Stability and Altitude Control of a Quadrotor Using Fuzzy Logic

R. L. S. M. Vilela [1,2] *, E. C. Silva [1,2] †
[1] Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro, PUC-Rio, Rio de Janeiro, Brazil
[2] AeroRio, Pontifical Catholic University of Rio de Janeiro, PUC-Rio, Rio de Janeiro, Brazil

ABSTRACT

The use of Unmanned Aerial Vehicles (UAVs) is becoming more common in many fields. The application spectrum varies from civil to military field, comprising environmental monitoring, border patrol, search and rescue operations, disaster relief, among others. In the next years, UAVs market is expected to provide an incoming of billions of dollars since it is rapidly growing in a lot of civilian and commercial industries such as agriculture, energy, utilities, mining, construction, real estate, news media and film production. Many of these applications require small and agile UAVs, capable to fly at low altitudes with a certain degree of maneuverability, controllability and stability, which requires well-tuned controllers. The most widely used controller for these applications is the PID (Proportional, Integral Derivative) controller. However, the tuning of this kind of controller can be very challenging. The objective of this paper is to present the development of a controller based on fuzzy logic to control the attitude angles and the altitude of a quadrotor UAV and compare its performance with a traditional PID control. The achieved results are shown and carefully discussed throughout the paper.

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have become an option to perform activities that would require a lot more effort when performed in traditional ways. The use of rotating wing UAVs have provided several benefits since they do not require a take-off and landing runway, with a VTOL (Vertical Take-off and Landing) vehicle system, and provide hovering capabilities. Also, the ease of control represents an advantage when compared to fixed-wing aircrafts. Furthermore, when it comes to financial value, the use of drones usually provide a viable alternative when compared to the traditional methods that are employed in the market[1].

Fixed-wings UAVs come in different geometries and sizes, that can be selected according to application and payload requirements. This paper discusses the modelling and control techniques for a quadrotor UAV in X configuration.

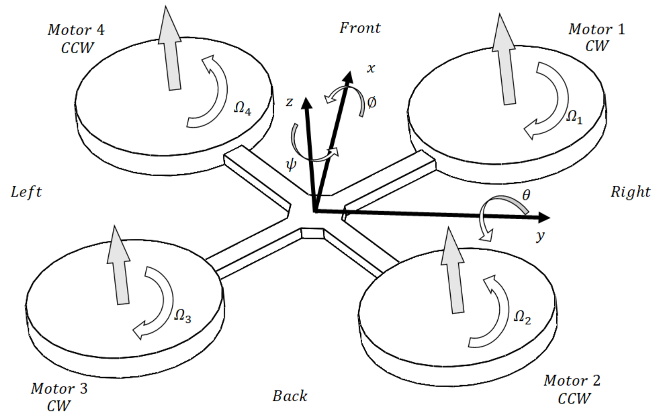

Figure 1: Quadcopter Representation

The principle of operation of multicopter vehicles consists in having their propellers generating enough lift to keep the vehicle in the air. To control the vehicle altitude, the amount of lift is increased or decreased, according to the desired movement. The position control in the xy plane is coupled with the roll and pitch angles, which means that a change in position is performed by changing the vehicles attitude. Finally, it is possible to control the heading of the vehicle with its yaw angle, that makes the drone rotate around its z-axis. Therefore, the vehicle dynamics can be described by changes in four main movements: altitude and roll, pitch and yaw angles.

The basis of the quadrotor dynamics consists in keeping the propellers spinning, aiming to generate enough lift to keep it in the air. Figure 1 shows a representation of a quadrotor configuration and will be used for a detailed explanation, throughout the paper. The pair of motors 1 and 3 spin on clockwise direction while motors 2 and 4 spin on counterclockwise direction. All four propellers generate thrust in the same direction, which requires inverted pitch in the blades attached to motors 1 and 3 in relation to the pitch of the blades connected to motors 2 and 4. The difference in the spinning direction is to counter balance the torque generated by the rotating propellers[2].

