

Study of Fuzzy Controllers Performance: Application on Embedded Systems

L. M. MUNIZ
CEFET-MG

Department of Eletroelectronics
José Peres Street, 558, Leopoldina
BRAZIL

M. J. CARMO
CEFET-MG

Department of Eletroelectronics
José Peres Street, 558, Leopoldina
BRAZIL

M. F. SANTOS
CEFET-MG

Department of Eletroelectronics
José Peres Street, 558, Leopoldina
BRAZIL

A. F. SANTOS NETO
CEFET-MG

Department of Eletroelectronics
José Peres Street, 558, Leopoldina
BRAZIL

P. MERCORELLI

Leuphana University of Lüneburg
Institute of Product and Process Innovation
D-21335, Lüneburg
GERMANY

Abstract: Nowadays the industrial field is testing more robust controllers to deal better with complex control processes, to guarantee high reliability and efficiency. In this context, this work aims to present the design of a Fuzzy Controller embedded in an Arduino Mega microcontroller, taking 2 experimental didactic plants into consideration: a resistor-capacitor circuit and a direct current motor assembly. Then, the Fuzzy Controller will have its effectiveness and also be compared with other topology: the parallel PID controller. In addition, an interesting study is done when the storage capacity of routines in the Arduino board is considered, regarding the dynamic memory when new membership functions and rules are increased. The Fuzzy controller was embedded obtaining satisfactory results when compared to the PID controller, taking non-intrusive performance indices as validation parameters.

Key-Words: Fuzzy Controller, Arduino Microcontroller, PID Controllers, Embedded Systems.

Received: November 28, 2019. Revised: March 27, 2020. Re-revised: April 27, 2020. Accepted: May 4, 2020. Published: May 12, 2020.

1 Introduction

There are different types of control systems, but the Proportional-Integral-Derivative (PID) controller is still one of the most used topology [1]. In some cases, where the PID does not present acceptable results (concerning dynamic behavior), Artificial Intelligence, Fuzzy logic and Neural Network have been gaining applicability.

In this context, Fuzzy logic is one of the approaches used and generally implemented in processes whose mathematical modeling is complex. Furthermore, this technique aims to create a system capable of representing complex, indefinite, controversial or incomplete knowledge, mathematically and logically [2].

In this type of controller, the most important part is the operator experience in the plant, once the language expressions relate the measured error to the controller performance in the industrial field controller. In other words, the Fuzzy controller tries to approximate the human behavior in a similar way as

possible in situations where the mathematical model is difficult to obtain and when there are uncertainties, time-varying or unknown parameters [3].

Some interesting works that applied Fuzzy logic in industrial systems are highlighted hereafter. Author of paper [4] developed a method capable of structuring a Fuzzy Logic Controller (FLC) based on decision tables applied to various industrial systems (tank level, Direct Current (DC) motor, temperature system), where all FLC parameters were tuned on-line; work [5] presents the design of signed distance method based on FLC, implemented on industrial scale polymerization reactor, which is the modified strategy to design a conventional FLC; [6] discusses the importance of FLC based on intelligent controller designs for temperature process control of an industrial furnace system comparing the conventional PID controller to the FLC; [7] proposed a FLC with adaptive nature to control the fan and compressor speed of air-conditioning, showing results with reduced power consumption; The authors from work [8]

presented a hierarchical force control framework consisting of a high-level control system based on Fuzzy logic and the existing motion system of a low level manipulator.

Still in this development context, this work seeks to develop a FLC to control two applications with distinct dynamics. Unlike the works mentioned, this one will bring small system applications, but with the differential of presenting the Fuzzy controller implementation specificities. Using a low cost microcontroller, this work presents the study of the FLC (type Mamdani Fuzzy). The FLC system developed will be tested in 2 didactic control plants: a Resistor-Capacitor (RC) circuit and a DC motor for speed control. Besides, the results will be compared to an industrial control system widely used in industry: the parallel PID controller. Finally, the conclusions will be depicted on the Fuzzy rule number influence in the developed prototypes, as well as for some aspects of memory allocation when increasing these rules.

This work is divided as follows: Section 2 shows the 2 prototypes developed to test the Fuzzy controller designed; Section 3 presents the Fuzzy logic, as well as its rules and pertinence functions; Section 4 displays the PID controllers just to compare with Fuzzy controller performance; Section 5 shows some experimental results; at the end, Section 6 concludes the results and present some future works.

