


AUTONOMOUS LANDING OF UAV: A COMPARISON BETWEEN TYPE-1 AND TYPE-2 FUZZY LOGIC SYSTEMS

ATERRISSAGEM AUTÔNOMA DE VANT: UMA COMPARAÇÃO ENTRE SISTEMAS LÓGICOS NEBULOSOS DO TIPO 1 E DO TIPO 2

Eduardo Ravaglia Campos Queiroz¹ 

Rodrigo Perobeli Silva Costa² 

Fernando Luiz Cyrino Oliveira³ 

André Luís Marques Marcato⁴ 

Eduardo Pestana de Aguiar⁵ 

Abstract: The Unmanned Aerial Vehicles (UAV) are an important technology with multiple applications. It is an object of study for researchers aiming to improve the performance of these vehicles, especially in flight stages as the landing. Therefore, this paper presents a method for the landing of a UAV based on Type-2 Fuzzy Logic System considering static targets. The advantage of this process is more precision and accuracy compared with Type-1 Fuzzy Logic System.

Keywords: UAV. Landing. Type-1 FLS. Type-2 FLS.

Resumo: Os Veículos Aéreos Não Tripulados (VANT) são uma tecnologia importante com múltiplas aplicações. Ele é um objeto de estudo para pesquisadores que buscam melhorar o desempenho desses veículos, principalmente em estágios de vôo como a aterrissagem. Portanto, este artigo apresenta um método para o pouso de um VANT baseado no Sistema Lógico Nebuloso do Tipo 2, considerando alvos estáticos. A vantagem deste processo é mais precisão e exatidão em comparação com o Sistema Lógico Nebuloso do Tipo 1.

Palavras-chave: VANT. Aterrissagem. Sistema lógico nebuloso do tipo 1. Sistema lógico nebuloso do tipo 2.

¹Mestrando em Engenharia de Produção, Pontifícia Universidade Católica do Rio de Janeiro, eduardo.campos@engenharia.ufjf.br.

²Mestrando em Modelagem Computacional, Universidade Federal de Juiz de Fora, rodrigo.costa2015@engenharia.ufjf.br.

³Doutor em Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, cyrino@puc-rio.br.

⁴Doutor em Engenharia Elétrica, Universidade Federal de Juiz de Fora, andre.marcato@ufjf.edu.br.

⁵Doutor em Engenharia Elétrica, Universidade Federal de Juiz de Fora, eduardo.aguiar@engenharia.ufjf.br.

1 INTRODUCTION

An Unmanned Aerial Vehicle (UAV) is an airframe and a computer system which combines sensors, Global Positioning System (GPS), servos, and Flight Control Unit (FCU) providing the aircraft's flight without human intervention. It is a revolutionary technology which is growing in the market, being a study object for researches around the world. Its applicability is present in several areas of society, as remote environmental research, pollution assessment and monitoring, fire-fighting management, security, border monitoring, agricultural and fishery applications, oceanography, communication relays for wide-band applications (PASTOR; LOPEZ; ROYO, 2007).

The UAV's operation is composed basically of five steps: takeoff, climb, cruise, descent, and landing. The last one is very important because it is the step which there can be accidents and, consequently, human and material losses. The automation of this stage is necessary because the many factors as the ability of operator and climate conditions can affect the success of the non-autonomous operation.

In some industrial applications, PID controllers are used because of their simple structure and the easy control design in the case when a process is linear or can be linearized at an operating point, however, there are processes that are not linear in the whole operating range but piecewise linearizable, for example, plants with nonlinear characteristics, chemical processes, robots, and UAVs (SOUZA *et al.*, 2019). The Fuzzy Logic System (FLS) is a good alternative of control that can manage highly nonlinear systems with uncertainties and do not need an accurate vehicle mathematical model (PRECUP; HELLENDORRN, 2011), which can be a problem for classical controllers. According to (SOUZA *et al.*, 2019), the FLS is real-time expert system implementing human experiences and knowledge, which can not be realized by PID and, in addition, the FLSs are the heuristic modular way of defining any non-linear control system, that flexibility is absent in PID.

