dh_g}[n+1] &= \frac{df_{g,4,1}}{dh_g} + \frac{df_{g,4,2}}{dg_4}\frac{dg_4}{dh_g}[n] + \frac{df_{g,4,2}}{dh_g} \end{aligned} \tag{92}$$

where the derivatives $df_{g,i,j}/dh_g$ corresponds to:

$$\frac{df_{gijp}}{dv_{gijp}} = \left(1 + e^{-\sigma_{gijp}(\chi - \gamma_{gijp})}\right)^{-1} \tag{93}$$

$$\begin{aligned} \frac{df_{gijp}}{d\sigma_{gijp}} &= v_{gijp}\left(1 + e^{-\sigma_{gijp}(\chi - \gamma_{gijp})}\right)^{-2} \\ &\quad e^{-\sigma_{gijp}(\chi - \gamma_{gijp})}(\chi - \gamma_{gijp}) \end{aligned} \tag{94}$$

$$\begin{aligned} \frac{df_{gijp}}{d\gamma_{gijp}} &= v_{gijp}\left(1 + e^{-\sigma_{gijp}(\chi - \gamma_{gijp})}\right)^{-2} \\ &\quad e^{-\sigma_{gijp}(\chi - \gamma_{gijp})}\sigma_{gijp} \end{aligned} \tag{95}$$

In the same way, for the derivatives $df_{g,i,2}/dg_i$ is obtained:

$$\frac{df_{g,i,2}}{d\chi} = \sum_{l=1}^{2} v_{gi2l}\left(1 + e^{-\sigma_{gi2l}(\chi - \gamma_{gi2l})}\right)^{-2} e^{-\sigma_{gi2l}(\chi - \gamma_{gi2l})}\sigma_{gi2l} \tag{96}$$

B. TRAINING EQUATIONS FOR THE INPUT MODULE

For the second plant part (input), having the parameters of this module h_w is obtained:

$$\begin{aligned} \frac{dy_1}{dh_w} &= \frac{dg_1}{dh_w} + \frac{dg_3}{dh_w} \\ \frac{dy_2}{dh_w} &= \frac{dg_2}{dh_w} + \frac{dg_4}{dh_w} \end{aligned} \tag{97}$$

$$\begin{aligned} \frac{dg_1}{dh_w}[n+1] &= \frac{df_{g,1,1}(z_1)}{dh_w}[n] + \frac{df_{g,1,2}(g_1)}{dh_w}[n] \\ \frac{dg_2}{dh_w}[n+1] &= \frac{df_{g,2,1}(z_1)}{dh_w}[n] + \frac{df_{g,2,2}(g_2)}{dh_w}[n] \\ \frac{dg_3}{dh_w}[n+1] &= \frac{df_{g,3,1}(z_2)}{dh_w}[n] + \frac{df_{g,3,2}(g_3)}{dh_w}[n] \\ \frac{dg_4}{dh_w}[n+1] &= \frac{df_{g,4,1}(z_2)}{dh_w}[n] + \frac{df_{g,4,2}(g_4)}{dh_w}[n] \end{aligned} \tag{98}$$

$$\begin{aligned}\frac{dz_1}{dh_w} &= w_3\frac{dw_1}{dh_w}+w_1\frac{dw_3}{dh_w}\\ \frac{dz_2}{dh_w} &= w_4\frac{dw_2}{dh_w}+w_2\frac{dw_4}{dh_w}\end{aligned}\tag{99}$$

$$\begin{aligned}\frac{dw_1}{dh_w}[n+1] &= \frac{df_{w,1,1}(u_1)}{dh_w}[n]+\frac{df_{w,1,2}(x_1)}{dh_p}[n]\\ \frac{dw_2}{dh_w}[n+1] &= \frac{df_{w,2,1}(u_1)}{dh_w}[n]+\frac{df_{w,3,2}(x_2)}{dh_p}[n]\\ \frac{dw_3}{dh_w}[n+1] &= \frac{df_{w,3,1}(u_2)}{dh_w}[n]+\frac{df_{w,2,2}(x_3)}{dh_p}[n]\\ \frac{dw_4}{dh_w}[n+1] &= \frac{df_{w,4,1}(u_2)}{dh_w}[n]+\frac{df_{w,4,2}(x_4)}{dh_p}[n]\end{aligned}\tag{100}$$

For this module, given the connection of the blocks, the function f_{gij} depends implicitly on parameters h_{wmnp}, therefore:

$$\frac{df_{gij}}{dh_{wmnp}} = \sum_{l=1}^{2} v_{wijl}\left(1+e^{-\sigma_{wijl}(\chi-\gamma_{wijl})}\right)^{-2} e^{-\sigma_{wijl}(\chi-\gamma_{wijl})}\sigma_{wijl}\frac{d\chi}{dh_{wmnp}}\tag{101}$$

In this way is obtained:

$$\begin{aligned}\frac{dg_1}{dh_w}[n+1] &= \frac{df_{g,1,1}}{dz_1}\frac{dz_1}{dh_w}[n]+\frac{df_{g,1,2}}{dg_1}\frac{dg_1}{dh_w}[n]\\ \frac{dg_2}{dh_w}[n+1] &= \frac{df_{g,2,1}}{dz_1}\frac{dz_1}{dh_w}[n]+\frac{df_{g,2,2}}{dg_2}\frac{dg_2}{dh_w}[n]\\ \frac{dg_3}{dh_w}[n+1] &= \frac{df_{g,3,1}}{dz_2}\frac{dz_2}{dh_w}[n]+\frac{df_{g,3,2}}{dg_3}\frac{dg_3}{dh_w}[n]\\ \frac{dg_4}{dh_w}[n+1] &= \frac{df_{g,4,1}}{dz_2}\frac{dz_2}{dh_w}[n]+\frac{df_{g,4,2}}{dg_4}\frac{dg_4}{dh_w}[n]\end{aligned}\tag{102}$$

where:

$$\frac{df_{gij}}{d\chi} = \sum_{l=1}^{2} v_{wijl}\left(1+e^{-\sigma_{wijl}(\chi-\gamma_{wijl})}\right)^{-2} e^{-\sigma_{wijl}(\chi-\gamma_{wijl})}\sigma_{wijl}\tag{103}$$

