

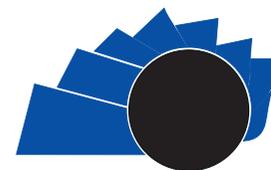


UNIVERSIDAD DISTRITAL  
FRANCISCO JOSÉ DE CALDAS

# Visión Electrónica

## Más que un estado sólido

<https://revistas.udistrital.edu.co/index.php/visele/index>



Visión Electrónica

A RESEARCH VISION

## Fuzzy control of quadrotor Ar. Drone 2.0 in a controlled environment

*Control difuso del quadrotor AR. Drone 2.0 en un ambiente controlado*

*Fernando Campos Archila<sup>1</sup>; Valentina Pinzón Saavedra<sup>2</sup>; Faiber Robayo Betancourt<sup>3</sup>*

### INFORMACIÓN DEL ARTÍCULO

#### Historia del artículo:

Enviado: 12/12/2018

Recibido: 21/12/2018

Aceptado: 02/01/2019

#### Keywords:

AR. Drone 2.0

Fuzzy logic

Height control

Matlab®



#### Palabras clave:

AR. Drone 2.0

Lógica difusa

Control de altura

Matlab®

### ABSTRACT

This paper aims to describe the design and implementation of the height control system for the quadrotor AR. Drone 2.0 making use of a fuzzy logic in a previously established environment. This device has a height control system both in simulation and in the real platform. Three controllers are developed by fuzzy logic whose parameters are obtained from the drone's sensors in such a way that it allows to control height and angles of orientation (Pitch, Roll and Yaw) as long as certain levels of battery charge are considered so that the system does not become unstable. For the visualization and interaction with the drone a Matlab® interface is designed and implemented, that allows communication between the user and all system functions in such a way that the mode of execution can be chosen, follow the reference parameters autonomously, store data for a later analysis, and visualize the displacements to observe the efficiency of system.

### RESUMEN

Este trabajo muestra el control de altura del quadrotor AR. Drone 2.0 aplicando lógica difusa en un ambiente previamente establecido, tanto en simulación como en la plataforma real. Se desarrollan tres controladores mediante lógica difusa cuyos parámetros son obtenidos de los sensores del dron, permitiéndose controlar altura y ángulos de orientación (Pitch, Roll y Yaw) siempre y cuando se consideren ciertos niveles de carga de batería para que el sistema no se vuelva inestable. Para la visualización e interacción con el dron se diseña e implementa una interfaz en Matlab® que permite la comunicación entre el usuario y todas las funciones del sistema, de tal manera que se puede elegir el modo de ejecución, seguir los parámetros de referencia de manera autónoma, almacenar datos para un posterior análisis, y visualizar los desplazamientos para determinar la eficiencia del sistema.

<sup>1</sup> BSc. In Electronic engineering, Universidad Surcolombiana, Colombia, E-mail: [u20131119489@usco.edu.co](mailto:u20131119489@usco.edu.co) ORCID: <https://orcid.org/0000-0003-1911-1475>

<sup>2</sup> BSc. In Electronic engineering, Universidad Surcolombiana, Colombia, E-mail: [u20122113296@usco.edu.co](mailto:u20122113296@usco.edu.co) ORCID: <https://orcid.org/0000-0002-1517-7605>

<sup>3</sup> BSc. In Electronic engineering, Universidad Surcolombiana, Colombia. MSc. In Control engineering, Universidad de Ibagué-Ghent, Colombia-Bélgica. Current position: Professor at Universidad Surcolombiana, Colombia. E-mail: [faiber.robayo@usco.edu.co](mailto:faiber.robayo@usco.edu.co) ORCID: <https://orcid.org/0000-0002-1048-8383>

## 1. Introduction

The Unmanned Aerial Vehicle (UAV) was developed some time ago. However, nowadays its use has increased exponentially due to its easy access. Drones are the most popular ones as they are tools that can carry out multiple applications which simplify human work. For example, they can be used in journalism to capture different perspectives [1], in crop spraying [2], for goods transport and shipment [3], in search and rescue, in agriculture, among others.

At the academic and research level, there has been an exponential progress in the importance of the implementation of new technologies that support the human being in different high-risk works that present a high risk, such as the use of UAV due to its versatility. The continuous research in the mechanics, functionality and applications of the UAV has made them more widely available, allowing knowing their classification that varies depending on their components. One of its classifications depends on the propellers the UAV has, in case of having four propellers it is known as quadrotor and one of the best known in the market is the AR. Drone 2.0.