To increase/decrease altitude, all four propellers in-

*Email address: renandelima95@gmail.com
†Email address: edusilva@ele.puc-rio.br

crease/decrease their rotation speed to increase/reduce the generated thrust, resulting a change in altitude. To change the roll angle, the pairs of motors 1 and 2 increase/decrease the rotation in comparison to the pair 3 and 4, generating a disbalanced thrust, and therefore changing the roll angle. To change the pitch angle, the process is analogous to the roll angle, however the pair of motors that changes the rotation speed are 1 and 4 or 2 and 3. On the other hand, to change the yaw angle, the pair of motors 1 and 3 changes its rotation in comparison to motors 2 and 4, increasing or decreasing the resultant torque applied to the vehicle.

2 QUADROTOR MODELLING

The quadrotor dynamic movements described in Section 3 are modelled based on Newton-Euler equations for 3D motion of rigid bodies and described in [3] and shown in Equation 1, where m is the drone‘s mass, I is a diagonal matrix with the drones inertia parameters around the x, y and z axis (I_{xx}, I_{yy}, I_{zz}), $\mathbf{v_B}$ is the vector with the drones velocity in the body-fixed reference system, $\omega_\mathbf{B}$ is the vector with the drones angular velocity in the body-fixed reference system and $\mathbf{f_B}$ and $\mathbf{m_B}$ are the vectors with the external forces and moments, respectively, applied to the drone.

$$\begin{cases} m(\dot{\mathbf{v_B}} + \omega_\mathbf{B} \wedge \mathbf{v_B}) = \mathbf{f_B} \\ I\dot{\omega_\mathbf{B}} + \omega_\mathbf{B} \wedge (I\omega_\mathbf{B}) = \mathbf{m_B} \end{cases} \tag{1}$$

The forces and moments generated by the rotation of the motors are described as in Equation 2 where l is the distance between the propeller and the vehicle CoG (center of gravity), b is the thrust coefficient of the motor-propeller setup, d is the aerodynamic drag coefficient, the Ω_n represent the rotation speed of the n-th motor, f_{Bx}, f_{By} and f_{Bz} make up the $\mathbf{f_B}$ vector and m_{Bx}, m_{By} and m_{Bz} make up the $\mathbf{m_B}$ vector, both from Equation 1.

$$\begin{cases} f_{Bx} = -mg\sin\theta \\ f_{By} = mg\cos\theta\sin\phi \\ f_{Bz} = mg\cos\theta\cos\phi + b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ m_{Bx} = \frac{\sqrt{2}}{2}lb(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \\ m_{By} = \frac{\sqrt{2}}{2}lb(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ m_{Bz} = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{cases} \tag{2}$$

The system kinematics is modelled using a ZYX Euler Rotation from the body fixed reference system to the inertial reference system.

Given both the dynamics and kinematics, the system equations can be arranged to form the state-space vector $x = [\dot{x} \quad \dot{y} \quad \dot{z} \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\psi} \quad]^T \in \mathbb{R}^6$. Taking into consideration the simplification for small angles of movement, $[\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T = [p \quad q \quad r]^T$ [4], where p, q and r make up the vector $\omega_\mathbf{B}$, the state-space equations are described in Equation 3.

$$\begin{cases} \ddot{x} = -(\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)\dfrac{f_{Bz}}{m} \\ \ddot{y} = -(-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi)\dfrac{f_{Bz}}{m} \\ \ddot{z} = (g - \cos\theta\cos\phi)\dfrac{f_{Bz}}{m} \\ \ddot{\phi} = \dfrac{I_{yy} - I_{zz}}{I_{xx}}\dot{\theta}\dot{\psi} + \dfrac{m_{Bx}}{I_{xx}} \\ \ddot{\theta} = \dfrac{I_{zz} - I_{xx}}{I_{yy}}\dot{\phi}\dot{\psi} + \dfrac{m_{By}}{I_{yy}} \\ \ddot{\psi} = \dfrac{I_{xx} - I_{yy}}{I_{zz}}\dot{\phi}\dot{\theta} + \dfrac{m_{Bz}}{I_{zz}} \end{cases} \tag{3}$$

3 TYPICAL CONTROL STATEGY

The quadrotors stability is achieved by using a closed loop control system, based on feedback from inertial measurements. Typically, UAVs have an embedded 9DOF IMU (9 Degrees of Freedom Inertial Measurement System), with three-axis accelerometers, three-axis gyroscopes and three-axis magnetometers that provide the attitude angles used to control its stability.

