

# Mathematical Model and Intelligent Control of a Quadcopter, with Non-conventional Membership Functions

Eduardo M. Bucio-Gallardo<sup>1</sup>, Ricardo Zavala-Yoé<sup>1</sup> and Ricardo A. Ramírez-Mendoza<sup>2</sup>

1. Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Ciudad de México, Escuela de Graduados en Ingeniería y Arquitectura, Calle del Puente 222, Mexico City 14380, Mexico

2. Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Monterrey, Departamento de Mecatrónica y Automatización, Av. Eugenio Garza Sada 2501 Sur, Monterrey 64849, Mexico

Received: June 29, 2016 / Accepted: July 13, 2016 / Published: October 31, 2016.

**Abstract:** This paper shows detailed steps for modeling a quadcopter with Euler-Lagrange equations, followed by controlling it with intelligent control that includes states decoupling. In addition, this control shows non-conventional membership functions for the most instable states, in order to get a fast and effective response.

**Key words:** Fuzzy control, hybrid control, intelligent control, states decoupling, quadcopter, Euler, Lagrange, Tait-Bryan.

## 1. Introduction

Nowadays, UAVs (unmanned aerial vehicles) are of great importance for the society. These have many applications such as security, monitoring, terrain recognition, shipping and many others. Nevertheless, they are also very useful for research. Being nonlinear systems, with big instabilities and with high states coupling, they become interesting subjects to test new control laws, intelligent algorithms or any other techniques that the researchers want to analyze.

The purpose of this paper is the nonlinear mathematical model of a symmetric quadcopter with respect to the XY-axis, using Euler-Lagrange equations. In addition, an efficient hybrid control will be implemented, in order to control and decouple its states.

## 2. Mathematical Model

### 2.1 Diagrams and Mathematical References

Fig. 1 shows direction of rotation of the motors,

---

**Corresponding author:** Ricardo Zavala-Yoé, Ph.D., research fields: mechatronics, math modelling and control.

their torques ( $\tau$ ), lift forces and the coordinate axes of the non-inertial frame.

Modeling assumptions:

- (1) The quadcopter is assumed as a rigid body.
- (2) Center of mass in the center of the rigid body ( $CM = (0, 0, 0)$ ).
- (3) The quadcopter is assumed as symmetric with respect to the XY-axis.
- (4) Coriolis force is neglected because the quadcopter will move in small areas and its reference frame, the Earth, is assumed as an inertial frame. In addition, most of the time its translation movement is not parallel to the centripetal force of the rotations around its axes  $X(\varphi)$ ,  $Y(\theta)$ ,  $Z(\psi)$ .
- (5) Centrifugal force is not considered because the modeling is based on the inertial frame, hence it does not have any effect.

Given that the quadcopter is symmetrical to the XY-axis, which crosses its center of mass located at  $CM = (0, 0, 0)$ , and the coordinated axes of the rigid body are fixed to it, while they rotate and translate along with it, the moments of inertia end up like this:

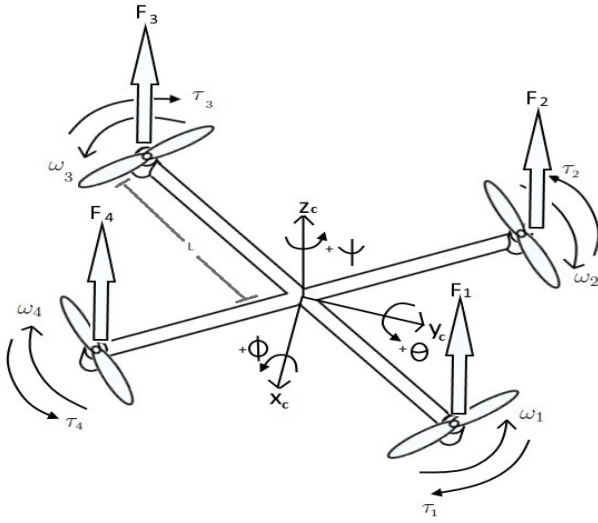


Fig. 1 Non-inertial frame diagram.

$$\begin{aligned} I_x &= \bar{I}_x = I_y = \bar{I}_y = \bar{I}_L \\ I_z &= \bar{I}_z \\ I_{xy} &= I_{xz} = I_{yz} = 0 \end{aligned}$$

The moments of inertia with a bar at the top are the moments of inertia with respect to the center of mass.