The constant technological advance has made that the AR. Drone 2.0, considered a highly complex system, can be used through intelligent control techniques that regardless of its complexity, do not require its mathematical model as proposed in the Universidad Tecnológica De La Mixteca, México with a fuzzy control of a quadrotor for the autonomous tracking of trajectories through fuzzy logic by means of the design and implementation of 4 controllers, one for each Euler Angle (Pitch, Roll and Yaw) and one last for the height; in addition, a control panel is presented from which the flight functions can be executed. In general, it was obtained a good response, although sensitive to disturbances, presenting multiple oscillations in the results obtained, despite that the controller's performance was of quality. [4].

At the national level, there is already research done in the field such as at the Universidad Católica de Manizales, where the design and implementation of the flight control system of a UAV are established. For the flight control a compass and artificial horizon using Arduino-Processing -MPU6050 are implemented. This software allowed the drone to perform a stable flight. [5].

In the Universidad Surcolombiana, the design and implementation of an inertial guidance

system for the autonomous flight control in a UAV is proposed. In order to achieve this, the inertial guidance system is designed making use of the different measurement instruments that the UAV is supplied with, so as to, by means of a DCM algorithm, calculate the data referring to the angles of Euler for its subsequent correction. Results were satisfactory, showing the good performance of the inertial guidance system. [6].

On the one hand, instrumentation is an important part when executing the control. Therefore, the control therefore it is indispensable to make use of sensors to measure the different variables related to the flight. The use of the Matlab® software [7] makes it possible to create a graphic interface that is intuitive and friendly in such a way that it allows communication between the model and the computer.

On the other hand, this work seeks to design and implement a height control system applying fuzzy logic for the quadrotor AR. Drone 2.0 in a controlled environment. The work is structured as follows: Initially the design and development are shown through the AR Drone Simulink Development-kit v1.1, the fuzzy control and the interface are designed through Matlab®. After the implementation and flight tests are carried out to finalize the conclusions.

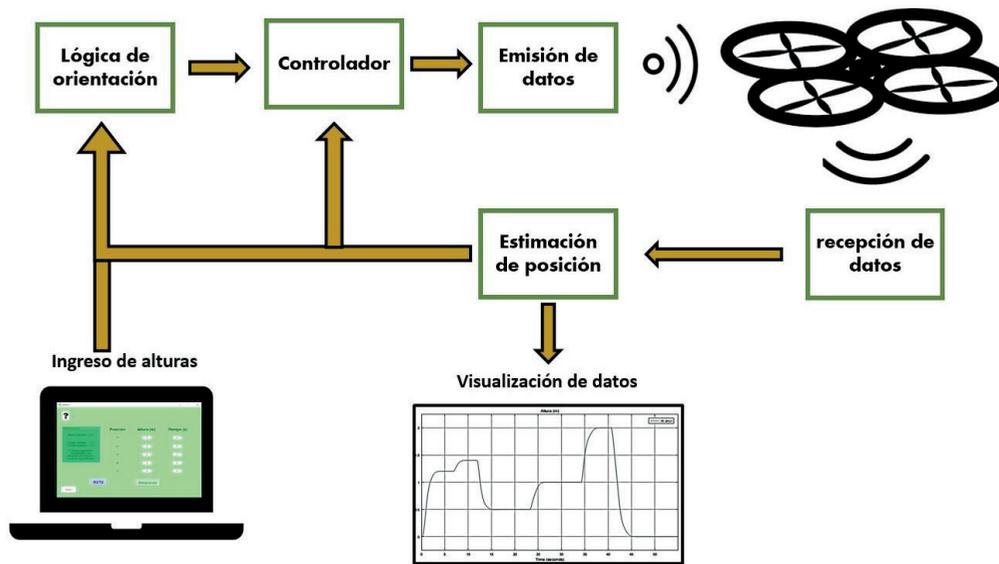
## 2. Design and development

Figure 1 shows a block diagram of the entire System implemented.