Several works have been published in the literature about UAVs vision-based landing motivated by the FLS characteristics. The authors in (OLIVARES-MENDEZ; KANNAN; VOOS, 2015), adopting the aruco eye augmented reality package, adopted four fuzzy controllers to perform the UAV autonomous landing which one controller for each three-dimensional linear displacement and other for yaw orientation. The authors in (BENAVIDEZ *et al.*, 2014), adopted two fuzzy controllers and the package used to recognize and estimate the landing site is the ar track alvar. Although, these multiple concurrent controllers approaches are not practical and simplified alternative when the goal is the embedded use. The authors in (SOUZA *et al.*, 2019) present an alternative method to train a multilayer perceptron neural network based on fuzzy Mamdani logic to control the landing of a UAV on an artificial marker. The works

(SOUZA *et al.*, 2019; OLIVARES-MENDEZ; KANNAN; VOOS, 2015; BENAVIDEZ *et al.*, 2014) adopted the markers composition for landing spot making it an interesting alternative since different sizes are proposed for detection at distinct vehicle height levels.

According to (KARNIK; MENDEL, 1998), a Type-2 FLS arose due to the insufficiency of Type-1 FLS in modeling the uncertainties inherent in defining the antecedents and consequent membership functions in a fuzzy inference system, becoming an extension of the traditional fuzzy set. Thus, a Type-2 FLS opens up an efficient way of developing improved control systems and for modeling human decision making. Consequently, this paper addresses the UAV's autonomous landing using a Type-2 FLS due to its potential capability to solve robotics problems that have uncertainty about the membership degree, uncertainty about the membership functions size or uncertainty on some parameters of membership functions.

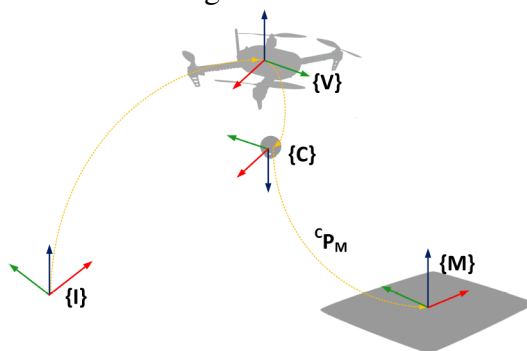
The main contribution of the present paper is an implementation of a UAV autonomous landing technique based on Type-2 FLS. The main goal is to realize the procedure with more precision and accuracy compared with Type-1 FLS considering static target.

This work is organized as follows: Section II presents the problem formulation, Section III presents the Interval Type-2 FLS, Section IV discusses the numerical results and Section V concludes the work.

2 PROBLEM FORMULATION

The procedure consists in detecting a landing spot, reducing the relative distance between the aircraft ($\{V\}$), and the marker ($\{M\}$). The image capture is done by a standard webcam and the computer vision algorithm returns the position of the marker (${}^C P_M$) relative to the camera frame ($\{C\}$). All the movements made by the aircraft are based on an inertial frame ($\{I\}$), the location where it is turned on. Figure 1 shows the frame coordinate system of the problem.

Figure 1: Frames configuration for vision-based landing

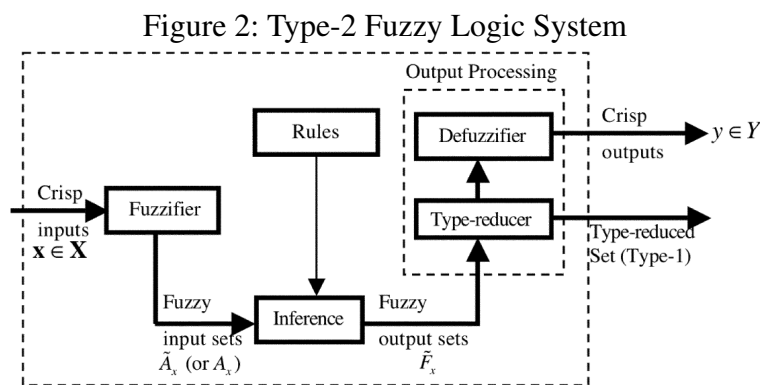


Source: (SOUZA *et al.*, 2019).

The landing logical is divided into two parts. The first part works with the horizontal adjustment to approximate the aircraft and the marker until it reaches a region acceptable for a landing spot that varies between -15% and 15% from the marker. At that moment, the second part works with the vertical adjustment until the height that the camera can not see and identify the marker due to the short distance and the landing is forced. If the aircraft goes away from the acceptable region during the vertical adjustment, the horizontal adjustment is performed again and this loop is repeated until the end of the process.