## 2.2 Euler Angles and Tait-Bryan Angles

The rotational angle around a coordinated axis is called Euler angle. A sequence of three rotations is called Euler angles sequence. These sequences have two rotations around the same axis, but not in a consecutive way. Therefore, the possible sequences are:  $XYX$ ,  $XZX$ ,  $YXY$ ,  $YZY$ ,  $ZXZ$  and  $ZYZ$ . Where, each rotation is given by Eqs. (1)-(3).

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (1)$$

$$R_y = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2)$$

$$R_z = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

After selecting a rotational matrix, coordinates of a rigid body can be mapped from one reference frame to another. In a reference frame where a counterclockwise rotation is positive and the Z-axis is positive going upwards, the rotations are calculated by a pre-multiplication of a column vector ( $v = [v_x \ v_y \ v_z]^T$ ) by a rotational matrix (e.g.  $R_{ZXZ} = R_z R_x R_z$ ). An example of a rotational equation is shown in Eq. (4).

$$v' = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (4)$$

Nevertheless, Euler rotations are not the best approach for aeronautics because they are more complicated to use with instrumentation. Consequently, Tait-Bryan rotations must be implemented. However, Euler rotations are also a valid solution, as presented in Ref [1]. Tait-Bryan rotation matrices do not repeat rotations in any coordinate axis, therefore the three axes are rotated. The possible rotation sequences are:  $XYZ$ ,  $XZY$ ,  $YXZ$ ,  $YZX$ ,  $ZXY$  and  $ZYX$ . This is also explained in Refs. [2, 3], but in Ref. [3] they are called ‘‘Symmetric Euler Angles Set’’.

The Tait-Bryan rotation used for the model studied in this paper is given by  $YXZ$ . Given that the lift force ( $F_s = k \sum_{n=1}^4 \omega_n^2$ , where  $k$  is the lift constant of the helices and  $\omega_n$  is the angular velocity of each helix) is pointing on the Z-axis direction of the non-inertial frame ( $\bar{F}_s = [0 \ 0 \ F_s]^T$ ), the rotation of this vector to the inertial frame ( $F_s'$ ) only uses the third column of the rotation matrix  $YXZ$ . Hence, the rotation matrix is given by Eq. (5).

$$F_s' = R_{YXZ} \bar{F}_s = k \sum_{n=1}^4 \omega_n^2 \begin{bmatrix} \cos(\phi) \sin(\theta) \\ -\sin(\phi) \\ \cos(\phi) \cos(\theta) \end{bmatrix} \quad (5)$$

The rotation matrix  $YXZ$  was selected because it is the only one that does not need the rotation angle

around Z,  $\psi$ , when rotating the non-inertial Z-axis, which contains  $F_s$ . This type of rotation is needed, as a result of the quadcopter geometry and dynamics. As it can be analyzed from Fig. 1, it does not matter how much the rigid body rotates around the Z-axis, the lift vector will never change direction neither sense. Whereas lift force is parallel to the non-inertial Z-axis,  $\psi$  should not affect at the moment of mapping this force from the non-inertial frame to the inertial frame. However, it is worth mentioning that there are different approaches, like the one presented in Ref. [4], which uses a different rotation matrix, as it can be analyzed from the dynamic model of the quadcopter they present.

### 2.3 Euler-Lagrange Equations

The general Euler-Lagrange equation is shown in Eq. (6).

$$\frac{d}{dt} \left( \frac{\delta T}{\delta \dot{q}_r} \right) - \frac{\delta T}{\delta q_r} = F_{q_r} \quad (6)$$

where,  $T$  is the kinetic energy of the system,  $F_{q_r}$  is the generalized force that generates changes in the kinetic energy and  $q_r$ , which is a generalized coordinate or in other words, one of the non-differentiated states that describe the system.

