*processes*

*Article*

# Novel Iterative Feedback Tuning Method Based on Overshoot and Settling Time with Fuzzy Logic

**Jacob Gonzalez-Villagomez [1], Carlos Rodriguez-Donate [1,*], Misael Lopez-Ramirez [1], Ruth I. Mata-Chavez [1] and Omar Palillero-Sandoval [2]**

[1] Departamento de Estudios Multidisciplinarios, División de Ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato, Yuriria C.P. 38944, Guanajuato, Mexico
[2] Center for Research in Engineering and Applied Science (CIICAp), Institute for Research in Pure and Applied Science (IICBA), UAEM. Av. Universidad No.1001 Col. Chamilpa, Cuernavaca C.P. 62209, Morelos, Mexico
* Correspondence: c.rodriguezdonate@ugto.mx

**Abstract:** Proportional–integral–derivative controllers are applied for solving a wide range of problems in industrial processes. They are preferred over computational techniques because of their implementation simplicity, low cost, and robustness against noise. In this paper, a novel iterative feedback tuning method is proposed using fuzzy logic, where the design parameters proposed by the user are the desired plant overshoot and settling time. In contrast to classical methods, the proposed technique does not require a precise and complex mathematical model for tuning the proportional–integral–derivative controllers through the plant, nor does it need an expert who knows the precise behavior of the system as in methods based on computational techniques. Furthermore, unlike iterative feedback tuning methods that use cost functions and require several experiments to perform the iterations, this proposal uses fuzzy logic to update the controller parameters, which facilitates its implementation in programmable hardware. The proposed method can be easily implemented in software considering three main stages: pre-processing, fuzzy logic system, and post-processing. The simulation and experimental results demonstrate the effectiveness of the proposed method for proportional–integral–derivative controller tuning; moreover, according to the performed comparison, the proposed method provides a trade-off between performance and robustness in comparison with other tuning techniques.

**Keywords:** fuzzy logic; overshoot; settling time; PID controller tuning



**Citation:** Gonzalez-Villagomez, J.; Rodriguez-Donate, C.; Lopez-Ramirez, M.; Mata-Chavez, R.I.; Palillero-Sandoval, O. Novel Iterative Feedback Tuning Method Based on Overshoot and Settling Time with Fuzzy Logic. *Processes* **2023**, *11*, 694. https://doi.org/10.3390/pr11030694

Academic Editor: Wen-Jer Chang

Received: 31 January 2023
Revised: 21 February 2023
Accepted: 22 February 2023
Published: 24 February 2023



## 1. Introduction

Proportional–integral–derivative (PID) controllers are often applied to solve a wide range of problems, mainly in industrial processes [1–4]. To date, the use of this type of controller predominates compared with controllers that are based on computational techniques [5–10], since they are cheap, easy to implement, and resilient to noise (robust). The performance of PID controllers is based on only three parameters: (i) proportional gain, (ii) integral gain, and (iii) derivative gain. Adjusting these gains in a PID controller is an important design process known as tuning, and it is necessary for the proper functioning of the controller [11]. Hence, several methods have been proposed to tune the PID gains according to the characteristics determined by the user [12], among which conventional methods based on computational techniques and iterative feedback tuning (IFT) methods can be highlighted.

Furthermore, the tuning of PID controllers using conventional methods and their variants [13–19] requires a transfer function describing the real dynamic behavior of the system to be controlled. The complexity of PID tuning depends on the mathematical model describing the system to be controlled; therefore, obtaining a simple mathematical model is recommended even if the precision of tuning is compromised [14]. In addition, it is widely known that in the industrial sector, it is not always possible to have a mathematical model describing the behavior of the system to be controlled [20].

On the other hand, tuning methods based on computational techniques such as fuzzy logic (FL) do not require an exact mathematical model of the plant [21], but they instead rely on the experience, knowledge, and linguistic definitions of the operator. It can be noticed that the most common input parameters in this type of tuning method are the error signal ($e$) and the derivative of the error ($\Delta e$), which are generated by the feedback loop and the reference signal. In this context, based on a model-free design method for fuzzy PI controllers, the fuzzy logic-supervised tuning system presented in [22] can be highlighted. In [21], a methodology is proposed to adjust the fuzzy PID controller gains using a non-linear optimization method to restrict the inputs and outputs of the closed-loop system. In [23], a fuzzy estimation algorithm with two inputs is used for evaluating the performance of the control system and providing a benchmark signal to adjust the PI control gain. In [24], a tuning approach is proposed for two-input fuzzy control systems, where the gray wolf optimizer (GWO) algorithm is used for solving the optimization problems within tuning parameter updating. In [25], a fuzzy PID controller structure is proposed, including a differential evolution algorithm to find out the optimal controller gains in real time. In addition to these previous works, adaptive fuzzy logic algorithms that require $e$ and $\Delta e$ to adjust the PID gains have been proposed [26–29]. It should be noticed that using a fuzzy system to tune a PID controller relies on the designer's or operator's experience and knowledge; otherwise, satisfactory results cannot be obtained [30].

Finally, iterative feedback tuning (IFT) methods consist of an approach based on the input and output gradients recorded by the closed-loop system, and these are based on cost functions for optimizing the controller parameters. These methods do not involve a parametric system model; they work with a PID controller structure, and three experiments are required to obtain enough data to be used in the cost function iterations; therefore, they lack robustness in their design [31,32]. In addition, different variants of iterative methods have been developed; for instance, in [33], virtual reference feedback tuning (VRFT) is proposed for optimizing the PID controller parameters by combining the closed-loop system response and a predefined reference path. In [34,35], fictitious reference iterative tuning (FRIT) for PID controllers is proposed using closed-loop system data in operation without any testing data; however, it is mandatory to know the control process to propose a reference model to which the parameters can be adjusted, complicating the application of this method in any process. In [36], a variant of the FRIT method for industrial chemical processes is introduced. This method, referred to as an extension of FRIT (E-FRIT), proposes the reference model and the PID parameters to be simultaneously adjusted, looking to minimize the control system errors; however, knowledge of the system is required to propose a reference model. To improve the performance of the IFT method, in [37], an approach based on the Box–Jenkins model algorithm is used for estimating the gradient with which the PID controller parameters are iteratively adjusted. In [38], the proposal is to include a restriction in the method (optimization parameter) by means of a penalty function to improve the robustness of the IFT approach. In [39], FL is used to replace the update law in IFT methods, thus minimizing the step size in the search direction for parameter optimization. On the other hand, in [40], a stability and convergence analysis based on the direct Lyapunov method is proposed, where tuning is carried out by means of an IFT algorithm for the design of Takagi–Sugeno–Kang PI fuzzy controllers (PI-FCs). Hence, as described above, to achieve acceptable PID controller tuning utilizing conventional methods, complex models are required, whereas fuzzy logic-based computational methods require the knowledge of an expert designer of control systems, and regarding IFT-based methods, a lot of data are required in each iteration and their design lacks robustness.

In this work, a new iterative feedback method for PID tuning using FL is proposed, with the design parameters being the desired plant overshoot and settling time. Different from conventional methods and FL-based methods, neither a complex mathematical model nor a system expert is required. The method only requires knowing the system response at each iteration, unlike conventional iterative methods, which require previous experimental results to perform the parameter optimization of PID controllers. Unlike most tuning methods based on the error and the error derivative of the system control signal, the proposed method is based on the real error of the overshoot characteristics and the response settling time of the control system. By providing only the overshoot time and the desired settling time of the plant, the error of the actual plant parameters is calculated and sent back to the FL-based tuning system to calculate the PID gains. The process is iterative and continues until the error is close to zero. The proposed method implementation is proposed in three main stages: pre-processing, FL system, and post-processing. Its validation was carried out using computational simulations; a comparison with other tuning techniques, where the proposed method provides a trade-off between performance and robustness; and an experimental test bench, which demonstrated the proposed method efficiency for PID controller tuning based on FL.

