

Article

Design and Optimization of a Neuro-Fuzzy System for the Control of an Electromechanical Plant

Helbert Espitia ^{1,*}, Iván Machón ² and Hilario López ²¹ Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas, Bogotá 11021-110231588, Colombia² Departamento de Ingeniería Eléctrica, Electrónica de Computadores y Sistemas, Universidad de Oviedo, Campus de Viesques, 33204 Gijón, Spain; machonivan@uniovi.es (I.M.); hilario@uniovi.es (H.L.)

* Correspondence: heespitiac@udistrital.edu.co

Abstract: One characteristic of neuro-fuzzy systems is the possibility of incorporating preliminary information in their structure as well as being able to establish an initial configuration to carry out the training. In this regard, the strategy to establish the configuration of the fuzzy system is a relevant aspect. This document displays the design and implementation of a neuro-fuzzy controller based on Boolean relations to regulate the angular position in an electromechanical plant, composed by a motor coupled to inertia with friction (a widely studied plant that serves to show the control system design process). The structure of fuzzy systems based on Boolean relations considers the operation of sensors and actuators present in the control system. In this way, the initial configuration of fuzzy controller can be determined. In order to perform the optimization of the neuro-fuzzy controller, the continuous plant model is converted to discrete time to be included in the closed-loop controller training equations. For the design process, first the optimization of a Proportional Integral (PI) linear controller is carried out. Thus, linear controller parameters are employed to establish the structure and initial configuration of the neuro-fuzzy controller. The optimization process also includes weighting factors for error and control action in such a way that allows having different system responses. Considering the structure of the control system, the optimization algorithm (training algorithm) employed is dynamic back propagation. The results via simulations show that optimization is achieved in the linear and neuro-fuzzy controllers using different weighting values for the error signal and control action. It is also observed that the proposed control strategy allows disturbance rejection.

Keywords: controller; electromechanical plant; neuro-fuzzy; optimization



Citation: Espitia, H.; Machón, I.; López, H. Design and Optimization of a Neuro-Fuzzy System for the Control of an Electromechanical Plant. *Appl. Sci.* **2022**, *12*, 541. <https://doi.org/10.3390/app12020541>

Academic Editor: Ludmila Dymova

Received: 11 November 2021

Accepted: 5 January 2022

Published: 6 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In control systems, fuzzy logic offers broad applicability given its flexibility to implement control strategies through a set of rules using membership functions [1–3]. It is possible to use training algorithms with neural networks when representing the fuzzy system as a neural network that grants higher accuracy in the fuzzy system performance. When representing a fuzzy system as a neural network, a neuro-fuzzy system is obtained [4,5].

An interesting application of fuzzy logic is the aided design process. In this regard, a remarkable application is observed for compliant mechanism design used in soft robotics, space, and bioengineering due to the advantages of free friction, monolithic structure, and minimal assembly. Considering this area, a hybrid methodology is presented in [6] for solving the multi-objective optimization design. Authors propose a hybridization through a combination of a finite element method, statistical technique, desirability function approach, fuzzy logic system, Adaptive Neuro-Fuzzy Inference System (ANFIS), and Lightning Attachment Procedure Optimization (LAPO). A bistable compliant mechanism is designed, where desirability values of two performance metrics are calculated and transferred into the fuzzy logic system to obtain the output in a single objective function. A related work

is presented in [7], proposing a method using statistics, numerical simulation, computational intelligence, and metaheuristics. A two-degree of freedom compliant mechanism is designed, where numerical datasets are collected by simulations. The sensitivity of design parameters is studied by analysis of variance and Taguchi technique. The results of sensitivity are employed to separate populations for lightning attachment procedure optimization. The values of two output mechanism performances are the inputs of the fuzzy logic model, and the output of this system is a single objective function, that is optimized using LAPO algorithm. Other advanced work is presented in [8], presenting an optimization framework to provide a systematic design method for a flexure gripper. The optimization strategy includes the topology, modeling, and size optimization phases. The gripper topology is determined via the solids isotropic material with penalization method considering stress constraint and force distribution. The modeling of the performances is implemented via an Enhanced Adaptive Neuro-Fuzzy Inference System (EANFIS). The optimization is performed to search for the best model parameters. Finally, a hybrid procedure of the Taguchi method is described in [9], using fuzzy logic, response surface method, and Moth-flame optimization algorithm to solve the design optimization of a flexure hinge to enhance three quality characteristics. Fuzzy modeling is proposed to interpolate the three objective functions into a unified objective function. An integrated regression equation is determined via response surface method and flexure hinge is optimized using Moth-flame optimization algorithm.

Regarding fuzzy control applications for electromechanical systems, there are relevant developments in electric engines control and robotics. With regard to fuzzy logic for motor control, reference [10] carries out a review of multi-motor control, which are multiple input and multiple output systems. The main purpose of the multiple motor control variators consists of achieving a coordinated operation in all motors. The literature reviewed shows that fuzzy systems are widely employed in multi-motor control. Meanwhile, in [11] a fuzzy controller is proposed using a motion capture system to operate the speed and orientation of a wheelchair considering the model of a Direct Current (DC) motor for the design process. Further, a fuzzy logic controller is used in [12], for an image-based navigation system of an autonomous underwater vehicle with a DC motor.

According to [13], due to high robustness and simple maintenance, Induction Motors (IM) are commonly used in household appliances and industries. Currently, advanced techniques are applied to traditional controls such as the Field-Oriented Control (FOC) and Direct Torque Control (DTC). Authors in [13] evaluate traditional FOC and DTC compared to two additional fuzzy and predictive controllers.

Concerning robotic applications using fuzzy logic, reference [14] employs a fuzzy control Proportional Integral (PI) in a robotic driving system for hybrid electric vehicles in a way that allows a reliable evaluation of energy efficiency. This depicts the relevance of the efficiency of fuel in hybrid vehicles. Thus, reliable measurements are required in order to have an accurate analysis of fuel efficiency; whereby, the authors propose a robotic system to perform driving tests to reduce the deviation when such tests are carried out by humans. Moreover, reference [15] proposes a Knowledge-Based Neural Fuzzy Controller (KNFC) for navigation control in mobile robots. Knowledge-Based Cultural Multi-Strategy Differential Evolution (KCMDE) is employed here to adjust the parameters of the KNFC. This is applied in mobile robots PIONEER 3-DX for automatic navigation systems and obstacle avoidance, employing the existing angle between the obstacle and the robot to determine if the robot enters in specific reference points to modify its behavior and thus skip dead ends. A related work is presented in [16] using a neuro-fuzzy system in path planning for a service robot in complex environments with the presence of humans as in a retirement home. As inputs for path planning, the system uses a 3D depth camera and multiple sonar sensors.

Meanwhile, reference [17] proposes a simple and practical decoupled control structure for a parallel robot positioning control using the Cable-Driven Parallel Robot (CDPR). This structure is employed with a non-linear Proportional Integral Derivative (PID) and classic

PID controllers. In addition, the control system structure is proposed through an analysis of cables involved in End-Effector (EE) robot's movement when operating independently for each axis; later, an adjustment of rules for fuzzy PID controllers was carried out and the Ziegler–Nichols method was applied to PID classic controllers.

Regarding recent research of neuro-fuzzy control systems, in [18] is developed a Terminal Sliding-Mode Control (TSMC) using a Fuzzy Double Hidden Layer Recurrent Neural Network (FDHLRNN). According to authors, the neuro-fuzzy system proposed is a combination of a radial basis neural network and a fuzzy neural network. To reduce the switching gain (associated with the TSMC), the FDHLRNN is employed to approximate the nonlinear sliding-mode equivalent control term. In order to observe the dynamic response and robustness, the control scheme is tested using a second-order nonlinear system. An additional work is observed in [19], where is designed a TSMC using FDHLRNN for a single-phase Active Power Filter (APF). Authors propose the TSMC to allow the tracking error converges to zero (in a finite time). The system is used in harmonic suppression first to approximate the equivalent control and second to suppress unknown disturbances. The equations to update the adaptive parameters of the FDHLRNN are obtained using a Liapunov function in order to ensure asymptotic stability. Finally, in [20] is displayed an approximation-based adaptive fractional sliding mode control. For approximating the system uncertainties and disturbance we used a Double Loop Recurrent Fuzzy Neural Network (DLRFNN). In this development, a fractional order term is incorporated into the sliding surface having the benefits of sliding mode control and fractional calculus. The DLRFNN uses the advantages of fuzzy systems to handle uncertain information and the neural networks to learn using data measured from the process.

In this work, the designed neuro-fuzzy controller uses the structure of a compact fuzzy system based on Boolean relations as presented in [21], which can be used for the identification and control of dynamic systems. Neuro-fuzzy systems based on Boolean relations are founded in a system design considering possible Boolean regions in the universe discourse, then being extended using fuzzy sets for implementation [22]. Various design considerations are presented in this scenario as well as the manipulation of system equations applying Boole and Kleene algebras. In this way, fuzzy systems with compact structures can be established and used for identification and control of dynamic systems [21]. As an example, the application of this type of system is shown in [23] to control an Automatic Voltage Regulator (AVR). In control applications, the Dynamic Back Propagation (DBP) training algorithm is used in cases when the plant's feedback signals and the controller are required for the neuro-fuzzy control system.

This document details the design and tuning of a neuro-fuzzy control system applied to an electromechanical plant. The neuro-fuzzy system design is made regarding the architecture of a linear controller. Subsequently, the training is performed similar to a neural network employing the dynamic back propagation algorithm [24–26].

The design can be applied in robotics since it is necessary to control the movements of the parts of a robot which are generally activated with DC motors. Another application can be a solar tracker usually employed to improve the use of solar energy. In previous works, the considered plant is similar to that employed to control a solar tracker mainly operated by a DC motor [27–29]. Regarding this plant, reference [30] carries out the optimization of a linear controller in discrete time.

Article Approach and Document Organization

The objective of this document is to show the design and optimization process of a fuzzy controller (based on Boolean relations), making the equivalence with a linear controller, for which a well-known plant (widely studied) is used to show the design process and optimization. Considering this work, the design process can be extended to more complex plants. Moreover, this document considers the design of a linear controller as reference since it allows determining the structure of the fuzzy controller as well as its initial configuration, which also allows the optimization using the dynamic back propagation

algorithm where the equations in closed-loop are deduced to train the linear and fuzzy controllers. The values from the linear controller are used to determine the initial configuration to optimize the fuzzy controller. A well-known plant composed of a coupled motor with a gear box and inertia with friction is employed to exhibit the design process and training of the controller. From the above, Figure 1 displays the methodology employed. First, the system modeling and analysis is performed, then the linear controller design used as the basis for designing the fuzzy controller (as mentioned above) is conducted. Later, the optimization of the controllers is carried out and finally the simulation of the closed-loop control system is performed.

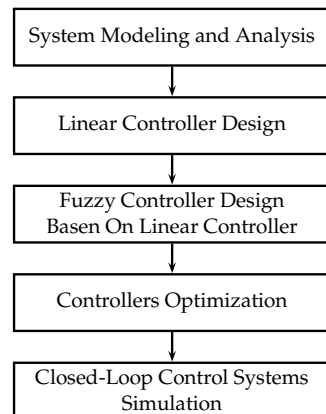


Figure 1. Methodology employed.

