

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Advantages of Fuzzy Control While Dealing with Complex/Unknown Model Dynamics: A Quadcopter Example

Luis Ibarra and Colin Webb

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62530>

Abstract

Commonly, complex and uncertain plants cannot be faced through well-known linear approaches. Most of the time, complex controllers are needed to attain expected stability and robustness; however, they usually lack a simple design methodology and their actual implementation is difficult (if not impossible). Fuzzy logic control is an intelligent technique which, on its basis, allows the translation from logic statements to a nonlinear mapping. Although it has been proven to effectively deal with complex plants, many recent studies have moved away from the basic premise of linguistic interpretability. In this work, a simple fuzzy controller is designed in a clear way, privileging design easiness and logical consistency of linguistic operators. It is simulated together to a nonlinear model of a quadcopter with added actuators variability, so the robust operation of the controller is also proven. Uneven gain, bandwidth, and time-delay variations are applied among quadcopter's motors, so the simulations results enclose those characteristics which could be found in reality. As those variations can be related to actuators' performance, an analysis can be driven in terms of the features which are not commonly included in mathematical models like power electronics drives or electric machinery. These considerations may shorten the gap between simulation and actual implementation of the fuzzy controller. Briefly, this chapter presents a simple fuzzy controller which deals with a quadcopter plant as a first approach to intelligent control.

Keywords: Fuzzy control, intelligent control, quadcopter, robust control, simulated uncertainty, artificial intelligence, non-parametric variability

1. Introduction

Power electronics systems are commonly comprised of nonlinear time-varying components which impose an overall complex dynamic behavior. As these systems are commonly integrated together as actuators with electric machinery or other transducers, the resulting plants are hard to analyze, model, and control. As linear models and linear controllers have been thoroughly studied, linear approximations to such systems are desired. However, such simplifications omit some important features of the physical media and lead to higher controller burdens or impossible realization [1]. Consequently, power electronic systems in plants are dealt with using three main methods: (1) facing their complexity by using difficult approaches, (2) simplifying their features and constraining controller design, or (3) finding nonparametric models which approximately capture the system behavior.

As control specifications cannot be followed by traditional controllers, artificial intelligence (AI) can be incorporated to the control strategy to achieve higher goals [2]. AI is a set of computational techniques which try to imitate human brain processing or practices found in nature to solve a given problem. For control systems, these methodologies can provide acceptable, or even improved, performance based on any amount of previous knowledge [3]. The increased usage of AI and its increased acceptance are mostly due to some important characteristics. Firstly, they can deal with uncertainty [4] as they can be adapted to a system's variations, sensor noise, or be defined in an uncertain manner [1]. Secondly, they are versatile, so mixed methods involving two or more AI techniques and traditional methods are readily found in literature as pointed in [2, 5, 6]. Finally, they are general, so their application to different case studies is possible and facilitates their continuous improvement. They have been proven to work properly as a direct substitution for linear controllers (as pointed in [7] on fuzzy systems) and as effective alternatives for controlling nonlinear, and uncertain plants [8].

Fuzzy systems (FS) compose one major part of the intelligent control spectrum. They are based on fuzzy logic first proposed by Zadeh [9], which expands set theory in such a way that an element can belong to a set with some membership value, instead of the traditional binary belongingness of "true" or "false". Although fuzzy control will be later discussed in detail, it is worth mentioning that it provides a way to deal with uncertainty through linguistic "elastic" categories and logic inferential rules. Narrowing the distance from human logic to numerical implementation has permitted control systems to be defined in terms of linguistic control rules from experts [6]. This is possible as fuzzy logic control (FLC) specifically converts linguistic rules into a nonlinear mapping [10], allowing dynamic modeling and controller description to be done as simple linguistic statements. The previously presented point of view related to designer experience represents only the most basic approach to FLC; however, it stands as an effective control technique which requires little knowledge about the plant [10, 11].

There are more specific reasons why FLC can be preferred over or together with traditional techniques. Linder and Shafai [12] specify that a quantitative controller's refinement tuning cannot be properly performed, as there is no guidance on the meaning of each parameter. They specify that the controller's refinement tuning can be properly executed when the controller

is built over linguistic rules that preserve their actual meaning. They also say that achieving a quantitative control of nonlinear systems is difficult and not quite realizable. This last statement is confirmed by Precup and Preitl [7] who use a model-based FLC constrained by H-norms. FLC can be implemented despite the complexity as commented by Khan et al. [13] who presented a fuzzy controller tuned through genetic algorithms: "...the unnecessary mathematical rigorosity, preciseness and accuracy involved with the design of the controllers have been a major drawback". They show that if FLC cannot cope with specific desired controller characteristics, there are intelligent ways in which the controller can be tuned, avoiding experts' possible limitations and preserving interpretability. There is no intention from the authors of this chapter to discredit advanced traditional control techniques; however, there are many works which point to real benefits of using intelligent control.

FLC has suffered of criticism for lacking a systematic design approach and a stability analysis method; in addition, the high cost of setting up a FLC as a replacement of an industrial standard PID has discouraged its usage [10]. The PID controller has been used not only as a standard controller but also as a reference for comparison when it comes to evaluating a controller's performance. Most control designers are familiar with PID, which is taken as an easy alternative and a starting point when facing a given plant. The familiarity to PID controller, its known performance, and its relative implementation feasibility have obscured some of the clear benefits of FLC. Nevertheless, Nithya et al. [14] shows that FLC exhibits an improved performance over PID controllers when dealing with first-order plus dead time systems; Gupta et al. [11] remark that nonlinear characteristics make PID unsuitable while intelligent control can cope with control specifications; and Linder and Shafai [12] and Kalavathi and Reddy [6] say that FLC greatly outperforms PI/PID controllers. In some cases, FLC is becoming an immediate choice when dealing with nonlinear, time-delayed, and high-order systems [15]; parameter variance has also been properly tackled by FLC [7]. Altogether, it is accepted as a better alternative than its PID counterpart [4]. As said by Precup and Preitl about fuzzy control: *"The use of linguistic rules and approximate reasoning is a simple and easy way to solve concrete control problems. Sophisticated models and control design tools are not necessary. It is sometimes the only way to begin to approach the control of a complex, uncertain, or ill-defined plant"* [7].

Some efforts have also been made to expand FLC application by finding similarities to traditional controllers, providing model-based approaches, integrating with other traditional/intelligent techniques, and incorporating features like adaptability to FLCs. Some examples of the aforementioned directions can be found in works like [2] where the equivalence of a FLC with a PID controller together with a relay is explained. Self-regulated fuzzy controllers which incorporate adaptability are also found in the literature as in [4], and robustness enhancement has been achieved by merging fuzzy logic with sliding-mode control as presented in [16] and by Feng [10] who presents a survey on model-based FLC. However, FLC is known as a robust alternative on its own merits [5]. Measuring noise can be incorporated directly if FLC is designed based on type-2 fuzzy logic [1], and simple applications are possible in industrial standard controllers as reported in [4] where an OMRON PLC is used.

Fuzzy logic, as a fundamental part of intelligent control, can face complex systems with relative ease, still providing robustness and logical interpretability. Quadcopters are a great example

of such systems which integrate uncertainty and parameter variability, electric machinery, and power electronics; due to their nature, they seem as a good test-bench to evaluate FLC performance. Quadcopters have gained some field in the research area as they are particularly well suited for unmanned tasks such as search and rescue, surveillance, remote inspection, and as military units [17]. These tasks involve parameters which are not easily obtained or estimated, and can result in uncertain modeling ([17, 18]) which must be alleviated by dependable controllers. Altogether, quadcopters represent a challenge for control designers nowadays [16].

Most of the time, mathematical models cannot fully represent the physical reality of a given plant, and in the case of quadcopters, the inner processing, power electronics, and electric machinery are often assumed not to impose restrictions to system performance. Actually, the quadcopter's motor information is normally unknown, so approximation techniques are needed as in [17]; their response to input desired values is normally assumed to be immediate, repeatable, and reliable even when complete studies about BLDC motors have shown the opposite as in [11]. All aforementioned subsystems are complex and would need a deep mathematical analysis to provide valid simulated results when integrated. Achteik et al. [18] propose a non-intelligent controller and specifically say that quadcopter modeling must avoid simplifications like linearization and that algorithms must be exact as the update rates of processing units are low. Li and Song [5] also identify the overall quality of the model as problematic for control design. Actual implementation also requires the measured data to be "clean," and sometimes redundancy is recommended as noise is unavoidable [19]. It is quite obvious that no control technique will tackle all of these problems at once due to increased complexity; however, model simplification together with uncertainty considerations could lead to a valid solution focusing on robustness against model uncertainties and sensors noise.

In this spirit, many works have simulated, tested, and compared traditional simple, complex, and intelligent techniques over quadcopter devices and models. A short survey about non-intelligent control techniques applied to quadcopters was presented by Li and Song [5] specifically covering the PID, LQR, H, sliding-mode, feedback linearization, adaptive control, and backstepping techniques, presenting their individual benefits and drawbacks. An adaptable controller based on a real-time quadcopter model was tested by Achteik et al. [18] showing great performance even with complete parametric ignorance; Mirzaei et al. [16] present a fuzzy indirect adaptive sliding-mode controller which provide robust performance toward noise and plant uncertainties.

It is not the main aim of this chapter to present a novel control technique or to solve all aforementioned issues regarding quadcopter control, but to test the effectiveness of simple FLC to face this complex plant, focusing on the effects of assuming quadcopter's subsystems to be non-ideal. As most traditional simple controllers require assumptions about the specific quadcopter dynamics, or limitations on the controllers to provide a mostly hovering behavior, the approach of this work is to show how linguistic interpretability can cope with less conservative control specifications while ensuring a certain amount of inherent robustness. For this purpose, a fuzzy controller will be designed and simulated together with an ideal quadcopter model so that a robustness comparison can be made later, when the processing

unit, power electronics, and electric machinery are assumed to incur a time delay and parametric variance. This analysis provides some insight to the “natural” benefits of intelligent control and its great potential to confront complex scenarios (besides its criticized heuristic approach), while presenting a simple and effective way to deal with quadcopter dynamics with some commentary facing actual implementation of the controller.