Meanwhile, for the function f_{wij} must be taken into account the dependency of the parameters h_{wmnp}. Considering the case where $i \neq m$ and $j \neq n$ is obtained:

$$\frac{df_{wij}}{dh_{wnml}} = 0\tag{104}$$

When $i = m$ and $j = n$ is stated that:

$$\frac{df_{wij}}{dh_{wijp}} = \frac{df_{wij1}}{dh_{wijp}}+\frac{df_{wij2}}{dh_{wijp}}\tag{105}$$

For the respective calculation it must be considered that the inputs u_1 and u_2 do not depend on h_w, that is f_{wi1p} only depends on h_{wi1p}, therefore, respectively for each parameter is obtained:

$$\frac{df_{wi1p}}{dv_{wi1p}} = \left(1+e^{-\sigma_{wi1p}(\chi-\gamma_{wi1p})}\right)^{-1}\tag{106}$$

$$\frac{df_{wi1p}}{d\sigma_{wi1p}} = v_{wi1p}\left(1+e^{-\sigma_{wi1p}(\chi-\gamma_{wi1p})}\right)^{-2} e^{-\sigma_{wi1p}(\chi-\gamma_{wi1p})}(\chi-\gamma_{wi1p})\tag{107}$$

$$\frac{df_{wi1p}}{d\gamma_{wi1p}} = v_{wi1p}\left(1+e^{-\sigma_{wi1p}(\chi-\gamma_{wi1p})}\right)^{-2} e^{-\sigma_{wi1p}(\chi-\gamma_{wi1p})}\sigma_{wi1p}\tag{108}$$

For the case when $p \neq k$ then:

$$\frac{df_{wi1k}}{dh_{wi1p}} = 0\tag{109}$$

Considering the internal signals $g_1, \ldots, g_4$, the function f_{wi2p} depends directly and implicitly on h_{wi2p}, therefore:

$$\begin{aligned}\frac{df_{wi2p}}{dv_{wi2p}} &= \left(1+e^{-\sigma_{wi2p}(\chi-\gamma_{wi2p})}\right)^{-1}\\ &\quad - v_{wi2p}\left(1+e^{-\sigma_{wi2p}(\chi-\gamma_{wi2p})}\right)^{-2}\\ &\quad e^{-\sigma_{wi2p}(\chi-\gamma_{wi2p})}(-\sigma_{wi2p})\frac{d\chi}{dv_{wi2p}}\end{aligned}\tag{110}$$

$$\begin{aligned}\frac{df_{wi2p}}{d\sigma_{wi2p}} &= -v_{wi2p}\left(1+e^{-\sigma_{wi2p}(\chi-\gamma_{wi2p})}\right)^{-2}\\ &\quad e^{-\sigma_{wi2p}(\chi-\gamma_{wi2p})}\left(\gamma_{wi2p}-\chi-\sigma_{wi2p}\frac{d\chi}{d\sigma_{wi2p}}\right)\end{aligned}\tag{111}$$

$$\begin{aligned}\frac{df_{wi2p}}{d\gamma_{wi2p}} &= -v_{wi2p}\left(1+e^{-\sigma_{wi2p}(\chi-\gamma_{wi2p})}\right)^{-2}\\ &\quad e^{-\sigma_{wi2p}(\chi-\gamma_{wi2p})}\left(\sigma_{wi2p}-\sigma_{wi2p}\frac{d\chi}{d\gamma_{wi2p}}\right)\end{aligned}\tag{112}$$

In addition, if $p \neq k$, then, f_{wi2k} depends implicitly on h_{wi2p} therefore:

$$\frac{df_{wi2k}}{dh_{wi2p}} = \left(1+e^{-\sigma_{wi2k}(\chi-\gamma_{wi2k})}\right)^{-2} e^{-\sigma_{wi2k}(\chi-\gamma_{wi2k})}\sigma_{wi2k}\frac{d\chi}{dh_{wi2p}}\tag{113}$$

In this way, in general terms, is obtained:

$$\begin{aligned}\frac{dg_1}{dh_w}[n+1] &= \frac{df_{w,1,1}}{dh_w}+\frac{df_{w,1,2}}{dg_1}\frac{dg_1}{dh_w}[n]+\frac{df_{w,1,2}}{dh_w}\\ \frac{dg_2}{dh_w}[n+1] &= \frac{df_{w,2,1}}{dh_w}+\frac{df_{w,2,2}}{dg_2}\frac{dg_2}{dh_w}[n]+\frac{df_{w,2,2}}{dh_w}\\ \frac{dg_3}{dh_w}[n+1] &= \frac{df_{w,3,1}}{dh_w}+\frac{df_{w,3,2}}{dg_3}\frac{dg_3}{dh_w}[n]+\frac{df_{w,3,2}}{dh_w}\\ \frac{dg_4}{dh_w}[n+1] &= \frac{df_{w,4,1}}{dh_w}+\frac{df_{w,4,2}}{dg_4}\frac{dg_4}{dh_w}[n]+\frac{df_{w,4,2}}{dh_w}\end{aligned}\tag{114}$$

The respective derivatives $df_{w,i,j}/dh_w$ correspond to:

$$\frac{df_{wijp}}{dv_{wijp}} = \left(1+e^{-\sigma_{wijp}(\chi-\gamma_{wijp})}\right)^{-1}\tag{115}$$

$$\frac{df_{wijp}}{d\sigma_{wijp}} = v_{wijp}\left(1+e^{-\sigma_{wijp}(\chi-\gamma_{wijp})}\right)^{-2} e^{-\sigma_{wijp}(\chi-\gamma_{wijp})}(\chi-\gamma_{wijp})\tag{116}$$

$$\frac{df_{wijp}}{d\gamma_{wijp}} = v_{wijp}\left(1+e^{-\sigma_{wijp}(\chi-\gamma_{wijp})}\right)^{-2} e^{-\sigma_{wijp}(\chi-\gamma_{wijp})}\sigma_{wijp}\tag{117}$$