2 The Two Prototypes Developed

Two plants were developed to analyze the Fuzzy controller performance. Both were tested using the Arduino Mega prototyping platform.

The Arduino has been widely used to control systems with low cost microcontrollers, and also for its simplicity of operation. Figure 1 illustrates the Arduino Mega board used to implement the Fuzzy controller and the Integrative Programming Environment (IDE) used.

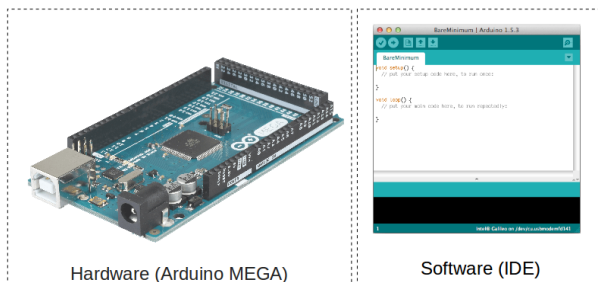


Figure 1: Arduino Mega board and IDE.

The first plant, represented in Figure 2, consists

of a system composed of 2 capacitors and 2 resistors, where the voltage in the second resistor is measured and controlled. In this plant, the circuit is supplied through the pin V_{in} , representing the control action coming from the controller. The pin marked V_{out} represents the controlled signal, i. e., the controlled variable.

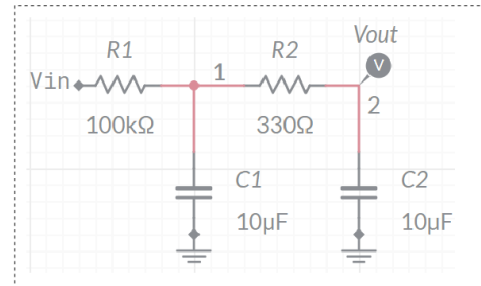
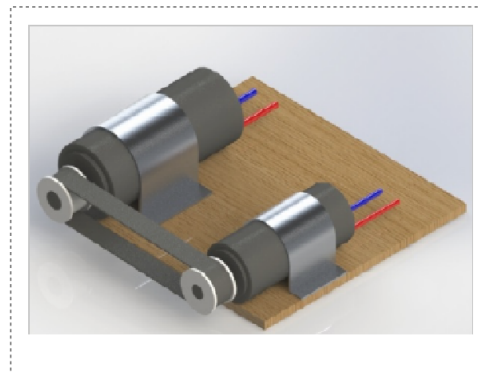
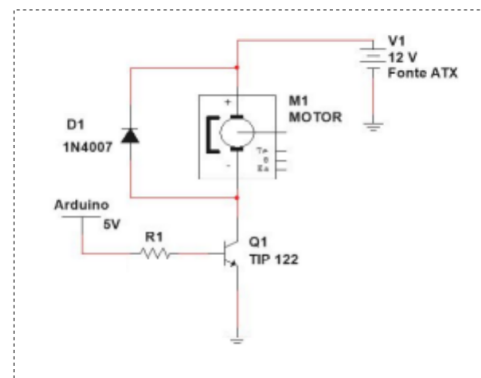


Figure 2: Electrical circuit prototype.

The second plant is a DC motor speed control made of 2 DC motors and the drive system electronic to operate one of the motors, which controls the velocity of the other one. Figure 3 shows the DC motor schematic prototype and the electronic drive system.



(a) Schematic of DC motors.



(b) Drive system electronic.

Figure 3: DC motor prototype.

As can be seen in Figure 3, the plant uses DC motors (Figure 3a), where the first motor operates as a DC motor, and the second one as a DC generator, driven by a plastic belt. The drive circuit (see Figure 3b) was necessary to integrate the Arduino into the system. In this way, a circuit composed of an amplifier with common-emitter was structured, in addition to a protection diode for reverse currents (“free wheel” diode). Also, Table 1 shows the drive circuit components and the DC motor characteristics.

Table 1: Drive circuit components.

Component	Specification
Transistor	TIP122
Resistor (R1)	330 Ω
Diode (D1)	1N4007
Source	ATX (12V/60W)

It is important to mention that the plant measurements are done through the analog pin available on the Arduino, which has an analog-to-digital converter of 10 bits, allowing the conversion of 0 to 5V. For the controller output, a digital pin of the (Pulse Width Modulation) PWM type was used.