### 2.1.AR. Drone Simulink development – kit V1.1

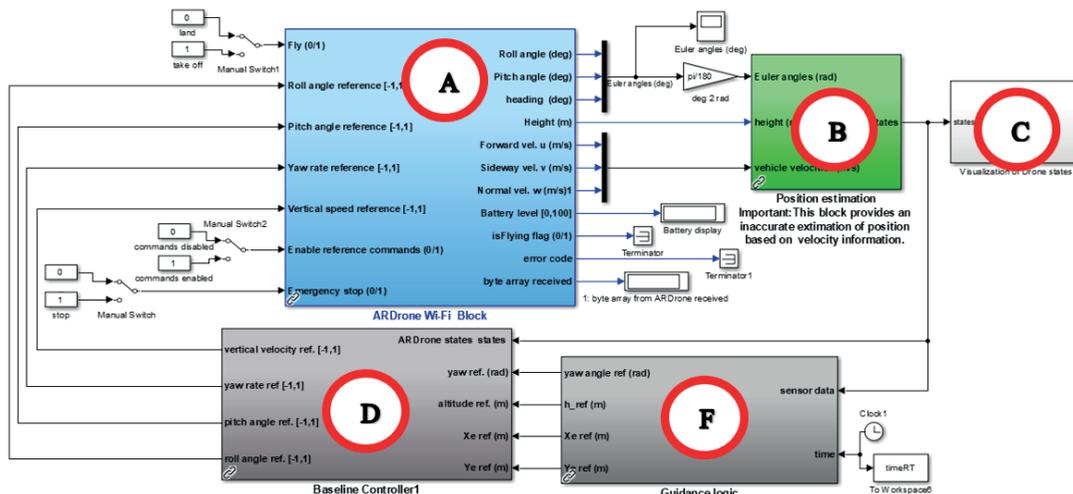
To develop the control system and its corresponding graphic interface it is decided to work with the Matlab® software, as it has a specific development package to work with the AR Drone 2.0. This kit [8] includes blocks and examples to simulate the control or implement it in the drone directly in real time as shown in Figure 2. The simulation blocks have an approximate model based on previous identifications of the vehicle, in turn the real control part allows to send and read the data of the necessary variables for the flight of the drone in real time. All of the above allows the user to control position, speed, tracking of trajectories in a simulated way as well as to execute them in a real way by means of the wifi control. This kit was developed in 2013 for the MathWorks research practice projects.

Figure 1. Diagram of hardware block.



Source: own.

Figure 2. Block diagram of the AR. Drone Simulink Development-kit V1.1.



Source: own.

A. AR. Drone Wi-Fi-Block/ AR.

B. Position Estimation

C. Visualization of Drone States.

D. Baseline Controller.

F. Guidance Logic.

The way to operate the real model of the development kit for the A.R Drone 2.0 can be summed up as follows:

1. Enter the desired trajectory in get Waypoints.m
2. Process the trajectory in the Guidance logic subsystem

3. Control the trajectory in the Baseline Controller subsystem
4. Send the corrected data to the drone from the A.R Drone Simulation Block subsystem
5. Receive the data of the drone from the A.R Drone Simulation Block subsystem
6. Estimate the position of the drone in the Position estimation subsystem
7. Visualize the position of the drone in the Visualization of Drone states subsystem
8. Send the data received from the drone to the Baseline Controller subsystem to correct these
9. Repeat everything from step 3 on the points are finished

## 2.2. Fuzzy control

For the control stage, fuzzy logic is used due to its useful advantages for the project as its ease of implementation, and in this particular case the Fuzzy Logic Toolbox of Matlab® / Simulink is used [9]. In addition, fuzzy logic has a great performance against non-linear systems or difficult modeling because it does not depend on complex and extensive mathematical equations and, most importantly, it is not necessary to know the mathematical model that governs the functioning of the system; As the drone is an aerial vehicle, it presents a complex and extensive modeling of its system. The use of the Fuzzy Logic Toolbox from Matlab® / Simulink allows to build and analyze the results of a fuzzy inference system (FIS) by means of its various components.

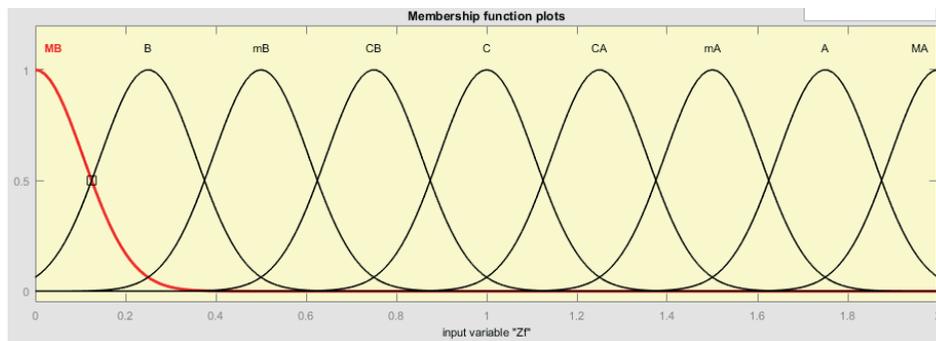
The Matlab® fuzzy logic tool allows to choose the kind of fuzzy system to work between Sugeno system and Mamdani system. In this case the last one is chosen since it has greater similarity with human analysis and greater acceptance; however, it should be pointed out that the Sugeno system is computationally more efficient.