Nevertheless, fuzzy control has also been used from more formal points of view which consider model-dependency, self-tuning, stability analysis, hybrid methodologies, etc. Chen et al. [20] present some major ways to incorporate FLC on robust control study area; Li and Ruan [4] show a very simple way to provide adaptability to a FLC, while Li et al. [2] provide adaptability to a PID controller through fuzzy logic and Chen et al. [21] to a smith predictor. Complex systems can also be modeled through Takagi-Sugeno (TS) fuzzy models [10] and some plants have been controlled by TS controllers and TS observers as shown in [3]. TS fuzzy systems are also behind gain scheduling topologies like those presented in [7] and [22]. Other intelligent techniques can be incorporated to enhance the design process, as in [13] where genetic algorithms are used to tune the FLC to guarantee optimal performance. Gupta et al. [11] show how FLC has been merged with artificial neural networks and traditional PID controllers to control BLDC motors. Lastly, Nithya et al. [14] show tuning schemes, auto-tuning, and model-based tuning FLC for first-order plus dead time models (FOPDT) in their cited literature; they propose a simple Mamdani FLC to deal with these kinds of systems.

We will use a Mamdani FLC which is opportunely simplistic in nature. It shows how it is adequate to take one of the simplest fuzzy topologies to deal with a complex (and current) plant, and it shows how the control design methodologies can come from basic logic premises. The simple FLC has been improved in many ways, so the perspective shown in this work should be taken as a starting point and as a sufficient statement to favor the usage of intelligent controllers alone or together with other techniques.

2. Fuzzy logic control

Fuzzy logic first intent as expressed by Zadeh [9] was to solve a membership problem. Including elements into a labeled set can be easy and straightforward as long as those elements can be clearly identified as members of the set. However, there will be some elements whose membership is ambiguous or unclear. The real issue here is that a label assigned to a set includes a meaning, subjected to human interpretation. Height measurements will not be easily classified in labeled sets as “Tall”, “Medium”, and “Short” because there is not a clear boundary between them, and their meaning is also subject to the experience and context of who interprets it. Consequently, crisp elements such as height measurements can belong to more than one set. Their membership should not be binary (true/false), but fuzzy [0, 1], representing how much an element belongs to each set attached to a particular linguistic meaning. The membership value of an element can be assigned depending on any kind of considerations regarding its relation to the set’s label; a common way to manage it is assigning a membership function $\mu(x) \in [0, 1]; x \in X$ where X represents the universe of discourse of the

measured variable. A different interpretation involves the set boundaries which, when fuzzy, can be told to be “elastic” [23]. It is similar to a blurry zone where included elements can still explain the set’s meaning to some extent.

The usual language used to describe fuzzy sets has led to many misconceptions which have obscured its acceptance and usage. As said by Zadeh [24] “...fuzzy logic is not fuzzy. Fuzzy logic is precise. Basically, fuzzy logic is a precise logic of imprecision.” In this way, applying fuzzy logic to a control problem permits the designer to make precise decisions about imprecise data. As variables are not forcefully fitted into crisp sets, the consequences of wrong classification are minimized, so it alleviates problems due to noise, disturbance, and overall uncertainty. As fuzzy logic appeals to human reasoning, a mathematical approach will difficultly find its way to describe it. This makes FLC to seem heuristic and restrictive. As this problem has been mitigated due to the research efforts made during the past 30 years, fuzzy logic acceptance on various computer and control science areas keeps growing and expanding the potential of current proposed solutions.

A standard representation of fuzzy sets is done through membership functions mapped on a n -dimensional space, where $(n - 1)$ dimensions are devoted to elements’ measured features and one extra dimension to depict their membership toward a given set. For elements described by only one feature, a 2-D mapping as shown in **Figure 1** can be used. A fuzzy set is told to be standard whenever it is described by a normalized (with values ranging $[0, 1]$), convex membership function. Most commonly used shapes for membership functions are piecewise linear (triangular, trapezoidal) and continuous (sigmoid, polynomial, and Gaussian). There are some works devoted to test and compare fuzzy systems built with different types of membership functions which try to elucidate the benefits and drawbacks of each one. Actually, there is not an “ultimate function” to describe memberships. Fuzzy systems are commonly very problem-dependent, and the main objective of placing a given problem in a fuzzy manner (at least for a non-model-based perspective) is to benefit from its uncertainty treatment and its linguistic inference relations. Both of the uncertainty treatment and linguistic inference relations are independent of the shape used.

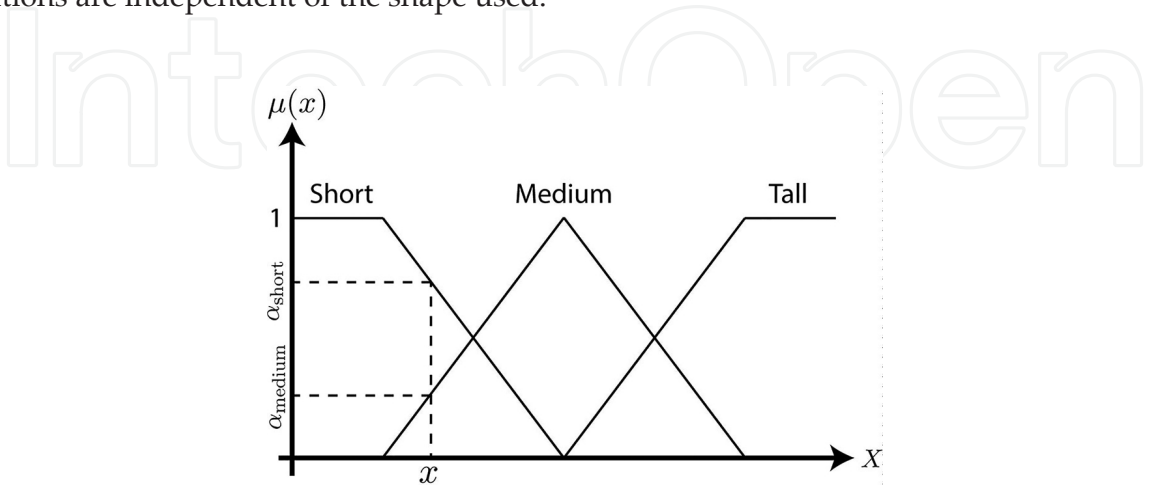


Figure 1. Standard representation of fuzzy sets.

A fuzzy inference system (the foundation of FLC) describes a relation between causes and consequences which follow from the inferential relation of (n -dimensional) input and output concepts. Each concept can be represented by a fuzzy set over a measurable feature in such a way that each input element is related to an output one, so a mapping is derived from linguistic rules. The linguistic rules have the following general form of an implication: *if x_1 is A_i^1 , x_2 is A_j^2 , ... AND x_n is A_k^n , then C_p^1 , C_p^2 , ... AND C_r^m are true.* The preceding statement considers n input elements x classified over their own universe of discourse, each separated by any number of labeled sets A . On the other hand, there are m outputs, each containing any number of consequences of the statement. As a practical example, a statement about quadcopter control can be as follows: If “altitude” is “normal” and “z-direction” is “falling”, then “motor1 speed” is set to “increase”, “motor2 speed” is set to “increase”, “motor3 speed” is set to “increase”, and “motor4 speed” is set to “increase”. It is noteworthy that any amount of inputs and outputs can be incorporated to a fuzzy inference system, and depending on the desired precision, fuzzy sets can be added or removed from a specific universe of discourse, e.g., adding “rapidly falling” and “slowly falling” to “z-direction”, or “increase a lot” to “motor speed”. A figure showing the general scheme of fuzzy inference is shown in **Figure 2**.

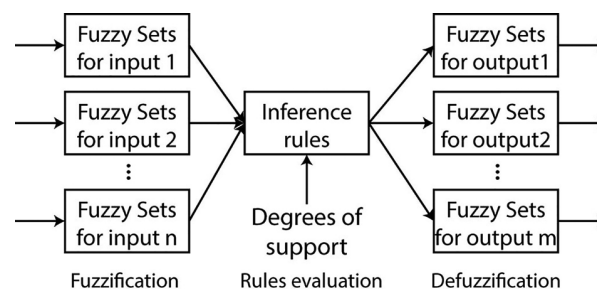


Figure 2. Standard Mamdani inference system.

A fuzzy inference system should never be seen as a set of binary conditionals. Many rules can activate at once (since set boundaries are not crisp, but fuzzy) with an individual rule-strength dependent on the membership values of each element to the labeled sets which compose the rule. A more adequate conceptual approach to fuzzy inference systems is understanding them as a nonlinear overlapping (blending, merging, weighing) of binary conditionals which result in a combined consequence. FLCs are fuzzy inference systems which commonly take the close-loop error as their input and provide a control signal as their output. The first approach of fuzzy logic to a control problem was developed by Mamdani [25] who controlled a steam engine by building the fuzzy inference machine, considering the output consequences to also be fuzzy sets. If consequences are also fuzzy sets, the general statement presented above implies that the control signal, found over a finite universe of discourse, will be also separated by fuzzy sets with meaningful labels. A FLC using this particular perspective about the controller outputs is normally referred as Mamdani-type fuzzy controller. Even if the actual numerical solution for a Mamdani FLC has not been given yet, there is one feature of this type

of controllers which has no evident solution: the controller output should not be a fuzzy set, so how can a crisp value be obtained from a fuzzy inference process?

Before answering the previous question, there is one different approach to look at which needs no fuzzy-crisp output conversion. When consequences are not fuzzy sets, but actual mathematical functions, the FLC is called to be of Takagi-Sugeno-type (TS), referring to the work of Takagi and Sugeno [26] who proposed this approach. Whenever the consequences are defined in this way, the whole FLC will work as a weighted nonlinear blending of those functions. The functions can grow in complexity from a simple constant to a dynamic model, so a TS-FLC is more general than a Mamdani one. It also expands fuzzy systems' potential as they could now perform gain-scheduling controllers or build fuzzy models for dynamic nonlinear systems, for example. Although a TS-FLC seems to absolutely outperform a Mamdani controller, it lacks of full linguistic interpretability. In consequence, the fine tuning of such a controller cannot be performed in terms of logical assumptions, and its construction would limit the fuzzy logic potential (in terms of artificial reasoning).

2.1. FLC algorithm

As noted in the preceding description, most of the inference process can be assumed transparently. However, conversion between crisp and fuzzy variables has yet to be discussed. There are various valid ways to perform such conversion; still, the standard approach will be presented in a more formal manner. Any Mamdani controller will require three steps in the inference process: fuzzification, rules evaluation, and defuzzification as shown in **Figure 2**.

