LANDING A UAV IN A DYNAMICAL TARGET USING FUZZY CONTROL AND COMPUTER VISION

João Pedro Carvalho*, Marco Jucá*, Alexandre Menezes*, Leonardo Olivi*, Alexandre Bessa*, André Marcato*

*Universidade Federal de Juiz de Fora
Programa de Pós Gradução em Engenharia Elétrica
Juiz de Fora, MG, Brasil

Emails: joao.pedro@engenharia.ufjf.br, marco.juca@engenharia.ufjf.br, alexandre.menezes@engenharia.ufjf.br , leonardo.olivi@ufjf.edu.br , alexandre.bessa@engenharia.ufjf.br , almmarcato@gmail.com

Abstract— Unmanned Aerial Vehicles (UAVs) are a recent brand of technological research field where multiple applications can be designed, such as surveillance, deliveries and others. The localization of the UAVs are done with Global Positioning System (GPS), which does not offer sufficient precision for landing, especially at dynamical landing points. To solve this imperative issue, many applications with computational vision and control theory have been developed. This paper presents a novel controller based on Mamdani-type fuzzy logic and augmented reality tag identification for landing of a UAV in a dynamical target point. Results are presented with simulation and a real drone, the IRIS+ with Odroid and ROS.

Keywords— Unmanned Aerial Vehicles, Fuzzy Control, Computer Vision, Augmented Reality, ROS, Odroid.

1 Introduction

Recently, Unmanned Aerial Vehicles (UAVs) such as drones and quadcopters have gained the attention of the research field community due to its many applications in all fields of robotics, such as surveillance of patrimony (Saska et al., 2012; Nagarjuna and Suresh, 2015), inspection of land areas such as forests, rivers and others for environmental analysis as deforestation and pollution (Teixeira et al., 2014), deliveries of small types of material, such as books, some enterprises such as Amazon proposes (Amazon.com, 2015).

The large amount of energy required to sustain these aerial robots in air is critical, and also a field of research, for optimization of the consumption of the batteries energy. This is reflected on the Amazon.com delivery system (Amazon.com, 2015), which reaches locations thirty minutes away from the warehouses, which makes a full flight of one hour. The researchers have been dealing with the motor's power consumption to improve its effectiveness, in order to extend the time of flight, such as control (Benavidez et al., 2014), landing bases to provide recharging locations (Wenzel et al., 2011; Saska et al., 2012) and many others.

A critical phase of this applications is the landing of the UAVs in a specific area. A landing area have, in many applications, the same size of the robot to fit itself in small locations, such as boats, other mobile robots, recharging platforms, and so on (Raza and Gueaieb, 2010; Benavidez et al., 2014). Many applications of path planning for UAVs consider Global Positioning System (GPS) coordinates for the position of the robot. The GPS data is indeed a good alternative for navigation in long ranges. However, the precision of the GPS is not sufficient to guarantee the landing of the UAV in a small location safely. Concerning the landing of a UAV, several works have been published in the literature to guarantee the precision and accuracy of the robot (Raza and Gueaieb, 2010; Wenzel et al., 2011; Benavidez et al., 2014; Nagarjuna and Suresh, 2015).

This paper presents a controller based on fuzzy logic and computer vision to land an IRIS+ from 3DRobotics(3DRobotics, 2015) controlled by Pixhawk and an Odroid computer (HardKernel, 2015) with Linux and ROS, through MAVROS, in a specific dynamic target. Section 2 has the related works in the area. Section 3 describes the whole system, such as the UAV, the vision system, etc. Section 4 describes the fuzzy controller and its particularities. Section 5 shows the results of the experiments. Finally, Section 6 concludes the paper.

2 Related works

The authors of (Saska et al., 2012) used a mobile robot as dynamic base for the landing of an UAV. With a P3AT from MobileRobots^{TM} as mobile base, an ArDrone quadcopter was used with classical controllers to control roll, pitch, yaw and altitude of the drone. The researchers used computer vision inputs for the controllers, which were processed in both aerial and terrestrial bases. The mobile base was embedded with actuators for shifting the drone in order to reduce the error at the landing.