The rest of the paper is organized as follows: Section 2 introduces the tuning method formulation. The proposed methodology and the FL-based system are given in Section 3. The simulation examples and experimental setup are provided in Section 4 to demonstrate the ability of the proposed tuning method. Finally, the paper is concluded in Section 5.

## 2. Background

### 2.1. PID Controller

The general structure of a PID controller can be represented by the transfer function $C_{PID}(s) = K_p(1 + (1/(T_i s)) + T_d s)$, where the gain ($K_p$), the integrative time ($T_i$), and the derivative time ($T_d$) must be tuned. In addition, the equation that describes the structure can be rewritten as in (1) for the development of the presented proposal.

$$C_{PID}(s) = \frac{K_p T_d s^2 + K_p s + \dfrac{K_p}{T_i}}{s} \tag{1}$$

For simplicity, the controller constants are replaced with $a$, $b$, and $c$, so that (1) takes the following form:

$$C_{PID}(s) = \frac{as^2 + bs + c}{s} \tag{2}$$

where $a = K_d = K_p T_d$, $b = K_p$, and $c = K_i = K_p/T_i$ are the gains of the controller.

### 2.2. PID Controller Behavior

To explain the tuning method, the behavior of controller parameters $a$, $b$, and $c$ in (2) is analyzed using the pole–zero diagram. To define a tuning plane such as the one presented in Figure 1a, the dominant poles of the transfer function of the system are located in a semicircle based on the pole cancellation method.

The variation in the parameters of (2) determines the radius of the semicircle and the range of variability of the real and imaginary axes of the pole-and-zero diagram of the PID controller. The radius of the semicircle ($r$) is defined by (3) as follows:

$$r = \sqrt{\frac{c}{a}} = \sqrt{\frac{\dfrac{K_p}{T_i}}{K_p T_d}} = \sqrt{\frac{1}{T_i T_d}} \tag{3}$$

Equation (2) is of second order; therefore, it has real and imaginary roots that can be determined with the general equation described in (4).

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{4}$$

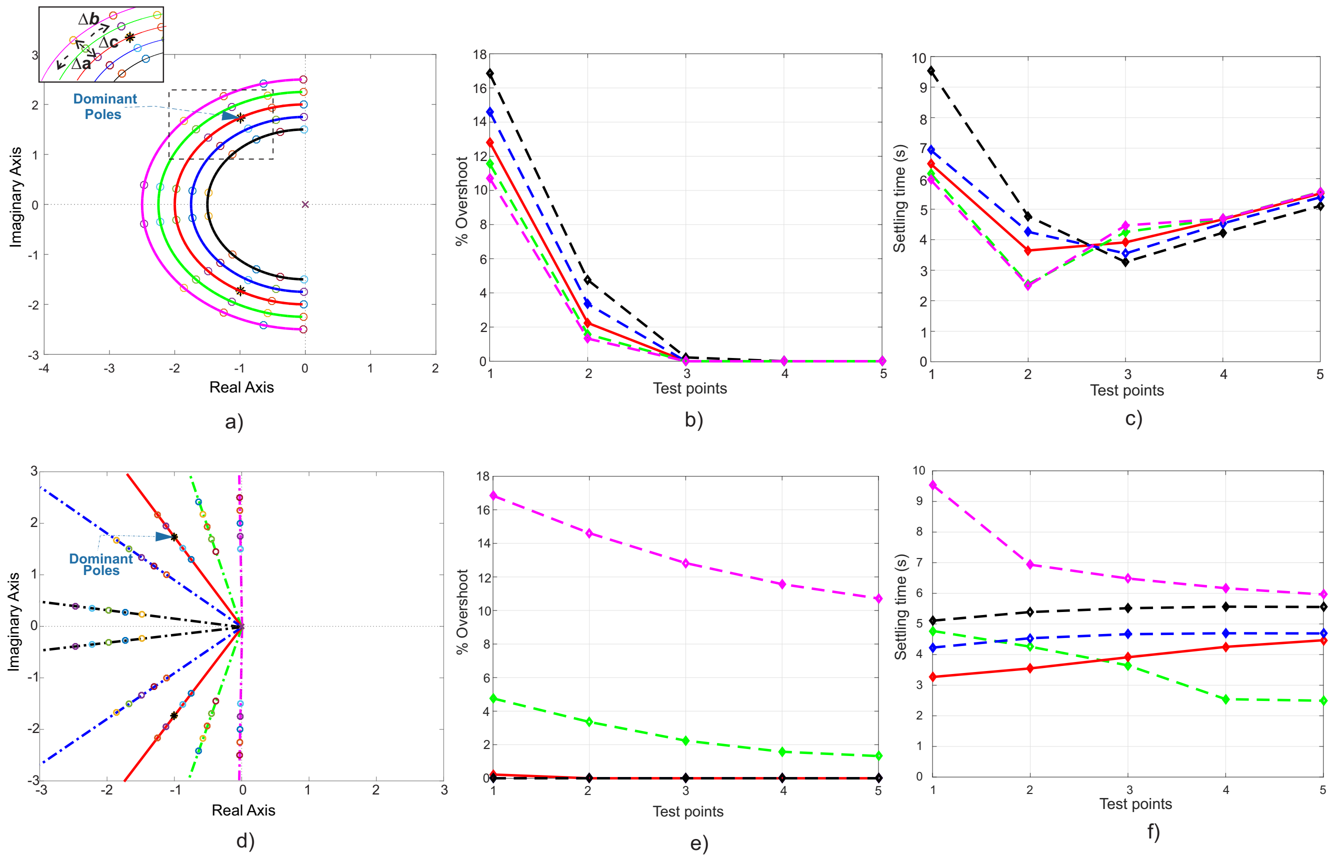


**Figure 1.** Analysis of the PID root behavior on Laplace plane. (**a**) Tuning plane on semicircle, (**b**) overshoot behavior on semicircle, and (**c**) settling time behavior on semicircle: central semicircle; semicircle at $-0.5$; semicircle at $-0.25$; semicircle at 0.25; semicircle at 0.5. (**d**) Tuning plane on straight line, (**e**) overshoot behavior on straight line, and (**f**) settling time behavior on straight line: central line; bottom line (near real axis); line between central line and bottom line; line between central and top line; top line (near imaginary axis).