The fuzzification process implies the conversion of crisp input values to fuzzy ones. Fuzzy values are obtained after evaluating the membership of a crisp value to all of the sets pertaining to its universe of discourse. All resulting values are important as they determine how strongly each rule should be executed. If more than one input is considered, there will be AND or OR operators along the inference rules' antecedents, so the evaluated membership of each input must be taken into account properly. Formally, for a universe of discourse where n fuzzy sets have been defined, the membership of an input value x to the sets can be written as $\mu_{A_i}(x) \in [0, 1]; i = 1, 2, \dots, n$. Commonly, μ_{A_i} can be used as a shortcut to express the membership value of a given input to the set A_i .

The rules evaluation is the core of inferential reasoning and it needs the values obtained in the previous step to correctly weight all of the rules in order to provide a logical response. Whenever a rule considers only one input, its triggering strength will be μ_{A_i} without any doubt. However, for rules with multiple inputs, there must exist some way to evaluate an AND operator for fuzzy values that are non-binary (OR operator can also be used, although it is not discussed in the present work). The different ways of determining the overall strengths of causes of inference rules are called "implication methods," and the work of Zhu et al. [27] can be reviewed for more details.

There are two main ways of evaluating the AND operator in fuzzy rules' antecedents. The first approach implies calculating the minimum of all the membership functions involved in the

rule causes so $\mu_{A_i} \text{ AND } \mu_{B_j} = \min(\mu_{A_i}, \mu_{B_j})$. The second approach uses the product of all rule causes as $\mu_{A_i} \text{ AND } \mu_{B_j} = \mu_{A_i} \mu_{B_j}$. Both approaches are conceptually similar and their usage is mostly designer-dependent as there are no drastic differences between them. It is worthwhile to mention that the purpose of a binary AND operator is to provide a true output only when all its inputs are true. As inputs' validity is not absolute, but graded, both aforementioned methods provide a similar behavior. If only two inputs are being evaluated, the result of the AND operator will be totally true if and only if both inputs are totally true; whenever any of the two inputs is totally false, the AND result will be totally false. For all of the cases in-between, i.e., (0, 1), the resulting validity will be decimated depending on how false the inputs are.

The cogency of the rule's consequences will be as much as the result of the operated inputs, which determine the overall rule strength. So, each rule will output a certain set of weights related to a set of consequences. It is easy to see from this point that different rules can output different weights for a single consequence, so a way to solve this issue is needed. The fundamental principle behind consequences aggregation is that it must satisfy all the rules which required it as an implication output. A simple approach would lead to take the maximum strength among the rules as it will surpass all needed values. From a different perspective, the aggregation can be seen as the sum of all related rules, limited to always be lesser or equal to one.

Lastly, the rules can be also weighted in terms of their importance to the problem. This is an additional weighting value called degree of support and is defined in a closed interval [0, 1]. Each rule can then be heuristically highlighted or obscured as desired. Actually, the tuning process can be based on the selection of proper degrees of support. Each consequence (fuzzy sets in this case) should be then lowered to meet the obtained overall weight. This process can be commonly driven in two different ways. The minimum method trims the membership function of the set by applying $\mu_{C_i} := \min(\mu_{C_i}(x), r_j) \forall x \in X$, where C_i is the consequent fuzzy set and r_j is the resulting overall strength, also called alpha-cut. The product method reduces the magnitude of the set by means of simple arithmetic, so $\mu_{C_i}(x) := r_j \mu_{C_i}$. Finally, the rule evaluation process can be divided in to three substeps: antecedents evaluation, which determines the rule strength due to inputs; rule implication, which logically links the antecedents to the consequences and adds the degree of support; consequences aggregation, which finally sets the overall rule strength and lowers the output sets. For the purposes pursued in terms of fuzzy implication, the resulting weighted consequences' union is the fuzzy output itself: A complex fuzzy set which represents the combination of all rules subjected to all inputs called the aggregated output fuzzy set. However, a controller cannot output a fuzzy set as a valid control signal, so an additional step is needed.

The conversion from a fuzzy set to a crisp value is commonly known as defuzzification. The most popular method used is the Centroid Calculation Method, which obtains a geometrical centre of the aggregated output fuzzy set with respect to the output universe of discourse. It is easy to find a conceptual support for this method as the whole output set represents all the values that are a valid output of the inference system, together with a measure of how possible it is they are the right output (their membership degree). The centroid method will find the

weighted average of all of outputs' possible values in terms of their resulting membership. Consequently, it will find the set's prototype value (a single value which effectively represents the whole set). The computation of this method is not different from a centre of mass calculation as shown in (1).

$$f(y) = \frac{\sum_{i=1}^n \mu_c(y_i) y_i}{\sum_{i=1}^n \mu_c(y_i)} \quad (1)$$

As can be noted above, the whole fuzzy inference system can effectively output a crisp value based on logic statements. The membership functions assigned to each set should effectively represent its meaning to guarantee that the process is consistent with reality or expected performance. Membership functions can be built in many different ways based on linguistic approaches, heuristic considerations, statistical consistency of data, or intelligently with the aid of an additional technique. Altogether, the adjustment of the design parameters presented above leads to a common tuning concept for fuzzy controllers. The number of sets to describe each universe of discourse, their membership functions, the rules assignments together with the methods chosen to evaluate the antecedents and consequences, the degree of support of each rule, and the defuzzification phase are variable parameters which lead to different performances (Fateh [8] shows the predominance of membership function shapes in tuning variations). Adjusting all these parameters in an analytical way or by an automatic process is very difficult, which somehow reduces FLCs acceptance, and their comparison and analysis potentials. Nonetheless, while a human logic-based approach to fuzzy tuning may be heuristic, it would effectively collect the desired features and specifications in a much easier way if compared with traditional approaches. *"Fuzzy control's main intent is to implement symbolic, linguistic control laws based on qualitative models of the plant, and control behaviours"* [12].

It is worthwhile to mention that a TS-FLC algorithm would follow almost the same steps, but will avoid the defuzzification process as its outputs are fully determined as functions. As functions are easier to manipulate through traditional mathematical well-known techniques, TS-FLCs are predominant in today's research trends and many hybrid techniques have been proposed to design, analyze, model, and tune TS-FLC ([4, 7, 6, 10]). On the other hand, improvements to the presented Mamdani's algorithm have been also proposed. For example, Li and Ruan [4] propose a way for the controller to find the inference rules automatically and Kahn et al. [13] fully tune a Mamdani controller by using genetic algorithms.

3. Quadcopter model

Quadcopters have built up much interest in recent years due to their physical characteristics and what their features allow them to do. As stated by Achtelik et al. [18], they can accomplish a range of tasks, including search and rescue, industrial inspection and observation, navigation and sensor data collection, and applications in the field of education and research. Additionally, they have virtually no legal requirements, and require minimal time and effort to begin

a test or flight. We chose to analyze quadcopters because of the range of applications they contain, and because they allow us to show how a complicated multivariable nonlinear system (the quadcopter model) can be controlled by simple fuzzy logic to produce a valid output.

There are many physical variables involved in quadcopters' flight that make them mathematically complex. They are classified as unmanned aerial vehicles (UAV) and as vertical take-off and landing (VTOL) devices. Their movement depends only on four vertical throttle forces whose individual variations and overall momentum conservation can determine the six degrees of freedom (DOFs) which rule any quadcopter [11]. The control design for this device is complicated, as it is for most of flying robots. A quadcopter is an under-actuated system subjected to big dynamic changes related to little parametric variations.

The quadcopter is controlled by changing the rotation speed of the propellers, where two of the propellers are spinning clockwise (CW) and two counterclockwise (CCW). The angular momentum is balanced in all axes if all of the propellers are spinning at the same speed, due to the CW and CCW spinning propellers' counteraction. To rotate about the vertical axis (yaw), for example, the CW spinning propellers are decelerated and the CCW spinning propellers are accelerated. Keeping the overall thrust constant, the quadcopter will turn right as a result of changing angular momentum. The cross (pitch) axis and the front (roll) axis are not affected because the CW and CCW spinning propellers are mounted on opposite sides. To rotate about the roll axis to the left side, for example, the right-side motors are sped up and the left-side motors are slowed down. Yaw is unaffected because of the balanced angular momentum about that axis. Accelerating or decelerating all motors equally results in ascending and descending, respectively, without affecting the pitch, roll, or yaw [18].

Because the quadcopter is complex and it is an under-actuated non-holonomic aircraft with fewer degrees of freedom than control variables, it is essential to simplify and extract the essential factors to create a kinematic model. Moreover, controlling this vehicle requires complex control algorithms, since the six degrees of freedom must be controlled only by four parallel force inputs [17].

The quadcopter model has many limitations that demand more precise control methods. While the model may roughly simulate the motion of the quadcopter, it is under-actuated, nonlinear, strongly coupled, and statically unstable. The model leaves air resistance, body gravity, driving force of propellers, and gyro force up to estimations by constants. This creates uncertainty in the model, which increases the difficulty in designing an accurate controller. Some components even lag or have delays while in use [5]. Additionally, quadcopters are typically propelled by brushless DC motors together with power electronics drivers, which are complicated to model. As a result, the model fails to account for the motor constants correctly [17]. These factors cause problems in modeling, as brushless DC motors, for example, are multivariable and nonlinear systems, where speed and position sensing of traditional machines with Permanent Magnet Brushless DC motors require a high degree of accuracy [11].

In a traditional model, all motor parameters would need to be detailed, making the model infeasible to replicate. In order to tackle power electronics and electric machinery uncertain and unknown behavior, a robust technique is needed. Given the limitations, an approximated

non-parametric model which accounts for time-delays, bandwidth variations, and gain uncertainties was used as an alternative together with a quadcopter traditional model. In addition, we use fuzzy logic with heuristic input-output relations to handle these vague and complex situations.