3 Fuzzy Logic

The main reason for the success of Fuzzy control technique was due to its definition by linguistic components, which allows the less familiar user with the process model to write complex algorithms without the system mathematical description [3].

As it is known, a Fuzzy controller contains 4 main steps: fuzzification, definition of basic rules, inference and defuzzification. These steps, illustrated in Fig. 4, allow non-fuzzy inputs, such as the error of a controlled variable, to be converted into intervention actions (non-fuzzy outputs) and control the system, when well defined.

The fuzzification step is where a real value will be converted to a Fuzzy value, i. e., it gets a degree of “truth” or pertinence value from 0 to 1 where each entry belongs to a certain Fuzzy set. In a Fuzzy set, an element cannot just be characterized as it belongs or does not belong, it can present any state, with a value within the real range $[0 - 1]$ [10].

The rule base can be defined by a specialist through test field or using numerical data. This is the part of how the system knowledge works, since it defines the control actions.

In the inference stage, the Mamdani method was used, which bases on the intersection of Fuzzy sets

through the logical operator “and”. Finally, the defuzzification will convert the Fuzzy output to a real value, to be driven to the system actuator.

For the adopted system, 2 different variables were used: the controlled variable error and its respective variation, shown in Fig. 5.

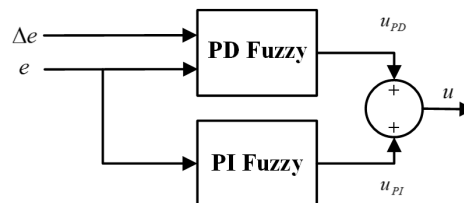


Figure 5: Illustration of the Fuzzy controller implemented.

where e represents the error, Δ_e the error variation, u_{PD} and u_{PI} are the derivative and integral control actions, respectively, and u is the resulting control action.

In both prototypes, several projects were created with different amounts of rules: 9, 25, 49, 81, 121, 169 and 225.

To simplify the rules creation procedure through the pertinence functions, it is shown in Table 2 the aspect of a system with only 3 pertinence functions for each variable (error and its variation), where it is possible to expand it to 225 rules.

Table 2: Example of rules for Fuzzy system: 3 pertinence functions.

		Variation of error		
		Low	Medium	High
Error	Low	Zero	Zero	Zero
	Medium	Zero	SetPoint	Max
	High	Max	Max	Max

From the procedure shown above, it is possible to see that when the error is close to zero, it is desired to follow the setpoint value; when it is negative, it is intended to have zero performance; and when there is a very big error, the objective is to perform the maximum system actuation. This definition will occur for all future expanded mappings.

It is worth adding that the code was implemented using the Arduino IDE. Also, another important information is about the sampling time, which must be enough for the Fuzzy controller to make its decision. Therefore, it will depend on the system time constant.

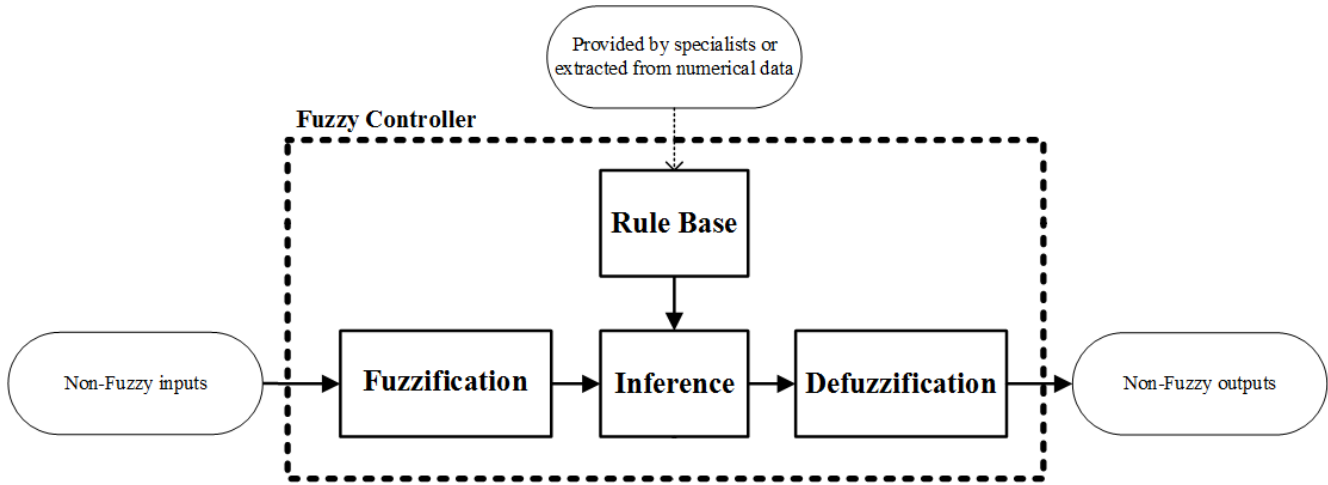


Figure 4: Fuzzy controller with the 4 main steps illustration.