Since attitude estimation requires numerical integration based on measured angular velocities, the systems are subjected to cumulative errors due to noisy measurements. Therefore, control systems include fusion filters that merges data from multiple sensors, such as linear complimentary filters and Extended Kalman Filters [2].

Nested PID controllers are the most widely used strategy for stability control of rotating wing UAVs. They actuate on the desired moments of the vehicle, based on feedback from the sensors and attitude commands. The attitude commands can be given directly by the pilot or be provided by an autonomous trajectory control algorithm, that relies on other ways to measure the vehicle position (i.e GPS) and provide the specific attitude commands for the desired trajectory.

Since this paper is not focused on trajectory control, but on stability and altitude control, it will be based on the case of a manual flight mode,as highlighted in red in Figure 2.

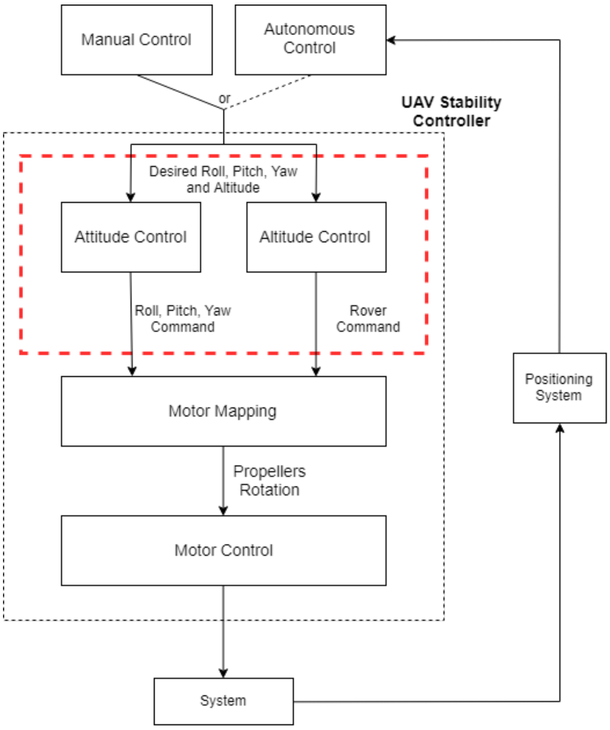

Figure 2: Overall Control Block Diagram

The altitude control, also known as altitude hold, is based on a PID controller actuating in the altitude error, that provides the base command for the motors in order to keep the quadrotor in hovering mode. Figure 3 shows the block diagram for altitude control.

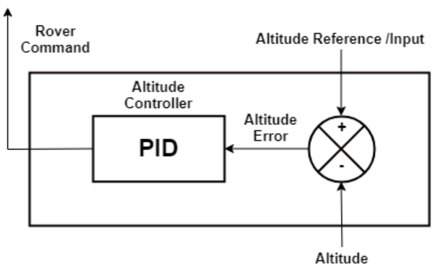

Figure 3: PID for Altitude Control

The stability control is based on a nested PID controller that actuates on the error of the attitude angles to provide the reference angular velocities and then actuates on the error of the angular velocities to provide a normalized actuation command for the motors. Figure 4 shows the block diagram for yaw angle control. Similar block diagrams are used for roll and pitch angles, varying only the inputs and controller gains.

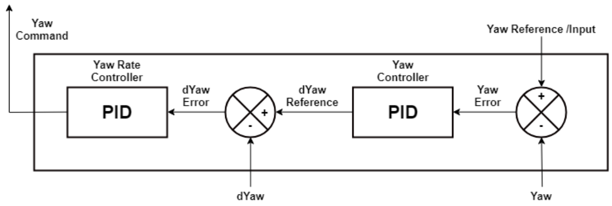

Figure 4: Nested PID for Yaw Control

4 FUZZY CONTROL

A fuzzy control system consists of 4 basic elements, shown in Figure 5. First, the fuzzification module, that is responsible for converting specific input values to fuzzy sets. The knowledge base is composed by the rules that define the control strategy for the system, which are usually extracted from a specialist. The inference system process the fuzzified inputs according to the rules from the knowledge base to infer the actions of the fuzzy controller. Finally, the defuzzification module converts the fuzzy sets, generated by the inference system, back to exact values that are used in the control process [5].