The document is organized as follows: Section 2 describes the plant model considering a simplified structure and the representation in discrete time. Then, in Section 3 the description is shown of the linear and fuzzy controllers employed. Section 4 displays the general process for controller parameter training. Section 5 describes the equations for linear controller parameter training and Section 6 the equations for fuzzy controller optimization. Then, Section 7 presents the results of the controller training process. Finally, Section 8 shows the discussion and conclusions are presented in Section 9.

2. Electromechanical Plant

Figure 2 displays the diagram of the electromechanical plant considered. This is composed of a DC motor, a gearbox, and a wheel (with moment of inertia) coupled to the gearbox. The analysis of this plant is presented in [31], corresponding to the example A-3-9 pages 95–97. The motor plays an important role in this plant since it is a fundamental part of the dynamic model, as it provides the required torque to move the inertia. The parameters of the DC motor are:

- K_t : Torque constant.
- K_e : Electromagnetic feedback constant.
- R_a : Shield resistance.
- L_a : Shield inductance.
- J_r : Motor moment of inertia.
- B_r : Rotor friction constant.

It is noteworthy that the diagram in Figure 2 is given as an example since in a real case the gearbox has more gears. The motor torque is proportional to the current of the shield for the modeling of the system as indicated in the following equation:

$$T_M = K_t i_a(t) \quad (1)$$

The counter-electromotive force is proportional to the angular speed of the motor:

$$e(t) = K_e \omega_r(t) \quad (2)$$

The equation that governs the electrical part is:

$$u(t) - e(t) = L_a \frac{di_a(t)}{dt} + R_a i_a(t) \tag{3}$$

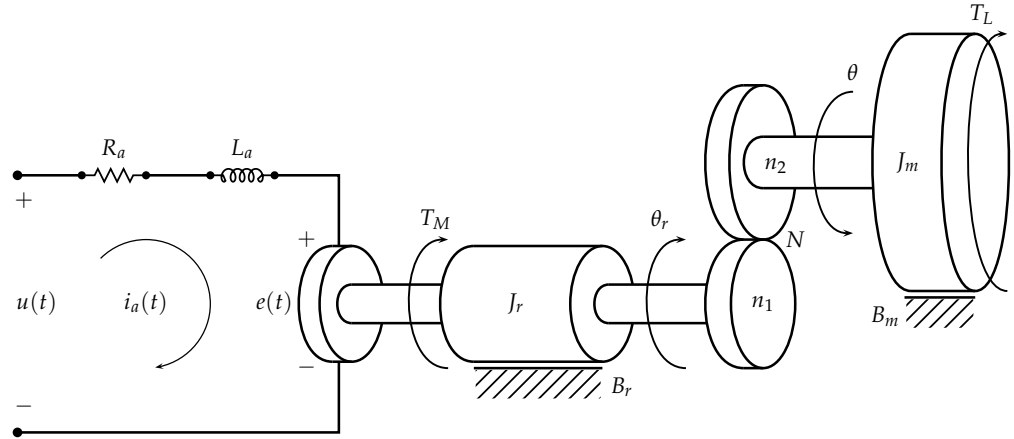


Figure 2. Electromechanical plant schematic diagram.

The torques in the motor shaft are considered to incorporate the mechanical part into the model. The relation between the generated torque and the load torque is obtained by performing a summation of torques on the motor shaft as shown below:

$$T_M(t) - N T_L(t) = J_e \frac{d\omega_r(t)}{dt} + B_e \omega_r(t) \tag{4}$$

where:

$$B_e = B_r + B_m N^2; \quad J_e = J_r + J_m N^2; \quad N = \frac{n_1}{n_2} \tag{5}$$

In these equations J_m, B_m are the inertia and friction of the mechanical part and N the transformation ratio of the gearbox such that $\theta(t) = N\theta_r(t)$. In addition, Equation (6) is employed to complete the model.

$$\omega_r = \frac{d\theta_r(t)}{dt} \tag{6}$$

2.1. System Block Diagram

In order to observe the presentation of this system as a transfer function, a Laplace transform \mathcal{L} is applied, obtaining:

$$T_M(s) = K_t I_a(s) \tag{7}$$

$$E(s) = K_e \Omega_r(s) \tag{8}$$

$$U(s) - E(s) = (R_a + sL_a) I_a(s) \tag{9}$$

$$T_M(s) - N T_L(s) = (B_e + sJ_e) \Omega_r(s) \tag{10}$$

$$\Theta_r(s) = \frac{\Omega_r(s)}{s} \tag{11}$$

In addition, taking $T_L = 0$ and $\Theta(s) = N\Theta_r(s)$ the system transfer function is:

$$\frac{\Theta(s)}{U(s)} = \frac{N K_t}{K_t K_e s + (L_a s + R_a)(J_e s + B_e)} \tag{12}$$

Considering the model equations, the block diagram in Figure 3 of the electromechanical system can be obtained.

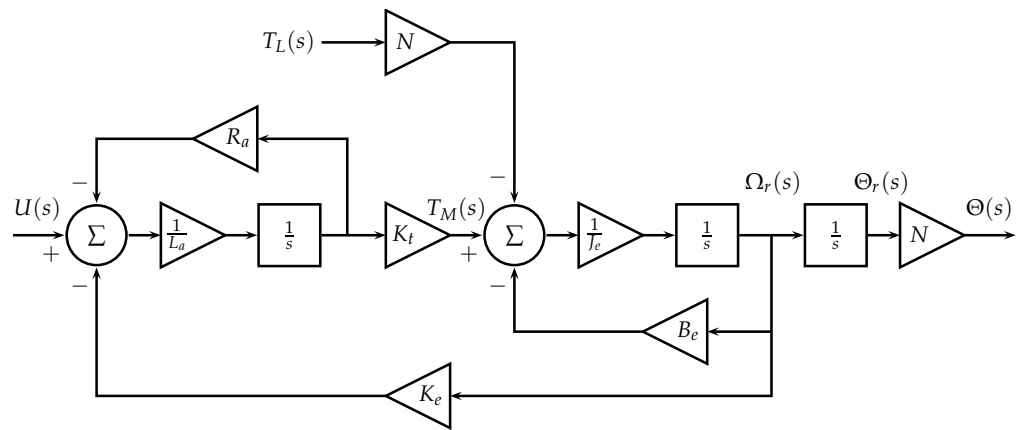


Figure 3. Electromechanical system block diagram.

2.2. Approximate Model

In some applications, a complete dynamic model of the plant may be impractical due to the difficulty in establishing the values of its parameter (in this case, the moment of inertia and the friction of the mechanical part). An alternative consists of a simplified model identifying its parameters considering the dynamic behavior of the plant [31].

According to [31], when considering $L_a \approx 0$ (very small) there is a simplified model of the plant; in this way, the diagram in Figure 3 can be represented as shown in Figure 4.

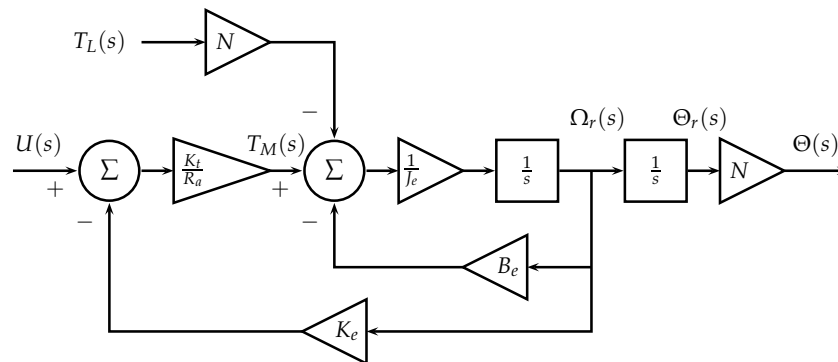


Figure 4. Block diagram of the electromechanical system considering $L_a \approx 0$.

Organizing the diagram of Figure 4, it is possible to obtain the representation of Figure 5 where $T_L(s)$ can be seen as a system disturbance.

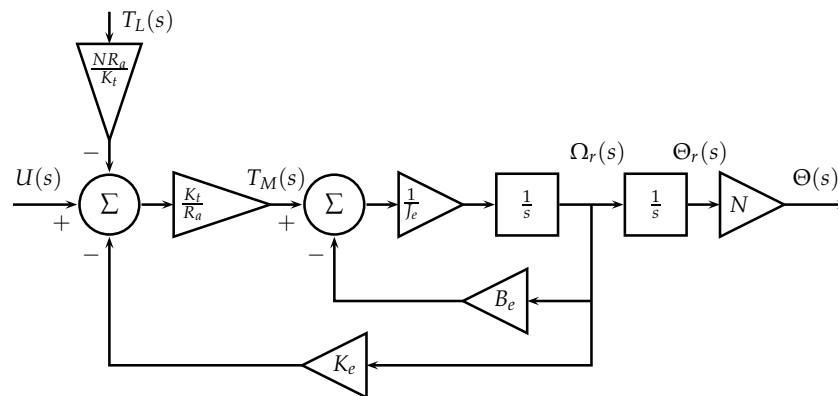


Figure 5. Simplified block diagram of the electromechanical system.

Equivalently, the diagram shown in Figure 6 is obtained where in the time domain the input $u(t) = \mathcal{L}^{-1}\{U(s)\}$ corresponds to the voltage; the output is the angular position $\theta(t) = \mathcal{L}^{-1}\{\Theta(s)\}$ and $p(t) = \mathcal{L}^{-1}\{P(s)\}$ is the disturbance given by the product

$(NR_a/K_t)T_L(t)$. In this way, the variable to sense and control corresponds to the angular position $\theta(t)$.

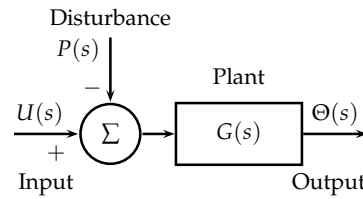


Figure 6. Equivalent block diagram of the plant.

On the other hand, considering the location of the poles, the criterion to establish the simplified model is shown in Figure 7 where the pole located in $s = -a$ corresponding to the mechanical part is the dominant pole (it governs the system dynamics) since $b \gg a$; therefore, to establish the approximate model, the pole located at $s = -b$ associated with the electromagnetic part can be neglected [31].

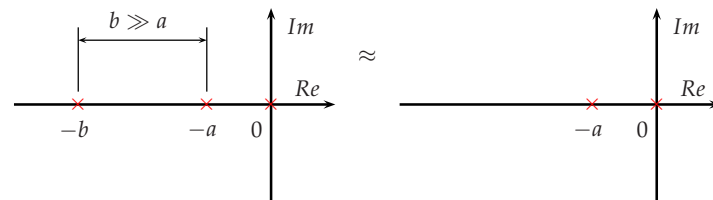


Figure 7. Consideration of pole location.

Equation (13) describes the simplified plant transfer function corresponding to a first-order system and an integrator, where the variable s is associated with the Laplace transform.

According to [31], the model of Equation (13) considers the parameters shown in Figure 5, namely $J_e, B_e, K_e, K_t, N,$ and R_a . In this way, through the data measured from the open-loop plant, the effect of the parameters on the plant operation is being directly considered in the identification process.

$$G(s) = \frac{K}{s(\tau s + 1)} \tag{13}$$

For this plant, the identification of parameters is carried out considering the average values of the speed and the settling time as shown in [27,28].