These are the coordinate axes ( $\xi = [X \ Y \ Z]^T$ ) and these, the attitude angles ( $\eta = [\phi \ \theta \ \psi]^T$ ).

As it can be seen in Ref. [5] on page 148, the general expression for kinetic energy in a free body is given by Eq. (7).

$$\begin{aligned} T = & \frac{1}{2} M v_O^2 + \frac{1}{2} [I_x \omega_x^2 + I_y \omega_y^2 + I_z \omega_z^2 - 2I_{xy} \omega_x \omega_y \\ & - 2I_{xz} \omega_x \omega_z - 2I_{yz} \omega_y \omega_z] + M [v_{Ox} (\omega_y \bar{z} - \omega_z \bar{y}) \\ & + v_{Oy} (\omega_z \bar{x} - \omega_x \bar{z}) + v_{Oz} (\omega_x \bar{y} - \omega_y \bar{x})] \end{aligned} \quad (7)$$

where,

$M$  : Mass of the free body.

$v_O$ : Translational velocity of the non-inertial frame's origin, with respect to the inertial frame's origin.

$v_{Ox}, v_{Oy}, v_{Oz}$  : Translational velocities of the non-inertial frame's origin, with respect to the inertial frame's coordinate axes.

$\omega_x, \omega_y, \omega_z$ : Angular velocities of the non-inertial frame, with respect to the inertial frame's coordinate axes.

$\bar{x}, \bar{y}, \bar{z}$  : Distances of the non-inertial frame's origin in each coordinate axis, to the center of mass of the rigid body.

Considering the modeling assumptions, these are equations are valid:

$$\bar{x} = \bar{y} = \bar{z} = 0$$

$$I_{xy} = I_{xz} = I_{yz} = 0$$

$$v_O = v_{CM}$$

In addition, it is known that  $v_{CM} = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$ ,  $\omega_x = \dot{\phi}$ ,  $\omega_y = \dot{\theta}$ ,  $\omega_z = \dot{\psi}$ . Therefore, if Eq. (7) is reorganized, Eq. (8) is obtained.

$$T = \frac{1}{2} M (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + \frac{1}{2} [\bar{I}_L (\dot{\phi}^2 + \dot{\theta}^2) + \bar{I}_z \dot{\psi}^2] \quad (8)$$

*Equation of motion for X:* If Eq. (8) is included in the Euler-Lagrange equation for X,

$\frac{d}{dt} \left( \frac{dT}{d\dot{x}} \right) - \frac{dT}{dx} = F_x$ , the simplified equation of motion for X is found,  $M\ddot{x} = F_x$ .

To obtain the generalized force  $F_x$ , virtual work must be used ( $dW_x = F_x dx$ ). This includes conservative and non-conservative forces. For X, the conservative forces are given by the propulsion of the helices and the non-conservative ones, by the air friction. Therefore, the virtual work for X is expressed in Eq. (9). However, as the last mention equation shows, the drag coefficient multiplies the trigonometric functions as absolute values. This is necessary because if it is analyzed without the absolute value, when the translational velocity of X is positive with angle of attack  $-\theta$ , the drag favors the movement instead of doing the opposite. This is why the absolute value is included in the equation.

$$dW_x = (k \sum_{n=1}^4 \omega_n^2 \cos \phi \sin \theta - B_x \dot{x} |\cos \phi \sin \theta|) dx \quad (9)$$

where,

$k$  : Lift coefficient of the helices.

$\omega_n$  : Angular velocity of each helix.

$B_x$  : Drag coefficient for  $X$ .

With the virtual work for  $X$ , the complete movement equation for  $X$  is developed as Eq. (10).

$$M\ddot{x} = k \sum_{n=1}^4 \omega_n^2 \sin \theta \cos \phi - B_x \dot{x} |\sin \theta \cos \phi| \quad (10)$$

Note: The movement equations for each state already include the rotation  $R_{YXZ}$ .