4 PID Controllers

One way to check the Fuzzy controller performance is to compare it with another controller. In this case, it was chosen the parallel PID topology. Basically, its transfer function is presented in (1):

$$C_{pid}(s) = K_P + \frac{K_I}{s} + K_D s \quad (1)$$

where K_P , K_I e K_D represent the proportional, integral and derivative gains, respectively.

To tune a PID controller mathematically (which is finding the parameters K_P , K_I e K_D) is necessary to have the system model, which can be obtained in two ways: from phenomenological modeling or from experimental procedures [11]. In this work, the open-loop identification method was adopted using the First Order plus Dead Time (FODT).

This model is widely used for modeling industrial systems and it is suitable for many 1st and 2nd order systems. It is also important to mention that this model proved to be adequate in tests carried out by the authors. Furthermore, this model allows the PID controller tuning through classic methods [11].

Also, the identification and tuning procedures will be presented separately according to each prototype. By the way, the PID controller gains can be obtained empirically, which can be dangerous in some industrial processes.

4.1 RC Circuit System

According to the controller tuning techniques chosen in this work, it is necessary to get a system transfer function in the FODT. Using Smith identification method, (3) is obtained [12]:

$$G(s) = \frac{K}{\tau s + 1} e^{-Ls} \quad (2)$$

$$= \frac{2.48}{1.98s + 1} e^{-0.02s} \quad (3)$$

where K is the static gain, L is the transport delay and τ is the time constant.

For the PID controller, the first method of Ziegler-Nichols was used, resulting in $K_P^{ZN} = 7.79$, $K_I^{ZN} = 20.5$ and $K_D^{ZN} = 0.59$ [13].

4.2 DC Motor System

Following the previews section methodology, the system open-loop transfer function was acquired taking (2) into account, also through the Smith identification method [12]:

$$G(s) = \frac{2.32}{0.073s + 1} e^{-0.029s} \quad (4)$$

For the system proposed in this section, a PI (Proportional-Integral) controller was used to control this plant. The choice is due to the DC motor system dynamic characteristics.

In this case, the classical methods considered were Chien, Hrones and Reswick (CHR) and Internal Model Control (IMC), where their respective gains are shown in the Table 3 [14, 15].

Table 3: Tuning of the PID controller: DC motor system.

Method	Proportional Gain	Integral Gain
CHR	$K_P^{CHR} = 0.37$	$K_I^{CHR} = 4.46$
IMC	$K_P^{IMC} = 0.64$	$K_I^{IMC} = 7.39$

4.3 Implementation

There are some forms to implement the PID controller in the Arduino board, such as using difference equation or through the “myPID” library (used in this work). Thus, to implement the controller, it is necessary to provide the its respective gains and setpoint. More information about the PID controller algorithm are presented in the Figure 6.

```
#include <PID_v1.h>
//Variables definition
double Setpoint, Input, Output;
double Kp = XX, Ki=YY, Kd = ZZ;
// Setting the Variables PID Function

PID myPID(&Input, &Output, &Setpoint,Kp,Ki,Kd, DIRECT);

void setup()
{
  Setpoint = 100;           // Define Setpoint
  myPID.SetMode(AUTOMATIC); // Initialize - PID mode
}

void loop()
{
  Input = analogRead(0);    // Sensor Read
  myPID.Compute();          // Compute PID output
  analogWrite(3,Output);    // System Intervention
}
```

Figure 6: PID controller algorithm.

5 Experimental Results

Three subsections are presented for a better arrangement of the results, one for each prototype and one displaying the computational effort required to run the tests regarding the microcontroller memory allocation performance.

All the tests are from experimental results, obtained through the Arduino analog pin. The experiments were developed by establishing a fixed setpoint, where the system always starts from zero voltage or from rest.