For the transfer function, in Equation (13) parameter K corresponds to the steady-state input–output relationship, which for this case is $K = \Delta\omega/\Delta u$ where ω is the angular velocity. Taking a voltage of 12 V, the average time taken by the system to travel an angle of $\pi/2$ is 0.06 rad/s; therefore, K parameter value corresponds to 0.005 rad/(V s) [28].

Parameter τ can be determined as one third of the system settling time (considering a band of 5%), which is 1.13 s, hence, τ corresponds to 0.38 s [28]. In this way, the plant model is given by Equation (14). It should be noted that the proposed control system is designed to regulate the angular position of the electromechanical system $\theta(t)$.

$$G(s) = \frac{\Theta(s)}{U(s)} = \frac{0.005}{s(0.38s + 1)} \tag{14}$$

2.3. Discrete Time Model

Since the control strategy to be implemented is PI, a continuous time PI controller is designed to establish the sampling time. Accordingly, the closed-loop system bandwidth ω_{bw} can be used to determine the sampling time.

The recommendations given in [32–34] are applied to design the PI controller. The continuous PI controller is given by Equation (15).

$$C(s) = K \frac{s + a}{s} \tag{15}$$

According to [34], the design procedure consists of locating the zero $s = -a$ close to the integrator pole so that the pole-zero pair of the feedback system “almost cancel”. As the objective in this section is to determine the sampling time, the goal is to have the highest bandwidth with an overshoot lower than 22% (as a criterion of the system response) to design the PI controller. Thus, using the root locus, it is established that $a = 0.132$ and $K = 492$. Figure 8 presents the root locus, the step response, and the Bode diagram of the closed-loop system with the continuous time PI control.

Thus, in Figure 8 it is observed that the bandwidth of the closed-loop system is $\omega_{bw} = 3.23$ rad/s. According to [35], the sample rate is usually taken between 10 and 20 times the bandwidth. For the design 20 times are taken; therefore, the sampling frequency is $\omega_s = 20\omega_{bw}$ and thus the sampling time $T_s = 2\pi/\omega_s$ is $T_s \approx 0.1$ s.

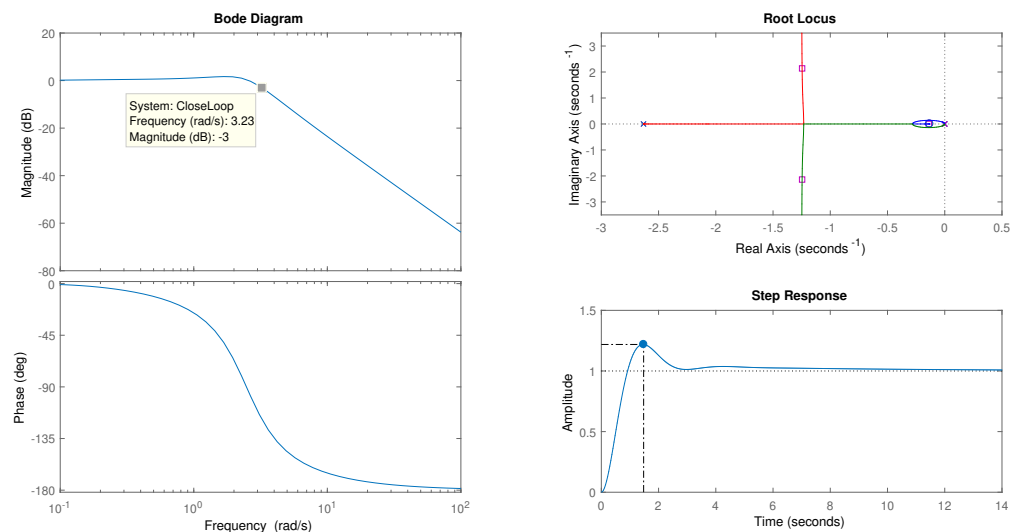


Figure 8. System characteristics with PI control.

The transfer function $G(s)$ given by Equation (14) is converted to $G(z)$ (discrete time system) using a sampling time of $T_s = 0.1$ s and the bilinear transformation method [30]. In this way, the model of the discrete time system is obtained given by Equation (16).

$$G(z) = 10^{-5} \frac{2.907 + 5.814z^{-1} + 2.907z^{-2}}{1 - 1.767z^{-1} + 0.767z^{-2}} \tag{16}$$

In general, this transfer function can be represented as:

$$G(z) = \frac{\Theta(z)}{U(z)} = \frac{B_0 + B_1z^{-1} + B_2z^{-2}}{1 + A_1z^{-1} + A_2z^{-2}} \tag{17}$$

The respective difference equation for this plant is:

$$\theta[n] = B_0u[n] + B_1u[n - 1] + B_2u[n - 2] - A_1\theta[n - 1] - A_2\theta[n - 2] \tag{18}$$

3. Fuzzy Control System

The neuro-fuzzy controller uses the structure of a compact fuzzy system based on Boolean relations such as that presented in [21], which can be used for the identification and control of dynamic systems. Fuzzy systems based on Boolean relations are based on a

system design considering possible Boolean regions in the discourse universes; then this system is extended using fuzzy sets for implementation, making it possible to consider the operation of sensors and actuators present in the control system [22]. The output of these systems can be calculated as:

$$y = \sum_{i=1}^N v_i Y_i \tag{19}$$

where v_i corresponds to the virtual actuator and Y_i the activation output that depends on the membership functions $\mu_{l,j}(x_l)$ associated with inputs x_l for $l = 1, 2, \dots, L$ and $j = 1, 2, \dots, M_l$.

In order to establish the activation functions Y_i , a truth table and various design considerations are used as well as the manipulation of the system equations using Boolean and Kleene algebras [22]. By using a compact structure, direct equivalence can be performed with the structure of a dynamic discrete time system, which allows the implementation of fuzzy controllers [21].

In this way, fuzzy control design consists of modifying (extending) a linear (discrete time) controller using fuzzy sets to model the controller non-linear relations. This design principle is used in [23] to control an automatic voltage regulator. The general diagram of the control system can be seen in Figure 9.

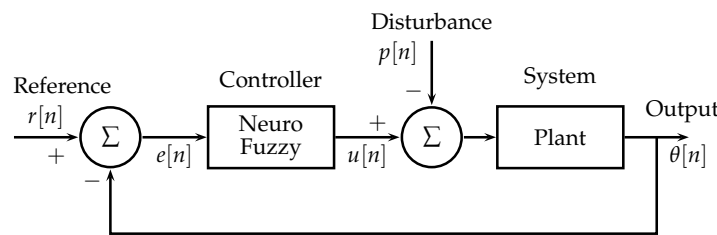


Figure 9. General diagram of the control system.

The fuzzy controller schema consists of modifying the non-linear relations that exist for the input and feedback signals. To carry out the design of the neuro-fuzzy controller, a linear PI (Proportional Integral) controller given by Equation (20) is considered.

$$C(s) = K_p + \frac{K_i}{s} = \frac{sK_p + K_i}{s} \tag{20}$$

Considering the controller in discrete time, the respective transfer function of the PI controller is:

$$C(z) = \frac{U(z)}{E(z)} = \frac{b_0 - b_1 z^{-1}}{1 - z^{-1}} \tag{21}$$

The difference equation of this controller corresponds to:

$$u[n] = b_0 e[n] - b_1 e[n - 1] + u[n - 1] \tag{22}$$

where the respective coefficients b_0, b_1 are constant. For the fuzzy controller, these constants are replaced by non-linear relations given by fuzzy membership functions, such that:

$$u[n] = f_1(e[n]) - f_2(e[n - 1]) + f_3(u[n - 1]) \tag{23}$$

Fuzzy sets shown in Figures 10 and 11 are employed for the fuzzy control system. Particularly, Figure 10 shows a sigmoidal fuzzy set to model positive values of the universe of discourse e and u , while in Figure 11 the negative values are represented for the error e and control action u .

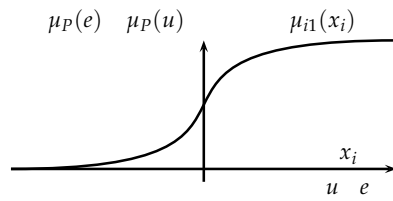


Figure 10. Membership function to model positive values.

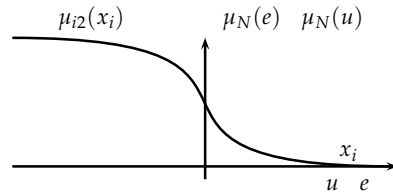


Figure 11. Membership function to model negative values.

Considering the fuzzy sets described in Figures 10 and 11, as well as the controller structure given by Equation (23), the scheme of Figure 12 is obtained, where the proposed fuzzy controller is shown.

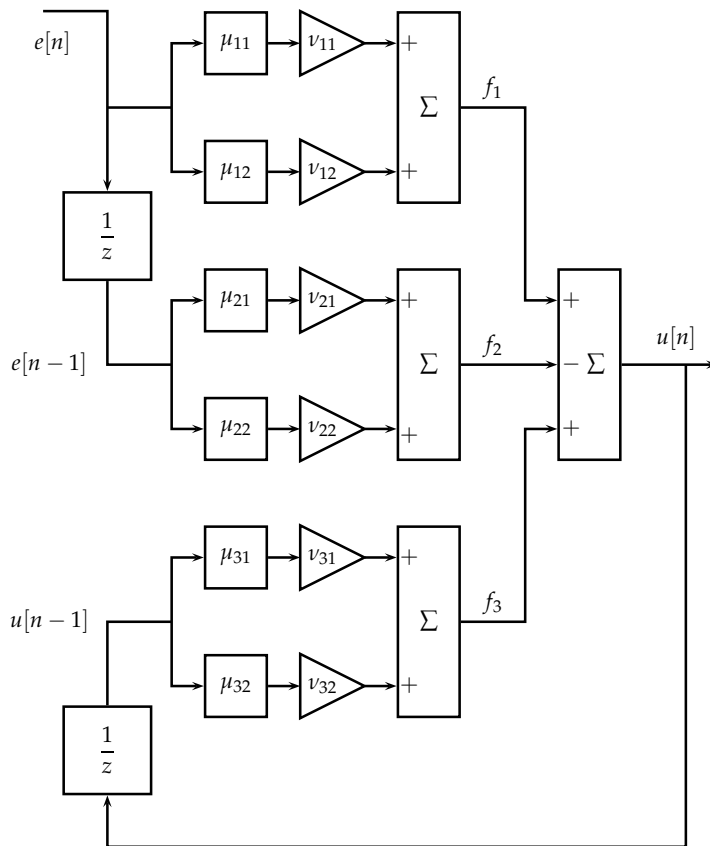


Figure 12. Neuro-fuzzy control system scheme.

The controller output can be calculated with Equation (24) where the inputs to the fuzzy system are $x_i \in \{e[n], e[n - 1], u[n - 1]\}$.

$$u[n] = \sum_{i=1}^3 \sum_{j=1}^2 v_{ij} \mu_{ij}(x_i) = \sum_{i=1}^3 f_i(x_i) \tag{24}$$

For each input x_i it is possible to define a function f_i as presented in Equation (25).

$$f_i = v_{i1}\mu_{i1}(x_i) + v_{i2}\mu_{i2}(x_i) \tag{25}$$

To carry out the respective calculations, the membership function is given by equation $\mu_{ij}(x_i) = (1 + e^{-\rho_{ij}(x_i - \chi_{ij})})^{-1}$; in this way, the set of controller parameters is $\mathbf{H}_c = \{v_{ij}, \rho_{ij}, \chi_{ij}\}$.