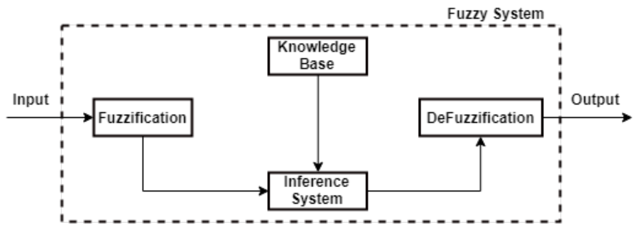

Figure 5: Fuzzy System Diagram

The fuzzy controller for the altitude control was developed based on the altitude error and the derivative of the altitude error. Figure 6 shows the block diagram of the fuzzy altitude controller.

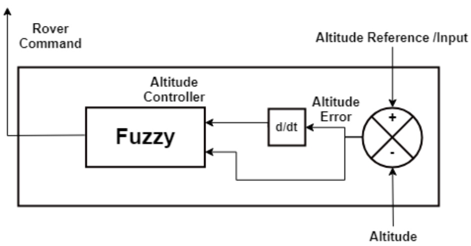

Figure 6: Fuzzy Control for Altitude Control

For the attitude angles, the same method of calculating the derivative of the error was used and only the first PID in the nested approach (Figure 4) was substituted for a fuzzy controller. Figure 7 shows the block diagram for the yaw angle fuzzy-PID controller. The same approach is used for roll and pitch angles as well, changing only the fuzzy controller.

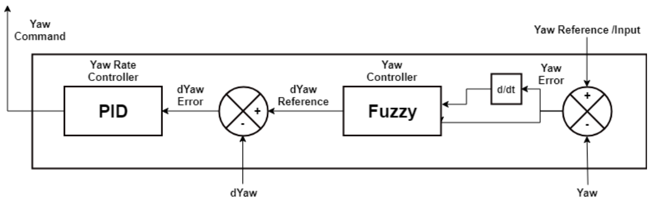

Figure 7: Nested Fuzzy-PID for Yaw Control

5 IMPLEMENTATION

The model was implemented in Simulink/Matlab, using the Simscape toolbox. The parameters used for the drone were extracted from the Crazyflie 2.0 Nano drone, a small quadrotor with open source software that allows the user to personalize its firmware and implement different control approaches. Therefore, developing the fuzzy control for this drone is a first step to implement it in a real setup.

The physical drone parameters were extracted from [6] and the PID gains, used during the initial simulations, were extracted from Crazyflie firmware. The Simscape toolbox creates a 3D simulation environment that allows proper visualization of the vehicles movements.

Fuzzy membership functions and rules were implemented by using the Fuzzy Logic Design toolbox, that provides visual interfaces for easy tuning of the parameters.

Sections 5.1, 5.2 and 5.3 shows the tuned membership functions and rules for the controllers developed. For all controllers, the centroid was used as defuzzification method while minimum was used as implication method.

5.1 Altitude Hold

The error in altitude input was divided in 5 membership functions: Negative Big (NB), Negative (N), Zero (Z), Positive (P) and Positive Big (PB). Its derivative was divided in 3 membership functions: Negative (N), Zero (Z) and Positive(P)