3.1. Quadcopter model

The specific kinematic model equations we will use in conjunction with fuzzy algorithms are from (2) to (7). A reference figure for equations consistency is shown in **Figure 3**.

$$m\ddot{x} = -k \sum_{i=1}^4 \omega_i^2 \sin \theta \cos \phi - B_x \dot{x} |\sin \theta \cos \phi| \quad (2)$$

$$m\ddot{y} = -k \sum_{i=1}^4 \omega_i^2 \sin \phi \cos \theta - B_y \dot{y} |\sin \phi \cos \theta| \quad (3)$$

$$m\ddot{z} = k \sum_{i=1}^4 \omega_i^2 \cos \theta \cos \phi - mg - B_z \dot{z} |\cos \theta \cos \phi| \quad (4)$$

$$I_L \ddot{\theta} = -B_\theta \dot{\theta} + kl (\omega_1^2 - \omega_3^2 - \omega_2^2 + \omega_4^2) \cos \pi / 4 \quad (5)$$

$$I_L \ddot{\phi} = -B_\phi \dot{\phi} + kl (\omega_1^2 - \omega_3^2 + \omega_2^2 - \omega_4^2) \cos \pi / 4 \quad (6)$$

$$I_Z \ddot{\psi} = -B_\psi \dot{\psi} + b (-\omega_1^2 - \omega_3^2 + \omega_2^2 + \omega_4^2) + I_M (-\dot{\omega}_1 - \dot{\omega}_3 + \dot{\omega}_2 + \dot{\omega}_4) \quad (7)$$

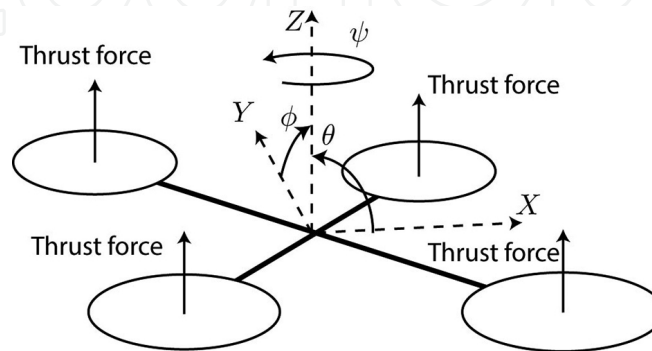


Figure 3. Quadcopter with its own reference frame.

where m is the total mass of the system, l represents the length of the arms, and each propeller speed is represented as ω_i and k is the lift constant. I_L stands for the moment of inertia of X and Y axes while I_Z is the moment of inertia with respect to the Z axis. The values B_i represent the air-drag i direction, b is the torque drag, and g stands for gravity. Please notice that all equations are given in terms of the quadcopter's reference frame so a tilt on the θ direction will directly modify the X axis. The model has not been referenced to ground as additional complexity would be added to the model without really aggregating dynamical reality. As the model could be later referenced to earth by simple trigonometric operations, this step is omitted in this work.

4. Testing variability

The presented quadcopter model represents a perfect model of how it would respond if the motor constants, air drag, and other variables were estimated exactly. However, a perfect model does not provide a valid representation of the motion of a real nonlinear system. As noted in the aforementioned text, we deal with lots of uncertainty in which the expected model deviates from the actual behavior of the system. Regarding the quadcopter system, many components exist in lag or delays while in actual use, which contributes to the deviation from the model [5]. Aspects like motor torque and rotational speeds and thrust of the motors vary to some degree and are averaged using test bench data, despite being unequal for each motor [18]. Other methods have also been implemented to counter or reduce uncertainty in systems.

One plain way of battling uncertainty is through proportional-integral-derivative controllers (PID). They are commonly used and useful in control theory and technology because they perform well for a wide class of processes. They give robust performance for specific operating conditions, and are easy to implement with analog or digital hardware, not to mention they are very familiar to engineers. However, plants that have time delays cannot be controlled effectively by a simple PID controller. Fuzzy logic controllers outperform PID controllers because they have no integral accumulator to increase the error of the naturally delayed system [14].

The nonlinearities of a quadcopter, and even of a quadcopter model, are hard to control for a PID as it is a linear controller in nature. Any linear control will be only capable of providing linear control commands to nonlinear error variations, so its reach cannot be extended to the whole operating space of a given plant. Consequently, using a linear controller over a quadcopter will restrict the system to maintaining a smooth hovering undisturbed flight. An additional drawback is that different linear controllers will be needed, as a single tuning will not suffice parametric inequalities among similar components of the quadcopter. This kind of consideration also increases the gap between simulation and actual implementation.

In order to face the difficulties exposed so far, the derived quadcopter model has not ignored relevant physical details apart from environmental wind speed and possible static parametric variations. Such uncertainty has a bigger impact if present at the propellers because controller's outputs are commonly given as thrust commands, assuming infinite bandwidth and no delay.

As explained earlier, the on-board sensors and processing unit, the power electronics driver, and the electric machinery composing each propeller should not be simulated. Incorporating a precise simulation of such elements would immensely increase model's complexity, resulting on an infeasible testbed. Consequently, a simplified model of the propeller is needed which still incorporates time delay, limited bandwidth, and non-ideal gain. In order to account for this deviation and represent uncertainties, we propose using a First-Order Plus Dead Time (FOPDT) model to make the quadcopter simulation more realistic and non-ideal. In addition, a statistical variation of FOPDT parameters can be simulated so the robustness of the FLC is tested.

Similar approaches have been used among literature to simplify model processing while still keeping the desired characteristics. Precup and Preitl [7] represent uncertainties by using a simplified plant model where nonlinear servo systems were used in conjunction with a simplified linearized model of a controlled plant. The controller even includes an inserted variable to notify changes from the nominal expected behavior. Rather than a formal dynamic approach to the system, they use a simplified model similar to FOPDT, as we do instead of simulating power electronics formally. In testing the system, we use a nonlinear model, the opposite of Preitl and Precup [28], whose sensitivity analysis is done over linearized models of the controller. Our tests are not restricted by the simplifications or assumptions they used in their linearized model.

As the introduced variations will result on different operating conditions, the FLC can be measured to be robust to some extent, e.g., a percentage variation of FOPDT parameters. The testing conditions are to be kept the same for all variations so its sensitivity to those changes can be evaluated. Altogether, a simple FLC can be proven to face a complex nonlinear system over a range of operating conditions for a tracking problem. It is clear that the usage of the FLC over the plant will describe an extended operating zone with tolerable performance. If the real plant is supposed to be within that zone, then the controller will be more likely to require minimum modifications for the implementation phase.

4.1. Robustness and boundedness

In design of linear controllers, robustness is achieved when the compensator is designed for a consistent qualitative plant model that encompasses all specified plant perturbations or variations. It is important to recognize the trade-off between maintaining stability robustness and maintaining an increased tracking performance [12]. Among literature, it is possible to confirm that a robust control design will need a previous analysis of plant variations. Variations are commonly bounded within some margins which represent the worst case situations the plant can exhibit. An effective robust controller can manage the whole area described by these variations.

There are ways in which this problem has become more manageable. In [20], a fuzzy observer-based linear controller provides rough tuning to approximate the nonlinear control system, while an optimal H attenuation provides precise tuning to make the robustness optimal. Here, both advanced tracking performance and improved robustness are sought. The paper also illustrates the use of bounding matrices to fit a nonlinear system inside certain bounds through

consideration of an approximate error. They also employ a bounded bandwidth to deal with robust performance control. To create the fuzzy controller, an optimization procedure uses the interior point method to minimize the quadratic boundary while testing stability, so that both controller is optimal and the boundary is minimal. The design of this controller allows the fuzzy algorithm to develop a robustness and function with a reduced effect of external disturbance [20].

In another example, robustness in the case of [4] is assumed to be better in FLC because the controller will adapt itself under varying conditions. However, here no expected performance is guaranteed. In [2], robustness is measured in terms of probability over an inference cell, based on the magnitude of the change of the inference output and the variation of each membership variation, taking the derivative of the inference output with respect to the corresponding alpha cut. This is integrated over the whole inference cell to finally evaluate the robustness [2].

Because control error increases with more conservative bounds, solving the robustness issue implies bounding model parameters. Bounding model parameters means assigning an expected range to the performance of the parameter. Not just any bounds will serve; it is important to find meaningful bounds based on knowledge of the uncertainties and the modelled system. For the quadcopter application, the parameters should be based on simulations or experimental results [18].

Physical constraints can impose boundaries to the controller parameter, so the controller output should not surpass some allowed characteristic of the actuator. In [8], an output scaling factor for motor voltage control does not exceed the maximum voltage allowed for that motor. Ferrero et al. [1] conclude that even measurement uncertainties should be bounded. In [3], they assume their system parameters to be uncertain but bounded to 19% around the nominal value. In [7], the variations are arbitrarily limited.

For a system that is not totally known, a set of systems can enclose it. Preitl and Precup [28] derive sensitivity models related to the step function of the control system for sensitivity analysis. With the plants described in [3], the system is proposed as $A + \Delta$ where Δ is the expected variation of the matrix used. The graphs from [7] demonstrate limits for different design parameters. Additionally, in [22], boundaries are also established so that the model lies within an expected zone. The controller is also designed considering bounded conditions.

The main idea is that a controller which faces a system set will control anything in between. In [12], stability is tested for parametric bounded variations. In [7], two controllers are tuned: (1) a nominal PI controller with respect to the set-point, and (2) a PI controller with respect to the disturbance inputs of the control system. The goal in this research was to blend the controllers to permit a bump-free transfer from one PI controller to the next, and good behavior in the operating zones covered by such blending. This guarantees the achievement of desired control system performance indices, overshoot, settling time, and robust stability [7].

5. Simulations and results

The chapter intentions will be expanded for sake of clarity. There are no models which thoroughly represent a given plant. Such incompleteness can cause many problems which take the implementation far from the expected simulated results. However, going into deep detail about subsystems comprising a plant would be inefficient (or even impossible) due to increased complexity. To tackle this problem, such systems can be approximated by equations which exhibit a similar behavior. As a single equation would hardly fit the system's response, those relegated characteristics can be assumed to be contained somewhere between marginal cases of the simplified equation. In consequence, not a model, but a set of models is proposed to enclose the system reality. A controller can be robustly tuned to face the whole set of simplified systems providing at least two benefits: (1) The real plant can be controlled as long as its features are actually bounded by the set of models, and (2) the marginal control specifications are known from worst-case simulations, so the real model can be controlled inside those borders.

Intelligent controllers, specifically FLC, seem to be a good option when dealing with complex plants and uncertain conditions. Besides FLC being formally analyzed and having model-based solutions that can be easily found in literature, its more remarkable feature is the "translation" of linguistic fuzzy rules and measurements to a nonlinear mapping. A fuzzy controller can be tuned by means of practical observation or experience, almost disregarding the plant's complexity. In this spirit, a complex nonlinear plant was modelled and further varied to show the advantages of a simple heuristically tuned FLC in terms of stability and robustness.