From (4), the factor that determines the limit between complex and real roots is $b^2 - 4ac$, where $b > 2\sqrt{ac}$ only has real values and $b < 2\sqrt{ac}$ only has complex values. Therefore, the range of variability ($Rn$) over the described semicircle is determined using (5).

$$Rn = 0 - 2\sqrt{ac} \tag{5}$$

From the above analysis, the behaviors of the overshoot and the settling time that are obtained when varying the position of the controller in the tuning zone can be defined in two ways: (i) with the arc of the semicircle described above (shown in Figure 1a) and (ii) with the straight line representing the radius of the semicircle described above (shown in Figure 1d). Firstly, considering different controller tuning cases, the overshoot behavior is shown in Figure 1b, and the settling time behavior is shown in Figure 1c. The settling time exhibits similar behavior when varying the controller and resetting the zeros of the PID controller. In Figure 1a, the central semicircle is positioned in the central part of the data of the other described semicircles on the tuning plane. Secondly, the overshoot behavior in the straight-line configuration is shown in Figure 1e, and the settling time behavior is shown in Figure 1f. To define a fuzzy logic system, the analysis is based on the central semicircle dominant poles, as described in Figure 1a,d, and the overshoot and settling time errors. They are estimated according to the system response and the desired response. In Figure 1b,e, if the overshoot error is greater than zero, the overshoot increases, so $a$ and $c$ decrease. This means that the semicircle has a minor radius. On the other hand, if the overshoot error is less than zero, the radius increases. For bigger overshoot values, the modified variable is $b$. This implies that the controller zero moves along the semicircle. In the same way, the settling time can be defined according to Figure 1c,f. After obtaining these behaviors, fuzzy logic rules are defined to tune the PID gains (Section 3.2).

## 3. Proposed Method

The new FL-based PID tuning method under an iterative scheme only requires two design parameters: the overshoot and the settling time. For the proposed system to carry out the tuning process, the step response is required to obtain the overshoot and settling time that are to be used to calculate the error functions that are to be minimized at every iteration of the fuzzy system; therefore, if these error functions are large, the fuzzy system generates new PID gains to estimate the new step response to obtain the error functions. If the error function is minimized, the tuning process is terminated; otherwise, this new method performs a new iteration. The design of the FL system that does not require an expert is based on the exhaustive numerical analysis of the PID root behavior on the Laplace plane presented in Section 2; its analysis allows us to define the input and output of membership functions of the FL system. In a detailed manner, the next subsections describe every step of the new tuning proposed method and its implementation.

### *3.1. Proposed Methodology*

The flow chart in Figure 2 shows the PID tuning methodology using FL to obtain the desired overshoot and settling time of a plant. It is described in the steps below.

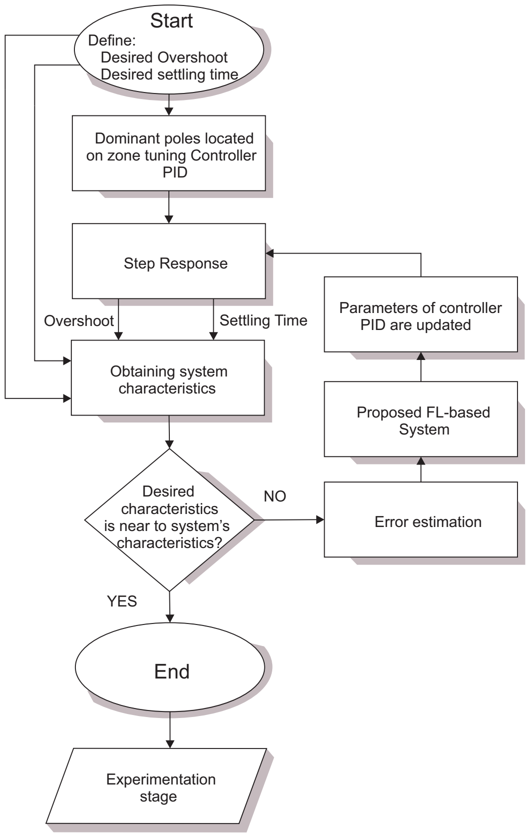


**Figure 2.** Flowchart of the proposed tuning method for PID controllers.

**Step 1**: The user proposes the desired overshoot and settling time characteristics of the system to be tuned.

**Step 2**: The parameters of overshoot and settling time of the plant are obtained to estimate the dominant poles and position them in the semicircle that crosses the tuning zone as described above and with which the initial gains of the PID controller are obtained.

**Step 3**: The system step response is obtained with the controller to estimate the actual values of overshoot and settling time.
**Step 4**: The errors between the desired parameters and the actual parameters are calculated.
**Step 5**: If the errors are small enough, PID adjustment is finished (step 8); otherwise, the estimated errors are used to calculate the new PID gains.
**Step 6**: From the errors obtained in step 4 and with the proposed system based on FL, the new PID gains are estimated.
**Step 7**: The PID gains are updated, and the process is repeated from step 3.
**Step 8**: The tuning process is finished and continues with the implementation of the controller.

At the end of the PID tuning stage, the controller is implemented to validate the design through an experimental or simulation stage.

### 3.2. Implementation of FL-Based System

Figure 3 shows the FL-based system for PID tuning performed in three stages: pre-processing, FL system, and post-processing. The inputs of the proposed system are the desired overshoot and settling time, which are entered into the pre-processing stage, where the error ($eOv$) is calculated with respect to the overshoot of the control system and the settling time error ($eST$) is calculated according to the settling time of the control system. These two error signals are the response of the closed-loop system and are the input to the FL system.

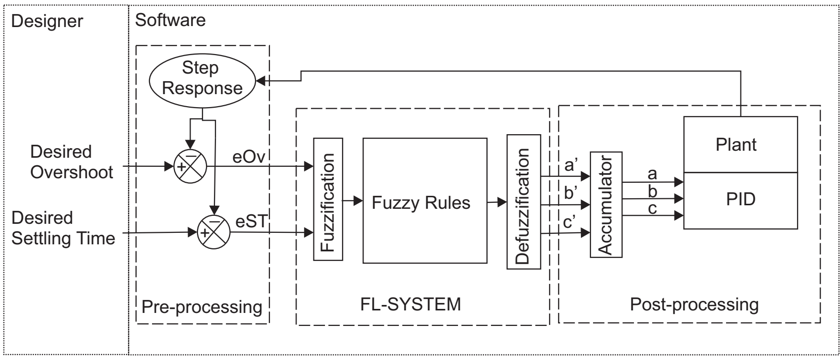


**Figure 3.** FL-based system for PID tuning.

In the next stage (FL system), the fuzzy inputs are converted into appropriate linguistic values, which are processed in the region of the fuzzy set that includes the membership functions (MFs). Different types of MF shapes can be used to design the FL-based system; however, the most popular types are the triangular and the trapezoidal types [20,41], represented by $AL$ for far left, $NL$ for near left, $N$ for near, $NR$ for near right, and $AR$ for far right. The notation of the MF outputs is the following: $VN$ for very negative, $N$ for almost negative, $VP$ for very positive, $P$ for positive, and $Z$ for zero. The MFs were designed using the fuzzy toolbox of MATLAB software. MF entries are shown in Figure 4.

The design of the FL system is based on the exhaustive numerical analysis carried out in Section 2. With the analysis of the PID root behavior on the Laplace complex plane, it is possible to identify the behaviors of the overshoot and settling time, which allows one to define a range of $-30$ to 30 for the input of the overshoot error of the response and a range of $-10$ to 10 for the input of the settling time error. Regarding the outputs, the fuzzy system considers one output for each variable of the PID controller ($a$, $b$, and $c$) defined by Equation (2). The MFs are shown in Figure 5.