For this project four fuzzy controllers are made, one for each Euler angle and additional for the height by means of three different fuzzy inference systems. In general, all systems share two inputs, one reference and the other is the signal from the drone. In the output a variable is shown which is sent to the vehicle. It also performs the process of defusification by the centroid method which determines the output of the fuzzy system finding the center of the area under the curve of the membership function that is relevant. At the moment it is emphasized that the height control system is designed for a maximum elevation of two meters. This height is chosen as a reference for the project, but the drone used can reach heights of 50 meters ideally.

As shown in Figure 3 for height, nine membership functions are used in both the inputs and the output. The functions of the input are of the Gaussian type (gaussmf) with a range of [0 2] expressing that the inputs are heights where their maximum value will be two meters, in addition, the functions are scattered uniformly throughout the range.

As shown in Figure 3, the nomenclature refers to the elevation states that are taken or divided the range to work: “Muy Bajo”, “Bajo”, “medianamente Bajo”, “Centro Bajo”, “Centro”, “Centro Alto”, “medianamente Alto”, “Alto” y “Muy Alto”, respectively.

**Figure 3.** Nine Gaussian functions for input variables.



Source: own.

As for the output as shown in Figure 4, the range handled is  $[-1.5 \ 1.5]$  which is the reference speed in meters per second (m/s). In this case, triangular signals are used, which in its lower area are wide at the ends, but as it gets closer to the center it narrow, since more abrupt changes are needed the further away from the reference point and the smoother the it is approaching the reference height.

In order to make the control work, 81 rules are used that relate all the membership functions to each other. It is also experimented with other similar control systems that differ in the number of membership functions used, the type of function and the number of rules. 11 membership functions are used in the inputs and the output applying the same reasoning used previously for the location of the functions. Simultaneously, so as not to make the nomenclature of these functions something tedious it is chosen to give these numerical values that go from  $-5$  through the center to  $5$ . The number of rules used is 121. The other control system implemented is similar to the previous one, but triangular signals are used in the inputs spread uniformly throughout the working range.

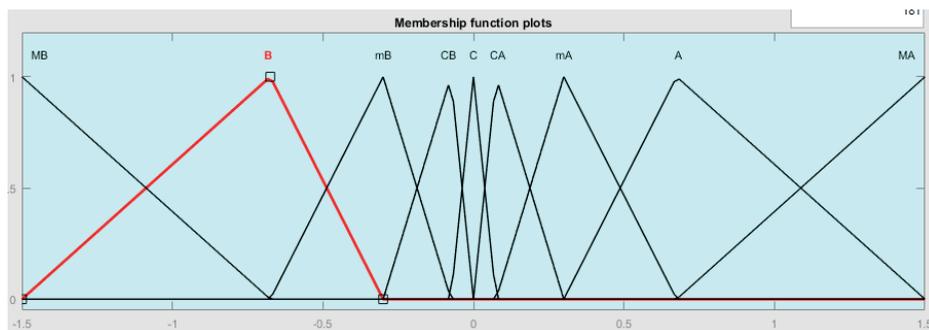
When controlling the steering angles the same methodologies mentioned above are used, but the input and output ranges vary as expressed below:

For the Pitch (as shown in Figure 5) the inputs are the speeds in m/s, in the front or rear direction which is represented in its advance along the X axis. The ranges worked are  $[-2 \ 2]$  for the input of the signal from the drone and  $[-3 \ 3]$  (m/s) for the reference signal. The output of the controller is the reference angle of the pitch that is sent to the drone and takes values of  $[-1.5 \ 1.5]$ .

For the Roll (as shown in Figure 6) the inputs are the speeds in m/s in the positive or negative lateral direction which is represented in its advance along the Y axis. The ranges used are  $[-1.5 \ 1.5]$  for the input of the signal from the drone and  $[-3 \ 3]$  for the reference signal. The output of the controller is the reference angle of the roll that is sent to the drone and takes values of  $[-1.5 \ 1.5]$ .

For the Yaw (as shown in Figure 7) the inputs are the angles in radians of the positive (or negative)

**Figure 4.** Nine triangular functions for the output variables.



Source: own.