A quadcopter is a UAV which incorporates sensors, a processing unit, power electronics drivers, and electric machinery. With its own complexity as a flying under-actuated device, its effective control is still a challenge nowadays. In this work, the quadcopter is modelled without the use of linearization processes, and a FLC is heuristically tuned to control it under ideal model conditions. In order to account for the variability present in the control process, a set of FOPDT approximation models are used to curtail actuator performance. As the quadcopter model mainly depends on actuators' speed, many model's parameters are being considered uncertain by varying the FOPDT equation. The FLC original tuning (without variations) is preserved and its performance is analyzed toward percentage variations of motors gains, bandwidth, and discrete time delays. Altogether, a good performance of the FLC would move the process forward to a successful implementation process. Additionally, a simple approach to intelligent control design could be proven to be sufficient for such a varying complex plant.

5.1. Methodology

The whole testbed was programmed in LabVIEW 2015 platform. The quadcopter model was solved through a custom Runge-Kutta of 4th order so that its response can be evaluated step-wise. This numerical method was chosen as further implementation on a restricted-size FPGA

is desired. The FLC was designed through the Fuzzy System Designer tool included as a part of the Control and Simulation toolkit for LabVIEW. The FOPDT model was discretized and solved as a recurrent equation dependent on the sampling time which, for all tests, was adjusted to 10 ms.

The ideal gains and bandwidths are set to 1 and 50 rad/s. They will be varied by a uniform random number generator limited to predefined relative limits of [0%, 2%, 5%, 10%, 15%] and [0%, 10%, 20%], respectively. The time delays will be managed as integer variations of sensor and actuator lags. For every simulated test, a time window of 20 s will be used and the squared error of all states will be measured. Each test will be repeated 200 times to allow the random generator to describe the variation uniformly and a final average will be computed per state.

A medium-sized quadcopter of 2 kg is considered. Its parameters are shown in **Table 1** below:

Constant	Abbreviation	Value	Units
Gravity	g	9.81	m/s^2
Arm length	l	0.225	m
X/Y moment of inertia	I_L	4.856E-3	$kg\ m^2$
Z Moment of inertia	I_Z	8.801E-3	$kg\ m^2$
Lift constant	k	2.1E-4	$kg\ m/rev^2$
Torque drag	b	1.14E-7	$kg\ m^2/rev^2$
X drag	B_x	.25	kg/s
Y drag	B_y	.25	kg/s
Z drag	B_z	.25	kg/s
θ drag	B_θ	.01	$kg\ m^2/s$
ϕ drag	B_ϕ	.01	$kg\ m^2/s$
ψ drag	B_ψ	5E-3	$kg\ m^2/s$
Rotors' moment of inertia	I_M	3.357E-5	$kg\ m^2$

Table 1. Considered quadcopter parameters.

For control tuning purposes, quadcopter actuators are assumed to be of infinite bandwidth, equal gain (among all 4 motors), and to exhibit no delay. In **Figure 4**, four main blocks are shown: (1) fuzzy MIMO controller, (2) a converter from tilt to actual motor power, (3) the quadcopter model, and (4) the bounding conditions block.

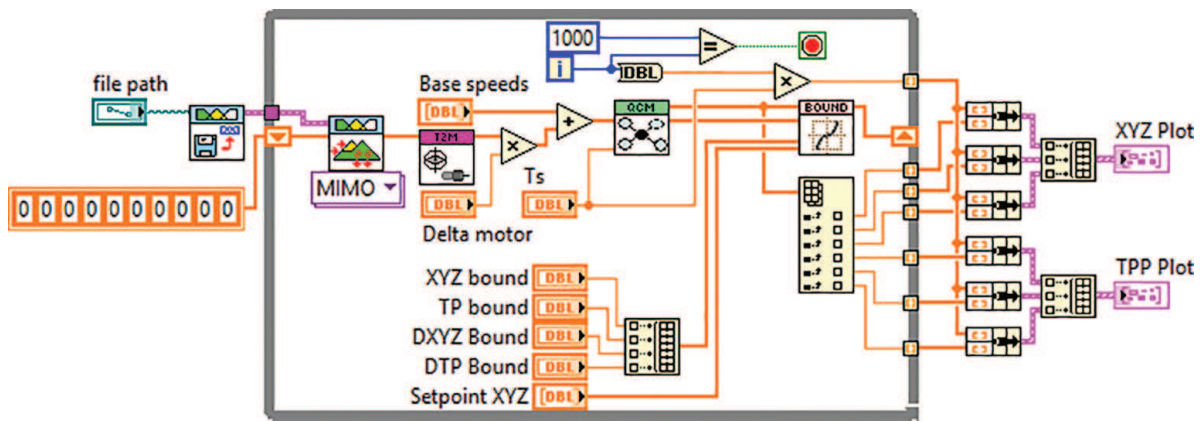


Figure 4. Testbed without FOPDT.

The FLC data are retrieved from a separate file and the control actions are executed in a black-box fashion. In search of a simple and logical controller, its output is designed in terms of forward tilt, right tilt, and overall thrust instead of individual motor power. It is clearer for a designer to identify whether the quadcopter needs to tilt rather than if an individual motor would provide the desired result; in addition, this simplifies the controller rules, as will be explained later. As the quadcopter still needs the individual motor power information, the second block converts the titling control signal into four separated increment or decrement actions. The number of fuzzy sets per input or output has been also arbitrarily determined to be of a maximum of three. The main interest here is showing a simple controller which is easy to understand despite possible better alternatives.

For improved versatility, all controller's inputs and outputs are normalized. This means that all considered universes of discourse have been limited to $[-1, 1]$. Pre- and post-processing scaling steps are needed to fit the actual needed intervals, and that is the main reason why the bounding block has been included. Having all the intervals normalized allows the designer to require a little effort in defining the membership functions as the scaling conditions will tune their reach. It is important that additional to scaling, a coerce condition is included so all the parameters can be represented within their universe of discourse. This implies that all the possible values are bounded (pre-known) by the designer.

In this case, the position control is attained by using one single FLC which requires the following inputs: X, Y, and Z error signals; X, Y, and Z speed signals; θ and ϕ angles; and $\dot{\theta}$ and $\dot{\phi}$ angular velocity signals. These inputs can recreate a basic proportional-derivative FLC; however, instead of taking the error's derivative, the velocity measurements are used so the rules are clearer. All the inputs are presented in **Table 2**.

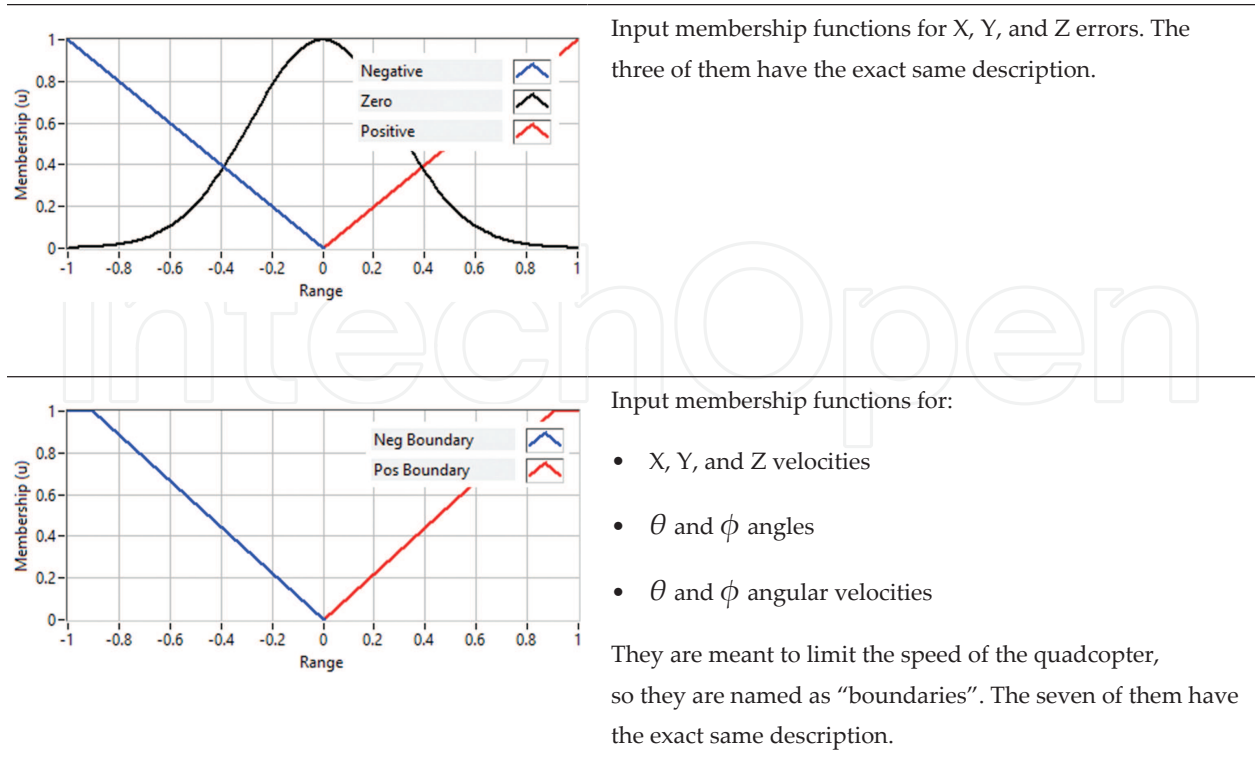


Table 2. Input membership functions.