3 INTERVAL TYPE-2 FUZZY LOGIC SYSTEM

A general Type-2 FLS is shown at Figure 2. It is composed of fuzzifier, rules, inference, type-reducer and defuzzifier, being necessary to have the presence of, at least, one Type-2 Fuzzy Set in one of the antecedents or in the consequent, which compose one of the rules that form the system. The specific case of Interval Type-2 FLS happens when all of the antecedents and consequent are Type-2 Fuzzy Sets, being this approach used in this work.



Source: (MENDEL; JOHN; LIU, 2006).

3.1 Input Variables

It was adopted for the controller the horizontal distance (dX and dY) between the UAV and the landing spot in the X axis and Y axis of UAV body frame, respectively. Figure 3(a) shows the membership functions for dX (dY have similar membership functions) which are divided into three fuzzy trapezoidal groups: negative (red), close (blue) and positive (green). The variables indicate velocity values between -100% and 100%, relative to the field of view of the camera used. The Equations (1) - (6) represent them listed above, where NS means negative superior, NI means negative inferior, CS means close superior, CI means close inferior, PS means positive superior and PI means positive inferior. The other shapes in fuzzy sets are outside the scope of this paper.

$$\mu_{NS}(dX) = \begin{cases} 1 & \text{if } dX \leq -10 \\ -0.1dX & \text{if } -10 \leq dX \leq 0 \\ 0 & \text{if } dX \geq 0 \end{cases} \quad (1)$$

$$\mu_{CS}(dX) = \begin{cases} 0 & \text{if } dX \leq -15 \\ 0.1dX + 1.5 & \text{if } -15 \leq dX \leq -5 \\ 1 & \text{if } -5 \leq dX \leq 5 \\ -0.1dX + 1.5 & \text{if } 5 \leq dX \leq 15 \\ 0 & \text{if } dX \geq 15 \end{cases} \quad (2)$$

$$\mu_{PS}(dX) = \begin{cases} 0 & \text{if } dX \leq 0 \\ 0.1dX & \text{if } 0 \leq dX \leq 10 \\ 1 & \text{if } dX \geq 10 \end{cases} \quad (3)$$

$$\mu_{NI}(dX) = \begin{cases} 0.5 & \text{if } dX \leq -15 \\ -0.1dX - 1 & \text{if } -15 \leq dX \leq -10 \\ 0 & \text{if } dX \geq -10 \end{cases} \quad (4)$$

$$\mu_{CI}(dX) = \begin{cases} 0 & \text{if } dX \leq -5 \\ 0.1dX + 0.5 & \text{if } -5 \leq dX \leq 0 \\ -0.1dX + 0.5 & \text{if } 0 \leq dX \leq 5 \\ 0 & \text{if } dX \geq 5 \end{cases} \quad (5)$$

$$\mu_{PI}(dX) = \begin{cases} 0 & \text{if } dX \leq 10 \\ 0.1dX - 1 & \text{if } 10 \leq dX \leq 15 \\ 0.5 & \text{if } dX \geq 15 \end{cases} \quad (6)$$

3.2 Output Variables

It was adopted for the controller the linear velocity (vX , vY and vZ) in the UAV frame. Figure 3(b) shows the membership functions for vX (vY and vZ have similar membership functions) which are divided into three fuzzy trapezoidal groups: negative (red), zero (blue) and positive (green). The variables indicate velocity values between -100% and 100%, which will later be denormalized for actuation on the plant, according to an established maximum velocity of the UAV. The Equations (7) - (12) represent them listed above, where NS means negative superior, NI means negative inferior, ZS means zero superior, ZI means zero inferior, PS means positive superior and PI means positive inferior. The other shapes in fuzzy sets are outside the

scope of this paper.