**Figure 5.** Pitch in the AR. Drone 2.0.



Source: own.

pitch (Yaw) which is represented in its frontal rotation or angular displacement in  $\varphi$  from the perspective of a spherical plane. The ranges used are  $[-\pi \pi]$  radians for the two inputs (signal coming from the drone and the reference signal). The output of the controller is the angular reference speed of the Yaw angle that is sent to the drone and values of  $[-6 \ 6]$  radians per second.

**Figure 6.** Roll in the AR. Drone 2.0.



Source: own.

**Figure 7.** Yaw in the AR. Drone 2.0.



Source: own.

Finally, it should be stressed that the quantity, type and distribution of the functions as well as the rules used are after a heuristic analysis according to the criteria of the designers and knowledge about the model to work.

### 3. Graphic interface

When designing the graphic interface, the “Guide” tool of Matlab® GUI [10] is used, which facilitates the design since it is possible to interact directly with the elements that will make part of the interface or it can also be modified by means of

code. To work in real time, it is chosen to perform in Matlab® as the entire control process is done through this software and migrating the data from one program to another could give rise to problems or undesirable delays.

Previously, it has been mentioned that the work is done in a simulated and real way, therefore, 2 work environments are created first for each form. Then an additional one is made where the user can freely enter the drone heights and time which he wishes to work with. Finally, a last view is designed which is displayed when the application starts and it is the one that allows the user to navigate between the first 3 views mentioned above. Next, it shows and explains the operation of each of these:

#### 3.1. General View

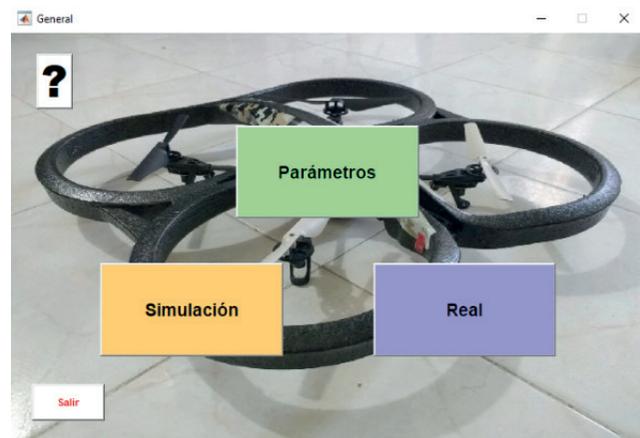
Each time a view is entered, a warning window opens automatically, which contains instructions for the proper use of the interface in which it is located.

This view, as shown in Figure 8, contains the buttons that allow entering the other panels, that is, it has the function of linking the different designed views.

#### 3.2. Parameters View

In this view, as shown in Figure 9, up to 5 reference heights with their respective flight times can be entered. Besides, there is a route button which loads the data from a single sample path of the trajectory tracking designed.

**Figure 8.** General view.



Source: own.

### 3.3. Simulation View

In the simulation window as shown in Figure 10, the buttons necessary to interact simulated with the drone model are included, as well as a space in which the drone performance on a graph can be visualized.

### 3.4. Real View

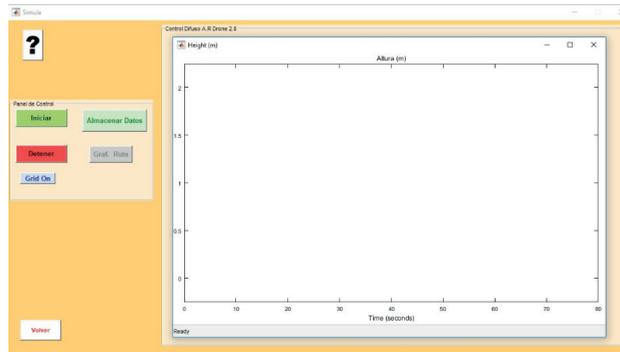
In the “Real” window, as shown in Figure 11, the user can interact directly and in real time with the drone by means of functions of takeoff, landing etc.

Figure 9. Parameters view.



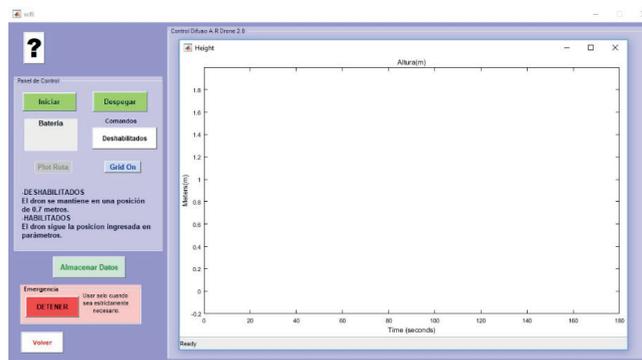
Source: own.