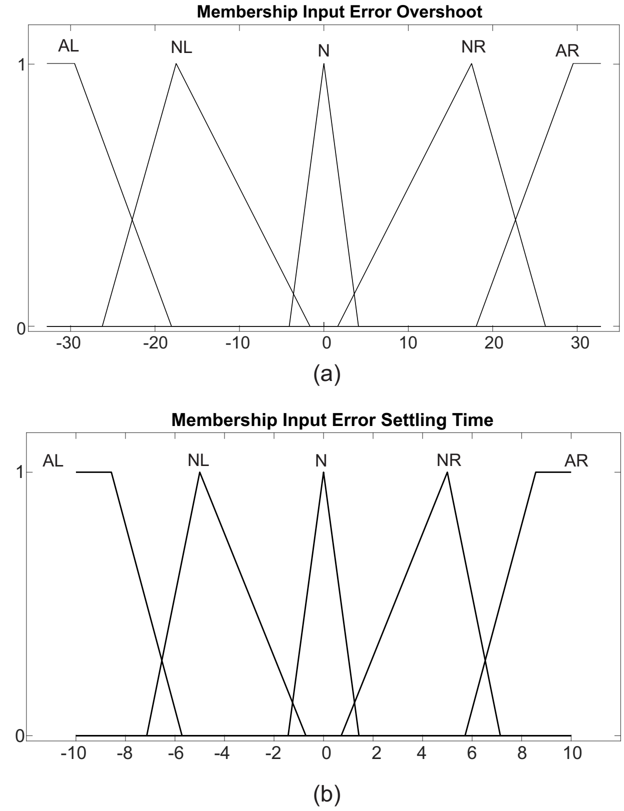


**Figure 4.** Inputs of 5 × 5 MFs designed in MATLAB/SIMULINK. (**a**) Error MF overshoot and (**b**) error MF settling time.

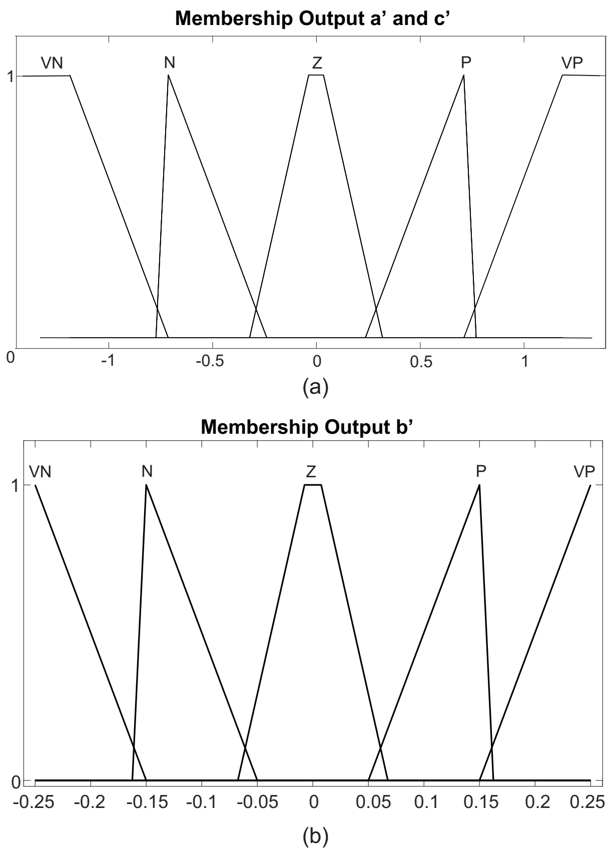


**Figure 5.** Outputs of 5 × 5 MFs designed in MATLAB/SIMULINK. (**a**) Variables a and c, and (**b**) variable b.

For the outputs of variables $a'$ and $c'$, the same structure of MFs as the structural behavior, a semicircle on the complex plane, is taken. The output values of variables $a'$ and $c'$ range from $-1.25$ to 1.25, while the output values of $b'$ range from $-0.25$ to 0.25; these ranges were heuristically determined by taking into account the numerical analysis presented in Section 2.

The qualitative relationship between the inputs and outputs of the system is illustrated by the design of fuzzy rule sets. These rules are developed based on the overshoot behavior of the closed-loop system shown in Figure 1b,e and the behavior of the settling time shown in Figure 1c,f. The rules based on the $5 \times 5$ MFs for variables $a'$ and $c'$ are given in Table 1, while Table 2 gives the matrix of fuzzy rules for variable $b'$.

**Table 1.** Fuzzy rule matrix for variables $a'$ and $c'$.

| e(ST) \ e(Ov) | AL | NL | N | NR | AR |
|---|---|---|---|---|---|
| **AL** | VP | P | Z | N | VN |
| **NL** | P | P | Z | N | VN |
| **N** | P | Z | Z | N | N |
| **NR** | N | N | Z | N | N |
| **AR** | VN | N | Z | P | VP |

**Table 2.** Fuzzy rule matrix for variable $b'$.

| e(ST) \ e(Ov) | AL | NL | N | NR | AR |
|---|---|---|---|---|---|
| **AL** | VN | N | Z | P | VP |
| **NL** | VP | P | Z | N | VN |
| **N** | P | Z | Z | Z | P |
| **NR** | VN | N | Z | P | P |
| **AR** | VN | N | Z | P | VP |

In the final stage, post-processing, outputs $a'$, $b'$, and $c'$ of the controller variables positioned at the dominant poles of the system to be tuned are added or subtracted; therefore, the controller variables to be evaluated with the closed-loop system are the following:

$$a = a + a' \tag{6}$$

$$b = b + b' \tag{7}$$

$$c = c + c' \tag{8}$$

With the updated values of the controller variables, the closed-loop system is evaluated, and the improved values of the overshoot and the settling time are captured. These serves as a reference for generating the error to be entered into the fuzzy system once again, and finally, to reach the desired value through iteration.

## 4. Results

This section presents the validation of the new proposed method to tune PID controllers based on FL. The proposed system was implemented in MATLAB software. The effectiveness of the method was evaluated with a simulation and an experimental setup.

### 4.1. Simulation

The plant represents a linearized model of aircraft pitch angle dynamics, described in reference [19], whose transfer function is given as follows:

$$G_1(s) = \frac{1.15s + 0.18}{s^3 + 0.74s^2 + 0.92} \tag{9}$$

The response of the closed-loop system is presented in Figure 6a. It can be seen that the system has overshoot of 0% and settling time of 34.53 s. Through the proposed FL-based algorithm for tuning PID controllers, it is desirable to have a system with overshoot of 5% and settling time of 10 s to improve the response of the system.

After defining the desired parameters, the next step is to place the zeros of the controller near the dominant poles, which generates the initial time response of the system with the controller, as shown in Figure 6b,c. After the FL-based tuning algorithm is performed, the zeros of the PID controller are moved at every iteration of the algorithm to obtain the desired system specifications, as depicted in Figure 6b, where the zero initial position and the zero final position are indicated by blue and red colors, respectively. In the same way, Figure 6c shows the evolution of the closed-loop response of the system in each iterative step until the desired range is reached. For this plant, as depicted in Figure 6b,c, the change is small in each iteration; however, the algorithm tries to approximate the desired value defined by the user, so that although the response is similar, the final response has better approximation than the initial one. According to (1) and (2), the resulting set of PID controller parameters while using the proposed method are as follows: $K_d$ = 1.124, $K_p$ = 1.4565, and $K_i$ = 1.044.