$$NS = \int_{-15}^0 \frac{vX \mu_{NS}(dX)}{\mu_{NS}(dX)} \quad (7)$$

$$ZS = \int_{-15}^{15} \frac{vX \mu_{CS}(dX)}{\mu_{CS}(dX)} \quad (8)$$

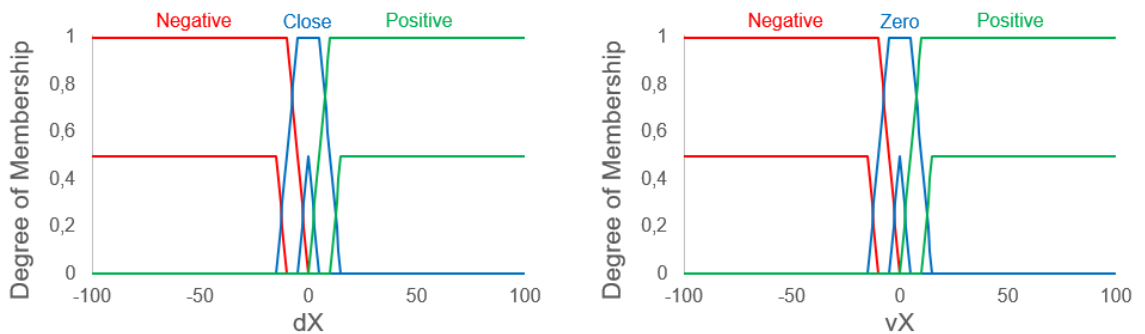
$$PS = \int_0^{15} \frac{vX \mu_{PS}(dX)}{\mu_{PS}(dX)} \quad (9)$$

$$NI = \int_{-15}^0 \frac{vX \mu_{NI}(dX)}{\mu_{NI}(dX)} \quad (10)$$

$$ZI = \int_{-15}^{15} \frac{vX \mu_{CI}(dX)}{\mu_{CI}(dX)} \quad (11)$$

$$PI = \int_0^{15} \frac{vX \mu_{PI}(dX)}{\mu_{PI}(dX)} \quad (12)$$

Figure 3: Membership Functions



(a) Input Variables (dX and dY)

(b) Output Variables (vX , vY and vZ)

Source: Author's own.

3.3 Fuzzifier

This module has the function of transforming crisp inputs (dX and dY) in Type-2 Fuzzy Sets. For this work singleton fuzzifier was used.

3.4 Rules

This module has the function of describing the relationship between linguistic variables, defining system's performance and behavior. Four experts worked for the build of Type-2 fuzzy sets. Its format is "if dX is ... and dY is ..., so vX (or vY , or vZ) is ...". As the combination of dX and dY includes the descent movement, the variable dZ did not participate on the rule base, being responsible for forcing the landing in a predetermined height.

Table 1: Rules

vX rules			vY rules			vZ rules		
Input		Output	Input		Output	Input		Output
dX	dY	vX	dX	dY	vY	dX	dY	vZ
Neg.	Neg.	Neg. (r01)	Pos.	Neg.	Neg. (r10)	Close	Close	Neg. (r19)
Neg.	Close	Neg. (r02)	Neg.	Neg.	Neg. (r11)	Pos.	Neg.	Zero (r20)
Neg.	Pos.	Neg. (r03)	Close	Neg.	Neg. (r12)	Pos.	Close	Zero (r21)
Close	Neg.	Zero (r04)	Pos.	Close	Zero (r13)	Pos.	Pos.	Zero (r22)
Close	Close	Zero (r05)	Neg.	Close	Zero (r14)	Neg.	Neg.	Zero (r23)
Close	Pos.	Zero (r06)	Close	Close	Zero (r15)	Neg.	Close	Zero (r24)
Pos.	Neg.	Pos. (r07)	Pos.	Pos.	Pos. (r16)	Neg.	Pos.	Zero (r25)
Pos.	Close	Pos. (r08)	Neg.	Pos.	Pos. (r17)	Close	Neg.	Zero (r26)
Pos.	Pos.	Pos. (r09)	Close	Pos.	Pos. (r18)	Close	Pos.	Zero (r27)

3.5 Inference

This module has the function of processing mathematically each rule proposition through approximate reasoning techniques, providing output from input combination. Considering the design of the FLS, it was adopted the minimum operator as the implication method. The Equations (13) - (21) represent the medium activation degrees for each one of the nine rules referring to vX .

$$f_1 = \frac{\min[\mu_{NS}(dX), \mu_{NS}(dY)] + \min[\mu_{NI}(dX), \mu_{NI}(dY)]}{2} \quad (13)$$

$$f_2 = \frac{\min[\mu_{NS}(dX), \mu_{CS}(dY)] + \min[\mu_{NI}(dX), \mu_{CI}(dY)]}{2} \quad (14)$$

$$f_3 = \frac{\min[\mu_{NS}(dX), \mu_{PS}(dY)] + \min[\mu_{NI}(dX), \mu_{PI}(dY)]}{2} \quad (15)$$

$$f_4 = \frac{\min[\mu_{CS}(dX), \mu_{NS}(dY)] + \min[\mu_{CI}(dX), \mu_{NI}(dY)]}{2} \quad (16)$$