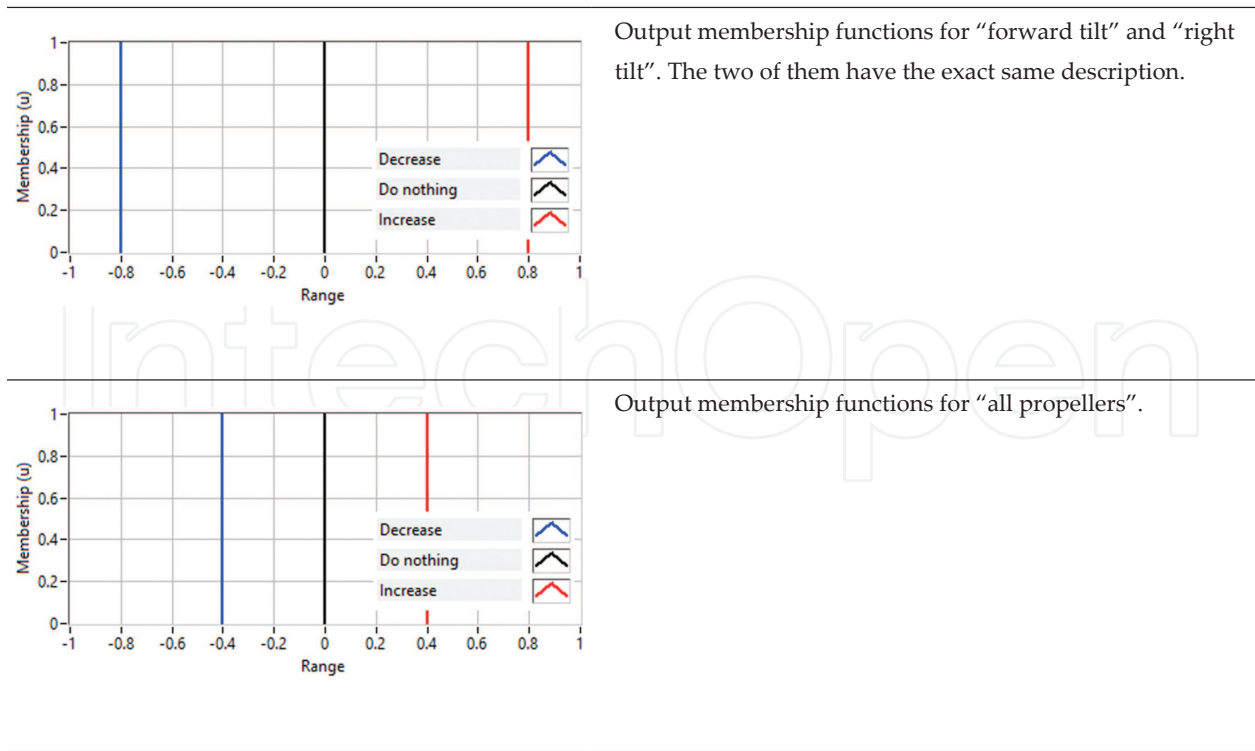


Table 3. Output membership functions.

The outputs of the controller were also furnished to be similar among them and are shown in **Table 3**. The tilt information is divided into two as there are two angles of interest with respect to the quadcopter. An additional output named “all propellers” permits the system to perform homogeneous changes in the Z axis. All outputs’ membership functions are defined using singletons; they are functions which only take the value of 1 at a particular point. As defuzzification is made based on weighing the influences of these functions, the resulting system is equivalent to a TS-FLC with constant outputs. Thus, each control signal will be a smooth transition between those constants.

As the motor inputs for the model are given on an interval [0, 155], the outputs of the FLC were multiplied by 10 and summed to a base speed. In this way, the fuzzy controller is providing correction variations to a predefined thrust which is close to the equilibrium force of the quadcopter. This computation permits the testbed to start from a condition where the vehicle is assumed to be already on flight.

Despite of the number of inputs, only 15 rules are needed to set the necessary logic to control the quadcopter. Nine of them are of a compensating nature and 6 are restrictive:

- Compensating nature:

- IF 'Zerr' IS 'Negative' THEN 'All propellers' IS 'Decrease'
- IF 'Zerr' IS 'Zero' THEN 'All propellers' IS 'Do nothing'
- IF 'Zerr' IS 'Positive' THEN 'All propellers' IS 'Increase'
- IF 'Xerr' IS 'Negative' THEN 'Tilt Right' IS 'Decrease'
- IF 'Xerr' IS 'Zero' THEN 'Tilt Right' IS 'Do nothing'
- IF 'Xerr' IS 'Positive' THEN 'Tilt Right' IS 'Increase'
- IF 'Yerr' IS 'Negative' THEN 'Tilt Forward' IS 'Decrease'
- IF 'Yerr' IS 'Zero' THEN 'Tilt Forward' IS 'Do nothing'
- IF 'Yerr' IS 'Positive' THEN 'Tilt Forward' IS 'Increase'

- Restrictive nature:

- IF 'DZ' IS 'Neg Boudary' THEN 'All propellers' IS 'Increase'
- IF 'DZ' IS 'Pos Boundary' THEN 'All propellers' IS 'Decrease'
- IF 'DX' IS 'Neg Boundary' OR 'Theta' IS 'Pos Boundary' OR 'DTheta' IS 'Pos Boundary' THEN 'Tilt Right' IS 'Increase'
- IF 'DX' IS 'Pos Boundary' OR 'Theta' IS 'Neg Boundary' OR 'DTheta' IS 'Neg Boundary' THEN 'Tilt Right' IS 'Decrease'
- IF 'DY' IS 'Neg Boundary' OR 'Phi' IS 'Pos boundary' OR 'DPhi' IS 'Pos Boundary' THEN 'Tilt Forward' IS 'Increase'

- IF 'DY' IS 'Pos Boundary' OR 'Phi' IS 'Neg Boundary' OR 'DPhi' IS 'Neg Boundary' THEN 'Tilt Forward' IS 'Decrease'

The FLC tuning is done through a trial-and-error process over the scaling factors of all inputs and outputs. The system is then configured as shown in **Table 4**.

X, Y, and Z errors	0.5
θ and ϕ boundaries	0.08
X, Y, and Z speed boundaries	0.48
θ and ϕ angular velocity boundaries	0.3
All outputs	10

Table 4. Tuning parameters for the FLC.

The resulting controller performance is shown in **Figure 5** for set-point values of [0.4, 0.3, 0.2] for [X, Y, Z], respectively. The tests regarding FOPDT variability are done for a common set-point of 0 (for which the ideal model exhibits no error operation). All of the tests evaluate how well the system can hold the zero reference, so the computed error accounts for the variations allowed by the controller. The modified testbed is shown in **Figure 6** and the error results in **Tables 5 to 8**, which include test conditions as follows:

- Bandwidth set to 50 rad/s, no sensor extra delay, no actuator delay
- Bandwidth set to 50 rad/s, one extra sensor delay, no actuator delay
- Bandwidth set to 50 rad/s, no sensor extra delay, one actuator delay
- Bandwidth set to 100 rad/s, no sensor extra delay, two actuator delays

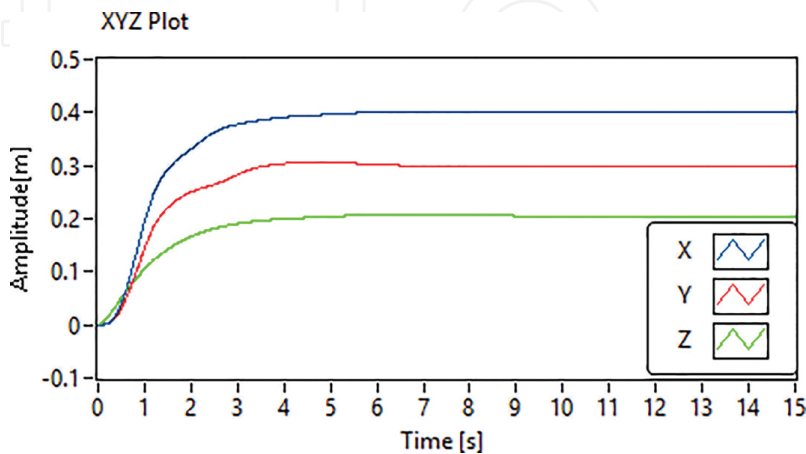


Figure 5. Test for tuned controller.

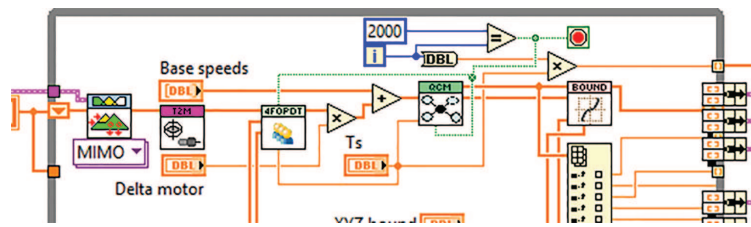


Figure 6. Testbed including FOPDT.

Delay	0 delay														
BW 50	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
0%	3.54	6.12	5.79	2.85	4.76	3.56	6.74	5.79	2.88	4.89	3.58	8.65	5.79	2.90	5.14
	E-06	E-08	E-02	E-07	E-08	E-06	E-08	E-02	E-07	E-08	E-06	E-08	E-02	E-07	E-08
2%	9.58	7.61	5.79	1.59	1.19	7.83	7.36	5.80	1.37	1.11	9.00	8.05	5.79	1.58	1.27
	E-05	E-05	E-02	E-06	E-06	E-05	E-05	E-02	E-06	E-06	E-05	E-05	E-02	E-06	E-06
5%	5.01	4.45	5.81	7.52	6.72	4.93	4.87	5.80	7.55	6.90	5.32	5.02	5.85	8.18	7.64
	E-04	E-04	E-02	E-06	E-06	E-04	E-04	E-02	E-06	E-06	E-04	E-04	E-02	E-06	E-06
10%	1.99	1.54	5.84	2.78	2.34	2.00	2.00	5.81	2.90	2.91	1.98	1.89	5.80	2.95	2.80
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05
15%	4.12	4.39	5.88	6.35	6.43	4.77	4.67	5.81	7.05	6.97	4.77	5.14	5.83	7.60	7.63
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05

Table 5. Reference results for 0 delay conditions.

Delay	1 extra sensor delay														
BW 50	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
0%	7.27	8.19	5.80	3.84	1.05	7.39	8.88	5.80	3.91	1.09	7.67	1.07	5.80	4.07	1.17
	E-06	E-08	E-02	E-07	E-07	E-06	E-08	E-02	E-07	E-07	E-06	E-07	E-02	E-07	E-07
2%	9.55	9.36	5.81	1.83	1.52	8.99	8.66	5.81	1.78	1.41	9.12	7.83	5.81	1.77	1.35
	E-05	E-05	E-02	E-06	E-06	E-05	E-05	E-02	E-06	E-06	E-05	E-05	E-02	E-06	E-06
5%	5.85	5.46	5.79	9.48	8.75	5.86	5.35	5.82	9.44	8.72	5.47	5.36	5.83	9.10	8.60
	E-04	E-04	E-02	E-06	E-06	E-04	E-04	E-02	E-06	E-06	E-04	E-04	E-02	E-06	E-06
10%	2.35	1.98	5.88	3.53	3.12	2.50	2.32	5.85	3.94	3.69	2.32	2.24	5.86	3.53	3.51
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05
15%	4.13	4.62	5.86	6.46	6.95	5.25	4.49	5.88	7.75	7.21	4.71	4.32	5.85	7.22	7.06
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05

Table 6. Results considering 1 extra sensor time delay.