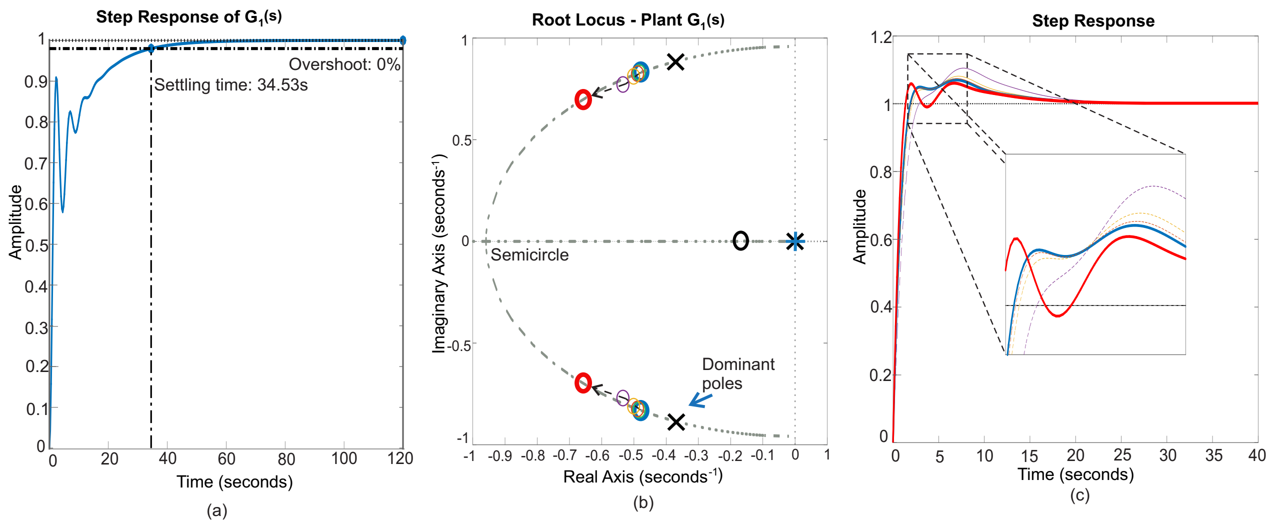


**Figure 6.** Plant $G_1$(s) response and tuning using the proposed tuning system. (**a**) Overshoot and settling time without controller, (**b**) results of movement of zeros of PID controller using the proposed method, and (**c**) step response per iteration: final controller tuning and initial controller tuning.

In order to make a comparison, Figure 7 shows the step response of both proposals with the controllers designed with each method: the design proposal in [19] and the design proposal described in this work. An evident improvement is observed in the settling time and overshoot of the plant described by (9). The quantitative data of controller performance are presented in Table 3 in the Discussion section.

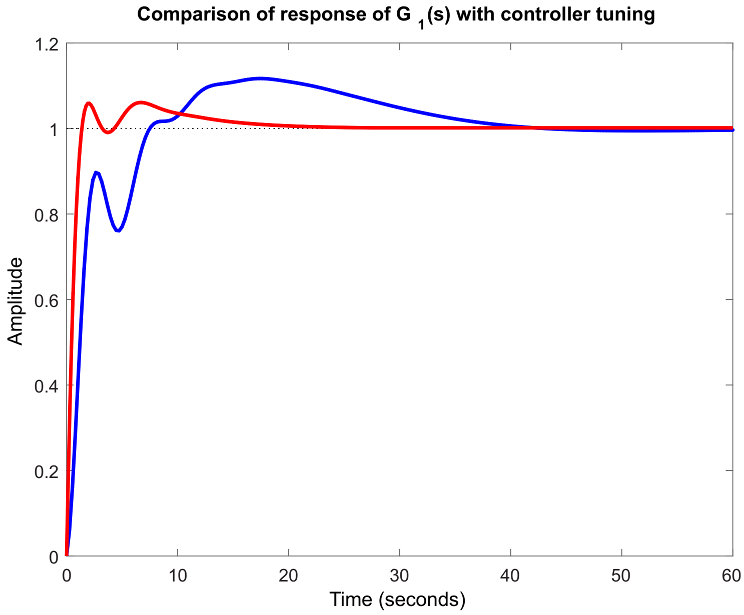


**Figure 7.** Comparison of response of G1(s) with both methods: red proposed tuning method and blue PI/PID tuning method [19].

### 4.2. Experimental Validation

The performance of the proposed technique was experimentally evaluated by controlling the x-axis of a Cartesian robot, which is a fundamental element in production and industrial processes for carrying out high-precision tasks. The experimental setup is shown in Figure 8; it consisted of a servomotor (model GYB41D5-RC2), which had a 20-bit serial encoder; a servo driver (RYH401F5-VV2); and a Terasic DE0-CV development board with an on-board Cyclone-IV field programmable gate array (FGPA; Altera). In the FPGA, the PID controller, the data acquisition system, and the PC communication system were embedded.

Before tuning the PID controller, the control system was relatively slow and had a settling time of 16.7 s without overshoot, as shown in Figure 9. Therefore, in order to make the system work faster, a settling time of 2 s was proposed with an overshoot of 20% to maintain a balance between them. Consequently, the proposed tuning method successfully updated these desired values to tune the PID controller with $a$ = 13.54, $b$ = 133, and $c$ = 471.2. Based on this design, Figure 10 shows the reference tracking response of the system in different steps, where the blue, dotted line represents the behavior of the system with the tuned controller and the red line is the given reference to follow. The qualitative response presented in Figure 10 demonstrates the efficiency and efficacy of the proposed system in reference tracking.

### 4.3. Result Discussion

The literature shows that FL has been widely used to self-tune PID controllers [21–30], where it is necessary to know the transfer function of the control system, since the inputs of the FL system depend on the control variable error. Hence, when performing the tuning process, a complex mathematical model of the control system must be used, since otherwise, the FL system would not be able to tune the PID controller. Additionally, an expert who knows the control system is required to design the MFs and the set of rules of the FL system. Regarding IFT, a system model to tune PID controllers is not required, but instead, three experiments are required to obtain enough data for the iterations; additionally, this method lacks robustness in its design [31,32]. Unlike the described methods, here, a new method to tune PID controllers is proposed; it requires the step response of the control system, from which the simple transfer function can be inferred, so the designer only needs to propose the desired overshot and settling time. Furthermore, the MFs and set of rules of the FL system are based on the analysis of the behavior of the PID transfer function on the Laplace plane; therefore, an expert of the control system is not required. Moreover, the FL system does not require complex input signal conditioning such as the one proposed in references [23–25] or the use of complex MFs such as those proposed in references [23,24,29], since this method is based on the error signal, as shown in Figure 3, and triangular MFs, which facilitates its implementation with low computational resources.

Additionally, the proposed method provides good performance in satisfying the specifications in terms of overshoot and settling time. It can be seen in Figure 6 that the proposed system adjusts the PID gain to obtain the desired parameters of the plant behavior.

It can also be noted that the adjustment of parameters such as overshoot and settling time is related, that is, the adjustment of one parameter influences the adjustment of the other. As defined in [21], fuzzy logic is approximate reasoning, so the proposed system is designed to approximate the desired values until equilibrium between the two parameters is reached. As a result, at the end of tuning, it is justified to expect errors in the obtained parameters.

On the other hand, Table 3 shows a quantitative comparison of the step response with different tuning techniques reported in the literature. The controller performance was measured with the integral absolute error (IAE), the integral square error (ISE), the integral time absolute error (ITAE), and the integral time square error (ITSE); additionally, in the table, rise time ($T_r$), settling time ($T_s$), and overshoot ($M_p$) are reported [42]. In this way, using the same simulation conditions presented in every case study reported, the proposed method was compared with case study $G_{P1}$(s) reported in [16]; cases studies $G_{Q1}$(s), $G_{Q2}$(s), and $G_{Q3}$(s) presented in [17]; case studies $G_{R1}$(s), $G_{R2}$(s), and $G_{R3}$(s) used in [18]; case studies $G_{S1}$(s) and $G_{S2}$(s) reported in [19]; and finally, case study $G_{T1}$(s) presented in [39]. According to this comparison table, smaller values of error criteria IAE, ISE, ITAE, and ITSE were achieved; in addition, the proposed method provided a faster response to $T_r$ and $T_s$ as well as smaller $M_p$. Therefore, the results indicate that the proposed method achieves an improvement in both performance and robustness in comparison with the other techniques considered. Finally, the experimental validation of the proposed method presented good performance in a qualitative way, as shown in Figure 10, and in a quantitative way, as indicated by the low values of the error criteria obtained (IAE = 0.1155; ISE = 0.0237; ITAE = 0.3234; ITSE = 0.0628), which ratifies the good performance and robustness of the method proposed in this article.