The authors of (Nagarjuna and Suresh, 2015) monitor a landing mechanism for an UAV by gyroscopes and accelerometers without image processing. The sensors are used to verify the altitude of the robot. Therefore, the altitude data is used to control the velocity of the motors, stabilizing and descending the robot to the desired point. Despite the landing control, the authors also establish a predefined path from a remote computer for the UAV, which follows the path for surveillance of the area. In the case of losing a motor during flight, failsafe measures are taken, were the other motors are put to work at 120% of their capacity to cover the loss.

In (Wenzel et al., 2011), the authors used a low-cost Wii infrared (IR) camera to capture light from light emitting diodes (LEDs) in order to control the take off, flight and landing of an UAV at a mobile robot base without the presence of direct sunlight. The control of the UAV considered several variables, and the experiments developed the tracking of the mobile base with a constant distance and also with circular movements.

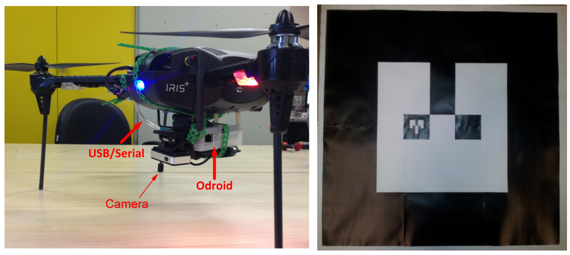
The use of artificial intelligence for the control of aerial robots has increased. Especially in the use of fuzzy logic, which can manage highly nonlinear systems with disturbances, where some classical approaches cannot (Pedrycz and Gomide, 2007; Raza and Gueaieb, 2010; Benavidez et al., 2014; Sureshkumar and Cohen, 2014). The authors of (Raza and Gueaieb, 2010) and (Sureshkumar and Cohen, 2014) described the control of a quadrotor UAV using fuzzy control. Authors used MATLAB™ for real time sensor data acquisition and timing of control loops. The authors were able to control the quadrotor's flight and landing through a set of fuzzy rules in simulation and hardware in the loop.

In (Benavidez et al., 2014), the authors describe the control of a Ardrone 2.0 quadcopter from Parrot™ with fuzzy logic to support a mobile base. The base acts as a launching and landing pad for the UAV, which provides environmental information through image to the base. The authors perform the simulation of altitude and landing control, where the input variables are extracted via computer vision.

In this paper a fuzzy logic controller is proposed to control a real quadrotor UAV IRIS+ with an Odroid U3 computer with Linux and ROS embedded. Results shown a good controller, dealing with all processes' nonlinearities and disturbances.

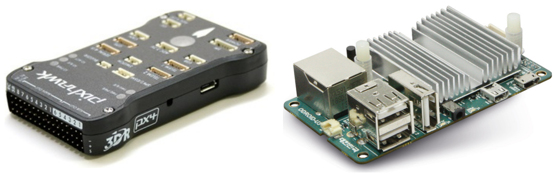
3 System description

In this work, an IRIS+ UAV is deployed in an environment containing a landing point, which is a set of printed tags from the ROS package *ar_track_alvar* in an appropriate scale. The UAV robot and the landing point are shown in Figure 1. The Figure 1a presents the IRIS+ robot with an embedded computer Odroid U3 (Figure 2b) and camera. The robot has its inherent controller board, Pixhawk (Figure 2a), which executes the robot's low-level control commands.

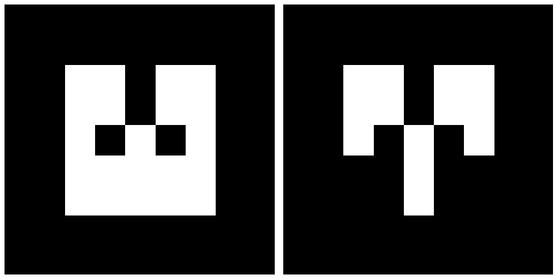


a) IRIS+ and vision setup. b) Landing point.