For both prototypes, the sensors measurements were treated through a moving average filter to attenuate noises. Also, the sampling time used for the entire Fuzzy controller decision process was 10ms, which includes the time constants of the RC circuit plant (1.98sec) and the DC motor system (78ms).

5.1 RC Circuit System

Figure 7 shows 7 Fuzzy systems developed to control the output voltage V_{out} .

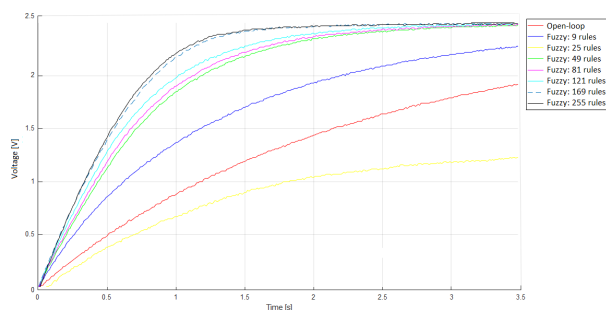


Figure 7: Fuzzy controller responses considering 7 configurations of rules.

Observing Fig. 7, it is verified how fast the system reacts, explained by the addition of Fuzzy rules and consequently the better mapping of the universe in error speech and its variation. It is also possible to notice that from 15 pertinence functions (225 rules), it is not noted a considerable difference with the mapping.

Adopting the response with 15 pertinence functions for each variable used (error and its variation), Fig. 8 illustrates a comparison between the experimental results of the Fuzzy system and the PID controller, described in Section 4.1.

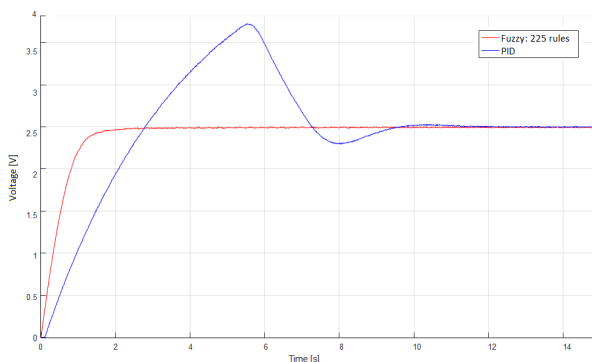


Figure 8: Comparison between Fuzzy and PID controlled responses considering 15 pertinence functions and 225 rules.

As can be seen above, the Fuzzy controller response showed to be faster and more adequate than the PID controller responses (considering the overshoot, it was approximately 48%).

5.2 DC Motor System

For this system, preliminary tests also showed that the responses with greater rules present better performances when compared to the others with fewer rules, as expected. Thus, the experimental result comparing the Fuzzy system and the CHR and IMC controllers are presented in Fig. 9.

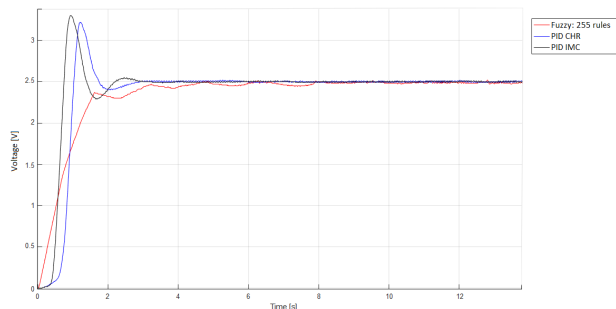


Figure 9: Comparison between Fuzzy and PID controlled responses considering 15 pertinence functions and 225 rules.

It can be seen that the Fuzzy controller did not obtain any overshoot, whereas the response generated by CHR and IMC controllers showed 24% and 28% overshoots, respectively. As for the rise time, both presented values close to each other. However, it can be emphasized that the Fuzzy controller showed to be more efficient.

5.3 Computational Effort to Perform the Tests

The insertion of pertinence functions and rules in the Fuzzy system resulted in a greater computational effort, adding new variables in the method, resulting in a greater use of the embedded microcontroller dynamic memory. This allocation data are presented in Table 4.

Table 4: Analysis of the memory allocation.

Allocation Memory (Bytes)	Number of Rules
7026	9
11924	25
20656	47
29926	81
43704	121
62758	169
83048	225

There was a considerable increase in the storage space of routines, which justifies the use of Arduino

Mega instead of Arduino Uno. Otherwise, the Arduino Uno would not be able to store the routines for 169 and 225 rules.