$$f_5 = \frac{\min[\mu_{CS}(dX), \mu_{CS}(dY)] + \min[\mu_{CI}(dX), \mu_{CI}(dY)]}{2} \quad (17)$$

$$f_6 = \frac{\min[\mu_{CS}(dX), \mu_{PS}(dY)] + \min[\mu_{CI}(dX), \mu_{PI}(dY)]}{2} \quad (18)$$

$$f_7 = \frac{\min[\mu_{PS}(dX), \mu_{NS}(dY)] + \min[\mu_{PI}(dX), \mu_{NI}(dY)]}{2} \quad (19)$$

$$f_8 = \frac{\min[\mu_{PS}(dX), \mu_{CS}(dY)] + \min[\mu_{PI}(dX), \mu_{CI}(dY)]}{2} \quad (20)$$

$$f_9 = \frac{\min[\mu_{PS}(dX), \mu_{NS}(dY)] + \min[\mu_{PI}(dX), \mu_{NI}(dY)]}{2} \quad (21)$$

3.6 Type-Reducer

This module has the function of finding a Type-1 Fuzzy Set that better represents Type-2 Fuzzy Set, in such a way that in the absence of uncertainty, the results of Type-2 FLS are reduced to Type-1 FLS (MENDEL; JOHN; LIU, 2006). For that, Karnik Mendel (KM) Algorithm (KARNIK; MENDEL, 2001) that can be executed on parallel and are monotonically and exponentially convergent is used.

The Equations (22) - (24) represents the KM Algorithm to obtaining the pertinence resultant functions to the left (L). The Equation (22) represents the vX^I initially estimated, whose result is refined through Equations (23) or (24), depending on the interval in which vX^I is placed. The variable vX^I receives this new value and this iterative process repeats until a new value of vX^I becomes equal to the previous one.

$$vX^I = \frac{NI \sum_{i=1}^3 f_i + ZI \sum_{j=4}^6 f_j + PI \sum_{k=7}^9 f_k}{\sum_{i=1}^3 f_i + \sum_{j=4}^6 f_j + \sum_{k=7}^9 f_k} \quad (22)$$

$$vX^I = \frac{NI \sum_{i=1}^3 f_i^S + ZI \sum_{j=4}^6 f_j^I + PI \sum_{k=7}^9 f_k^I}{\sum_{i=1}^3 f_i^S + \sum_{j=4}^6 f_j^I + \sum_{k=7}^9 f_k^I}, \text{ if } NI \leq vX^I \leq ZI \quad (23)$$

$$vX^I = \frac{NI \sum_{i=1}^3 f_i^S + ZI \sum_{j=4}^6 f_j^S + PI \sum_{k=7}^9 f_k^I}{\sum_{i=1}^3 f_i^S + \sum_{j=4}^6 f_j^S + \sum_{k=7}^9 f_k^I}, \text{ if } ZI \leq vX^I \leq PI \quad (24)$$

The Equations (25) - (27) represents the KM Algorithm to obtaining the pertinence resultant functions to the right (R). The Equation (25) represents the vX^S initially estimated, whose result is refined through Equations (26) or (27), depending on the interval in which vX^S is placed. The variable vX^S receives this new value and this iterative process repeats until a new value of vX^S becomes equal to the previous one.

$$vX^S = \frac{NS \sum_{i=1}^3 f_i + ZS \sum_{j=4}^6 f_j + PS \sum_{k=7}^9 f_k}{\sum_{i=1}^3 f_i + \sum_{j=4}^6 f_j + \sum_{k=7}^9 f_k} \quad (25)$$

$$vX^S = \frac{NS \sum_{i=1}^3 f_i^I + ZS \sum_{j=4}^6 f_j^S + PS \sum_{k=7}^9 f_k^S}{\sum_{i=1}^3 f_i^I + \sum_{j=4}^6 f_j^S + \sum_{k=7}^9 f_k^S}, \text{ if } NS \leq vX^S \leq Z \quad (26)$$

$$vX^S = \frac{NS \sum_{i=1}^3 f_i^I + ZS \sum_{j=4}^6 f_j^I + PS \sum_{k=7}^9 f_k^S}{\sum_{i=1}^3 f_i^I + \sum_{j=4}^6 f_j^I + \sum_{k=7}^9 f_k^S}, \text{ if } ZS \leq vX^S \leq PS \quad (27)$$

3.7 Defuzzification

This module has the function of transforming Type-1 Fuzzy Sets in crisp output (vX , vY and vZ). For the present work, centroid defuzzification was used. The Equation (28) represents this transformation.

$$vX = \frac{vX^I + vX^S}{2} \quad (28)$$

The step by step for calculating vY and vZ is the same as for vX .