Figure 1: The UAV and the landing point.



a) Pixhawk. b) Odroid U3.



c) *Ar_track_alvar* tag 00. d) *Ar_track_alvar* tag 15.

Figure 2: Set for control and computer vision algorithms.

The Odroid U3 has ROS installed and executes the high-level control and vision routines. The *ar_track_alvar* package is an augmented reality software which recognizes predefined tags and, then, can deliver the camera's position and orientation with respect to the tags through 3D positions and a quaternion for orientation. The *ar_track_alvar* tags are shown in Figures 2. The Odroid U3 is needed for the vision algorithm, once Pixhawk doesn't have sufficient computing power.

The landing point, shown in Figure 1b is compound by two of the several *ar_track_alvar* tags. These two tags are the number 00 and 15, respectively, and are shown in Figures 2c and 2d. All *ar_track_alvar* tags are squared and binary images. In this work, as can be seen in Figure 1b, the tags are printed in different scales. The bigger tag, number 00, has 55 centimeters of side and the smaller, number 15, has 5.5 centimeters of side. These scales must be informed for *ar_track_alvar*, so the software calculates both on the same plane at the same distance from the camera.

The tag 15 is embedded in 00 as some strategies propose through literature (Benavidez et al., 2014), for a technical purpose. When the UAV is at its higher altitudes during the mission, the software identifies only the bigger tag in the images. As the robot approximates the landing point, the bigger tag, eventually, will not be able for the camera's aperture because of its size. Without a tag, *ar_track_alvar* cannot calculate the robot's position. Therefore, the smaller tag will take place for the landing algorithm finishes the control loop.

To command the UAV with an external controller, another package from ROS is used, *MAVROS*. The *MAVROS* package is the compilation of the *MAVlink* (Micro Air Vehicle Link) protocol library for ROS. Therefore, the data provided by *ar_track_alvar* is used as input for the controller, which generates the control law for the UAV. This control law is applied with *MAVROS* in the control loop. The robot's mission is to land on the landing point with sufficient precision, because this base can be placed, for instance, on a boat or on another mobile robot, then, the UAV cannot miss the landing.

4 The fuzzy controller

The fuzzy controller described in this section receives the inputs from the augmented reality software *ar_track_alvar*.

4.1 The input variables

The *ar_track_alvar* provides the UAV's three-dimensional position $p_{x,y,z}$ and its orientation given by a quaternion $q_{w,x,y,z}$ relative to the tag's position. Also, robot has a heading concerning its front in the global frame p_θ, which is the Euler angle for heading or rotation in z-axis, given by $p_\theta = atan2(2q_yq_w - 2q_xq_z, 1 - 2q_y^2 - 2q_z^2)$. For this work, the tag's position is at the origin of the system. From these parameters we extract three input variables for the controller: ρ is the distance between the UAV and the landing point at the same horizontal plane; α is the heading angle between the robot and the landing point in the plane of the variable ρ; and h is the altitude of the UAV relative to the landing point.

The input parameters are described in Figure 3 and their equations in robot's frame are:

$$\rho = \sqrt{p_x^2 + p_y^2} \tag{1}$$

$$\alpha = atan2\left(p_y, p_x\right) - p_\theta \tag{2}$$

$$h = p_z \tag{3}$$

The variables ρ and α treat the UAV as a bi-dimensional robot moving in a plane. This happens because the robot will not start landing until it reaches a region straight above to the base. This strategy avoids collision with vegetation and other

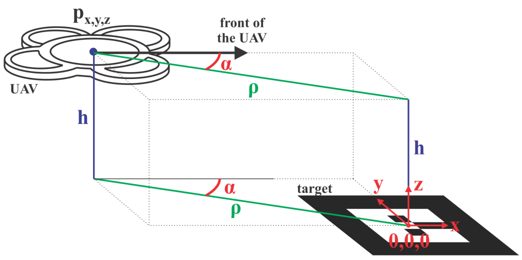


Figure 3: Set for computer vision algorithms.

obstacles near the landing point. The regions considered for the UAV are shown in Figure 4.