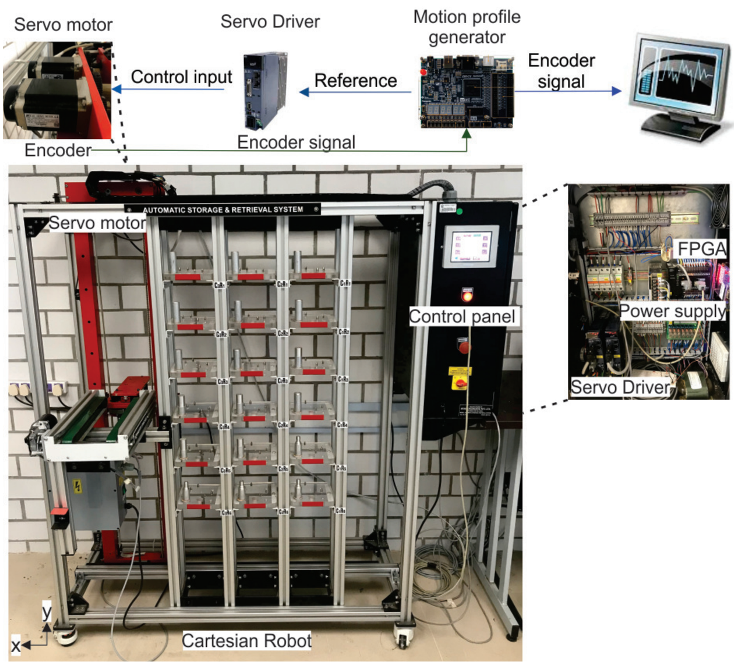


**Figure 8.** Experimental setup.

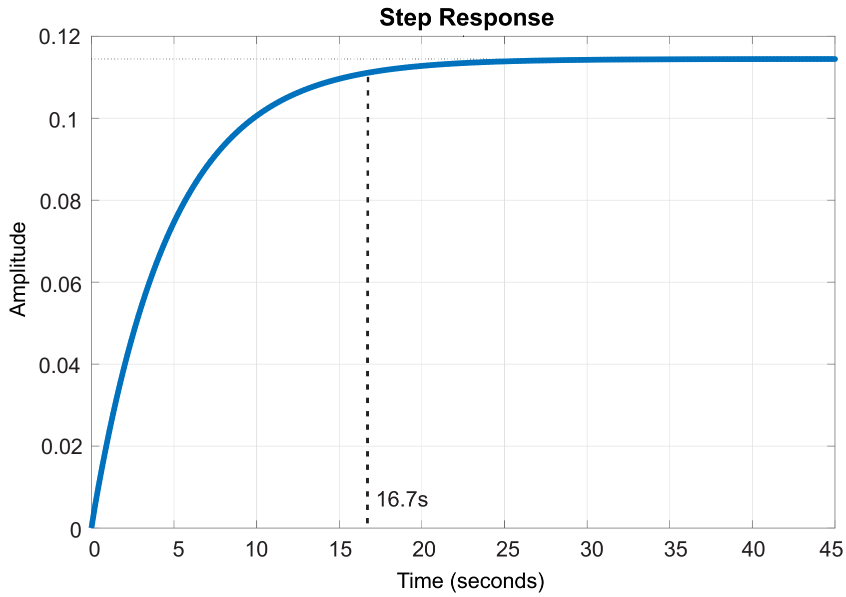


**Figure 9.** Step response of the experimental system.

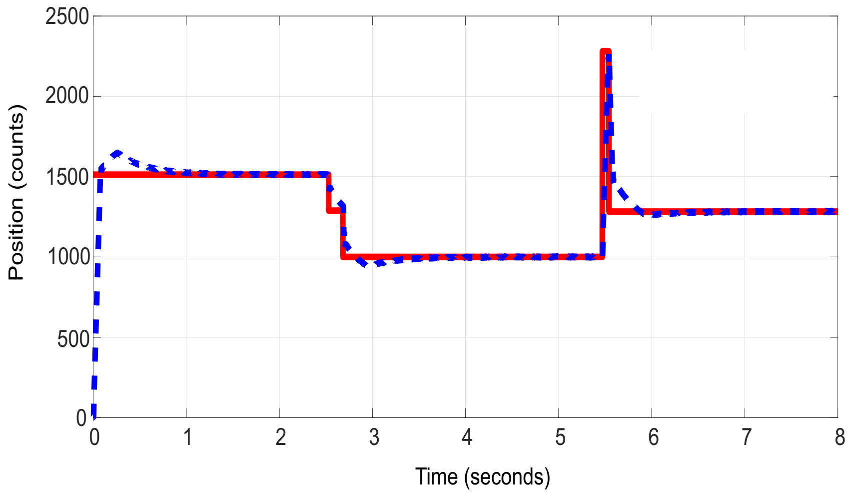


**Figure 10.** Tracking test with different reference positions of servo motor: - - tuned system response with proposed method and — reference signal.