Figure 10. Simulation view.



Source: own.

Figure 11. Real view.



Source: own.

## 4. Implementation

For the real implementation, a suitable location is looked for to carry out the tests as it is established from the beginning that it is in a controlled environment. It was decided to work in 3 places that were not affected to a large extent by the various disturbances such as the nearby wifi networks,

**Figure 12.** Drone and its interface.



Source: own.

objects in the field and mainly the wind. Also, it is known that the drone makes use of its lower chamber to detect displacement changes in any direction by means of image processing. Therefore, the ground of the place must have figures or shapes that the drone can differentiate in case of displacement. In the same way, the luminosity of the place should be appropriate, as shown in Figures 12 and 13.

**Figure 13.** Drone in height execution.



Source: own.

## 5. Results

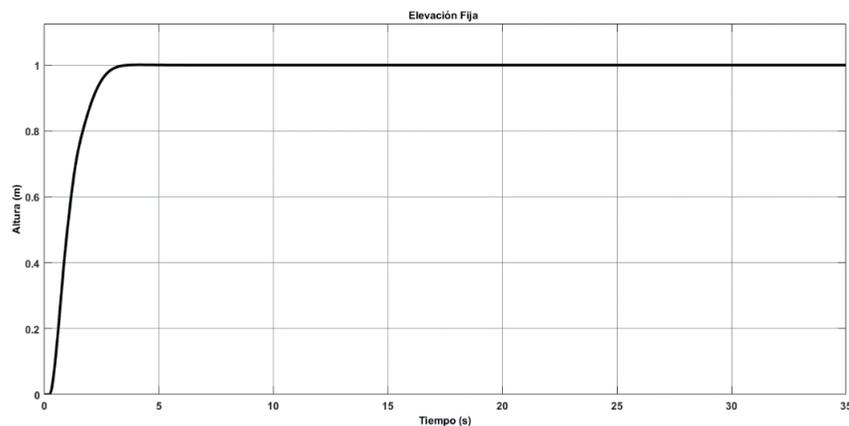
In order to control the elevation of the drone in flight, three fuzzy logic inference systems are designed. These are also used to make trajectory control. At the time of the Real and Simulated implementation, the systems are executed in three different ways for each one; The first form is the most basic and consists of adding a single reference point, that is, a flight with a fixed elevation. The second form consists of multiple

consecutive reference points and the third way is to enter reference points in both height and space for the drone to follow a pre-established route.

### 5.1. Stationary flight

The simulated response with a fixed reference point obtained is shown in Figure 14, the settling time is quite short and the steady state error is zero, in addition it does not show any overshoots.

**Figure 14.** Simulated stationary flight response.



Source: own.

In Figure 15, the real response of the height is shown and differs with the simulated response in Figure 14 since the former has a longer set-up time due to the communication delays between the computer and the drone. Also, although there are oscillations, their peaks do not exceed 10% of the total signal, thus getting a minimum and acceptable error. This behavior take place constantly in the flights performed.

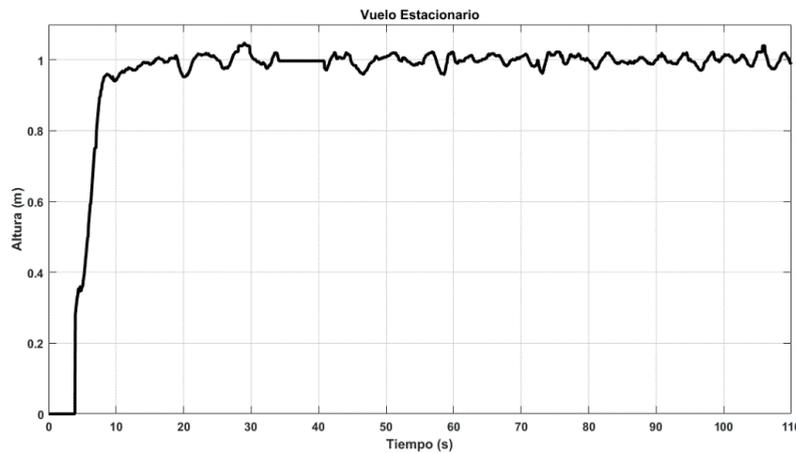
### 5.2. Flight multiple points

For the execution of the flight in multiple elevation reference points, three heights (1.7

meters, 1.2 meters and 0.7 meters) are used. The simulated response with multiple reference points is shown in Figure 16 where the good performance of the control system is shown under ideal parameters, thus having a near-zero error and a short establishment time.