Delay	1 actuator delay—20s														
BW 50	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
0%	7.37	7.64	5.80	3.41	9.14	7.35	8.37	5.80	3.44	9.57	7.14	1.02	5.80	3.45	9.83
	E-06	E-08	E-02	E-07	E-08	E-06	E-08	E-02	E-07	E-08	E-06	E-07	E-02	E-07	E-08
2%	9.41	9.18	5.81	1.75	1.51	9.51	9.00	5.80	1.85	1.50	1.10	9.04	5.81	2.08	1.63
	E-05	E-05	E-02	E-06	E-06	E-05	E-05	E-02	E-06	E-06	E-04	E-05	E-02	E-06	E-06
5%	5.72	5.01	5.82	8.71	7.92	6.04	5.12	5.81	9.29	8.17	5.80	5.82	5.80	9.36	9.53
	E-04	E-04	E-02	E-06	E-06	E-04	E-04	E-02	E-06	E-06	E-04	E-04	E-02	E-06	E-06
10%	2.27	2.28	5.81	3.50	3.58	2.23	2.00	5.85	3.41	3.14	2.49	2.15	5.82	3.86	3.51
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05
15%	4.95	4.99	5.88	7.54	7.64	4.56	4.65	5.82	7.21	7.06	4.95	5.18	5.87	7.95	8.14
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05

Table 7. Results considering 1 actuator time delay.

Delay	2 actuator delay—20s														
BW 100	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
0%	1.92	1.21	6.44	9.92	8.97	1.87	1.05	6.52	9.94	9.04	2.05	1.27	6.79	1.06	9.63
	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-01	E-02
2%	2.04	1.22	6.46	9.89	8.95	2.02	1.18	6.49	1.01	9.11	2.04	1.37	6.81	1.04	9.46
	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-01	E-02	E-02	E-02	E-02	E-01	E-02
5%	2.47	1.79	6.49	9.81	8.88	2.41	1.98	6.57	1.02	9.21	2.67	2.29	6.89	1.05	9.43
	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-01	E-02	E-02	E-02	E-02	E-01	E-02
10%	4.12	3.33	6.54	9.62	8.69	3.95	3.50	6.67	9.77	8.81	3.95	3.80	6.85	1.05	9.47
	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-01	E-02
15%	5.52	4.71	6.72	9.36	8.39	5.35	4.85	6.78	9.64	8.55	5.08	4.87	7.04	9.73	8.82
	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02

Table 8. Results considering 100 rad/s bandwidth and 2 actuator time delays.

6. Analysis and conclusions

There are two things to comment about the controller before continuing with data analysis. It does not include an accumulator, so steady-state errors are expected. In addition, the scaling of X, Y, and Z errors could be constrained into a lower margin to force the controller to react toward smaller inputs. However, errors were assumed to be bounded on 0.05 m, so lowering this value would saturate the controller whenever higher set-points are requested. The quadcopter model is started with a base motor power of 76.5, equivalent to the 49% of the total

thrust. This is enough to make the quadcopter to rise before the controller action compensates for the error. For ideal conditions and without the FOPDT model, the system response presents a steady-state error of about 5 mm as shown in **Figure 7**. As motors are assumed to be identical, no variations appear on X and Y axes.

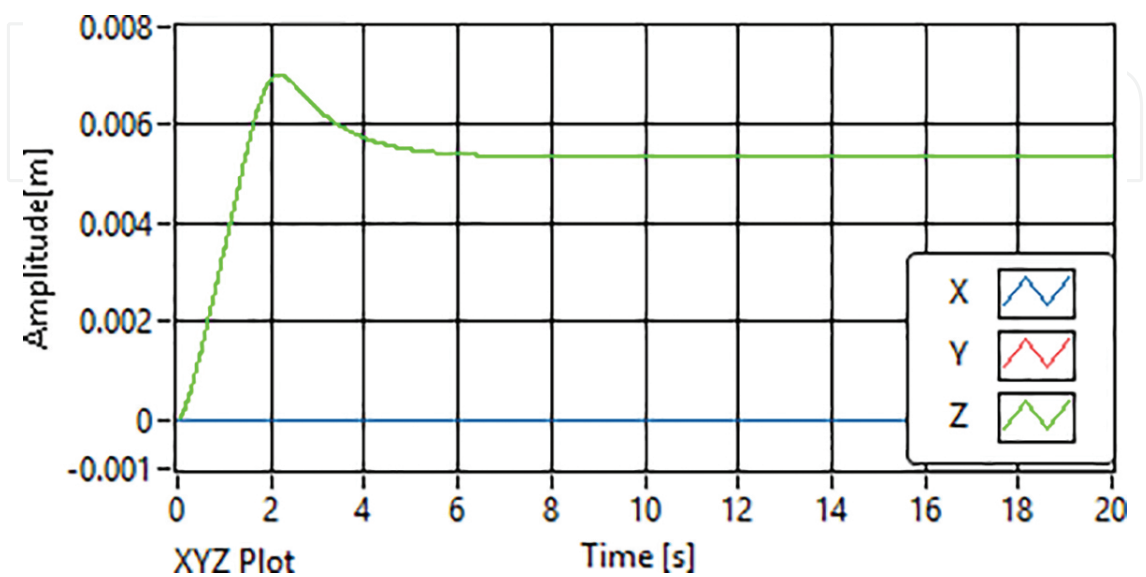


Figure 7. Reference test and steady state error of Z axis.

After performing all the tests with incorporated variability, the controller preserved stability in all cases, except for a 2 delay on the actuator side for which no data were collected. This particular case will be discussed afterwards. In order to ease analysis, four more tables (from **Tables 9** to **12**) are shown below. Each has the following rows: “Gain diff” measures the absolute difference in error calculation due to gain variations; “Col Avg” computes the average error value of a given state; “Delta State” compares each state with respect to the previous lower bandwidth test; and “Delta Ref” compares the “Col Avg” row with those of the test performed with no delays.

Delay	0 delay														
BW 50	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
Gain Diff	4.12	4.39	8.10	6.32	6.43	4.77	4.67	1.73	7.02	6.96	4.77	5.14	3.16	7.57	7.62
	E-03	E-03	E-04	E-05	E-05	E-03	E-03	E-04	E-05	E-05	E-03	E-03	E-04	E-05	E-05
Col Avg	1.34	1.29	5.82	2.01	1.91	1.47	1.45	5.80	2.18	2.14	1.48	1.52	5.81	2.31	2.26
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05
Delta State						8.70%	10.77%	-0.31%	7.44%	10.42%	0.37%	4.96%	0.13%	5.88%	5.64%
Delta Ref	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 9. Analysis for reference conditions.

Delay	1 extra sensor delay														
BW 50	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
Gain Diff	4.13	4.62	5.50	6.42	6.94	5.24	4.49	7.35	7.71	7.19	4.71	4.32	4.72	7.18	7.05
	E-03	E-03	E-04	E-05	E-05	E-03	E-03	E-04	E-05	E-05	E-03	E-03	E-04	E-05	E-05
Col Avg	1.43	1.45	5.83	2.23	2.22	1.69	1.49	5.83	2.57	2.38	1.54	1.44	5.83	2.38	2.32
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05
Delta State						14.9	2.54%	0.06%	13.1	6.81%	-9.80%	-3.50%	-0.01%	-8.22%	-2.99%
						2%			9%						
Delta Ref	6.86	12.	0.09%	10.	16.	14.	2.67%	0.45%	18.	11.	4.04%	-5.	0.31%	2.80%	2.28%
	%	14%		87%	13%	67%			21%	63%		72%			

Table 10. Analysis for first test.

Delay	1 actuator delay—20s														
BW 50	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
Gain Diff	4.94	4.99	7.67	7.51	7.63	4.55	4.65	2.19	7.18	7.06	4.94	5.18	7.28	7.92	8.13
	E-03	E-03	E-04	E-05	E-05	E-03	E-03	E-04	E-05	E-05	E-03	E-03	E-04	E-05	E-05
Col Avg	1.58	1.57	5.82	2.43	2.44	1.50	1.45	5.82	2.35	2.24	1.63	1.60	5.82	2.60	2.56
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05
Delta State						-5.31	-8.54	-0.13	-3.03	-8.93	7.97	9.37	0.11	9.36	12.
						%	%	%	%	%	%	%	%	%	51%
Delta Ref	17.	21.	0.01	20.	27.	1.95%	0.22%	0.19%	8.21%	4.65%	10.	5.10%	0.17%	12.	12.
	59%	92%	%	45%	24%						36%			36%	87%

Table 11. Analysis for second test.

Delay	2 actuator delay - 20s														
BW 100	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
Gain Diff	3.60	3.50	2.73	-5.59	-5.73	3.48	3.80	2.66	-3.07	-4.88	3.02	3.60	2.48	-9.09	-8.11
	E-02	E-02	E-03	E-03	E-03	E-02	E-02	E-03	E-03	E-03	E-02	E-02	E-03	E-03	E-03
Col Avg	3.21	2.45	6.53	9.72	8.78	3.12	2.51	6.61	9.93	8.95	3.16	2.72	6.88	1.04	9.36
	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-01	E-02
Delta State						-3.02%	2.42%	1.18%	2.10%	1.91%	1.26%	7.63%	3.94%	4.13%	4.43%
Delta	229	179	12.13%	482	458	202	163	13.82%	456	418	204	168	18.33%	447	413
Ref	3.99%	9.51%		505	437	1.63%	6.89%		177	668	0.72%	7.06%		866	368
				.67%	.12%				.34%	.30%				.68%	.83%

Table 12. Analysis for last test.

It can be seen that the uneven gain variation is responsible for the highest increase of error in all tests. On the other hand, bandwidth modifications do not generate an evident trend as error is sometimes even lower than previous ones. It is clear from this point that bandwidth unevenness is not quite relevant from a statistical point of view; however, angle variation shows an increasing error most of the time. This is due to a constant need to change the propellers' speed because command signals are not fully applied simultaneously. For strongly varying angles, an overall vibration effect will be visible on the quadcopter.

It is worth to mention that the actuator delay will affect the controller performance more strongly than a sensor delay. Those actuator time-related issues may exist due to power electronic drives and integrated close-loops. They can also appear if the motors' information is not collected properly due to sensing or measuring problems on the inner loop and filtering delays. An additional feature observed throughout the simulations is that Z remains almost unaltered, mostly due to its greater independence to little angle variations.

As can be noted, an additional test was performed considering the motors bandwidth to be of 100 rad/s. Stability was not preserved for 2 actuator time delays and 50 rad/s bandwidth, so it was doubled to investigate the effect of improved actuator characteristics. A slow motor performance causes the controller signals to wait some iterations before their effect is fully applied. If the motors vary in bandwidth, some control signals will be delivered on time and some will not, increasing the error seen by the controller. This effect would lead to wrong compensation signals which, if delayed, will hardly keep a stable plant. Remember that the FLC tuning was not modified, so its success will depend on the quadcopter's overall bandwidth.