4. Controller Parameter Training

The structure of the closed-loop neuro-fuzzy controller can be assimilated as a recurrent neural network; according to [36], there are different alternatives to perform the training process. One way is to calculate the respective recurrent equation that updates each parameter using the respective fitness function derivatives. In this way, the system simulation will be carried out with the current parameters without modification; in the same simulation the adjustment of parameters is carried out in auxiliary variables, since these parameters are not used in the current simulation. When finishing the simulation, the system parameters are updated with those stored in the auxiliary variables and a new simulation is carried out until the value of the fitness function is reduced.

To carry out the controller training, first the structure of the plant and the controller are established; thus, for a closed-loop system, the control action corresponds to:

$$u[n] = f_c(e[n], e[n - 1], u[n - 1], \mathbf{H}_c) \tag{26}$$

Likewise, the system's output employed for controller training is:

$$\theta[n] = f_p(u[n], u[n - 1], \theta[n - 1], \theta[n - 2], \mathbf{H}_p) \tag{27}$$

where \mathbf{H}_c corresponds to the set of controller parameters and \mathbf{H}_p to the plant parameters. Considering h_c as one of the control system parameters, the adaptation of the controller parameters is carried out using Equation (28).

$$h_c(k + 1) = h_c(k) - \eta \frac{dJ}{dh_c} \tag{28}$$

In Equation (28), parameter η is the learning rate and J corresponds to the adjustment function defined as:

$$J = \frac{1}{2} [Q(e[n])^2 + R(u[n])^2] \tag{29}$$

where $e[n] = r[n] - \theta[n]$ and the derivative of J depending on the adjustment parameters is given by Equation (30). Considering these equations, Figure 13 shows the algorithm employed for controller training.

$$\frac{dJ}{dh_c} = Qe[n] \frac{de}{dh_c} + Ru[n] \frac{du}{dh_c} \tag{30}$$

The first step consists of choosing the initial controller's parameter configuration; then, the next step involves the system output calculation using the model of the plant. Subsequently, the parameter's adjustment occurs employing the respective equations involved in the dynamics in the control system and the derivatives of the parameters (equations de/dh_c , du/dh_c and subsequent). It is noteworthy that during the simulation the adjusted parameters are stored in auxiliary variables since during this process these values are not used by the controller. To continue the simulation and parameter adjustment, the algorithm returns to the step where the control system is evaluated with $n = n + 1$, repeating this process until simulation time N_T is complete. Once the simulation time is completed, the parameters of the controller are updated with the optimized values to return to the evaluation step of the control system to start a new iteration $k = k + 1$ until

the objective function $J_T(k) = (1/N_T) \sum_{n=1}^{N_T} J(k, n)$ is less than a given ϵ or until k is equal to a given number K_T .

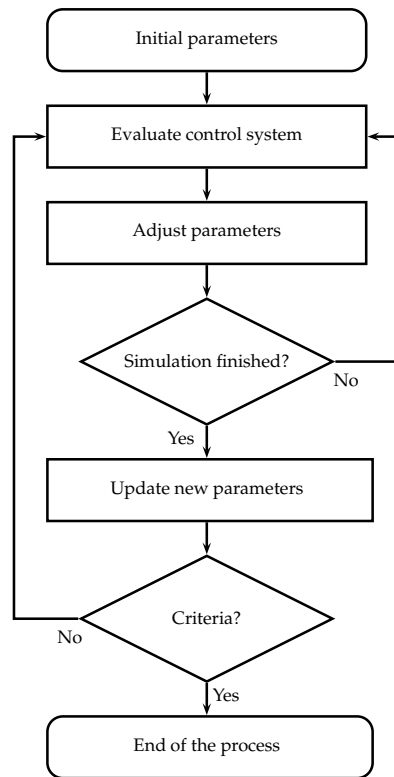


Figure 13. Algorithm for controller training.

5. Equations for Linear Controller Parameter Training

The implementation of the training algorithm (dynamic back propagation) uses difference equations of each parameter in the adaptation Equation (28). On aspects to bear in mind in [37,38], the stability analysis is presented for this kind of algorithm used for training recurrent neural networks.

In order to determine the equations for linear controller training, first, the error equation is $e[n] = r[n] - \theta[n]$ where the output of the plant is given by Equation (31).

$$\theta[n] = B_0u[n] + B_1u[n - 1] + B_2u[n - 2] - A_1\theta[n - 1] - A_2\theta[n - 2] \tag{31}$$

Secondly, the difference equation of the controller is:

$$u[n] = b_0e[n] - b_1e[n - 1] + u[n - 1] \tag{32}$$

Therefore, the training equations of the parameters $h_c \in \{b_0, b_1\}$ are the following:

$$\frac{de}{dh_c}[n] = -\frac{d\theta}{dh_c}[n] \tag{33}$$

$$\frac{d\theta}{dh_c}[n] = B_0\frac{du}{dh_c}[n] + B_1\frac{du}{dh_c}[n - 1] + B_2\frac{du}{dh_c}[n - 2] - A_1\frac{d\theta}{dh_c}[n - 1] - A_2\frac{d\theta}{dh_c}[n - 2] \tag{34}$$

The derivative equation of $u[n]$ based on the controller parameter b_0 is:

$$\begin{aligned} \frac{du}{db_0}[n] &= \frac{d}{db_0}(b_0e[n]) - b_1\frac{de}{db_0}[n - 1] + \frac{du}{db_0}[n - 1] \\ &= b_0\frac{de}{db_0}[n] - b_1\frac{de}{db_0}[n - 1] + \frac{du}{db_0}[n - 1] + e[n] \end{aligned} \tag{35}$$

Therefore, for the parameter b_1 the equation is:

$$\begin{aligned} \frac{du}{db_1}[n] &= b_0 \frac{de}{db_1}[n] - \frac{d}{db_1}(b_1 e[n-1]) + \frac{du}{db_1}[n-1] \\ &= b_0 \frac{de}{db_1}[n] - b_1 \frac{de}{db_1}[n-1] + \frac{du}{db_1}[n-1] - e[n-1] \end{aligned} \tag{36}$$

In this way, using a learning rate η the following equation is used to update the parameters:

$$h_c(k+1) = h_c(k) - \eta \left[Qe[n] \frac{de[n]}{dh_c} + Ru[n] \frac{du[n]}{dh_c} \right] \tag{37}$$

6. Equations for Fuzzy Controller Parameter Training

In this case, the training equations are established for the fuzzy controller based on Boolean relations; in this order, the equation of the plant is:

$$\theta[n] = B_0 u[n] + B_1 u[n-1] + B_2 u[n-2] - A_1 \theta[n-1] - A_2 \theta[n-2] \tag{38}$$

Considering the controller architecture (Figure 12) and the expression for f_i given by Equation (25), the controller dynamics is given by:

$$u[n] = f_1(e[n]) - f_2(e[n-1]) + f_3(u[n-1]) \tag{39}$$

Additionally,

$$e[n] = r[n] - \theta[n] \tag{40}$$

is used to calculate the error variable.

The respective derivative of the plant output $\theta[n]$ for a controller parameter $h_{ij} \in \mathbf{H}_c$ is:

$$\frac{d\theta}{dh_{ij}}[n] = B_0 \frac{du}{dh_{ij}}[n] + B_1 \frac{du}{dh_{ij}}[n-1] + B_2 \frac{du}{dh_{ij}}[n-2] - A_1 \frac{d\theta}{dh_{ij}}[n-1] - A_2 \frac{d\theta}{dh_{ij}}[n-2] \tag{41}$$

In the same way, the derivative of the control action $u[n]$ with respect to h_{ij} is:

$$\frac{du[n]}{dh_{ij}} = \frac{df_1(e[n])}{dh_{ij}} - \frac{df_2(e[n-1])}{dh_{ij}} + \frac{df_3(u[n-1])}{dh_{ij}} \tag{42}$$

Meanwhile, the derivative of the error $e[n]$ for a controller parameter h_{ij} corresponds to:

$$\frac{de[n]}{dh_{ij}} = \frac{dr[n]}{dh_{ij}} - \frac{d\theta[n]}{dh_{ij}} = - \frac{d\theta[n]}{dh_{ij}} \tag{43}$$

To determine the respective derivatives, it is considered that:

$$\frac{df_l(x_l)}{dh_{ij}} = \frac{d}{dh_{ij}}(v_{l1}\mu_{l1}(x_l)) + \frac{d}{dh_{ij}}(v_{l2}\mu_{l2}(x_l)) \tag{44}$$

where $l = 1, \dots, 3, i = 1, \dots, 3$ and $j = 1, 2$; therefore, there are different cases for values of i and l . According to the values of i, l and j the possible cases are:

- Case: $l \neq i$, where f_l does not depend directly on h_{ij} .
- Case: $l = i$ and $j = 1$, where f_i directly depends on h_{i1} .
- Case: $l = i$ and $j = 2$, where f_i directly depends on h_{i2} .

In the case when $l \neq i$ it is obtained:

$$\frac{df_l(x_l)}{dh_{ij}} = \frac{d}{dx_l}(v_{l1}\mu_{l1}(x_l)) \frac{dx_l}{dh_{ij}} + \frac{d}{dx_l}(v_{l2}\mu_{l2}(x_l)) \frac{dx_l}{dh_{ij}} \tag{45}$$

$$\frac{df_l(x_l)}{dh_{ij}} = \left[\frac{d}{dx_l}(v_{l1}\mu_{l1}(x_l)) + \frac{d}{dx_l}(v_{l2}\mu_{l2}(x_l)) \right] \frac{dx_l}{dh_{ij}} \tag{46}$$

For $m = 1, 2$ the respective derivatives are:

$$\frac{d}{dx_l}(v_{lm}\mu_{lm}(x_l)) = \frac{d}{dx_l} \left(v_{lm} \left(1 + e^{-\rho_{lm}(x_l - \chi_{lm})} \right)^{-1} \right) \tag{47}$$

$$\frac{d}{dx_l}(v_{lm}\mu_{lm}(x_l)) = L_{lm} = v_{lm} \left(1 + e^{-\rho_{lm}(x_l - \chi_{lm})} \right)^{-2} e^{-\rho_{lm}(x_l - \chi_{lm})} \rho_{lm} \tag{48}$$

Meanwhile, when $l = i$ and $j = 1$ it is established that:

$$\frac{df_i(x_i)}{dh_{i1}} = \frac{d}{dh_{i1}}(v_{i1}\mu_{i1}(x_i)) + \frac{d}{dx_i}(v_{i2}\mu_{i2}(x_i)) \frac{dx_i}{dh_{i1}} \tag{49}$$

Now in the case where $l = i$ and $j = 2$ it is presented that:

$$\frac{df_i(x_i)}{dh_{i2}} = \frac{d}{dx_i}(v_{i1}\mu_{i1}(x_i)) \frac{dx_i}{dh_{i2}} + \frac{d}{dh_{i2}}(v_{i2}\mu_{i2}(x_i)) \tag{50}$$