In the same way, for derivatives $df_{w,i,2}/dg_i$ is obtained:

$$\frac{df_{w,i,2}}{d\chi} = \sum_{l=1}^{2} v_{wi2l}\left(1+e^{-\sigma_{wi2l}(\chi-\gamma_{wi2l})}\right)^{-2} e^{-\sigma_{wi2l}(\chi-\gamma_{wi2l})}\sigma_{wi2l} \qquad (118)$$

C. TRAINING ALGORITHM

As observed, there are internal feedbacks present in the structure of the plant model; therefore, a set of recurrence equations is utilized for parameters adjustment. The algorithm steps used to identify the plant are the following:

1) Define an initial configuration of the parameters for the plant's model (neuro-fuzzy system).
2) In the respective training iteration k, for the current simulation step n and employing the input data (u_1, u_2), the output of the neuro-fuzzy system is calculated (y_1, y_2).
3) Using input-output data pairs (u_1, y_{d_1}) and (u_2, y_{d_2}), and the data obtained from the model (y_1, y_2), the neuro-fuzzy system parameters adjustment is made using equations (77), (79), and subsequent. Here, it is important to point out that the adjusted parameters are stored in auxiliary variables since the neuro-fuzzy system does not employ such values in this step.
4) Return to step 2 for the next simulation step $n = n + 1$ (where the neuro-fuzzy system output is calculated) until n is equal to a defined value (simulation time).
5) When completing the simulation time, the parameters are updated and then return to step 2 to a new iteration $k = k + 1$ until the error $J_p(k)$ is less than a ε defined, or until k is equal to a determined value.

VIII. EXPERIMENTAL RESULTS

To observe the behavior of the adaptive control system, a set of experimental tests are carried out with the real MIMO plant, in a way that the iterative adjustments can be appreciated so that the plant output reaches the desired reference value. First, the performance metrics used to evaluate the performance of the controller considering different configurations of the controller are described, then the results obtained for different reference values are shown; finally some observations are presented considering these results.

The performance index considered to evaluate the system response corresponds to the shown in the Equation (119), where r_1, r_2 are the desired reference values, y_1, y_2 the outputs of the control system and N_T the total number of data.

$$J = \frac{1}{N_T}\sum_{n=1}^{N_T}\left[(r_1[n]-y_1[n])^2 + (r_2[n]-y_2[n])^2\right] \qquad (119)$$

Also, this equation can be expressed as $J = J_1 + J_2$, where:

$$J_1 = \frac{1}{N_T}\sum_{n=1}^{N_T}(r_1[n]-y_1[n])^2 \qquad (120)$$

TABLE 6. **Summary of the configurations, according to f_j from Fig. 15.**

Configuration	f_1	f_2	f_3
CO1	Yes	No	Yes
CO2	Yes	Yes	Yes
CO3	No	Yes	Yes

$$J_2 = \frac{1}{N_T}\sum_{n=1}^{N_T}(r_2[n]-y_2[n])^2 \qquad (121)$$

Taking into account the data obtained of the plant in open loop (Fig. 5) and the results reported in [53], the desired behavior corresponds to first-order systems with settling time of 60 seconds; therefore, reference models are:

$$G_1(s) = \frac{1}{15s+1} \qquad (122)$$

$$G_2(s) = \frac{1}{15s+1} \qquad (123)$$

Using the reference values of Table 7, the respective response of the reference models can be seen in Fig. 21.

For the experimental tests, three neuro-fuzzy configurations shown in Table 6 are considered. Here, the presence or absence of f_j functions in Fig. 15 are considered to implement the BBR subsystems based for the MIMO controller.

- CO1: is proposed as the equivalence with a transfer function with one pole and gain.

$$C_i(z) = \frac{b_1}{1-a_0z^{-1}} \qquad (124)$$

The equation for implementing the BBR subsystems is $w[n] = f_1(e[n]) + f_3(w[n-1])$.

- CO2: is considered as an analogy of a transfer function with zero, pole, and gain.

$$C_i(z) = \frac{b_1 + b_0z^{-1}}{1-a_0z^{-1}} \qquad (125)$$

The equation for the implementation of the BBR subsystems is $w[n] = f_1(e[n]) + f_2(e[n-1]) + f_3(w[n-1])$.

- CO3: is an analogy with a transfer function with pole, gain, and a zero at the origin.

$$C_i(z) = \frac{b_0z^{-1}}{1-a_0z^{-1}} \qquad (126)$$

The equation for implementing the BBR subsystems is $w[n] = f_2(e[n-1]) + f_3(w[n-1])$.

As previously mentioned, the fuzzy sets from Fig. 12 are considered for modeling positive and negative values. In order to determine the parameters configuration of the BBR subsystems, each function f_j of Fig. 22 can be set to obtain a behavior similar to the one displayed in Fig. 23. Such approach allows defining the parameters to each BBR subsystem through an analogy with the SISO linear system considered as based. Taking into account the range of values of the input variables, for this case $[-100, 100]$, the BBR subsystem is configured to get a similar behavior to the linear system in this range.

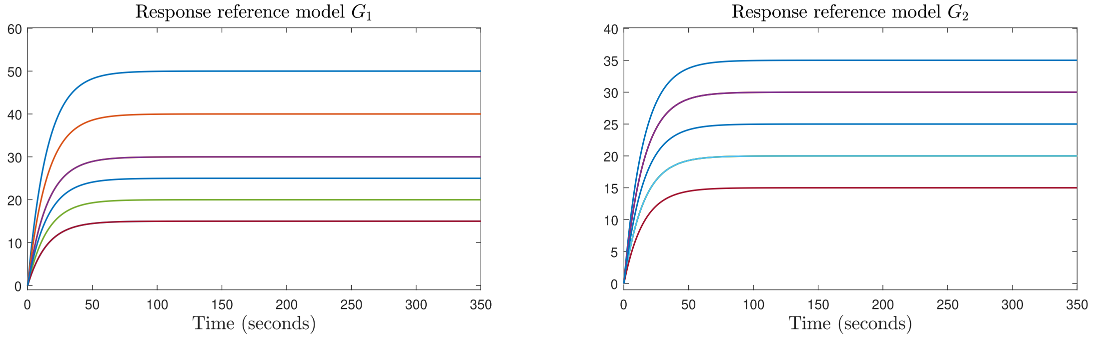

FIGURE 21. Simulation of the reference models.