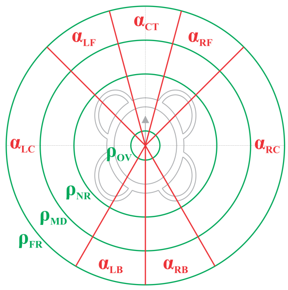


Figure 4: Fuzzy regions for the UAV.

The variable ρ has four regions concerning the distance between the UAV and the landing point: OV (Over), NR (Near), MD (Medium) and FR (Far). The variable α has seven regions concerning the angle between the UAV and the landing point: CT (Center), LF (Left Front), LC (Left Center), LB (Left Behind), RF (Right Front), RC (Right Center) and RB (Right Behind). The variable h has four regions concerning the altitude to the ground plane: GD (Ground), LW (Low), MD (Medium) and HG (High).The regions at Figure 4 shape the fuzzy sets designed for the input variables, shown in Figures 5a, 5b and 5c.

The range of the variable ρ is within $[0, 100\%]$, which the calculated distance must be normalized by a predefined maximum distance ρ_{max} with the relation $\rho_{crisp} = min(100\rho/\rho_{max}, 100)$. For the variable α, its range is in $[-180°, 180°]$ with no need for normalization. The altitude h behaves in the same way as ρ, within the range $[0, 100\%]$, with a maximum allowed altitude h_{max} with the relation $h_{crisp} = min(100h/h_{max}, 100)$.

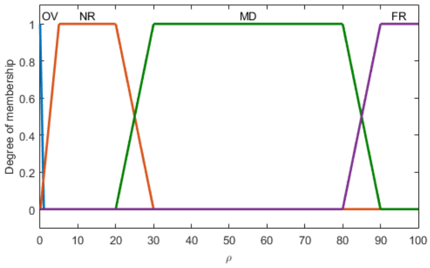
4.2 The output variables

The UAV, through *MAVROS* package, offers many variables to design control. This work considers four output variables of displacement: three of linear displacement Δx, Δy, Δz and one of angular displacement $\Delta\theta_Z$ for heading. All variables are in the frame of the robot, not the global.

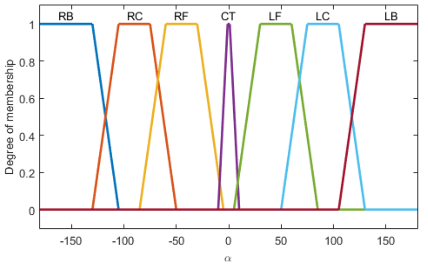
The variables Δx, Δy, Δz are configured with the same fuzzy sets, all shown in Figure 6a to provide the same in linear movement for the UAV

ISSN 2525-8311

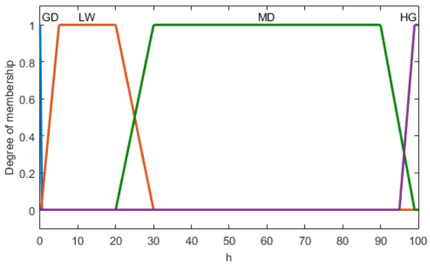
in all directions. The fuzzy sets are: NH (Negative High), NM (Negative Medium), NS (Negative Small), NT (Neutral), PS (Positive Small), PM (Positive Medium) and PH (Positive High).



a) Fuzzy sets for the input variable ρ.



b) Fuzzy sets for the input variable α.



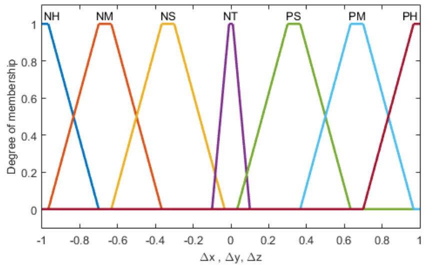
c) Fuzzy sets for the input variable h.

Figure 5: Input variables ρ, α and h.