In the first place, for the derivatives with respect to x_i is obtained:

$$\frac{d}{dx_i}(v_{ij}\mu_{ij}(x_i)) = L_{ij} = v_{ij} \left(1 + e^{-\rho_{ij}(x_i - \chi_{ij})} \right)^{-2} e^{-\rho_{ij}(x_i - \chi_{ij})} \rho_{ij} \tag{51}$$

Second, for the derivatives with respect to h_{ij} it has:

$$\frac{d}{dh_{ij}}(v_{ij}\mu_{ij}(x_i)) = \frac{d}{dh_{ij}} \left(v_{ij} \left(1 + e^{-\rho_{ij}(x_i - \chi_{ij})} \right)^{-1} \right) \tag{52}$$

In this way, for the respective parameters v_{ij} , ρ_{ij} and χ_{ij} it is obtained that:

$$\frac{d}{dv_{ij}}(v_{ij}\mu_{ij}(x_i)) = \left(1 + e^{-\rho_{ij}(x_i - \chi_{ij})} \right)^{-1} + v_{ij} \left(1 + e^{-\rho_{ij}(x_i - \chi_{ij})} \right)^{-2} e^{-\rho_{ij}(x_i - \chi_{ij})} \rho_{ij} \frac{dx_i}{dv_{ij}} \tag{53}$$

$$\frac{d}{d\rho_{ij}}(v_{ij}\mu_{ij}(x_i)) = -v_{ij} \left(1 + e^{-\rho_{ij}(x_i - \chi_{ij})} \right)^{-2} e^{-\rho_{ij}(x_i - \chi_{ij})} \left(\chi_{ij} - x_i - \rho_{ij} \frac{dx_i}{d\rho_{ij}} \right) \tag{54}$$

$$\frac{d}{d\chi_{ij}}(v_{ij}\mu_{ij}(x_i)) = -v_{ij} \left(1 + e^{-\rho_{ij}(x_i - \chi_{ij})} \right)^{-2} e^{-\rho_{ij}(x_i - \chi_{ij})} \left(\rho_{ij} - \rho_{ij} \frac{dx_i}{d\chi_{ij}} \right) \tag{55}$$

In general terms for h_{ij} these equations can be written as:

$$\frac{d}{dh_{ij}}(v_{ij}\mu_{ij}(x_i)) = T_{h_{ij}} + C_{ij} \frac{dx_i}{dh_{ij}} \tag{56}$$

Considering the parameters v_{ij} , ρ_{ij} and χ_{ij} it has:

$$\frac{d}{dv_{ij}}(v_{ij}\mu_{ij}(x_i)) = T_{v_{ij}} + C_{ij} \frac{dx_i}{dv_{ij}} \tag{57}$$

$$\frac{d}{d\rho_{ij}}(v_{ij}\mu_{ij}(x_i)) = T_{\rho_{ij}} + C_{ij} \frac{dx_i}{d\rho_{ij}} \tag{58}$$

$$\frac{d}{d\chi_{ij}}(v_{ij}\mu_{ij}(x_i)) = T_{\chi_{ij}} + C_{ij} \frac{dx_i}{d\chi_{ij}} \tag{59}$$

where:

$$C_{ij} = v_{ij} \left(1 + e^{-\rho_{ij}(x_i - \chi_{ij})}\right)^{-2} e^{-\rho_{ij}(x_i - \chi_{ij})} \rho_{ij} \tag{60}$$

$$T_{v_{ij}} = \left(1 + e^{-\rho_{ij}(x_i - \chi_{ij})}\right)^{-1} \tag{61}$$

$$T_{\rho_{ij}} = -v_{ij} \left(1 + e^{-\rho_{ij}(x_i - \chi_{ij})}\right)^{-2} e^{-\rho_{ij}(x_i - \chi_{ij})} (\chi_{ij} - x_i) \tag{62}$$

$$T_{\chi_{ij}} = -v_{ij} \left(1 + e^{-\rho_{ij}(x_i - \chi_{ij})}\right)^{-2} e^{-\rho_{ij}(x_i - \chi_{ij})} \rho_{ij} \tag{63}$$

Considering the above for the Equation (46) (case $l \neq i$), it can be written in the form:

$$\begin{aligned} \frac{df_l(x_l)}{dh_{ij}} &= \left[\frac{d}{dx_l} (v_{l1}\mu_{l1}(x_l)) + \frac{d}{dx_l} (v_{l2}\mu_{l2}(x_l)) \right] \frac{dx_l}{dh_{ij}} \\ &= [L_{l1} + L_{l2}] \frac{dx_l}{dh_{ij}} \end{aligned} \tag{64}$$

On the other hand, for Equation (49) (case $l = i, j = 1$) the following representation is stated:

$$\begin{aligned} \frac{df_i(x_i)}{dh_{i1}} &= \frac{d}{dh_{i1}} (v_{i1}\mu_{i1}(x_i)) + \frac{d}{dx_i} (v_{i2}\mu_{i2}(x_i)) \frac{dx_i}{dh_{i1}} \\ &= T_{h_{i1}} + C_{i1} \frac{dx_i}{dh_{i1}} + L_{i2} \frac{dx_i}{dh_{i1}} \\ &= T_{h_{i1}} + [C_{i1} + L_{i2}] \frac{dx_i}{dh_{i1}} \end{aligned} \tag{65}$$

Meanwhile for the Equation (50) (case $l = i, j = 2$) it is obtained:

$$\begin{aligned} \frac{df_i(x_i)}{dh_{i2}} &= \frac{d}{dx_i} (v_{i1}\mu_{i1}(x_i)) \frac{dx_i}{dh_{i2}} + \frac{d}{dh_{i2}} (v_{i2}\mu_{i2}(x_i)) \\ &= L_{i1} \frac{dx_i}{dh_{i2}} + T_{h_{i2}} + C_{i2} \frac{dx_i}{dh_{i2}} \\ &= T_{h_{i2}} + [L_{i1} + C_{i2}] \frac{dx_i}{dh_{i2}} \end{aligned} \tag{66}$$

Finally, Equations (67)–(69) are employed to update the parameters, here η corresponds to the learning rate.

$$v_{ij}(k + 1) = v_{ij}(k) - \eta \left[Qe[n] \frac{de[n]}{dv_{ij}} + Ru[n] \frac{du[n]}{dv_{ij}} \right] \tag{67}$$

$$\rho_{ij}(k + 1) = \rho_{ij}(k) - \eta \left[Qe[n] \frac{de[n]}{d\rho_{ij}} + Ru[n] \frac{du[n]}{d\rho_{ij}} \right] \tag{68}$$

$$\chi_{ij}(k + 1) = \chi_{ij}(k) - \eta \left[Qe[n] \frac{de[n]}{d\chi_{ij}} + Ru[n] \frac{du[n]}{d\chi_{ij}} \right] \tag{69}$$

7. Results

Using the equations described in Sections 5 and 6, the respective optimization of the control system is carried out for both the linear and fuzzy controllers. The settings for the Q and R values are as follows:

- QR1: $Q = 1, R = 0.00005$.
- QR2: $Q = 1, R = 0.0001$.

- QR3: $Q = 1, R = 0.0005$.
- QR4: $Q = 1, R = 0.001$.

After the training process Figure 14 shows the results when considering different reference values using the PI discrete linear controller.

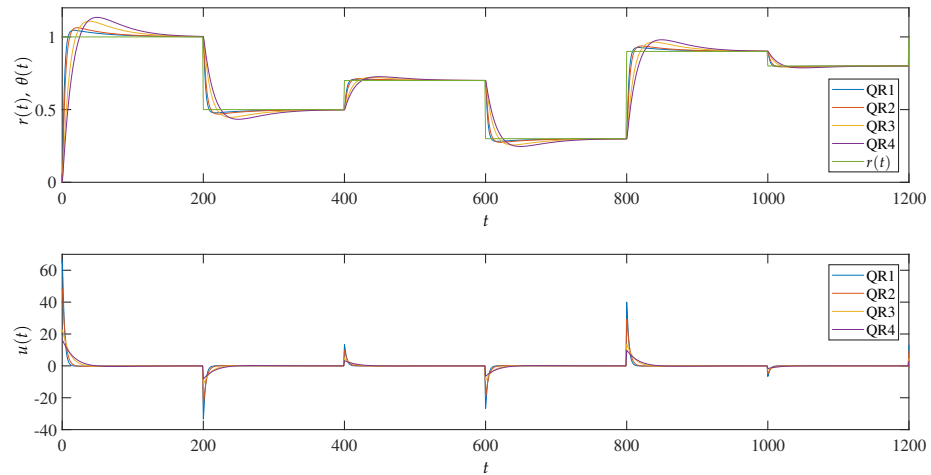


Figure 14. System response with the linear controller.

The transfer functions of the PI controller obtained to Q and R values are the following:

- QR1:

$$C(z) = \frac{66.79z - 66.67}{z - 1} = \frac{66.79(z - 0.9982)}{(z - 1)}$$

- QR2:

$$C(z) = \frac{48.48z - 48.39}{z - 1} = \frac{48.48(z - 0.9981)}{(z - 1)}$$

- QR3:

$$C(z) = \frac{22.66z - 22.62}{z - 1} = \frac{22.66(z - 0.9982)}{(z - 1)}$$

- QR4:

$$C(z) = \frac{16.11z - 16.08}{z - 1} = \frac{16.11(z - 0.9981)}{(z - 1)}$$

The stability analysis for linear controller consists of determining the closed-loop poles of the control system (controller and plant). In this order, if the poles are in the unit circle the system is stable. For each case the transfer function $T(z)$ in a closed loop is:

- QR1:

$$T(z) = \frac{0.001942z^3 + 0.001945z^2 - 0.001935z - 0.001938}{1.002z^3 - 2.765z^2 + 2.532z - 0.7689}$$

- QR2:

$$T(z) = \frac{0.001409z^3 + 0.001412z^2 - 0.001404z - 0.001407}{1.001z^3 - 2.766z^2 + 2.533z - 0.7684}$$

- QR3:

$$T(z) = \frac{0.0006587z^3 + 0.0006599z^2 - 0.0006564z - 0.0006576}{1.001z^3 - 2.766z^2 + 2.533z - 0.7677}$$

- QR4:

$$T(z) = \frac{0.0004683z^3 + 0.0004692z^2 - 0.0004666z - 0.0004674}{z^3 - 2.767z^2 + 2.534z - 0.7675}$$

The respective pole localization is presented in Table 1. It is observed that the poles are within the unit circle. Therefore, all the systems in a closed loop are stable.

Table 1. Pole localization for each QR case.

QR1	QR2	QR3	QR4
0.9981	0.9980	0.9978	0.9971
0.9639	0.9756	0.9905	0.9947
0.7978	0.7881	0.7762	0.7734

Additionally, as observed, the controllers display the zero near to $z = 0.998$, while the gain K has the values 66.79, 48.48, 22.66 and 16.11. Moreover, when incrementing the value of R (which weighs the action control), the gain value of the K controller is reduced. Furthermore, it is seen that the algorithm establishes the zero near the pole, therefore the pole and zero “almost cancel” which is consistent with reference [34] for the PI controller design.

As results analysis, Figure 15 shows the root locus for the PI control system. This figure also displays the movement of the roots as the gain K increases.