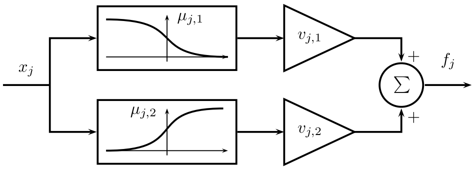

FIGURE 22. Configuration for partial output f_j.

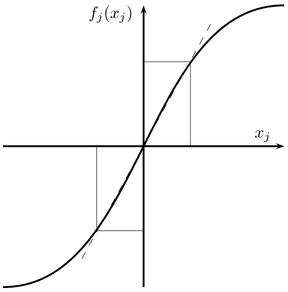

FIGURE 23. Configuration for the positive slope.

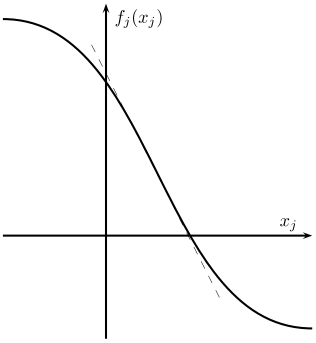

FIGURE 24. Configuration for the negative slope corresponding to $100 - u_2$.

TABLE 7. References used.

Outputs	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
y_1	50	40	30	30	20	15	15	25
y_2	35	30	20	30	20	20	15	25

Meanwhile, the values obtained from the parameter's identification shown in Section II are considered for the neuro-fuzzy plant model initialization. The configuration shown in Fig. 19 is proposed for having an equivalence regarding the diagram of Fig. 4; whereby, the initial configuration considers that $g_2 = 0$ and $g_3 = 0$. Besides, the intermediate output $z_1 = w_1 w_3$ is equivalent to $u_1(100 - u_2)$, and the other partial output $z_2 = w_2 w_4$ is equivalent to $u_1 u_2$. The function $100 - u_2$ is achieved through the configuration of f_j shown in Fig. 24. The configurations of $BBR_{2,1}$ and $BBR_{2,4}$ allow equivalent behaviors to first order systems with a settling time of 100 seconds similar to the real plant, in this order, for these subsystems is used the configuration of f_j shown in Fig. 23.

The reference values shown in Table 7 are considered for the experimental tests. In this order, all results (outputs) of the system can be seen in Fig. 25, where the associated control action can be also observed. To better observe the results, these are presented in separated groups. Particularly, in Fig. 26, the results are displayed for references R_1, R_3 and R_7. Meanwhile, Fig. 27 presents the results for references R_2, R_6 and R_8. Finally, Fig. 27 shows the results for references R_4 and R_5.

In these results, it can be seen that the system presents greater variation for high reference values, it can be also observed that every 100 seconds plant identification and adaptation process of the controller are carried out, the first adaptation process being where the greatest variation occurs. It can be also observed that as more adaptation processes are carried out, the outputs approach the references. In these results, it should be noted that the CO1 and CO3 configurations are the ones that present the greatest variation in most of the references taken.

Also, in these results is observed that the output y_1 presents more noise compared to y_2, which is associated with the location of tank $D1$ at the bottom (see Fig. 2). Despite the above, the control system manages to reach the reference values for both outputs y_1 and y_2.

Performing the respective error calculation for each reference, Table 8 shows the values for J_1 and Table 9 the results for J_2; therefore, Table 10 shows the values of J for

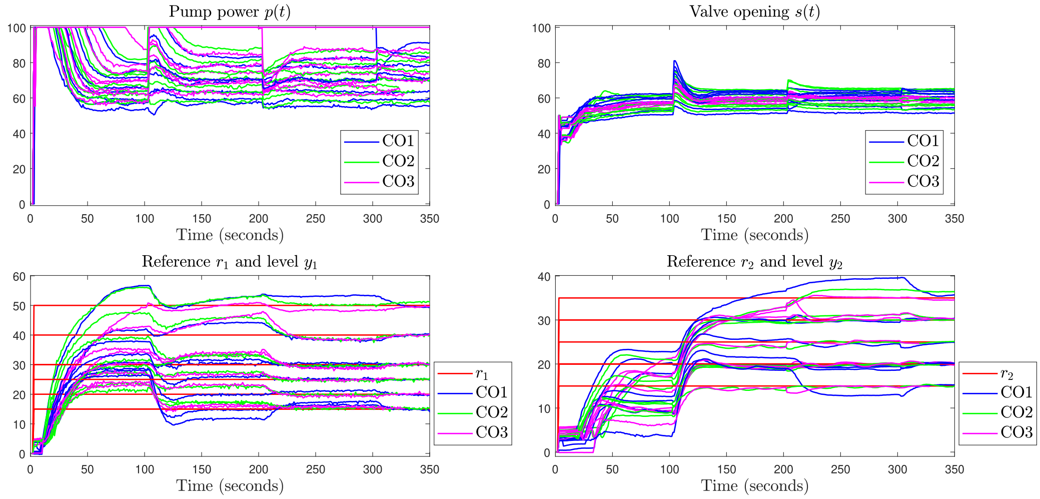

FIGURE 25. Adaptive control system results.