**Table 3.** Performance of the PID controller tuning method.

| Tuning Method | PID Parameter | | | Dynamic Performance Specification | | | ISE | IAE | ITSE | ITAE |
|---|---|---|---|---|---|---|---|---|---|---|
| | $K_p$ | $K_d$ | $K_i$ | $T_r$ | $T_s$ | $M_p$ (%) | | | | |
| [16] $G_{P1}(s)$ | 1.6 | 3.2 | 0.061 | 15.1 | 27.7 | 2.57 | 6.51 | 10.39 | 24.15 | 132.2 |
| Proposed | 4.4006 | 2.6139 | 0.0906 | 7.14 | 10.9 | 0.633 | 5.27 | 6.14 | 9.97 | 24.79 |
| [17] $G_{Q1}(s)$ | 0.567 | 0.49 | 0.0766 | 1.97 | 22.8 | 64 | 3.84 | 6.37 | 14.26 | 36.62 |
| Proposed | 0.725 | 0.38 | 0.055 | 2.13 | 30 | 31.6 | 2.92 | 5.87 | 9.30 | 41.70 |
| [17] $G_{Q2}(s)$ | 0.279 | 0.493 | 0.0235 | 3.49 | 26.6 | 54.3 | 5.48 | 9.02 | 29.26 | 70.26 |
| Proposed | 0.3086 | 0.4993 | 0.0022 | 3.52 | 44.2 | 32.7 | 4.08 | 6.81 | 12.26 | 46.11 |
| [17] $G_{Q3}(s)$ | 0.27 | 0.34 | 0.0233 | 1.88 | 23 | 53.7 | 6.58 | 8.98 | 24.26 | 59.68 |
| Proposed | 0.2344 | 0.1822 | 0.0007 | 2.93 | 14.7 | 22.4 | 5.23 | 6.46 | 12.02 | 28.50 |
| [18] $G_{R1}(s)$ | 1 | 1.2 | 0.2 | 0.679 | 10.6 | 48.5 | 1.66 | 2.6698 | 1.85 | 6.88 |
| Proposed | 0.9566 | 1.3564 | 0.2301 | 0.616 | 10.4 | 47.2 | 1.62 | 2.6607 | 1.79 | 7.31 |
| [18] $G_{R2}(s)$ | 1.106 | 1.1856 | 0.1558 | 1.4 | 15.6 | 10.5 | 0.694 | 1.837 | 0.739 | 8.797 |
| Proposed | 2.425 | 1.74 | 0.395 | 0.471 | 11 | 8.69 | 0.4636 | 1.1 | 0.306 | 3.746 |
| [18] $G_{R3}(s)$ | 0.569 | 0.4995 | 0.081 | 1.77 | 11.2 | 47.3 | 2.51 | 4.31 | 6.14 | 17.32 |
| Proposed | 0.6053 | 0.6771 | 0.1135 | 1.56 | 11.3 | 47.3 | 2.35 | 4.15 | 5.49 | 16.5 |
| [19] $G_{S1}(s)$ | 0.75 | 0.12 | 0.1488 | 5.9 | 35.6 | 11.7 | 1.18 | 4.12 | 4.62 | 50.36 |
| Proposed | 1.4565 | 1.124 | 1.044 | 0.982 | 13.5 | 6.07 | 0.33 | 1.026 | 0.22 | 4.86 |
| [19] $G_{S2}(s)$ | $3.11 \times 10^{-6}$ | 0.41 | 17.083 | 0.167 | 0.288 | 0 | 0.021 | 0.058 | $8.6 \times 10^{-4}$ | 0.005 |
| Proposed | 1 | 1044 | $1.09 \times 10^{6}$ | $4.37 \times 10^{-6}$ | $7.67 \times 10^{-6}$ | 0.072 | $9.9 \times 10^{-7}$ | $1.9 \times 10^{-6}$ | $9.9 \times 10^{-13}$ | $3.8 \times 10^{-12}$ |
| [39] $G_{T1}(s)$ | 0.0027 | 0 | $1.09 \times 10^{-4}$ | 0.115 | 0.312 | 3.64 | 0.055 | 0.084 | 0.0020 | 0.0053 |
| Proposed | 1 | 0.1 | 0.01 | $1.57 \times 10^{-5}$ | $2.8 \times 10^{-5}$ | 0 | $3.6 \times 10^{-6}$ | $7.2 \times 10^{-6}$ | $1.3 \times 10^{-11}$ | $5.2 \times 10^{-11}$ |

## 5. Conclusions

This work presents a new iterative method to tune PID controllers using fuzzy logic, where the user or designer only needs to propose the desired overshoot and settling time of the plant to be controlled. The method assumes that the plant overshoot and settling time are known. It is also specified that fuzzy logic PID tuning is designed based on the analysis of the pole-and-zero diagram of the PID controller. This new method requires the plant step response at each iteration to obtain the real settling time and overshoot of the plant to compare them with the desired parameters. To make its easy implementation in software possible, the method is carried out in three stages: pre-processing, FL system, and post-processing. In contrast to classical methods, the proposed method requires no exact nor complex mathematical model for the plant to tune PID controllers, nor an expert who must know the precise behavior of the system as in methods based on fuzzy logic. Unlike IFT methods, the proposal does not require the stage of optimization of the PID gains using cost functions, since the design of the FL system is based on the behavior of the pole-and-zero diagram of the PID controller, with which the controller gains are adjusted. Additionally, by using triangular membership functions, the FL system facilitates implementation on a programmable card and an online self-tuning system.

On the other hand, the tuning method is based on the analysis of the zeros of the PID controller on a Laplace plane synthesized with fuzzy logic and the iterative feedback tuning technique. Effectiveness was verified by means of a simulation and experiments, which were carried out on a test bench for servomotor position control. The embedded control system in the FPGA was adjusted with this method. Furthermore, the method showed better performance and robustness than other tuning techniques, which is evident from the IAE, ISE, ITAE, and ITSE values.

Finally, future work on this method could improve the location of dominant poles and the tuning zone control, or else, the method tends to diverge. Additionally, the use of other design parameters such as the response time or the overshoot time could be explored.

**Author Contributions:** J.G.-V. contributed to all steps of the research method, conceptualization, methodology, validation, software, and writing—original draft; C.R.-D. contributed to conceptualization, resources, supervision, and investigation; all authors contributed to writing—review and editing, visualization, and formal analysis. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by National Council on Science and Technology (CONACYT), Mexico, under scholarship 781337.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The author thanks the financial support of National Council on Science and Technology (CONACYT), Mexico, under scholarship 781337.

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

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| PID | Proportional–integral–derivative |
| FL | Fuzzy logic |
| IFT | Iterative feedback tuning |
| MF | Membership functions |