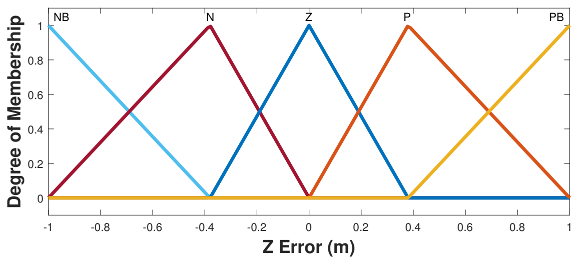

Figure 8: Input Membership Function - Z Error

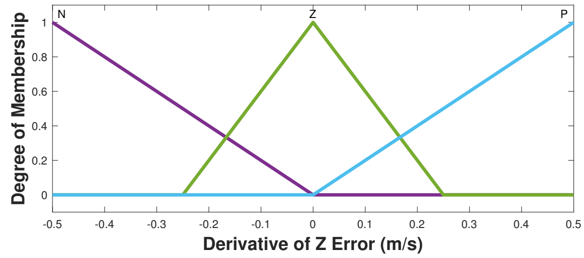

Figure 9: Input Membership Function - Derivative of Z Error

The output for the altitude controller was divided in 5 membership functions: Down Big (DB), Down (D), Maintain (M), Up (U) and Up Big (UB).

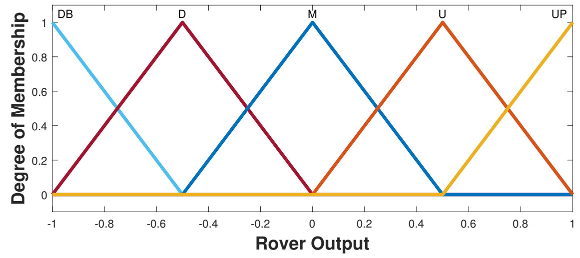

Figure 10: Output Membership Function - Rover

Table 1 show the set of rules used in the knowledge base of the fuzzy inference system for the altitude controller.

Derivative \ Error	NB	N	Z	P	PB
N	DB	D	D	M	U
Z	DB	D	M	U	UB
P	D	M	U	UB	UB

Table 1: Fuzzy Rules for Altitude Control

5.2 Roll/Pitch Angle

Since the model being used is symmetrically in the x and y direction, the roll and pitch movements presents the same dynamic. Therefore, the same controller was capable of controlling both angles.

The error in the roll and pitch angle input was divided in 5 membership functions: Negative Big (NB), Negative (N), Zero (Z), Positive (P) and Positive Big (PB). Its derivative was divided in 3 membership functions: Negative (N), Zero (Z) and Positive(P)

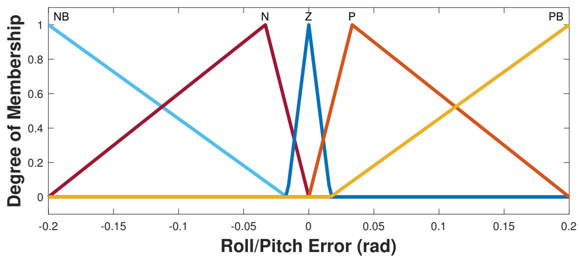

Figure 11: Input Membership Function - Roll/Pitch Error

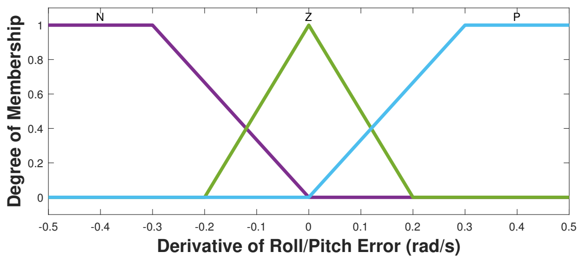

Figure 12: Input Membership Function - Derivative of Roll/Pitch Error

The output for the roll/pitch derivative reference was divided in 5 membership functions: Negative Big (NB), Negative (N), Zero (Z), Positive (P) and Positive Big (PB).

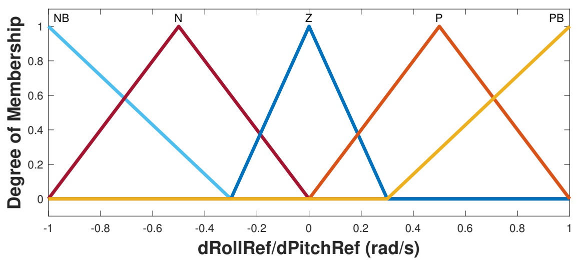

Figure 13: Output Membership Function - Reference for Derivative of Roll/Pitch

Table 2 shows the set of rules used in the knowledge base of the fuzzy inference system for the roll and pitch controller.