4 EXPERIMENTAL RESULTS

Software In The Loop (SITL) consists of running the real PX4 firmware and the proposed algorithms in a virtual experiment. It is realized in Gazebo software, a simulation environment that simulates physics and dynamics of bodies close to reality. The code is implemented in C++ along with the ROS framework and the experiment is proposed with a static target. The landing procedure started with the UAV surrounding the marker about 8 meters from the ground. By the time the procedure starts, the proposed algorithm takes control of the aircraft, initiating a horizontal displacement towards the center of the marker. Then, the vertical error is reduced by moving the aircraft downwards until reaching the landing marker, while simultaneously correcting any horizontal disturbances. The stability analysis is outside the scope of this paper.

The simulation experimentation was repeated 30 from same starting points ($dX = 2m$,

$dY = 2\text{m}$ and $dZ = 8\text{m}$) and the results are shown in Table 2, where dX means the distance of the target on the X axis and dY means the distance of target on the Y axis. The radial error is calculated through Pythagorean Theorem at Equation (29).

$$\text{Radial Error} = \sqrt{dX^2 + dY^2} \quad (29)$$

The landing based on Type-2 FLS returned an average radial error, from the center of mass of the vehicle to the central marker of the landing spot, of 0.13 ± 0.06 m. The results for Type-1 FLS under the same conditions was 0.23 ± 0.09 m. The completed experimental results of Type-1 FLS can be founded in (SOUZA *et al.*, 2019). Aiming to compare these error values between the algorithms a hypothesis test is adopted.

The two-sample t -test allows us to infer assumptions from two independent data samples and to verify their validity statistically. This statistical test is represented by Equation (30) (MOORE; KIRKLAND, 2007), where $\overline{G_1}$ and $\overline{G_2}$ are the mean values of the samples G_1 and G_2 , sG_1 and sG_2 are the standard deviations of the samples, n and m correspond to the size of the sample sets sG_1 and sG_2 , respectively. The degree of freedom is defined as $n + m - 2$.

$$t = \frac{\overline{G_1} - \overline{G_2}}{\sqrt{\frac{sG_1^2}{n} + \frac{sG_2^2}{m}}} \quad (30)$$

In addition to the determination of t , it becomes important to infer the hypotheses. The hypotheses are presented in Equation (31), where H_0 is the null hypothesis (which indicates that both algorithms have obtained the same mean radial error value) and H_1 is the alternative hypothesis (which indicates that both algorithms have obtained the distinct mean radial error value).

$$\begin{cases} H_0 : G_1 = G_2 \\ H_1 : G_1 \neq G_2 \end{cases} \quad (31)$$

Given a significance level α , the p -value is calculated from t and represents the lowest value of α to reject the null hypothesis. Thus, values of the p -value below α show that the null hypothesis is not true (MOORE; KIRKLAND, 2007).

For the landing problem, it is intended by the t -test to verify that the mean radial errors obtained by the experiments with Type-2 FLS and Type-1 FLS are not equivalent. For this, the sets of samples G_1 and G_2 of Equation (30), are the radial error data obtained for the 30 landings made in SITL for each one FLS. As the degree of freedom is relatively high, equal to

58, it is not necessary to verify the normality of errors distributions (MOORE; KIRKLAND, 2007).

For $\alpha = 0.05$, $p\text{-value} = 2.4683e - 6$. As $p\text{-value}$ is less than α , it is possible to reject the null hypothesis of equal means for landings, that is, there is statistical difference between the mean radial error generated by landings of Type-2 FLS and Type-1 FLS.

The Equation (32) is used to calculate the radial error percentage of Type-2 FLS in relation to Type-1 FLS. The term $radial.error_{t2}$ is the radial error of Type-2 FLS and the term $radial.error_{t1}$ is the radial error of Type-1 FLS obtained in (SOUZA *et al.*, 2019). The results was 45.53%.

$$\%error = \frac{radial.error_{t2} - radial.error_{t1}}{radial.error_{t1}} \quad (32)$$

5 CONCLUSION

This paper addressed a comparison between two controllers for autonomous landing of UAV: Type-2 FLS and Type-1 FLS. Numerical results obtained through computational simulations revealed that there is an extremely statistically significant between Type-1 FLS and Type-2 FLS and it had a better performance in relation to the precision and accuracy of the landing.