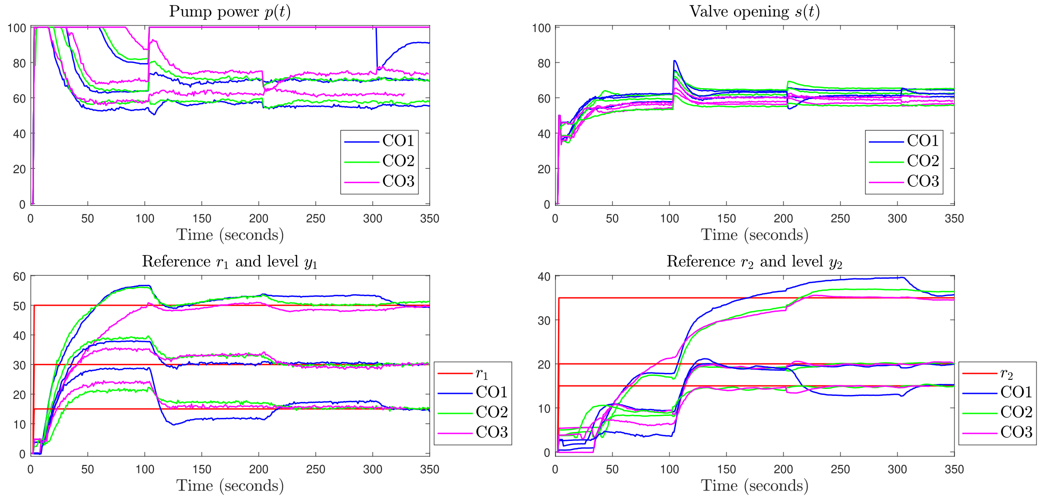

FIGURE 26. Adaptive control system results for references R_1, R_3 and R_7.

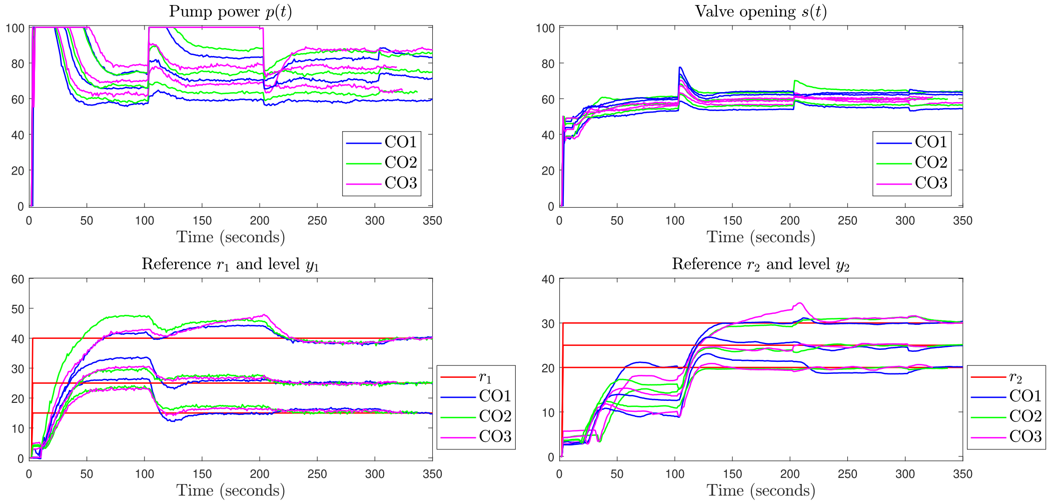

FIGURE 27. Adaptive control system results for references R_2, R_6 and R_8.

different controller configurations and references. In these results, it can be seen that the best value of J_1, J_2 and J is obtained for configuration CO2.

Considering the results of Table 10 and taking as a reference the total best value obtained for J, the difference of the configuration CO2 in relation to CO1 is 11.5% and compared

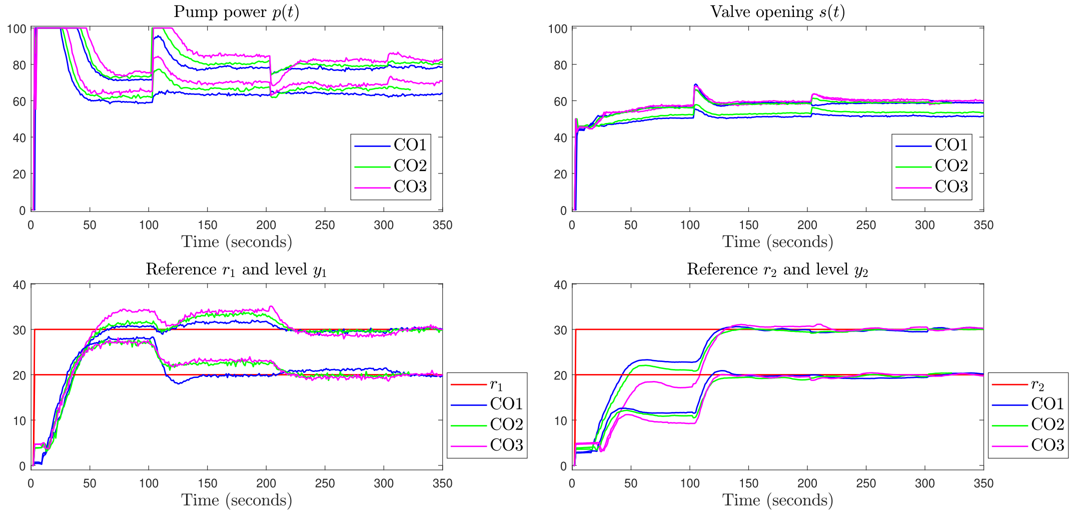

FIGURE 28. Adaptive control system results for references R_4 and R_5.

TABLE 8. Results obtained for J_1 on a scale of 1×10^4.

Configuration	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	Total
CO1	4.0317	3.0330	1.5658	1.8523	0.9244	1.1357	1.6300	1.2699	15.443
CO2	3.4896	2.5275	1.5308	1.7981	0.8279	0.7300	0.5484	1.0653	12.518
CO3	5.6787	2.8151	1.3731	1.6940	0.8448	0.6466	0.7030	1.1250	14.880

TABLE 9. Results obtained for J_2 on a scale of 1×10^4.