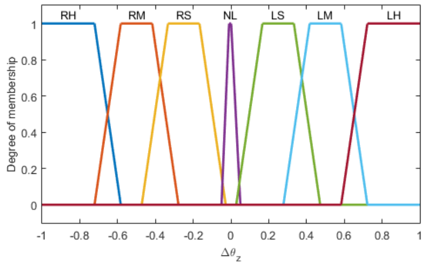
The variable $\Delta\theta_Z$ is the rotational movement in the z-axis, which means the heading movement. Their fuzzy sets are shown in Figure 6b and are: RH (Right High), RM (Right Medium), RS (Right Small), NL (Null), LS (Left Small), LM (Left Medium) and LH (Left High). All output variables are normalized in the range $(-1, 1)$.

4.3 The fuzzy rules

As described in previous sections, the control strategy will treat the UAV as a bi-dimensional robot while the distance ρ is not placed in the fuzzy set OVER. This strategy maintains the robot in its maximum altitude to avoid the collision with medium or small height obstacles, such as tree branches, which might be in the way. Therefore, if altitude h is High (HG), the fuzzy sets are shown in Table 1. There rules are designed to align the UAV with its target and navigate towards it. As result, the robot will reach the distance ρ_{OV} (Over), which means the drone



a) Fuzzy sets for the variables $\Delta x, \Delta y, \Delta z$.



b) Fuzzy sets for the output variable $\Delta\theta_z$.

Figure 6: Output variables $\Delta x, \Delta y, \Delta z$ and $\Delta\theta_z$.

is over the target and can simply land.

Table 1: Fuzzy rules if h is *HG*.

		ρ		
		NR	MD	FR
α	RB	Δx=NS Δy=NS Δz=NT $\Delta\theta_z$=NL	Δx=NT Δy=NT Δz=NT $\Delta\theta_z$=RH	Δx=NT Δy=NT Δz=NT $\Delta\theta_z$=RH
	RC	Δx=NT Δy=NS Δz=NT $\Delta\theta_z$=NL	Δx=NT Δy=NT Δz=NT $\Delta\theta_z$=RM	Δx=NT Δy=NT Δz=NT $\Delta\theta_z$=RM
	RF	Δx=PS Δy=NS Δz=NT $\Delta\theta_z$=NL	Δx=NT Δy=NT Δz=NT $\Delta\theta_z$=RS	Δx=NT Δy=NT Δz=NT $\Delta\theta_z$=RS
	CT	Δx=PS Δy=NT Δz=NT $\Delta\theta_z$=NL	Δx=PM Δy=NT Δz=NT $\Delta\theta_z$=NL	Δx=PH Δy=NT Δz=NT $\Delta\theta_z$=NL
	LF	Δx=PS Δy=PS Δz=NT $\Delta\theta_z$=NL	Δx=NT Δy=NT Δz=NT $\Delta\theta_z$=LS	Δx=NT Δy=NT Δz=NT $\Delta\theta_z$=LS
	LC	Δx=NT Δy=PS Δz=NT $\Delta\theta_z$=NL	Δx=NT Δy=NT Δz=NT $\Delta\theta_z$=LM	Δx=NT Δy=NT Δz=NT $\Delta\theta_z$=LM
	LB	Δx=NS Δy=PS Δz=NT $\Delta\theta_z$=NL	Δx=NT Δy=NT Δz=NT $\Delta\theta_z$=LH	Δx=NT Δy=NT Δz=NT $\Delta\theta_z$=LH

If the robot's distance ρ is *OV*, the Table 2 shows the rules for a smooth descent at the landing point. This is the phase in which all commands of the robot where correctly executed and the disturbances were not able insert effective error on the UAV's performance.