Derivative \ Error	NB	N	Z	P	PB
N	NB	NB	N	Z	P
Z	NB	N	Z	P	PB
P	N	Z	P	PB	PB

Table 2: Fuzzy Rules for Roll and Pitch Control

5.3 *Yaw Angle*

The error in the yaw angle input was divided in 5 membership functions: Negative Big (NB), Negative (N), Zero (Z), Positive (P) and Positive Big (PB). Its derivative was divided in 3 membership functions: Negative (N), Zero (Z) and Positive(P)

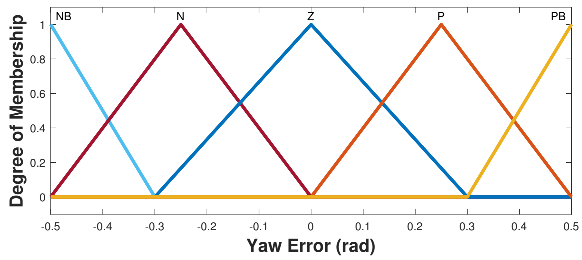

Figure 14: Input Membership Function - Yaw Error

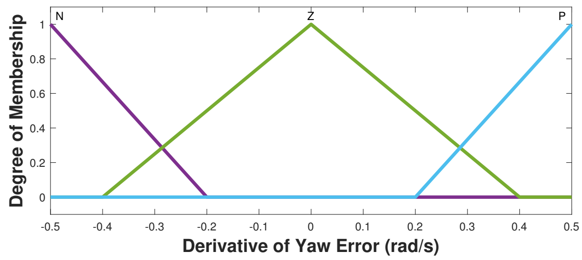

Figure 15: Input Membership Function - Derivative of Yaw Error

The output for the yaw derivative reference was divided in 5 membership functions: Negative Big (NB), Negative (N), Zero (Z), Positive (P) and Positive Big (PB).

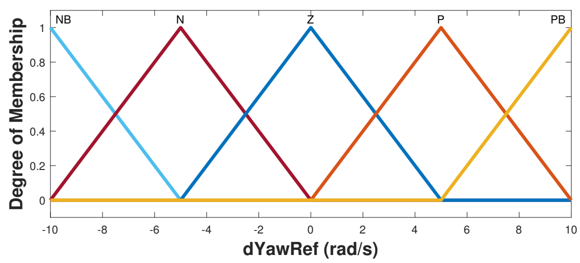

Figure 16: Output Membership Function - Reference for Derivative of Yaw

Table 3 shows the set of rules used in the knowledge base of the fuzzy inference system for the yaw controller.

Derivative \ Error	NB	N	Z	P	PB
N	NB	NB	N	Z	P
Z	NB	N	Z	P	PB
P	N	Z	P	PB	PB

Table 3: Fuzzy Rules for Yaw Control

6 RESULTS

After completing the tuning for the Fuzzy membership functions and rules, simulations were performed by giving reference values for the 4 variables simultaneously. The values given for reference were selected considering reasonable maneuvers for a quadcopter. Figures 17, 18, 19 and 20 shows the results obtained.

To provide a benchmark to evaluate the performance of the designed fuzzy controller, simulations were performed using the traditional PID controller with gains set according to the standard values that come out-of-the-box with the Crazyflie 2.0 Nano.