It is also noteworthy that even if control signals are executed fast enough to keep stability, chattering may appear due to late compensation. This effect (show in **Figure 8**) can be incorrectly identified as noise, which can damage control performance.

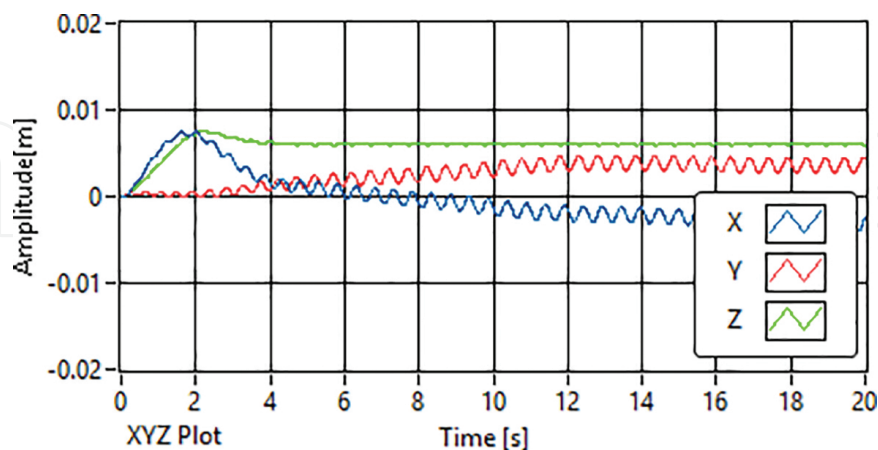


Figure 8. Chattering effect due to time delay of actuators.

A simple FLC has been proven to control a quadcopter nonlinear model subjected to variability (FOPDT) on its actuators. It is a very simple controller which seeks clarity and simplicity and

that still exhibits robust performance when tested under model variations. The controller seems unimpaired by uneven gains and bandwidth variations among quadcopter motors, as well as single time delays present before or after controller block execution. It is possible to say that unevenness on motors' bandwidth does not have a drastic effect. However, an overall low bandwidth will greatly restrict the system's stability.

The controller allowed a maximum 15 mm absolute error among tests and kept the simulated quadcopter stable for motor-unbalanced variations up to 15% in gain and 20% in bandwidth. The controller is capable of dealing with time delay too. However, the actuators' response speeds seem to be a major restriction of this. This analysis permits the FLC system presented to be considered as a simple option for actual implementation. Its tuning and further improvement through more dependable techniques can lead to a realizable controller which observes better specifications.

Author details

Luis Ibarra^{1*} and Colin Webb²

*Address all correspondence to: ibarra.luis@itesm.mx

1 Tecnológico de Monterrey, Mexico City, Mexico

2 Massachusetts Institute of Technology, Cambridge, USA

References

- [1] A. Ferrero, A. Federici, S. Salicone. Instrumental uncertainty and model uncertainty unified in a modified fuzzy inference system. *IEEE Transactions on Instrumentation and Measurement*. 2010;59(5):1149-1157. DOI: 10.1109/TIM.2010.2044257
- [2] H-X. Li, L. Zhang, K-Y. Cai, G. Chen. An improved robust fuzzy-PID controller with optimal fuzzy reasoning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*. 2005;35(6):1283-1294. DOI: 10.1109/TSMCB.2005.851538
- [3] J. Zhang, M. Fei. Analysis and design of robust fuzzy controllers and robust fuzzy observers of nonlinear systems. In: *Sixth World Congress on Intelligent Control and Automation*; 2006; Dalian, China. IEEE; 2006. pp. 3767-3771. DOI: 10.1109/WCICA.2006.1713075
- [4] X.Z. Li, D. Ruan. Comparative study of fuzzy control, PID control, and advanced fuzzy control for simulating a nuclear reactor operation. *International Journal of General Systems*. 2000;29(2):263-279. DOI: 10.1080/03081070008960933

- [5] Y. Li, S. Song. A survey of control algorithms for quadrotor unmanned helicopter. In: 2012 IEEE Fifth International Conference on Advanced Computational Intelligence; 2012; Nanjing, China. IEEE; 2012. pp. 365-369. DOI: 10.1109/ICACI.2012.6463187
- [6] M.S. Kalavathi, C.S.R. Reddy. Performance evaluation of classical and fuzzy logic control techniques for brushless DC motor drive. In: 2012 IEEE International Power Modulator and High Voltage Conference; 2012; San Diego, USA. IEEE; 2012. pp. 488-491. DOI: 10.1109/IPMHVC.2012.6518787
- [7] R.-E. Precup, S. Preitl. PI-Fuzzy controllers for integral plants to ensure robust stability. *Information Sciences*. 2007;177(20):4410-4429. DOI: 10.1016/j.ins.2007.05.005
- [8] M.M. Fateh. Robust fuzzy control of electrical manipulators. *Journal of Intelligent & Robotic Systems*. 2010;69(3):415-434. DOI: 10.1007/s10846-010-9430-y
- [9] L.A. Zadeh. Fuzzy sets. *Information and Control*. 1965;8(3):338-353. DOI: 10.1016/S0019-9958(65)90241-X
- [10] G. Feng. A survey on analysis and design of model-based fuzzy control systems. *IEEE Transactions on Fuzzy Systems*. 2006;14(5):676-697. DOI: 10.1109/TFUZZ.2006.883415
- [11] R.A. Gupta, R. Kumar, A.K. Bansal. Artificial intelligence applications in permanent magnet brushless DC motor drives. *Artificial Intelligence Review*. 2010;33(3):175-186. DOI: 10.1007/s10462-009-9152-3
- [12] S.P. Linder, B. Shafai. Behavior-based robust fuzzy control. In: *Intelligent Control (ISIC)*, 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS); 1998; Gaithersburg, USA. IEEE; 1998. pp. 233-238. DOI: 10.1109/ISIC.1998.713666
- [13] S. Kahn, S.F. Abdulazeez, L.W. Adetunji, A.H.M.Z. Alam, M.J.E. Salami, S.A. Hameed, A.H. Abdalla, M.R. Islam. Design and implementation of an optimal fuzzy logic controller using genetic algorithm. *Journal of Computer Science*. 2008;4(10):799-806.
- [14] S. Nithya, A.S. Gour, N. Sivakumaran, N. Anantharaman. Measurement and control of process using soft computing. *Instrumentation Science and Technology*. 2008;36(2):194-208. DOI: 10.1080/10739140701850944
- [15] R.-E. Precup, R.-C. David, E.M. Petriu, M.-B. Rădac, S. Preitl, J. Fodor. Evolutionary optimization-based tuning of low-cost fuzzy controllers for servo systems. *Knowledge-Based Systems*. 2013;38(Advances in Fuzzy Knowledge Systems: Theory and Application):74-84. DOI: 10.1016/j.knosys.2011.07.006
- [16] Mojtaba Mizaei, Mohammad Eghtesad, Mohammad Alishahi. A new robust fuzzy method for unmanned flying vehicle control. *Journal of Central South University*. 2015;22(6):2166-2182. DOI: 10.1007/s11771-015-2741-1
- [17] J.A. Kunzumann, C.A. Berry. Investigation in the control of a four-rotor aerial robot. In: B. Morrison, M.H. Hamza, editors. *Proceedings of the 2nd IASTED International*

- Conference on Robotics, Robo 2011; 2011; Pittsburgh, USA. ACTA Press; 2011. pp. 181-188. DOI: 10.2316/P.2011.752-055
- [18] M. Achteik, T. Bierling, J. Wang, L. Höcht, F. Holzapfel. Adaptive control of a quadcopter in the presence of large/complete parameter uncertainties. In: AIAA Infotech at Aerospace Conference and Exhibit 2011; 2011; St. Louis, USA. ARC; 2011. pp. AIAA 2011-1485.
- [19] S. Gupte, P.I.T. Mohandas, J.M. Conrad. A survey of quadrotor unmanned aerial vehicles. In: 2012 Proceedings of IEEE Southeastcon; 2012; Orlando, USA. IEEE; 2012. pp. 1-6. DOI: 10.1109/SECon.2012.6196930
- [20] B.-S. Chen, C.-S. Tseng, H.-J. Uang. Robustness design of nonlinear dynamic systems via fuzzy linear control. IEEE Transactions on Fuzzy Systems. 1999;7(5):571-585. DOI: 10.1109/91.797980
- [21] H. Chen, Z. Zouaoui, Z. Chen. Application of fuzzy logic to reduce modelling errors in PIDSP for FOPDT process control. In: 19th Mediterranean Conference on Control Automation (MED); 2011; Corfu, Grece. IEEE; 2011. pp. 1478-1483. DOI: 10.1109/MED.2011.5983205
- [22] D. Senthilkumar, C. Mahanta. Identification of uncertain nonlinear systems for robust fuzzy control. ISA Transactions. 2010;49(1):27-38. DOI: 10.1016/j.isatra.2009.07.005
- [23] L.A. Zadeh. Fuzzy logic—a personal perspective. Fuzzy Sets and Systems. 2015;281(Celebrating the 50th Anniversary of Fuzzy Sets):4-20. DOI: 10.1016/j.fss.2015.05.009
- [24] L.A. Zadeh. A summary and update of "fuzzy logic". In: Proceedings of the 2010 IEEE International Conference on Granular Computing; 2010; San Jose, USA. IEEE; 2010. pp. 42-44. DOI: 10.1109/GrC.2010.144
- [25] E.H. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. Proceedings of the Institution of Electrical Engineers. 1974;121(12):1585-1588. DOI: 10.1049/piee.1974.0328
- [26] T. Takagi, M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man and Cybernetics. 1985;15(1):116-132. DOI: 10.1109/TSMC.1985.6313399
- [27] X. Zhu, Z. Huang, S. Yang, G. Shen. Fuzzy implication methods in fuzzy logic. In: Fourth International Conference on Fuzzy Systems and Knowledge Discovery; 2007; Haikou, China. IEEE; 2007. pp. 154-158. DOI: 10.1109/FSKD.2007.327
- [28] S. Preitl, R.-E Precup. Sensitivity study of a class of fuzzy control systems. Periodica Polytechnica, Electrical Engineering. 2006;50(3-4):255-268.