*Equation of motion for  $Y$ :* As an analogy of  $X$ , the equation of motion for  $Y$  is obtained in Eq. (11).

$$M\ddot{y} = -(k \sum_{n=1}^4 \omega_n^2 \sin \phi - B_y \dot{y} |\sin \phi|) \quad (11)$$

*Equation of motion for  $Z$ :* Virtual work of  $Z$ ,  $F_z$ , is given by Eq. (12).

$$dW_z = ((k \sum_{n=1}^4 \omega_n^2 - B_z \dot{z}) \cos \phi \cos \theta - Mg) dz \quad (12)$$

where,

$g$ : Acceleration of gravity.

$B_z$ : Drag coefficient for  $Z$ .

Therefore, the complete equation of motion for  $Z$  is represented by Eq. (13). In this equation there are no absolute values due to the cosine functions, which are even functions.

$$M\ddot{z} = (k \sum_{n=1}^4 \omega_n^2 - B_z \dot{z}) \cos \phi \cos \theta - Mg \quad (13)$$

*Equation of motion for  $\theta$ :* The equation of motion for  $\theta$  is given by  $\bar{I}_L \ddot{\theta} = F_\theta$ . Its generalized torque, as well as the previous generalized forces, is obtained from the inertial frame. Fig. 2 shows the quadcopter on the  $XY$  of the inertial frame. With this image, it is possible to analyze with trigonometry how rotations around  $Z$  ( $\psi$ ), affect torques on  $\theta$  and  $\phi$ .

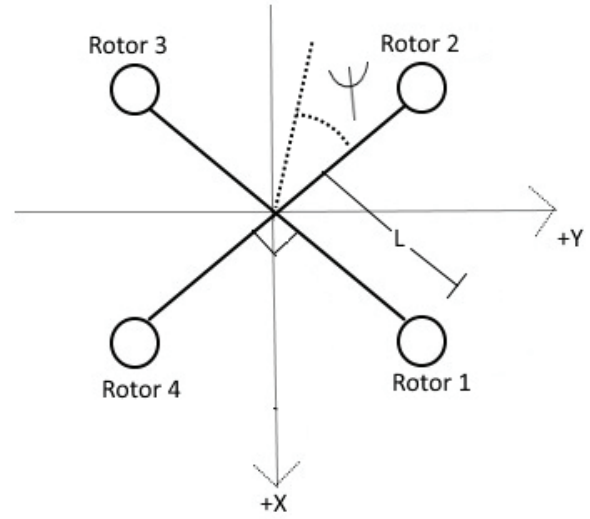


Fig. 2 Quadcopter diagram for torques analysis.

The general equation of motion for  $\theta$  is given by Eq. (14).

$$\bar{I}_L \ddot{\theta} = -B_\theta \dot{\theta} + kl [-(\omega_1^2 - \omega_3^2) \cos(\pi/4 + \psi) + (\omega_2^2 - \omega_4^2) \cos(\pi/4 - \psi)] \quad (14)$$

where,

$l$ : Quadcopter arm length.

$B_\theta$ : Drag coefficient for  $\theta$ .

*Equation of motion for  $\phi$ :* As an analogy of  $\theta$ , the complete equation of motion of  $\phi$  is obtained Eq. (15).

$$\bar{I}_L \ddot{\phi} = -B_\phi \dot{\phi} + kl [(\omega_1^2 - \omega_3^2) \cos(\pi/4 - \psi) + (\omega_2^2 - \omega_4^2) \cos(\pi/4 + \psi)] \quad (15)$$

where,

$B_\phi$ : Drag coefficient for  $\phi$ .

*Equation of motion for  $\psi$ :* As it is mentioned in Ref. [6], the torque that affects  $\psi$ , is given by the reaction generated by the velocities and accelerations of the rotors. However, in this last reference, the acceleration effect is neglected as it is assumed as a small value. Nonetheless, the present model includes the acceleration effect as a result of many computational simulations, which showed that it represents approximately 10% of the total torque around the  $Z$ -axis. Considering this, the complete equation of motion for  $\psi$  is developed as Eq. (16).

$$\ddot{\psi} = -B_{\psi}\dot{\psi} + \frac{b}{I_Z}[-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2] + \frac{I_M}{I_Z}[-\dot{\omega}_1 + \dot{\omega}_2 - \dot{\omega}_3 + \dot{\omega}_4] \quad (16)$$

where,

$B_{\psi}$ : Drag coefficient for  $\psi$ .