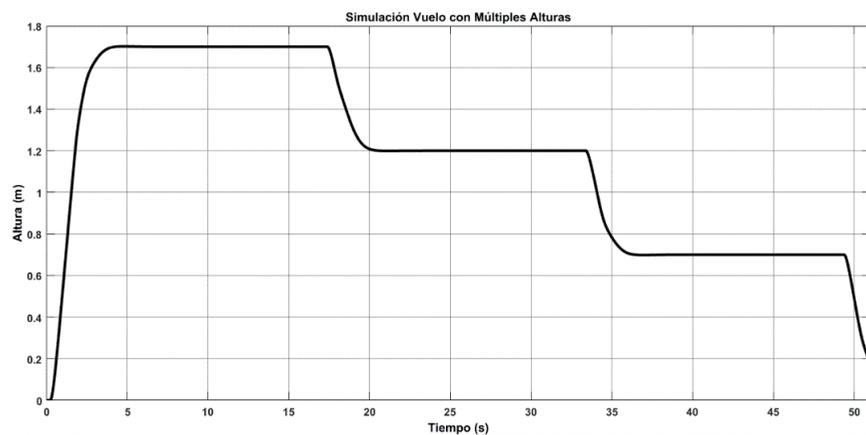
In Figure 17, the real response is shown, although the reference points are at different elevations, the performance of this is still optimal as the steady state error is less than 10% and the establishment times are minimum.

**Figure 15.** Real stationary flight response.

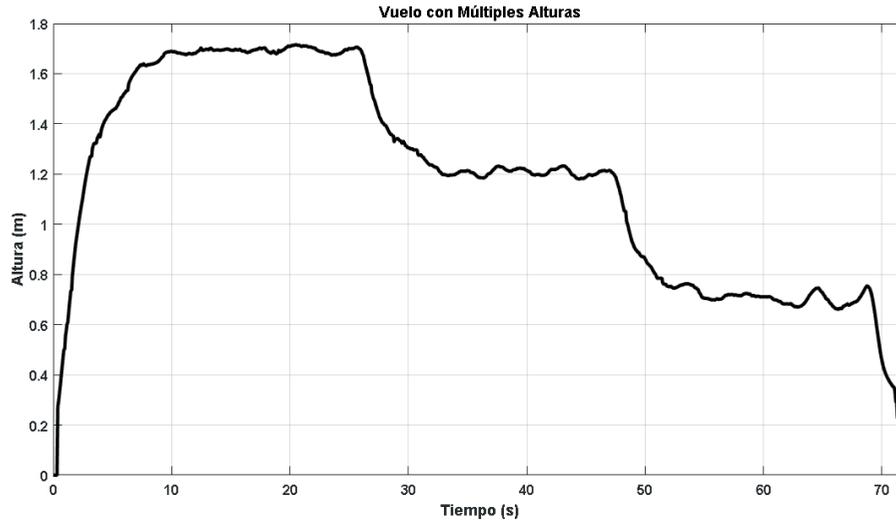


Source: own.

**Figure 16.** Simulated response with multiple points.



Source: own.

**Figure 17.** Real response with multiple points.

Source: own.

## 6. Conclusions

A height control system was successfully designed and implemented using fuzzy logic for the quadrotor AR Drone 2.0. The use of fuzzy logic does not require the mathematical model. In addition, a Mamdani type system was designed as it is more accepted at an academic level, although if it is required that the system has a higher computational efficiency, it is recommended to use a Sugeno system. To use the systems of fuzzy logic inference it is necessary to have a wide knowledge of the system to work with since this eases the development of input parameters, output parameters, the quantity and the type of membership functions per variable, the number of rules to be used and how it relates to each other so that the response of the system is as expected.

Fuzzy inference systems can rely on gains located at the inputs and outputs of its system. However, a good determination of the ranges to be used can substitute these gains. The membership functions were used in such a way that they were transposed to each other to reduce the uncertainty of the system as well as use functions such as triangular or Gaussian type one or type two. It is also advisable since the data is fully delimited. Nevertheless, the use of Gaussian-type membership functions is considered superior since the most relevant section will be a set of values and not a single value as in the triangular functions.

In terms of the number of membership functions to be used if there are few ranges of values contained should be very broad which can trigger the system has greater uncertainty. On the other hand, if too many functions are used for membership it can be solved as each level will be very well defined by a minimum range. However, since the fuzzy systems work by evaluating the parameters of the inputs to determine an optimal output if the functions are large, the decision making takes more time as it will have to evaluate more cases and the computational cost will be higher. For this reason, it is advisable to work with the largest number of membership functions for greater accuracy but always taking care that the equipment where the control system is running can support the computing.