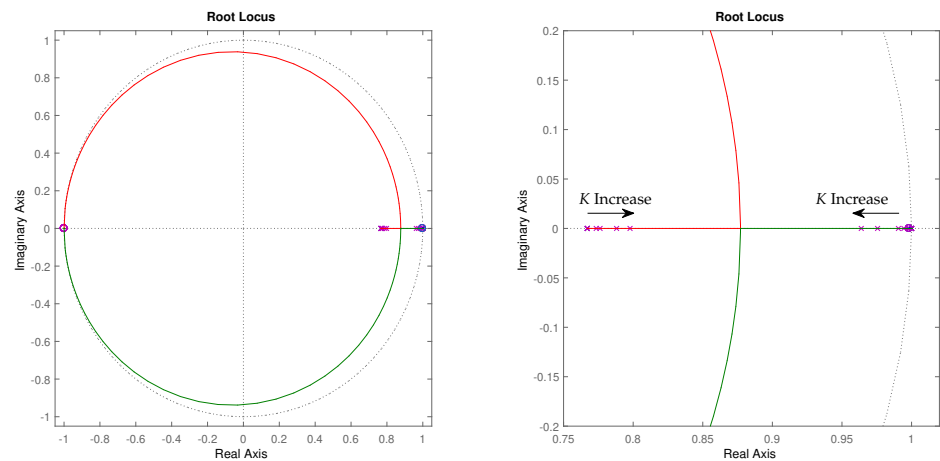


Figure 15. Root locus for the linear controllers obtained.

Employing configurations of functions $f_i(x_i)$ shown in Figure 16, it is possible to use the values obtained from the optimization of the linear controller to define an initial configuration to optimize the fuzzy controller. Thus, the slope of f_i in the linear region is set to be equivalent with each parameter of the linear controller.

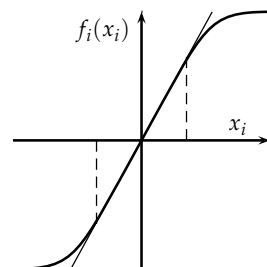


Figure 16. Configuration employed for $f_i(x_i)$.

The fuzzy controller parameters obtained, with which the simulation of the closed-loop system can be performed, are shown in Table 2. As can be seen, there are different parameters considering the QR case. In this table, i corresponds to rows and j to columns.

Table 2. Fuzzy controller parameters for QR cases.

Fuzzy controller parameters for QR1 case					
v_{ij}		ρ_{ij}		χ_{ij}	
-1.3358×10^4	1.3358×10^4	-0.0101	0.0101	-0.3382×10^{-5}	-0.3382×10^{-5}
1.3334×10^4	-1.3334×10^4	-0.0099	0.0099	0.3446×10^{-5}	0.3446×10^{-5}
-0.0200×10^4	0.0200×10^4	-0.0100	0.0100	0	0
Fuzzy controller parameters for QR2 case					
v_{ij}		ρ_{ij}		χ_{ij}	
-9.6960×10^3	9.6960×10^3	-0.0100	0.0100	-0.1066×10^{-5}	-0.1066×10^{-5}
9.6780×10^3	-9.6780×10^3	-0.0100	0.0100	0.1064×10^{-5}	0.1064×10^{-5}
-0.2000×10^3	0.2000×10^3	-0.0100	0.0100	0	0
Fuzzy controller parameters for QR3 case					
v_{ij}		ρ_{ij}		χ_{ij}	
-4.5320×10^3	4.5320×10^3	-0.0100	0.0100	0.1086×10^{-6}	0.1086×10^{-6}
4.5240×10^3	-4.5240×10^3	-0.0100	0.0100	-0.1081×10^{-6}	-0.1081×10^{-6}
-0.2000×10^3	0.2000×10^3	-0.0100	0.0100	0	0
Fuzzy controller parameters for QR4 case					
v_{ij}		ρ_{ij}		χ_{ij}	
-3.2220×10^3	3.2220×10^3	-0.0100	0.0100	0.6994×10^{-7}	0.6994×10^{-7}
3.2160×10^3	-3.2160×10^3	-0.0100	0.0100	-0.6982×10^{-7}	-0.6982×10^{-7}
-0.2000×10^3	0.2000×10^3	-0.0100	0.0100	0	0

After performing the fuzzy controller training, Figure 17 shows the system response for different reference values, as well as the value of the control signal; additionally, Figure 18 shows in detail the system response for a unitary reference. According to simulations, the suitable configurations correspond to QR2 or QR3 considering the values of control action and settling time.

On the other hand, Table 3 presents the values obtained for J_T in the initial and the final iteration (30 iterations) when performing the optimization process. These results show that the value of the objective function J_T increases when increasing the value of R .

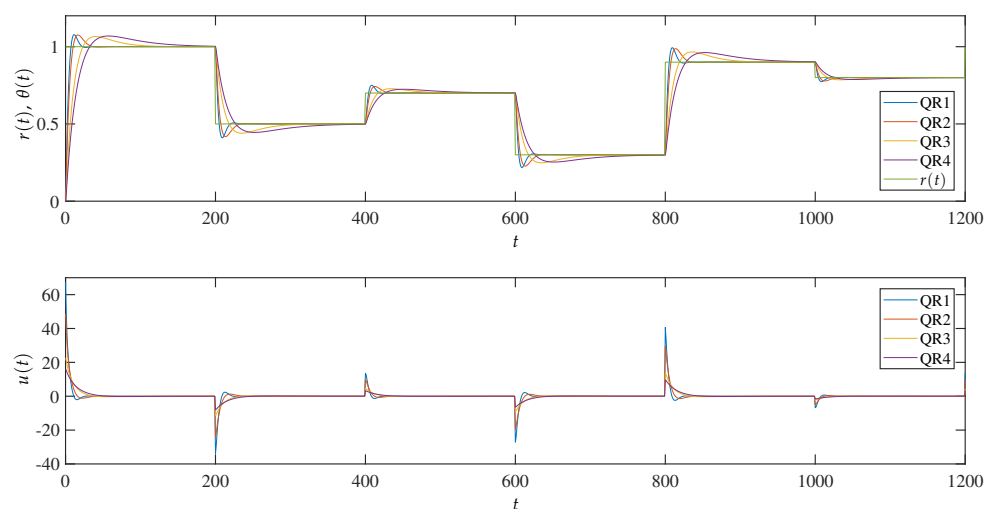


Figure 17. System response for different reference values.

In addition, Table 3 shows that linear and fuzzy controls obtain similar values of J_T . These results highlight that the design and choice of the initial configuration of the fuzzy controller were satisfactory since it presents a similar result to the linear controller.

To statistically observe the performance of the linear and fuzzy controllers for different values of Q and R 10 simulations were performed with randomly generated $r(t)$ reference values (six variations in each simulation). Results of J_T for 10 simulations are shown in Table 4, while Table 5 displays the mean, Standard Deviation (STD), maximum and minimum values.

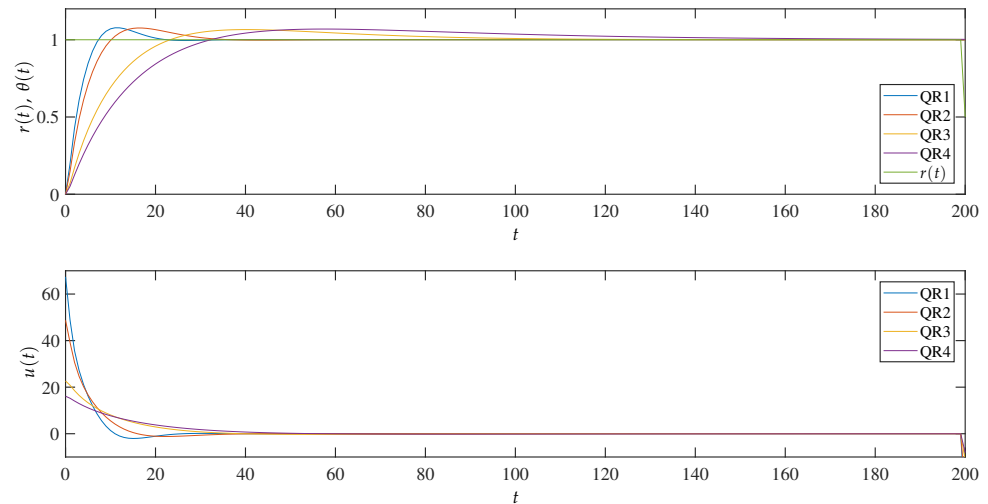


Figure 18. Control signal and response for a unitary reference.

Table 3. Initial and final values for J_T in each configuration.

QR	Linear		Fuzzy	
	Initial	Final	Initial	Final
QR1	0.0240293	0.0077754	0.0082463	0.0077858
QR2	0.0301406	0.0101258	0.0103760	0.0101451
QR3	0.1424850	0.0197969	0.0198750	0.0198434
QR4	0.4415530	0.0269572	0.0270372	0.0270370

Table 4. Results of J_T for 10 simulations.

Simulation	QR1		QR2		QR3		QR4	
	Linear	Fuzzy	Linear	Fuzzy	Linear	Fuzzy	Linear	Fuzzy
1	0.0032	0.0033	0.0044	0.0044	0.0092	0.0090	0.0130	0.0126
2	0.0018	0.0019	0.0024	0.0025	0.0051	0.0050	0.0071	0.0069
3	0.0043	0.0044	0.0058	0.0058	0.0122	0.0119	0.0172	0.0166
4	0.0032	0.0032	0.0043	0.0043	0.0090	0.0088	0.0127	0.0123
5	0.0046	0.0046	0.0062	0.0062	0.0130	0.0127	0.0184	0.0178
6	0.0017	0.0018	0.0023	0.0023	0.0048	0.0047	0.0067	0.0066
7	0.0018	0.0018	0.0024	0.0025	0.0051	0.0051	0.0073	0.0070
8	0.0009	0.0010	0.0013	0.0013	0.0026	0.0026	0.0037	0.0036
9	0.0009	0.0010	0.0012	0.0013	0.0026	0.0026	0.0037	0.0036
10	0.0025	0.0026	0.0034	0.0034	0.0069	0.0068	0.0097	0.0094

Table 5. Numerical measures of J_T for 10 simulations.

Numerical Measure	QR1		QR2		QR3		QR4	
	Linear	Fuzzy	Linear	Fuzzy	Linear	Fuzzy	Linear	Fuzzy
Mean	0.0025	0.0026	0.0034	0.0034	0.0070	0.0069	0.0099	0.0096
STD	0.0013	0.0013	0.0017	0.0017	0.0037	0.0036	0.0052	0.0050
Max	0.0046	0.0046	0.0062	0.0062	0.0130	0.0127	0.0184	0.0178
Min	0.0009	0.0010	0.0012	0.0013	0.0026	0.0026	0.0037	0.0036

Regarding the statistical tests application [39], the Kruskal–Wallis test is used to determine statistically significant differences between two or more experimental groups, obtaining a p value of 3.7814×10^{-7} , which indicates differences, then it the Bonferroni procedure is employed to perform multiple comparisons, the results are presented in Figure 19, where we observe no significant difference between the linear and fuzzy implementations for the Q and R configurations. We also observed a significant difference between cases QR1 and QR4.