Configuration	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	Total
CO1	7.3540	3.1550	1.7838	2.2425	1.2314	1.6137	1.5872	2.4866	21.454
CO2	6.8454	4.3796	1.9408	2.4945	1.3056	1.2068	0.5229	1.8525	20.548
CO3	7.2573	3.6674	1.6123	3.4419	1.5170	1.4623	0.8969	2.2196	22.075

TABLE 10. Results obtained for J on a scale of 1×10^5.

Configuration	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	Total
CO1	1.1386	0.6188	0.3350	0.4095	0.2156	0.2749	0.3217	0.3757	3.6897
CO2	1.0335	0.6907	0.3472	0.4293	0.2133	0.1937	0.1071	0.2918	3.3066
CO3	1.2936	0.6482	0.2985	0.5136	0.2362	0.2109	0.1600	0.3345	3.6955

to CO3 is 11.7%. Also, in Table 10 it is seen that CO1 presents the best results for R_2, and R_4; the CO2 configuration obtains the best result for R_1, R_5, R_6, R_7, and R_8, meanwhile, with CO3 the best result is achieved for R_3.

IX. CONCLUSION

The proposed neuro-fuzzy MIMO control scheme allows iterative adjustment of the controller to reach the reference values. For this, progressive adjustments of the plant identification are also made.

In this work, BBR subsystems are proposed considering the analogy with linear systems used for the implementation of MIMO configurations. In this way, a MIMO system is designed for the identification of the plant and another for the controller; in addition, the training equations (DBP) for these systems are deduced.

The identification of the plant is an essential aspect for the functioning of the adaptive control system; therefore, a progressive adjustment is made for the plant identification and the controller optimization.

As observed, a limited amount of data is available for the identification of the plant obtained during the operation of the control system, which influences the fit of the plant model. Consequently, the adjustment is carried out progressively in an adaptive way.

By the experimental results it is possible to verify that the proposed adaptive scheme allows reaching the desired reference values. It is observed that in the first adaptation process the largest variation occurs.

The experimental test was carried out considering various configurations of the MIMO controller, based on BBR subsystems. In future work, other possible configurations of these subsystems used to implemented the MIMO controller can be considered.

In further work, compact fuzzy systems can be used in other control strategies such as sliding mode control and passivity-based control, where the adaptability of the controller parameters could be included.

ACKNOWLEDGMENT

A recognition for the Department of Systems Engineering and Automation, Universidad de Oviedo, and the Engineering Faculty, Universidad Distrital Francisco José de Caldas.

REFERENCES

[1] R. Belohlavek, J. W. Dauben, and G. J. Klir, *Fuzzy Logic and Mathematics: A Historical Perspective*. London, U.K.: Oxford Univ. Press, 2017.

[2] H. T. Nguyen, N. R. Prasad, C. L. Walker, and E. A. Walker, *A First Course in Fuzzy and Neural Control*. London, U.K.: Chapman & Hall, 2003.

[3] H. Ibbi, ''Optimization of Tsukamoto fuzzy inference system using fuzzy grid partition,'' *Int. J. Comput. Sci. Netw.*, vol. 5, no. 5, pp. 786–791, 2016.

[4] J. E. Whitesitt, *Boolean Algebra and Its Applications* (Dover Books on Computer Science). New York, NY, USA: Dover, 2012.

[5] P. O. J. Kaltjob, *Control of Mechatronic Systems: Model-Driven Design and Implementation Guidelines*. Hoboken, NJ, USA: Wiley, 2021.

[6] H. E. Espitia, H. R. Chamorro, and J. J. Soriano, ''Fuzzy controller design using concretion based on Boolean relations (CBR),'' in *Proc. 12th UK Workshop Comput. Intell. (UKCI)*, 2012, pp. 1–8.

[7] H. Espitia, J. Soriano, I. Machón, and H. López, ''Design methodology for the implementation of fuzzy inference systems based on Boolean relations,'' *Electronics*, vol. 8, no. 11, p. 1243, Oct. 2019.

[8] H. Espitia, J. Soriano, I. Machón, and H. López, ''Compact fuzzy systems based on Boolean relations,'' *Appl. Sci.*, vol. 11, no. 4, p. 1793, Feb. 2021.

[9] R. J. Sternberg, *Adaptive Intelligence: Surviving and Thriving in Times of Uncertainty*. Cambridge, U.K.: Cambridge Univ. Press, 2021.

[10] K. Passino, *Biomimicry for Optimization, Control, and Automation*. London, U.K.: Springer-Verlag, 2005.

[11] I. D. Landau, R. Lozano, M. M'Saad, and A. Karimi, *Adaptive Control, Algorithms, Analysis and Applications*. London, U.K.: Springer-Verlag, 2011.

[12] P. Ioannou and J. Sun, *Robust Adaptive Control*. New York, NY, USA: Dover, 2012.

[13] H. Dhiman, D. Deb, V. Muresan, and V. Balas, ''Wake management in wind farms: An adaptive control approach,'' *Energies*, vol. 12, no. 7, p. 1247, Apr. 2019.

[14] Y. Zhang, G. Tao, M. Chen, L. Wen, and Z. Zhang, ''A matrix decomposition based adaptive control scheme for a class of MIMO non-canonical approximation systems,'' *Automatica*, vol. 103, pp. 490–502, May 2019.

[15] L. Wen, G. Tao, and B. Jiang, ''Adaptive compensation of persistent actuator failures of nonlinear systems,'' *Int. J. Adapt. Control Signal Process.*, vol. 35, no. 3, pp. 373–400, Mar. 2021.

[16] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence and Robustness* (Dover Books on Electrical Engineering Series). New York, NY, USA: Dover, 2011.

[17] N. T. Nguyen, *Model-Reference Adaptive Control: A Primer* (Advanced Textbooks in Control and Signal Processing). Cham, Switzerland: Springer, 2018.

[18] J. Cervantes, W. Yu, S. Salazar, and I. Chairez, ''Takagi–Sugeno dynamic neuro-fuzzy controller of uncertain nonlinear systems,'' *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1601–1615, Dec. 2017.