For futures works, the next steps consists in to compare Type-2 FLS and Type-1 FLS controllers for a target in linear motion, circular motion and path with obstacles.

ACKNOWLEDGMENTS

The authors acknowledges the support from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Table 2: Experimental Results of Type-2 Fuzzy

Test	dX(m)	dY(m)	Radial Error(m)
1	0.16510	0.09151	0.18877
2	0.13331	0.01687	0.13437
3	0.10378	0.09699	0.14204
4	0.11228	0.02003	0.11405
5	0.14863	0.00485	0.14871
6	0.11865	-0.00039	0.11865
7	0.05312	0.02824	0.06016
8	0.04253	0.12603	0.13301
9	-0.15710	-0.18476	0.24252
10	-0.05812	-0.07576	0.09548
11	-0.09553	-0.10375	0.14103
12	0.10867	0.01706	0.11000
13	0.07743	0.03033	0.08316
14	0.05721	0.03733	0.06831
15	0.14052	0.01637	0.14147
16	0.04936	0.05312	0.07252
17	-0.03210	0.02738	0.04219
18	-0.11266	-0.25297	0.27692
19	0.01616	-0.23323	0.23379
20	0.07177	-0.01858	0.07414
21	0.11472	0.06255	0.13066
22	0.15428	0.04237	0.15999
23	0.09052	0.05433	0.10557
24	0.11080	-0.06562	0.12877
25	0.14394	0.08766	0.16853
26	0.16185	0.05016	0.16945
27	0.07899	0.07911	0.11179
28	-0.14431	-0.05949	0.15609
29	-0.03696	-0.00234	0.03703
30	0.07653	-0.06818	0.10249
Average	-	-	0.12972 ± 0.05607

REFERENCES

- BENAVIDEZ, P. *et al.* Landing of an ardrone 2.0 quadcopter on a mobile base using fuzzy logic. In: IEEE. *2014 World Automation Congress (WAC)*. [S.l.], 2014. p. 803–812.
- KARNIK, N. N.; MENDEL, J. M. Introduction to type-2 fuzzy logic systems. In: IEEE. *1998 IEEE international conference on fuzzy systems proceedings. IEEE world congress on computational intelligence (Cat. No. 98CH36228)*. [S.l.], 1998. v. 2, p. 915–920.
- KARNIK, N. N.; MENDEL, J. M. Centroid of a type-2 fuzzy set. *information SCIences*, Elsevier, v. 132, n. 1-4, p. 195–220, 2001.
- MENDEL, J. M.; JOHN, R. I.; LIU, F. Interval type-2 fuzzy logic systems made simple. *IEEE transactions on fuzzy systems*, IEEE, v. 14, n. 6, p. 808–821, 2006.
- MOORE, D. S.; KIRKLAND, S. *The basic practice of statistics*. [S.l.]: WH Freeman New York, 2007. v. 2.
- OLIVARES-MENDEZ, M. A.; KANNAN, S.; VOOS, H. Vision based fuzzy control autonomous landing with uavs: From v-rep to real experiments. In: IEEE. *2015 23rd Mediterranean Conference on Control and Automation (MED)*. [S.l.], 2015. p. 14–21.
- PASTOR, E.; LOPEZ, J.; ROYO, P. Uav payload and mission control hardware/software architecture. *IEEE Aerospace and Electronic Systems Magazine*, IEEE, v. 22, n. 6, p. 3–8, 2007.
- PRECUP, R.-E.; HELLENDORRN, H. A survey on industrial applications of fuzzy control. *Computers in industry*, Elsevier, v. 62, n. 3, p. 213–226, 2011.
- SOUZA, J. P. C. de *et al.* Autonomous landing of uav based on artificial neural network supervised by fuzzy logic. *Journal of Control, Automation and Electrical Systems*, Springer, v. 30, n. 4, p. 522–531, 2019.

Edição especial - XXII ENMC (Encontro Nacional de Modelagem Computacional) e X ECTM (Encontro de Ciência e Tecnologia dos Materiais)

Enviado em: 27 mai. 2020

Aceito em: 10 ago. 2020

Editor responsável: Rafael Alves Bonfim de Queiroz