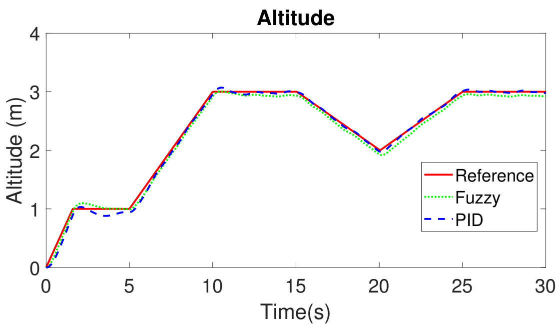

Figure 17: Altitude Control

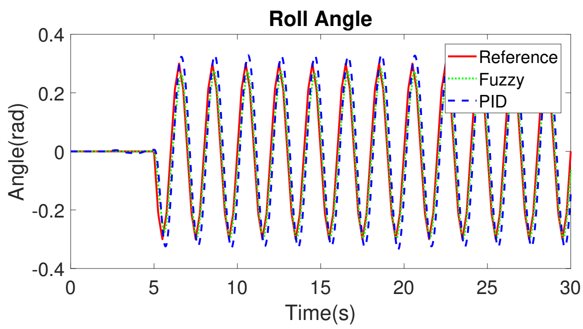

Figure 18: Roll Angle Control

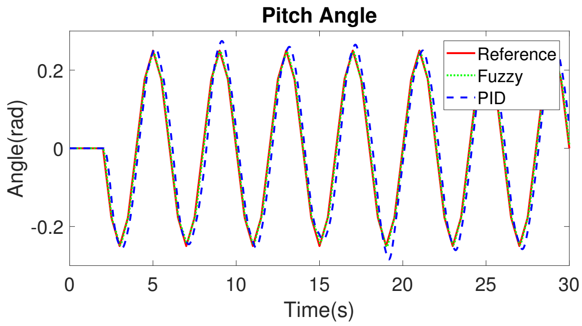

Figure 19: Pitch Angle Control

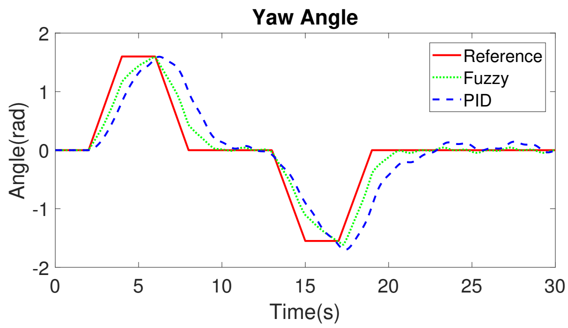

Figure 20: Yaw Angle Control

7 DISCUSSION AND CONCLUSION

The implementation of the fuzzy controller in the altitude and attitude control of a quadrotor UAV was successful, outperforming the standard PID controller, used as benchmark, in several cases.

For the altitude controller, there was a reduction in the over/undershoot performance compared to the PID controller. The roll and pitch controllers provided very good results, regardless of being based on PID or fuzzy control techniques, with different reference derivatives however. The yaw fuzzy controller obtained a significantly better performance compared to the PID performance, reducing the error during the whole control process.

As a future work, still in the stability control, it is possible to substitute the second PID controller for the derivative of the attitude angles for Fuzzy controllers. Also it is possible to expand this work to enable a full Fuzzy autonomous trajectory control. Besides, considering the good results predicted by the performed simulations, a next step would be to implement the fuzzy controller in the real Crazyflie 2.0 Nano quadcopter, aiming to evaluate its experimental behavior and also compare to the results obtained in the simulations.

Aiming to further enhance the simulation, a proper modelling of the sensors can be performed to analyze the influence of the sensor noise in the controllers performance. Also, one thing that needs to be taken into consideration when it comes to hardware performance is the implementation of the fuzzy controller, since it normally requires more calculations than the PID controllers.

ACKNOWLEDGEMENTS

We would like to thank the Brazilian funding agencies CNPq, FAPERJ and CAPES for continued support and supplied resources.

REFERENCES

[1] S. Maharana. Commercial drones. *IJASCET*, 5:96 – 101, 2017.

[2] Q. Quan. *Introduction to multicopter design and control*. Springer, Singapore, 2017.

[3] F. Sabatino. *Quadrotor control: modeling, nonlinear control design, and simulation*. PhD thesis, 2015.

[4] A. Das, K. Subbarao, and F. Lewis. Dynamic inversion with zero-dynamics stabilisation for quadrotor control. *IET Control Theory Applications*, 3(3):303–314, March 2009.

[5] A. C. Gomide, F. Gudwin, and R. Tanscheit. Conceitos fundamentais da teoria de conjuntos fuzzy, logica fuzzy e aplicacoes. 12 2018.

[6] J. Forster. *System Identification of the Crazyflie 2.0 Nano Quadrocopter*. PhD thesis, 2015.