Table 2: Fuzzy rules if ρ is OV.

		h			
		HG	MD	LW	GD
ρ	OV	Δx=NT Δy=NT Δz=NH $\Delta\theta_z$=NL	Δx=NT Δy=NT Δz=NM $\Delta\theta_z$=NL	Δx=NT Δy=NT Δz=NS $\Delta\theta_z$=NL	Δx=NT Δy=NT Δz=NT $\Delta\theta_z$=NL

However, several situations were disturbances can cause harm to the robot must be considered in order to protect the robot. For instance, as wind blows, noise may alter the robot's altitude, or at small heights, collision is a possibility. Tables 3 and 4 treat the cases. If the robot is far from the landing point, navigating with low altitude is dangerous, then the controller increases UAV's altitude with no displacements in the x and y-axis for safety. However, if the UAV is near the landing point, it is counter-productive to increase the robot's altitude, and then, the robot will track the mission's point from its current altitude.

Table 3: Recovery strategy when ρ is FR or MD.

		h		
		MD	LW	GD
ρ	FR	Δx=NT Δy=NT Δz=PS $\Delta\theta_z$=NL	Δx=NT Δy=NT Δz=PM $\Delta\theta_z$=NL	Δx=NT Δy=NT Δz=PH $\Delta\theta_z$=NL
	MD	Δx=NT Δy=NT Δz=PS $\Delta\theta_z$=NL	Δx=NT Δx=NT Δz=PM $\Delta\theta_z$=NL	Δx=NT Δy=NT Δz=PH $\Delta\theta_z$=NL

5 Results

For calibration and analyses of the algorithms are proposed simulations. The simulations were with ROS and hardware in the loop performed on jMAVSim Simulator (Project, 2015). All fuzzy sets and rules were calibrated with MATLAB™ Fuzzy Logic Toolbox due to its facility in design and debug, and the controller was implemented in *C++* on Odroid with the Fuzzy Lite (Rada-Vilela, 2014), which is a user-friendly *C++* library.

The experiments were realized in three stages. Landing in a static goal, in a target moving in straight line trajectory with constant linear velocity and with a target moving in a circular trajectory with constant linear and angular velocities.

5.1 Simulation results

Results obtained with simulation establish the ground truth of the full capabilities of the software developed. In this results, the robot starts

Table 4: Recovery strategy when ρ is NR.

		h
		MD or LW
α	RB	Δx=NS Δy=NS Δz=NT $\Delta\theta_z$=NL
	RC	Δx=NT Δy=NS Δz=NT $\Delta\theta_z$=NL
	RF	Δx=PS Δy=NS Δz=NT $\Delta\theta_z$=NL
	CT	Δx=PS Δy=NT Δz=NT $\Delta\theta_z$=NL
	LF	Δx=PS Δy=PS Δz=NT $\Delta\theta_z$=NL
	LC	Δx=NT Δy=PS Δz=NT $\Delta\theta_z$=NL
	LB	Δx=NS Δy=PS Δz=NT $\Delta\theta_z$=NL

far from the landing point. The UAV has to take off with a heading error of 180°, to test all fuzzy rules developed in Section 4.

For all experiments, the maximum allowed altitude h_{max} is 10 meters, linear displacements $\Delta x, \Delta y, \Delta z$ (outputs from the fuzzy controller) are scaled by 1 meter and angular displacement $\Delta\theta_z$ is scaled by 3° after calibration. The linear velocity of the landing point in the dynamical experiments was fixed in 0.1 meters/second and the radius was 10 meters.

The UAV results are shown in the Figures 7, 8 and 9. The experiments were performed with several starting points in order to evaluate the repeatability and accuracy of the controller, and to acquire measurements for statistical analysis of the landing error to the target point.

In Figure 7, it is possible to see the robot's tree-dimensional trajectory in red and bi-dimensional in gray. The target point is at the origin of the camera's frame, and additional frame homogeneous transformations are done with *tf* ROS package.

The figure shows the UAV's take off to its maximum allowed altitude, evidencing the behavior forced by rules from Tables 3 and 4. Then, the robot actuates on its orientation, evidencing the rules of Table 1. When the robot is above the target, the landing procedure starts, showing the action of rules from Table 2. In this experiment, the average landing error was 6.0840 cm with 1.83 cm of standard deviation (σ).