[19] T. C. Lin, C. H. Kuo, T. Y. Lee, and V. E. Balas, ''Adaptive fuzzy H^∞ tracking design of SISO uncertain nonlinear fractional order time-delay systems,'' *Nonlinear Dyn.*, vol. 69, pp. 1639–1650, Sep. 2012.

[20] M. J. Kochenderfer and T. A. Wheeler, *Algorithms for Optimization*. Cambridge, MA, USA: MIT Press, 2019.

[21] F. Sherwani, B. S. K. K. Ibrahim, and M. M. Asad, ''Hybridized classification algorithms for data classification applications: A review,'' *Egyptian Informat. J.*, Aug. 2020, doi: 10.1016/j.eij.2020.07.004.

[22] R. Qi, G. Tao, and B. Jiang, *Fuzzy System Identification and Adaptive Control* (Communications and Control Engineering). Springer, 2019.

[23] N. Siddique and H. Adeli, *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing*. Hoboken, NJ, USA: Wiley, 2013.

[24] P. Tooranjipour and R. Vatankhah, ''Adaptive critic-based quaternion neuro-fuzzy controller design with application to chaos control,'' *Appl. Soft Comput.*, vol. 70, pp. 622–632, Sep. 2018.

[25] C. Wu, J. Liu, X. Jing, H. Li, and L. Wu, ''Adaptive fuzzy control for nonlinear networked control systems,'' *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 8, pp. 2420–2430, Aug. 2017.

[26] S. Song, B. Zhang, J. Xia, and Z. Zhang, ''Adaptive backstepping hybrid fuzzy sliding mode control for uncertain fractional-order nonlinear systems based on finite-time scheme,'' *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 4, pp. 1559–1569, Apr. 2020.

[27] Y. Lu, ''Adaptive-fuzzy control compensation design for direct adaptive fuzzy control,'' *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 6, pp. 3222–3231, Dec. 2018.

[28] E. D. S. Oliveira, R. H. C. Takahashi, and W. M. Caminhas, ''Online neuro-fuzzy controller: Design for robust stability,'' *IEEE Access*, vol. 8, pp. 193768–193776, 2020.

[29] R. W. Soares, L. R. Barroso, and O. A. S. Al-Fahdawi, ''Response attenuation of cable-stayed bridge subjected to central US earthquakes using neuro-fuzzy and simple adaptive control,'' *Eng. Struct.*, vol. 203, Jan. 2020, Art. no. 109874.

[30] E. Essien, H. Ibrahim, M. Mehrandezh, and R. Idem, ''Adaptive neuro-fuzzy inference system (ANFIS)—Based model predictive control (MPC) for carbon dioxide reforming of methane (CDRM) in a plug flow tubular reactor for hydrogen production,'' *Thermal Sci. Eng. Prog.*, vol. 9, pp. 148–161, Mar. 2019.

[31] M. J. Khodaei, M. H. Balaghi I., A. Mehrvarz, and N. Jalili, ''An adaptive multi-critic neuro-fuzzy control framework for intravenous anesthesia administration,'' *IFAC-PapersOnLine*, vol. 51, no. 34, pp. 202–207, 2019.

[32] F. G. Areed, A. Y. Haikal, and R. H. Mohammed, ''Adaptive neuro-fuzzy control of an induction motor,'' *Ain Shams Eng. J.*, vol. 1, no. 1, pp. 71–78, Sep. 2010.

[33] E. Kamal, L. Adouane, R. Abdrakhmanov, and N. Ouddah, ''Hierarchical and adaptive neuro-fuzzy control for intelligent energy management in hybrid electric vehicles,'' *IFAC PapersOnLine*, vols. 1–50, pp. 3014–3021, Jul. 2017.

[34] G. V. Lakhekar, L. M. Waghmare, P. G. Jadhav, and R. G. Roy, ''Robust diving motion control of an autonomous underwater vehicle using adaptive neuro-fuzzy sliding mode technique,'' *IEEE Access*, vol. 8, pp. 109891–109904, 2020.

[35] S. Alavandar and M. J. Nigam, ''New hybrid adaptive neuro-fuzzy algorithms for manipulator control with uncertainties–comparative study,'' *ISA Trans.*, vol. 48, no. 4, pp. 497–502, Oct. 2009.

[36] E. Kayacan, E. Kayacan, H. Ramon, and W. Saeys, ''Adaptive neuro-fuzzy control of a spherical rolling robot using sliding-mode-control-theory-based online learning algorithm,'' *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 170–179, Feb. 2013.

[37] D. M. Atia and H. T. El-Madany, ''Analysis and design of greenhouse temperature control using adaptive neuro-fuzzy inference system,'' *J. Electr. Syst. Inf. Technol.*, vol. 4, no. 1, pp. 34–48, May 2017.

[38] S. Mohamed and I. A. Hameed, ''A GA-based adaptive neuro-fuzzy controller for greenhouse climate control system,'' *Alexandria Eng. J.*, vol. 57, no. 2, pp. 773–779, Jun. 2018.

[39] S. Song, B. Zhang, X. Song, Y. Zhang, Z. Zhang, and W. Li, ''Fractional-order adaptive neuro-fuzzy sliding mode H_∞ control for fuzzy singularly perturbed systems,'' *J. Franklin Inst.*, vol. 356, pp. 5027–5048, Jul. 2019.

[40] T.-C. Lin, T.-Y. Lee, and V. E. Balas, ''Adaptive fuzzy sliding mode control for synchronization of uncertain fractional order chaotic systems,'' *Chaos, Solitons Fractals*, vol. 44, no. 10, pp. 791–801, Oct. 2011.

[41] S. Masumpoor, H. Yaghobi, and M. A. Khanesar, ''Adaptive sliding-mode type-2 neuro-fuzzy control of an induction motor,'' *Expert Syst. Appl.*, vol. 42, no. 19, pp. 6635–6647, Nov. 2015.

[42] S. P. Moustakidis, G. A. Rovithakis, and J. B. Theocharis, ''An adaptive neuro-fuzzy tracking control for multi-input nonlinear dynamic systems,'' *Automatica*, vol. 44, no. 5, pp. 1418–1425, May 2008.