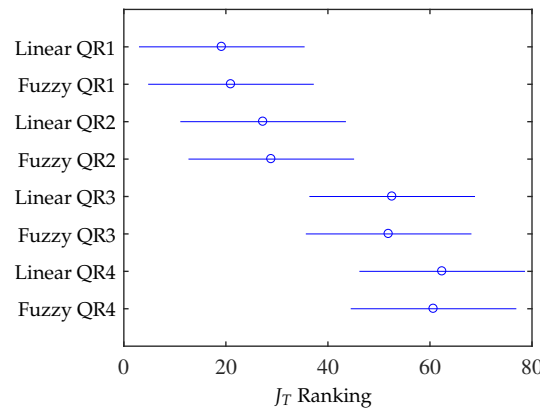


Figure 19. Multiple comparison using Bonferroni procedure.

Meanwhile, Figure 20 shows the system behavior for $\theta(t)$ when there are disturbances $p(t)$. This figure also displays the responses for different configurations in Q and R. It can be observed that the disturbance effect is more noticeable for the controllers that employ a small signal control, as in the cases QR3 and QR4 (according to Figures 17 and 18).

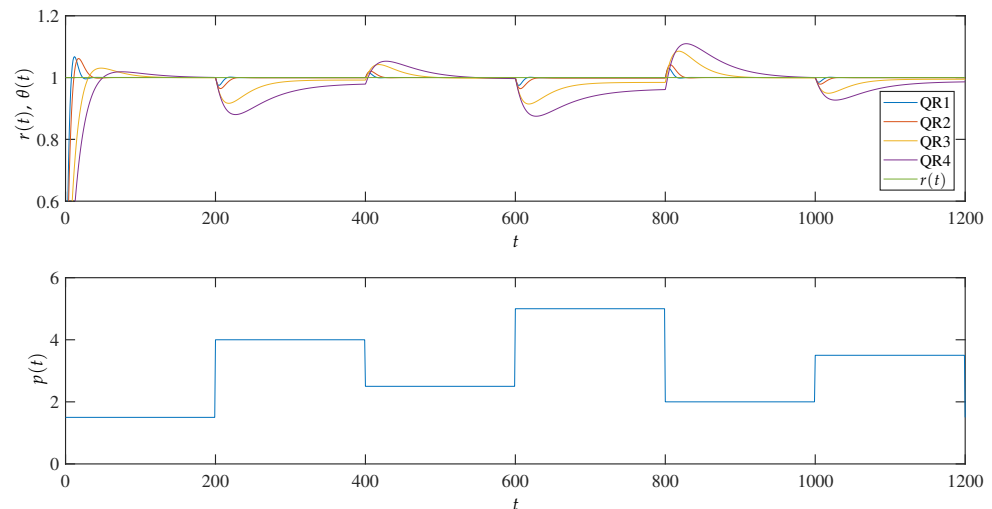


Figure 20. Response of the system for different disturbance values.

In Figure 21, the behavior comparison of the linear and fuzzy controllers is presented considering different perturbation values. For the disturbance values considered (Figure 18), some fuzzy controller configurations decrease the disturbance rejection capacity.

In order to statistically observe the performance of the linear and fuzzy controllers for disturbance rejection (considering different values of Q and R) 10 simulations were performed with randomly generated $p(t)$ disturbance values (six variations in each simulation). Results of J_T for 10 simulations are shown in Table 6, in this way, Table 7 displays the mean, standard deviation, and maximum and minimum values.

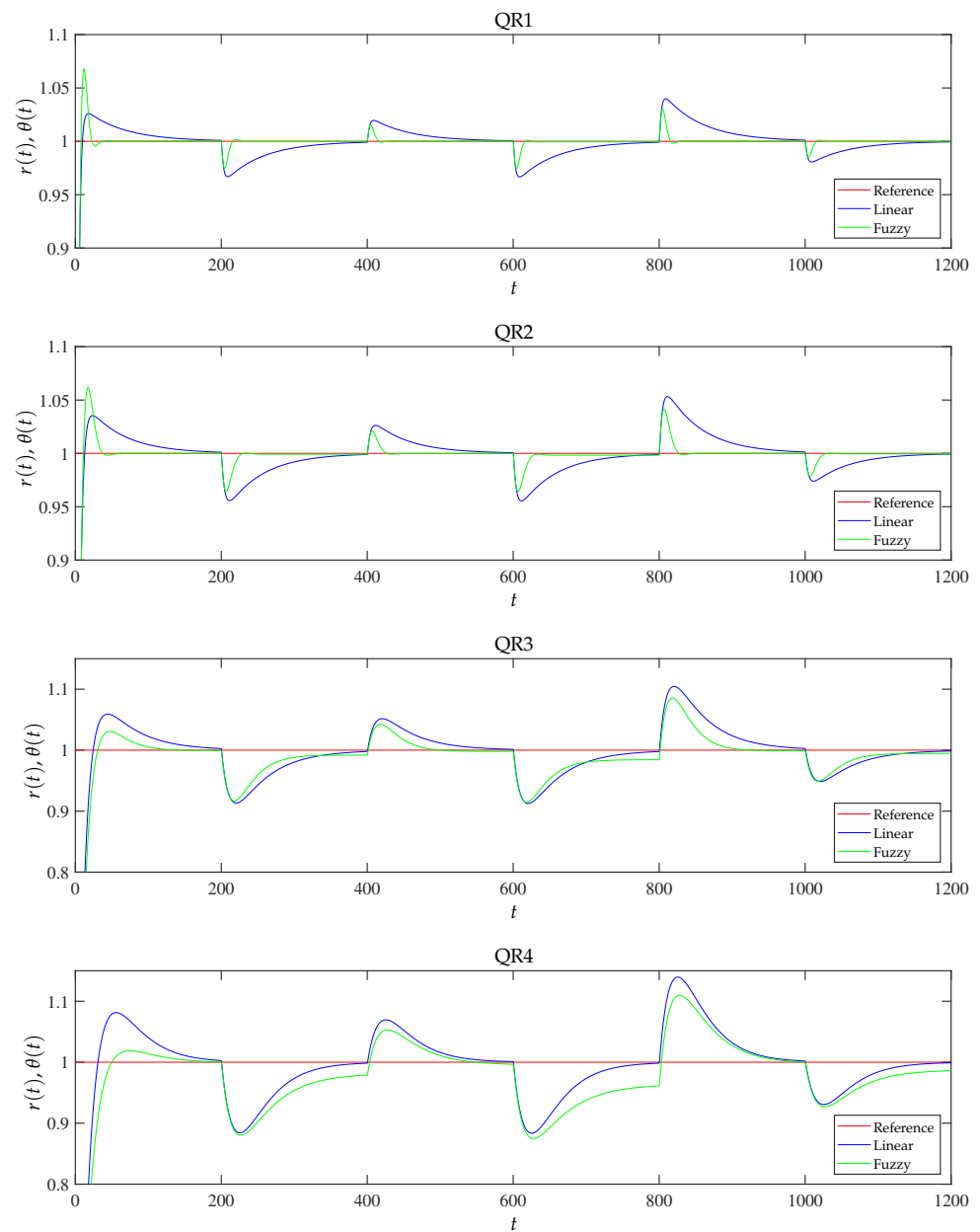


Figure 21. Comparison for the response of the system using linear and fuzzy controllers for different disturbance values.

Considering Tables 6 and 7, to determine statistically significant differences among the groups, we used the Kruskal–Wallis test, having a p -value of 2.8463×10^{-13} , which indicates differences. Considering the above result, the multiple comparisons using Bonferroni procedure is performed, obtaining Figure 22 observing no significant difference between the linear and fuzzy controllers for the configurations of Q and R . In addition, Figure 22 allows us to observe differences among cases QR1, QR2, QR3, and QR4, for example, when the value of R is increased the value of J_T also increases.

Table 6. Disturbance rejection results of J_T for 10 simulations.

Simulation	QR1		QR2		QR3		QR4	
	Linear	Fuzzy	Linear	Fuzzy	Linear	Fuzzy	Linear	Fuzzy
1	0.0025	0.0024	0.0038	0.0036	0.0119	0.0116	0.0208	0.0213
2	0.0024	0.0024	0.0036	0.0035	0.0109	0.0107	0.0189	0.0190
3	0.0023	0.0023	0.0034	0.0033	0.0101	0.0100	0.0172	0.0178
4	0.0027	0.0028	0.0043	0.0044	0.0147	0.0153	0.0264	0.0289
5	0.0023	0.0023	0.0034	0.0034	0.0103	0.0102	0.0176	0.0180
6	0.0023	0.0023	0.0034	0.0033	0.0099	0.0099	0.0168	0.0178
7	0.0020	0.0020	0.0029	0.0028	0.0075	0.0072	0.0120	0.0120
8	0.0024	0.0024	0.0036	0.0036	0.0111	0.0114	0.0192	0.0205
9	0.0022	0.0022	0.0031	0.0031	0.0088	0.0086	0.0145	0.0147
10	0.0022	0.0022	0.0032	0.0031	0.0090	0.0090	0.0149	0.0157

Table 7. Numerical measures of J_T for 10 simulations considering disturbance rejection.

Numerical	QR1		QR2		QR3		QR4	
	Linear	Fuzzy	Linear	Fuzzy	Linear	Fuzzy	Linear	Fuzzy
Mean	0.0023	0.0023	0.0035	0.0034	0.0104	0.0104	0.0178	0.0186
STD	0.0002	0.0002	0.0004	0.0004	0.0020	0.0022	0.0039	0.0045
Max	0.0027	0.0028	0.0043	0.0044	0.0147	0.0153	0.0264	0.0289
Min	0.0020	0.0020	0.0029	0.0028	0.0075	0.0072	0.0120	0.0120

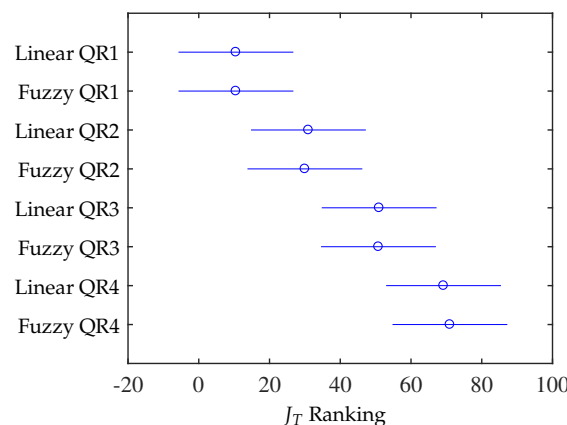


Figure 22. Multiple comparison using Bonferroni procedure.

8. Discussion

As seen, for linear controller the algorithm locates the zero near the pole in such a way that the pole and zero “almost cancel” occurs, which is a PI controller design method as presented in [34]. Frequently, it is possible to create a design by placing the zero of the controllers near the pole of origin in a way that both pole and zero of the system in closed loop “almost cancel” each other. Considering [33,34], a pair pole-zero that almost cancels has an insignificant effect on the time response; thus, the transitory response of the closed-loop system with the controller is approximately the same as in the closed-loop system without the controller. As a result, PI controller offers little deterioration in the transitory response with a significant improvement in steady state error.

According to [40], a prefilter can be employed to improve the system's dynamic response. In the linear controller, the prefilter allows the cancellation of the zero effect; also, the system overshoot can be reduced with an adequate prefilter design. The effect of the prefilter might be included in the training algorithm for the optimization of the neuro-fuzzy system. The prefilter may have a linear design and then extend it to a fuzzy design.