To establish the rules, it is expected that the membership functions relate to each other, covering all possible cases, resulting in an optimal response for the system; therefore, the greater the number of membership functions, the greater the number of rules that relate them.

At the moment of controlling the system, it was intended to do so only on the Z variable since it is responsible for the elevation of the drone. However, when working with it, it is realized that the elevation is strongly related to the behavior of the other variables, that is, Pitch, Roll and Yaw. This takes place since the elevation only determines the power of the propellers of the drone so that the

angles of Pitch and Roll and maintain the desired spatial position the vehicle without unnecessary displacements and the Yaw so that the drone does not rotate on its same axis arbitrarily.

The surface on which the drone is located for the execution of the flight should be even as the drone determines its height based on the most immediate measurement below it. Then, if there are uneven or blunt objects they can directly affect the performance of the control system. Additionally, the surface must have colors, figures or shapes that identify it because the drone detects its displacement changes by means of the camera located at the bottom, if the ground is plain or the light conditions are not optimal, the drone will not perceive the changes of position or at least not quickly which will cause the control system is not executed efficiently.

The drone is made of lightweight materials which makes it not a very robust system especially in the presence of the wind which disturbs it considerably, making it deviates from its reference points during the flight; nearby wireless networks can affect the communication with the drone, another very important factor is the level of the battery because when this is less than 35% the whole system is compromised by the lack of power which will greatly complicate the controllability of the vehicle.

The registration and presentation of height levels traversed by the quadrotor in real time is achieved through a graphic interface made with Matlab® which, by allowing the data to be stored, makes it easier for the user to manipulate or analyze these later.

Data obtained from the tests performed show that when using a greater number of membership functions, the response of the system improves. For this particular case the data obtained from the flights that are implemented in the fuzzy control system with 11 membership functions are better than those obtained with the fuzzy system that only has 9 functions since the range of the variable to be used is divided into a greater number of spaces which makes the system more accurate.

Finally, it is worth mentioning that the performance of the three fuzzy controllers was as expected since their responses did not show an error over 10% and their establishment time was not very high.

## References

- [1] A. Gynnild, T. Uskali, “Responsible Drone Journalism”, London: Taylor & Francis Group, 2018. <https://doi.org/10.4324/9781315163659>
- [2] I. D. Pharne, S. Kanase, S. Patwegar, P. Patil, A. Pore and Y. Kadam, “Agriculture Drone Sprayer”, *International Journal of Recent Trends in Engineering & Research (IJRTER)*, vol. 4, no. 3, 2018. <https://doi.org/10.23883/ijrter.2018.4113.tvnqj>
- [3] T. Jones, “International Commercial Drone Regulation and Drone Delivery Services”, Corporación RAND, 2017. <https://doi.org/10.7249/rr1718.3>
- [4] A. A. Casanova, “Control Difuso del Quadrotor AR. Drone 2.0. Para el Seguimiento Autónomo de Trayectorias”, thesis, Universidad Tecnológica de la Mixteca, Oaxaca, 2015.
- [5] F. N. Castañeda, R. Henao and F. Valencia, “Diseño e Implementación del Sistema de Control de Vuelo de un UAV”, thesis, Universidad Católica de Manizales, Colombia, 2016.
- [6] A. F. Lasso, A. Murcia, “Diseño e Implementación de un Sistema de Guía Inercial para el Control de Vuelo Autónomo en un UAV”, thesis, Universidad Surcolombiana, Huila, 2012.
- [7] MathWorks, “Matlab, the Mathworks Inc.”, 1992. [Online]. Available: [https://la.mathworks.com/products/matlab.html?s\\_tid=hp\\_ff\\_p\\_matlab](https://la.mathworks.com/products/matlab.html?s_tid=hp_ff_p_matlab)
- [8] D. Escobar, “AR Drone Simulink Development Kit V1. 1”, 2013. [Online]. Available: <https://la.mathworks.com/matlabcentral/fileexchange/43719-ar-drone-simulink-development-kit-v1-1>
- [9] Simulink, “Simulation and Model-Based Design”, 2005. [Online]. Available: [https://www.mathworks.com/tagteam/73559\\_9320v07\\_overview.pdf](https://www.mathworks.com/tagteam/73559_9320v07_overview.pdf)
- [10] MathWorks, “Guide of Matlab®”. [Online]. Available: <https://la.mathworks.com/discovery/matlab-gui.html>