## References


1. Chen, P.; Luo, Y. A Two-Degree-of-Freedom Controller Design Satisfying Separation Principle with Fractional Order PD and Generalized ESO. *IEEE ASME Trans. Mechatron.* **2021**, *27*, 137–148. [CrossRef]
2. Moreno, J.; Perez, R.; Guerrero, M.; Dzul, A. Nonlinear PID-type controller for quadrotor trajectory tracking. *IEEE ASME Trans. Mechatron.* **2008**, *23*, 2436–2447. [CrossRef]
3. Zaky, M.S. A self-tuning PI controller for the speed control of electrical motor drives. *Electr. Power Syst. Res.* **2015**, *119*, 293–303. [CrossRef]
4. Wang, Y.; Xia, Y.; Zhou, P. Fuzzy-model-based sampled-data control of chaotic systems: A fuzzy time-dependent Lyapunov–Krasovskii functional approach. *IEEE Trans. Fuzzy Syst.* **2017**, *25*, 1672–1684. [CrossRef]
5. Ozyetkin, M.M.; Onat, C.; Tan, N. PID tuning method for integrating processes having time delay and inverse response. *IFAC-PapersOnLine* **2018**, *51*, 274–279. [CrossRef]
6. Zhang, J.; Guo, L. Theory and design of PID controller for nonlinear uncertain systems. *IEEE Control Syst. Lett.* **2019**, *3*, 643–648. [CrossRef]
7. Torres, W.L.; Araujo, I.B.; Filho, J.B.M.; Junior, A.G.C. Mathematical modeling and PID controller parameter tuning in a didactic thermal plant. *IEEE Lat. Am. Trans.* **2017**, *15*, 1250–1256. [CrossRef]
8. Wang, J.S.; Yang, G.H. Data-driven approach to accommodating multiple simultaneous sensor faults in variable-gain PID systems. *IEEE Trans. Ind. Electron.* **2009**, *66*, 3117–3126. [CrossRef]
9. Liu, J.; Li, T.; Zhang, Z.; Chen, J. NARX prediction-based parameters online tuning method of intelligent PID system. *IEEE Access* **2020**, *8*, 130922–130936. [CrossRef]
10. Zhou, R.; Tan, W. Analysis and tuning of general linear active disturbance rejection controllers. *IEEE Trans. Ind. Electron.* **2019**, *66*, 5497–5507. [CrossRef]
11. Verma, B.; Padhy, P.K. Optimal PID controller design with adjustable maximum sensitivity. *IET Control Theory Appl.* **2018**, *12*, 1156–1165. [CrossRef]
12. Nguyen, N.H.; Nguyen, P. Overshoot and settling time assignment with PID for first-order and second-order systems. *IET Control Theory Appl.* **2018**, *12*, 2407–2416. [CrossRef]
13. Åström, K.J.; Hägglund, T.; Hang, C.C.; Ho, W.K. Automatic tuning and adaptation for PID controllers-a survey. *Control Eng. Pract.* **1993**, *1*, 699–714. [CrossRef]
14. Maddahi, A.; Sepehri, N.; Kinsner, W. Fractional-order control of hydraulically powered actuators: Controller design and experimental validation. *IEEE ASME Trans. Mechatron.* **2019**, *24*, 796–807. [CrossRef]
15. Bharat, S.; Ganguly, A.; Chatterjee, R.; Basak, B.; Sheet, D.; Ganguly, A. A review on tuning methods for PID controller. *Asian J. Converg. Technol. (AJCT)* **2019**, *5*, 818–827.
16. Wang, L.; Cluett, W.R. Tuning PID controllers for integrating processes. *IEEE Proc. Control Theor. Appl.* **1997**, *144*, 385–392. [CrossRef]
17. Vrančić, D.; Huba, M.; Oliveira, P.M. PID controller tuning for integrating processes. *IFAC-PapersOnLine* **2018**, *51*, 586–591. [CrossRef]
18. Atic, S.; Cokmez, E.; Peker, F.; Kaya, I. PID controller design for controlling integrating processes with dead time using generalized stability boundary locus. *IFAC-PapersOnLine* **2018**, *51*, 924–929. [CrossRef]
19. Bazanella, A.S.; Pereira, L.F.A.; Parraga, A. A new method for PID tuning including plants without ultimate frequency. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 637–644. [CrossRef]
20. Farah, N.; Nizam, H.; Shahida, N.; Abdullah, Q.; Ibrahim, Z.; Binti, J.; Jidin, A. A Novel Self-Tuning Fuzzy Logic Controller Based Induction Motor Drive System: An Experimental Approach. *IEEE Access* **2019**, *7*, 68172–68184. [CrossRef]
21. Gil, P.; Lucena, C.; Cardoso, A.; Palma, L.B. Gain tuning of fuzzy PID controllers for MIMO systems: A performance-driven approach. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 757–768. [CrossRef]
22. Yordanova, S.; Merazchiev, D.; Jain, L. A two-variable fuzzy control design with application to an air-conditioning system. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 474–481. [CrossRef]
23. Khorramabadi, S.S.; Bakhshai, A. Critic-based self-tuning PI structure for active and reactive power control of VSCs in microgrid systems. *IEEE Trans. Smart Grid* **2015**, *6*, 92–103. [CrossRef]
24. Precup, R.E.; David, R.C.; Petriu, E.M. Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity. *IEEE Trans. Ind. Electron.* **2017**, *64*, 527–534. [CrossRef]
25. Osinski, C.; Leandro, G.V.; da Costa Oliveira, G.H. Fuzzy PID controller design for LFC in electric power systems. *IEEE Lat. Am. Trans.* **2019**, *17*, 147–154. [CrossRef]
26. Ali, J.A.; Hannan, M.A.; Mohamed, A.; Abdolrasol, M.G. Fuzzy logic speed controller optimization approach for induction motor drive using backtracking search algorithm. *Measurement* **2015**, *78*, 49–62. [CrossRef]
27. Thao, N.G.M.; Uchida, K.; Kofuji, K.; Jintsugawa, T.; Nakazawa, C. An automatic-tuning scheme based on fuzzy logic for active power filter in wind farms. *IEEE Trans. Control Syst. Technol.* **2019**, *27*, 1694–1702. [CrossRef]
28. Chen, J.; Xu, C.; Wu, C.; Xu, W. Adaptive fuzzy logic control of fuel-cell-battery hybrid systems for electric vehicles. *IEEE Trans. Ind. Inform.* **2018**, *14*, 292–300. [CrossRef]
29. Kumbasar, T. Robust stability analysis and systematic design of single-input interval type-2 fuzzy logic controllers. *IEEE Trans. Fuzzy Syst.* **2016**, *24*, 675–694. [CrossRef]

30. Reza, A.H.; Javadi, S.; Reza, S.N. DC motor speed control by self-tuning fuzzy PID algorithm. *Trans. Inst. Meas. Control* **2014**, *37*, 164–176. [CrossRef]
31. Hjalmarsson, H.; Gunnarsson, S.; Gevers, M. A convergent iterative restricted complexity control design scheme. In Proceedings of the 1994 33rd IEEE Conference on Decision and Control, Lake Buena Vista, FL, USA, 14–16 December 1994; pp. 1735–1740. [CrossRef]
32. Hjalmarsson, H.; Gevers, M.; Gunnarsson, S.; Lequin, O. Iterative feedback tuning: Theory and applications. *IEEE CS-M* **1988**, *18*, 26–41. [CrossRef]
33. Campi, M.C.; Lecchini, A.; Savaresi, S.M. Virtual reference feedback tuning: A direct method for the design of feedback controllers. *Automatica* **2002**, *38*, 1337–1346. [CrossRef]
34. Soma, S.; Kaneko, O.; Fujii, T. A new approach to parameter tuning of controllers by using one-shot experimental data—A proposal of fictitious reference iterative tuning. *Trans. ISCIE* **2004**, *17*, 528–536. [CrossRef]
35. Kaneko, O.; Soma, S.; Fujii, T. Fictitious reference iterative tuning in the two-degree of freedom control scheme and its application to a facile closed loop system identification. *Trans. ISCIE* **2006**, *42*, 17–25. [CrossRef]
36. Kano, M.; Tasaka, K.; Ogawa, M.; Takinami, A.; Takahashi, S.; Yoshii, S. Extended fictitious reference iterative tuning and its application to chemical processes. In Proceedings of the 2011 International Symposium on Advanced Control of Industrial Processes (ADCONIP), Hangzhou, China, 23–26 May 2011; pp. 379–384.
37. Masuda, S. Iterative Controller Parameters Tuning Using Gradient Estimate of Variance Evaluation. In Proceedings of the 2019 International Conference on Advanced Mechatronic Systems (ICAMechS), Kusatsu, Japan, 26–28 August 2019; pp. 272–277. [CrossRef]
38. Marcel, F.H.; Bart, V.V. Constrained iterative feedback tuning for robust control of a wafer stage system. *IEEE Trans. Control Syst. Technol.* **2016**, *24*, 56–66. [CrossRef]
39. Precup, R.E.; Preitl, Z.; Preitl, S. Iterative feedback tuning approach to development of PI-fuzzy controllers. In Proceedings of the 2007 IEEE International Fuzzy Systems Conference, London, UK, 23–26 July 2007; pp. 1–6. [CrossRef]
40. Precup, R.E.; Radac, M.B.; Tomescu, M.; Petriu, E.M.; Pretil, S. Stable and convergent iterative feedback tuning of fuzzy controllers for discrete-time SISO systems. *Expert Syst. Appl.* **2012**, *40*, 188–199. [CrossRef]
41. Khokhar, S.; Peng, Q.; Asif, A.; Yasir, M.; Inam, A. A Simple Tuning Algorithm of Augmented Fuzzy Membership Functions. *IEEE Access* **2020**, *8*, 35805–35814. [CrossRef]
42. Sarif, B.M.; Kumar, V.A.; Rao, V.G. Comparison Study of PID Controller Tuning using Classical/Analytical Methods. *Int. J. Appl. Eng. Res.* **2018**, *13*, 5618–5625.