The Figures 8 and 9 show the dynamical tracking experiments. As figures show, the UAV tracks and lands at the target points with success. The average error for the linear trajectory was 10.21 cm with σ =2.33 cm. The circular trajectory had 16.61 cm of average error with σ =2.72 cm. The average variation for the maximum altitude was 10.02 meters with 0.045 cm of standard deviation.

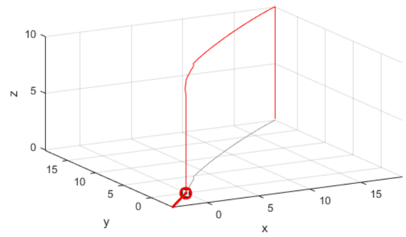


Figure 7: Simulation for a static target.

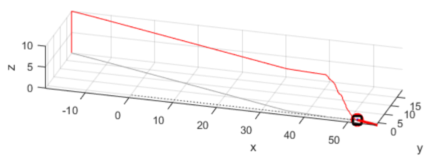


Figure 8: Simulation for linear trajectory.

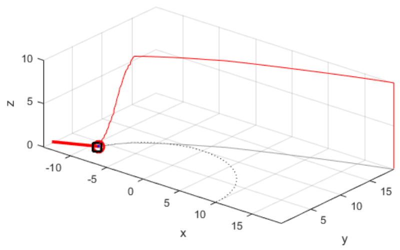


Figure 9: Simulation for circular trajectory.

6 Conclusions

This paper presented the integration of various systems to control the flight and landing of a UAV IRIS+ with a fuzzy logic controller. For simulation we used jMAVSim with hardware in the loop, associated with an IRIS+ embebbed with Odroid U3 for command (*MAVROS*) and camera for image processing with *ar_track_alvar*.

The fuzzy controller proposed had success in all experiments, landing the robot inside the target area. All errors were smaller than the landing point with good statistical deviations, showing accuracy and reliability from the controller. For future works we will implement more features for the controller to reduce the error and improve the performance.

Acknowledgment

The authors would like to thank the Duke Energy Electricity Company, FAPEMIG, CNPq and FADEPE/UFJF.

References

3DRobotics (2015). Iris+ quadrotor, http://store.3drobotics.com/products/iris.

Amazon.com (2015). Amazon prime air, http://www.amazon.com/b?node=8037720011.

Benavidez, P., Lambert, J., Jaimes, A. and Jamshidi, M. (2014). Landing of an ardrone 2.0 quadcopter on a mobile base using fuzzy logic, *World Automation Congress*, pp. 803–812.

HardKernel (2015). Odroid, http://www.hardkernel.com/.

Nagarjuna, K. and Suresh, G. (2015). Design of effective landing mechanism for fully autonomous unmanned aerial vehicle, *International Conference on Signal Processing, Communication and Networking (lCSCN)*, pp. 1–6.

Pedrycz, W. and Gomide, F. (2007). *Fuzzy Systems Engineering: Toward Human-Centric Computing*, Wiley.

Project, P. A. (2015). jmavsim, https://pixhawk.org/dev/hil/jmavsim.

Rada-Vilela, J. (2014). fuzzylite: a fuzzy logic control library, http://www.fuzzylite.com.

Raza, S. A. and Gueaieb, W. (2010). *Intelligent Flight Control of an Autonomous Quadrotor*, InTech.

Saska, M., Krajnik, T. and Pfeucil, L. (2012). Cooperative microuav-ugv autonomous indoor surveillance, *International Multi-Conference on Systems, Signals and Devices*, pp. 1–6.

Sureshkumar, V. and Cohen, K. (2014). Intelligent fuzzy flight control of an autonomous quadrotor uav, *AIAA SciTech*, Vol. 52, pp. 1–10.

Teixeira, J., Ferreira, R., Santos, M. and Teichrieb, V. (2014). Teleoperation using google glass and ar, drone for structural inspection, *XVI Symposium on Virtual and Augmented Reality*, pp. 28–36.

Wenzel, K. E., Masselli, A. and Zell, A. (2011). Automatic take off, tracking and landing of a miniature uav on a moving carrier vehicle, *Journal of intelligent and robotic systems*, Vol. 61, pp. 221–238.