[43] P. Sarhadi, B. Rezaie, and Z. Rahmani, ''Adaptive predictive control based on adaptive neuro-fuzzy inference system for a class of nonlinear industrial processes,'' *J. Taiwan Inst. Chem. Engineers*, vol. 61, pp. 132–137, Apr. 2016.

[44] S. Song, B. Zhang, X. Song, and Z. Zhang, ''Adaptive neuro-fuzzy backstepping dynamic surface control for uncertain fractional-order nonlinear systems,'' *Neurocomputing*, vol. 360, pp. 172–184, Sep. 2019.

[45] M. Abdelrahman and S.-Y. Park, "Spacecraft attitude control via a combined state-dependent Riccati equation and adaptive neuro-fuzzy approach," *Aerosp. Sci. Technol.*, vol. 26, no. 1, pp. 16–28, Apr. 2013.

[46] V. Hosseinnezhad, M. Shafie-Khah, P. Siano, and J. P. S. Catalao, "An optimal home energy management paradigm with an adaptive neuro-fuzzy regulation," *IEEE Access*, vol. 8, pp. 19614–19628, 2020.

[47] M. Ahmed, A. Vahidnia, M. Datta, and L. Meegahapola, "An adaptive power oscillation damping controller for a hybrid AC/DC microgrid," *IEEE Access*, vol. 8, pp. 69482–69495, 2020.

[48] H. E. Espitia, I. Machón-González, H. López-García, and G. Díaz, "Proposal of an adaptive neurofuzzy system to control flow power in distributed generation systems," *Complexity*, vol. 2019, pp. 1–16, Mar. 2019.

[49] M. H. E. Balaghi, R. Vatankhah, M. Broushaki, and A. Alasty, "Adaptive optimal multi-critic based neuro-fuzzy control of MIMO human musculoskeletal arm model," *Neurocomputing*, vol. 173, pp. 1529–1537, Jan. 2016.

[50] M. M. Saafan, M. M. Abdelsalam, M. S. Elksas, S. F. Saraya, and F. F. G. Areed, "An adaptive neuro-fuzzy sliding mode controller for MIMO systems with disturbance," *Chin. J. Chem. Eng.*, vol. 25, no. 4, pp. 463–476, Apr. 2017.

[51] V. Nekoukar and A. Erfanian, "Adaptive fuzzy terminal sliding mode control for a class of MIMO uncertain nonlinear systems," *Fuzzy Sets Syst.*, vol. 179, no. 1, pp. 34–49, Sep. 2011.

[52] P. Gil, T. Oliveira, and L. Palma, "Adaptive neuro–fuzzy control for discrete-time nonaffine nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 8, pp. 1602–1615, Aug. 2019.

[53] I. Machón-González, H. López-García, and I. Bocos-Barranco, "Dynamics identification and control of nonlinear MIMO coupled plant using supervised neural gas and comparison with recurrent neural controller," *Neural Comput. Appl.*, vol. 32, no. 24, pp. 18123–18142, Dec. 2020.

[54] D. H. Shah and D. M. Patel, "Design of sliding mode control for quadruple-tank MIMO process with time delay compensation," *J. Process Control*, vol. 76, pp. 46–61, Apr. 2019.

[55] P. Navrátil, L. Pekař, and R. Matušů, "Control of a multivariable system using optimal control pairs: A quadruple-tank process," *IEEE Access*, vol. 8, pp. 2537–2563, 2020.

[56] *Global Optimization Toolbox User's Guide*, MathWorks, Natick, MA, USA, 2015.

[57] R. Patel, D. Deb, R. Dey, and V. E. Balas, "Introduction to adaptive control," in *Adaptive and Intelligent Control of Microbial Fuel Cells* (Intelligent Systems Reference Library), vol. 161. Cham, Switzerland: Springer, 2020.

[58] G. Tao and G. Song, "Higher order tracking properties of model reference adaptive control systems," *IEEE Trans. Autom. Control*, vol. 63, no. 11, pp. 3912–3918, Nov. 2018.

[59] K. L. Du and M. N. S. Swamy, *Neural Networks and Statistical Learning*. London, U.K.: Springer, 2013.

[60] H. T. Nguyen and E. A. Walker, *A First Course in Fuzzy Logic*, 3rd ed. London, U.K.: Chapman & Hall, 2006.

[61] M. Singh, I. Singh, and A. Verma, "Identification on non linear series-parallel model using neural network," *Int. J. Electr. Instrum. Eng.*, vol. 3, no. 1, pp. 21–23, 2013.

[62] A. Alshejari, "Neuro-fuzzy based intelligent approaches to nonlinear system identification and forecasting," Ph.D. dissertation, Univ. Westminster, London, U.K., 2018.

HELBERT ESPITIA received the B.S. degree in electronic engineering from the Universidad Distrital Francisco José de Caldas, Bogotá, Colombia, the B.S. degree in mechatronics engineering from the Universidad Nacional de Colombia, the M.Sc. degree in industrial engineering from the Universidad Distrital Francisco José de Caldas, and the M.Sc. degree in mechanical engineering and the Ph.D. degree in systems engineering and computer science from the Universidad Nacional de Colombia. He is currently a Professor with the Faculty of Systems Engineering, Universidad Distrital Francisco José de Caldas. His research interests are in the areas of control, intelligent systems, and cybernetics.

IVÁN MACHÓN received the B.S. and Ph.D. degrees in industrial engineering from the Universidad de Oviedo, Gijón, Spain. He is currently a Professor with the Department of Electric Engineering, Electronic of Computers and Systems, Universidad de Oviedo. His research interests are in the areas of nonlinear control and intelligent systems.

HILARIO LÓPEZ received the B.S. and Ph.D. degrees in industrial engineering from the Universidad de Oviedo, Gijón, Spain. He is currently a Professor with the Department of Electric Engineering, Electronic of Computers and Systems, Universidad de Oviedo. His research interests are in the areas of nonlinear control and intelligent systems.