It is also noticed that even increasing the routine storage space and consequently the dynamic memory through global variables, it was still more interesting to use 15 pertinence functions and 225 rules for the best case. In other words, the use of more memory allocation and space did not affect data processing, which implies that the control with more rules and functions became feasible.

6 Conclusions

The industrial controllers are continually changing, and new techniques are emerging and increasing, thanks to methodologies that use AIs to optimize the results, making the professionals to be aware and continuously updated to the different projects arising. It is in this context that control education in universities has a significant focus on the new PID control technique dissemination, where new typologies (such as FLC) have taken an important place in processes with difficult mathematical modelling.

The current work had the objective of implementing a Fuzzy controller in 2 different systems: a RC circuit and a DC motor. The 2 systems can be considered linear, but with the insertion of the microcontroller device, the system saturation from 0 to 5V inserts nonlinear characteristics, for example.

In the PID controller system, many tuning were tested, not counting the need to perform its mathematical modelling, but to obtain a precise and stable response, since it became a nonlinear system.

In the Fuzzy controller, the tuning process was shorter, since it is not affected by the system nonlinearities as the PID controller is. It is worth mentioning that the change in the PID controller tuning can improve or worse its performance, and may not always have inferior answers to Fuzzy, as it highly depends on the operator/user experience.

Although the systems became nonlinear, the Fuzzy controller succeeded in its purpose by finding more satisfactory results, taking the error to zero in both studied systems.

The main difference between the implemented techniques is the Fuzzy controller ease implementation, since it does not require system modelling as accurate as is in the PID.

Another benefit of the Fuzzy Control technique is the facilitated controller configuration, either by inserting pertinence functions or by adding new rules to even another system performance.

6.1 Future Works

The conclusion of this work opens some future works:

- Through the insertion of new types of pertinence functions, new inferences techniques and even different defuzzification methods in the Fuzzy system can be studied, also in order to measure its improvement;
- Use of new plants to test the implemented Fuzzy system reliability;
- Compare the implemented Fuzzy system with others PID Controller techniques.

Acknowledgment

The authors would like to thank CEFET-MG and Lephana University of Lüneburg for the financial support.

References:

- [1] M. Bauer, A. Horch, L. Xie, M. Jelali, and N. Thornhill, "The current state of control loop performance monitoring—a survey of application in industry," *Journal of Process Control*, vol. 38, pp. 1–10, 2016.
- [2] B. Bouchon-Meunier, "La logique floue et ses applications." 1995.
- [3] M. G. Simoes and I. S. Shaw, "Controle e modelagem fuzzy," *Editor Blucher: FAPESP, São Paulo*, 2007.
- [4] L. Mastacan and C. Dosofoi, "Fuzzy logic controller design and implementation for industrial applications," 2014.
- [5] A. Singh and P. S. Londhe, "Design of signed distance method based fuzzy logic controller for tito," 2017.
- [6] B. V. Murthy, Y. V. P. Kumar, and U. V. R. Kumari, "Fuzzy logic intelligent controlling concepts in industrial furnace temperature process control," 2012.
- [7] N. U. Ahamed, Z. B. Taha, I. B. M. Khairuddin, M. F. Rabbi, S. A. M. M. Rahaman, and K. Sundaraj, "Fuzzy logic controller design for intelligent air-conditioning system," 2016.
- [8] S.-T. Lin and A.-K. Huang, "Hierarchical fuzzy force control for industrial robots," *IEEE Transactions on Industrial Electronics*, 1998.
- [9] J. Sobota, R. PiŠl, P. Balda, and M. Schlegel, "Raspberry pi and arduino boards in control education," *IFAC Proceedings Volumes*, vol. 46, no. 17, pp. 7–12, 2013.
- [10] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.
- [11] K. J. Åström and T. Hägglund, *PID controllers: theory, design, and tuning*. Instrument society of America Research Triangle Park, NC, 1995, vol. 2.
- [12] C. L. Smith, "Digital computer process control," *Scranton, PA, EUA, International Textbook Co*, 1972.
- [13] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Transaction of ASME*, vol. 64, no. 11, 1942.
- [14] K. L. Chien, "On the automatic control of generalized passive systems," *Trans. Asme*, vol. 74, pp. 175–185, 1972.
- [15] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control: PID controller design," *Industrial & engineering chemistry process design and development*, vol. 25, no. 1, pp. 252–265, 1986.