This application shows the possibility of achieving a non-linear fuzzy controller taking as reference the design of a linear controller (linear controllers with excellent performance can be used for the plant). As the plant considered is well known, the article focuses on showing how a linear controller can be extended to a fuzzy controller (starting from a behavior close to the linear controller). After having the first implementation of the fuzzy controller, nonlinearities can be included to improve the behavior of the control system when having more complex plants with nonlinearities, a subject that will be addressed in later work. In addition, it can be considered a broader comparison with other techniques using different plants.

Further, in this work, the stability analysis of the closed-loop system was performed with the linear controller, in a later work it is expected to extend the stability analysis for this type of fuzzy controllers by means of a Liapunov analysis.

9. Conclusions

Linear controller design and optimization allow the design of the fuzzy controller as well as the definition of the initial optimization setup of this controller.

For the linear controller (in discrete time), since the pole of the controller is placed in $z = 1$, the employed algorithm aims at finding the optimal location for the zero. Then, the fuzzy controller is optimized by considering these values for the initial setup; therefore, similar behavior is obtained. This demonstrates that it is possible to carry out the design and optimization of the fuzzy controller having a linear controller as a reference.

The fuzzy controller permits to include nonlinear relations through membership functions. Adequate configurations of these functions also allow the definition of a linear region in a way that it is possible to determine equivalence between the linear and fuzzy controllers.

The use of values Q and R permits obtaining different behaviors considering the action values that may be supplied for the plant. When testing the fuzzy controller for different reference values, the behavior of the control system is similar for the considered reference values.

It is observed that keeping the value of Q constant and increasing the value of R , the gain of the controller is reduced in a way that the energy of the plant also decreases.

The fuzzy controller scheme can be used for the implementation of an adaptive control strategy, in such a way that when a variation of the plant parameters occurs, the controller can be adjusted to obtain the desired behavior.

Author Contributions: Conceptualization, H.E., I.M. and H.L.; methodology, H.E., I.M. and H.L.; project administration, H.L.; supervision, H.L.; validation, I.M.; writing—original draft, H.E.; writing—review and editing, H.E., I.M. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: In this work were not carried out test on individuals (humans).

Informed Consent Statement: The study presented in this work does not involve human beings.

Data Availability Statement: No external data were required for this work.

Acknowledgments: The authors express gratitude to the Universidad de Oviedo, and the Universidad Distrital Francisco José de Caldas.

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

References

1. Feng, G. *Analysis and Synthesis of Fuzzy Control Systems: A Model-Based Approach*; CRC Press: Boca Raton, FL, USA, 2010.
2. Barragán, J.; Al-Hadithi, B.M.; Andújar, J.M.; Jiménez, A. Metodología formal de análisis del comportamiento dinámico de sistemas no lineales mediante lógica borrosa. *Rev. Iberoam. Autom. Inform. Ind.* **2015**, *12*, 434–445. [[CrossRef](#)]
3. Ross, T.J. *Fuzzy Logic with Engineering Applications*; John Wiley & Sons: West Sussex, UK, 2017.
4. Zhang, B.; Shin, Y.C. A Data-Driven Approach of Takagi-Sugeno Fuzzy Control of Unknown Nonlinear Systems. *Appl. Sci.* **2021**, *11*, 62. [[CrossRef](#)]
5. Chen, D.; Cai, J.; Huang, Y.; Lv, Y. Deep Neural Fuzzy System Oriented toward High-Dimensional Data and Interpretable Artificial Intelligence. *Appl. Sci.* **2021**, *11*, 7766. [[CrossRef](#)]
6. Chau, N.L.; Tran, N.T.; Dao, T.P. A multi-response optimal design of bistable compliant mechanism using efficient approach of desirability, fuzzy logic, ANFIS and LAPO algorithm. *Appl. Soft Comput.* **2020**, *94*, 106486. [[CrossRef](#)]
7. Chau, N.L.; Tran, N.T.; Dao, T.P. Design optimization for a compliant mechanism based on computational intelligence method. *Neural Comput. Appl.* **2021**, *33*, 9565–9587. [[CrossRef](#)]
8. Dinh, V.B.; Tran, N.T.; Dao, T.P. An integration framework of topology method, enhanced adaptive neuro-fuzzy inference system, water cycle algorithm with evaporation rate for design optimization for a flexure gripper. *Neural Comput. Appl.* **2021**, 1–26. [[CrossRef](#)]
9. Chen, Z.S.; Dang, M.P.; Le, H.G.; Chau, N.L.; Dao, T.P. Optimization for a flexure hinge using an effective hybrid approach of fuzzy logic and moth-flame optimization algorithm. *Math. Probl. Eng.* **2021**, *2021*, 6622655. [[CrossRef](#)]
10. Jerkovic, V.; Varga, T.; Benšić, T.; Barukcic, M. A Survey of Fuzzy Algorithms Used in Multi-Motor Systems Control. *Electronics* **2020**, *9*, 1788. [[CrossRef](#)]
11. Callejas-Cuervo, M.; González-Cely, A.X.; Bastos-Filho, T. Design and Implementation of a Position, Speed and Orientation Fuzzy Controller Using a Motion Capture System to Operate a Wheelchair Prototype. *Sensors* **2021**, *21*, 4344. [[CrossRef](#)]
12. Lin, Y.-H.; Yu, C.-M.; Wu, C.-Y. Towards the Design and Implementation of an Image-Based Navigation System of an Autonomous Underwater Vehicle Combining a Color Recognition Technique and a Fuzzy Logic Controller. *Sensors* **2021**, *21*, 4053. [[CrossRef](#)]
13. Ortega-García, L.E.; Rodríguez-Sotelo, D.; Nuñez-Perez, J.C.; Sandoval-Ibarra, Y.; Perez-Pinal, F.J. DSP-HIL Comparison between IM Drive Control Strategies. *Electronics* **2021**, *10*, 921. [[CrossRef](#)]
14. Hwang, K.-H.; Park, J.-H.; Kim, H.-J.; Kuc, T.-Y.; Lim, S.-J. Development of a Simple Robotic Driver System (SimRoDS) to Test Fuel Economy of Hybrid Electric and Plug-In Hybrid Electric Vehicles Using Fuzzy-PI Control. *Electronics* **2021**, *10*, 1444. [[CrossRef](#)]
15. Chen, C.-H.; Lin, C.-J.; Jeng, S.-Y.; Lin, H.-Y.; Yu, C.-Y. Using Ultrasonic Sensors and a Knowledge-Based Neural Fuzzy Controller for Mobile Robot Navigation Control. *Electronics* **2021**, *10*, 466. [[CrossRef](#)]
16. Chien, J.-C.; Dang, Z.-Y.; Lee, J.-D. Navigating a Service Robot for Indoor Complex Environments. *Appl. Sci.* **2019**, *9*, 491. [[CrossRef](#)]
17. Carpio, M.; Saltaren, R.; Viola, J.; Calderon, C.; Guerra, J. Proposal of a Decoupled Structure of Fuzzy-PID Controllers Applied to the Position Control in a Planar CDPR. *Electronics* **2021**, *10*, 745. [[CrossRef](#)]
18. Fei, J.; Chen, Y.; Liu, L.; Fang, Y. Fuzzy Multiple Hidden Layer Recurrent Neural Control of Nonlinear System Using Terminal Sliding-Mode Controller. *IEEE Trans. Cybern.* **2021**, 1–16. [[CrossRef](#)]
19. Fei, J.; Chen, Y. Fuzzy Double Hidden Layer Recurrent Neural Terminal Sliding Mode Control of Single-Phase Active Power Filter. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 3067–3081. [[CrossRef](#)]
20. Fei, J.; Wang, Z.; Liang, X.; Feng, Z.; Xue, Y. Fractional Sliding Mode Control for Micro Gyroscope Based on Multilayer Recurrent Fuzzy Neural Network. *IEEE Trans. Fuzzy Syst.* **2021**. [[CrossRef](#)]
21. Espitia, H.; Soriano, J.; Machón, I.; López, H. Compact Fuzzy Systems Based on Boolean Relations. *Appl. Sci.* **2021**, *11*, 1793. [[CrossRef](#)]
22. Espitia, H.; Soriano, J.; Machón, I.; López, H. Design Methodology for the Implementation of Fuzzy Inference Systems Based on Boolean Relations. *Electronics* **2019**, *8*, 1243. [[CrossRef](#)]
23. Espitia, H.; Machón, I.; López, H. Optimization of a Fuzzy Automatic Voltage Controller Using Real-Time Recurrent Learning. *Processes* **2021**, *9*, 947. [[CrossRef](#)]
24. Gupta, M.; Jin, L.; Homma, N. *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory*; John Wiley & Sons: Hoboken, NJ, USA, 2003.
25. Ioannou, P.A.; Pitsillides, A. *Modeling and Control of Complex Systems*; CRC Press: Boca Raton, FL, USA, 2008.
26. Olivas, E.S.; Guerrero, J.D.M.; Martínez-Sober, M.; Magdalena-Benedito, J.R.; Serrano López, A.J. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques: Algorithms, Methods, and Techniques*; Information Science Reference (IGI Global): Hershey, PA, USA, 2010.
27. Espitia, H.; Sierra, F. Optimización de un controlador análogo para un seguidor solar empleando algoritmos genéticos y enjambres de partículas. In Proceedings of the IEEE SIFAE Simposio Internacional en Fuentes Alternativas de Energía y Calidad Energética, Barranquilla, Colombia, 25–26 October 2012.
28. Espitia, H.; Sierra, J. Diseño e implementación de controladores análogos para un seguidor solar. *Visión Electrónica* **2013**, *7*, 118–135.
29. Espitia, H.; Sierra, F. Controller optimization for a solar tracking system using differential evolution. *Tecciencia* **2015**, *7*, 7–13. [[CrossRef](#)]

30. Espitia, H.; Machón, I.; López, H. Optimization of a linear controller using dynamic back-propagation. *Visión Electrónica* **2019**, *13*, 273–279. [[CrossRef](#)]
31. Ogata, K. *Modern Control Engineering*, 5th ed.; Instrumentation and Controls Series; Prentice Hall: Boston, MA, USA, 2010.
32. Gajic, Z.; Lelic, M. *Modern Control System Engineering*; Prentice Hall: Hoboken, NJ, USA, 1996.
33. Nise, N.S. *Control Systems Engineering*, 6th ed.; John Wiley & Sons: Hoboken, NJ, USA, 2011.
34. Fadali, M.S. Antonio Visioli, *Digital Control Engineering: Analysis and Design*; Elsevier: Waltham, MA, USA, 2013.
35. Ogata, K. *Sistemas de Control en Tiempo Discreto*; Prentice Hall Hispanoamericana: Mexico City, Mexico, 1996.
36. Mak, M.W.; Ku, K.W.; Lu, Y.L. On the improvement of the real time recurrent learning algorithm for recurrent neural networks. *Neurocomputing* **1999**, *24*, 13–36. [[CrossRef](#)]
37. Steil, J. Online stability of backpropagation-decorrelation recurrent learning. *Neurocomputing* **2006**, *69*, 642–650. [[CrossRef](#)]
38. Perrusquía, A.; Yu, W. Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview. *Neurocomputing* **2021**, *438*, 145–154. [[CrossRef](#)]
39. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [[CrossRef](#)]
40. Forrai, A. *Embedded Control System Design: A Model Based Approach*; Springer: New York, NY, USA, 2013.