$b$ : Torque drag coefficient.

$I_M$ : Moment of inertia of the rotors.

$\dot{\omega}_i$ : Angular acceleration of rotor “ $i$ ”.

Given this, the system inputs are not just the angular velocities of the rotors, but also their angular acceleration. Hence, the inputs are  $U = [\omega_1^2 \ \omega_2^2 \ \omega_3^2 \ \omega_4^2 \ \dot{\omega}_1 \ \dot{\omega}_2 \ \dot{\omega}_3 \ \dot{\omega}_4]^T$ .

#### 2.4 Hybrid Controller Fuzzy-Crisp

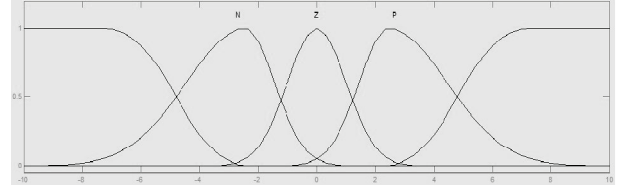
Sampling period is 10 ms.

**Controller for Z:** A PD fuzzy controller + I crisp controller is implemented. The proportional error constant is  $K_{P_z} = 5$ , the derivative constant is unitary and the integral constant is  $K_{I_z} = 0.0035$ . The limits of the integrator are  $\in [-0.2, 0.2]$  and it includes anti windup.

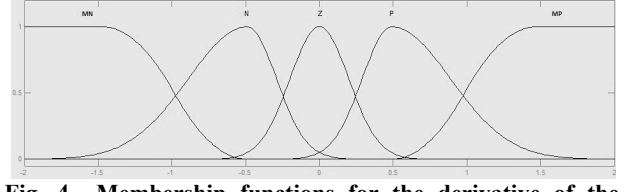
The input membership functions are shown in Figs. 3 and 4.

The output membership functions (Fig. 5) are normalized and its crisp output must be multiplied by the maximum virtual  $\frac{rev}{s}$ . These are obtained from the states decoupling by algebraic restrictions.

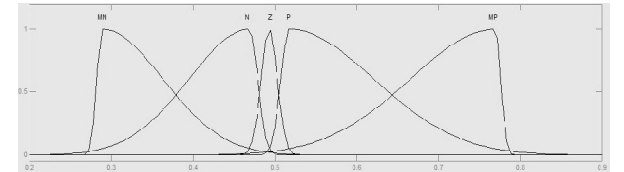
The acceleration function of Z is  $\ddot{Z} = \frac{k}{m} \sum_{n=1}^4 \omega_n^2 \cos \theta \cos \phi - g$ . If during the quadcopter translation, the system must not lose altitude and therefore, decouple states,  $\ddot{Z} = 0$  is needed. If the equation of motion of Z is cleared when equal to zero, the rotors' angular velocities that generate a lift equal to the weight are obtained (Eq. (17)). However, as the fuzzy output is normalized and the center of the membership function “Z” is localized at “CenterZ”, the maximum virtual  $\frac{rev}{s}$  is acquired



**Fig. 3 Membership functions for the error of Z.**



**Fig. 4 Membership functions for the derivative of the error of Z.**



**Fig. 5 Membership functions for the output of Z.**

by Eq. (18). It is worth mentioning that each system has its own maximum allowed  $\frac{rev}{s}$ ; In this case, it is  $155 \frac{rev}{s}$ . In addition, for this model, Center Z = 0.493 because it generates  $76.42 \frac{rev}{s}$ , which gives a lift that equals the weight when  $\phi$  and  $\theta$  are 0.

It should be noted that all the controllers use bisector method for defuzzification (explained in Ref. [7]).

$$\omega_Z = \sqrt{\frac{mg}{4k \cos \theta \cos \phi}} \quad (17)$$

$$\omega_{MaxVirtual} = \frac{\omega_Z}{CenterZ} \quad (18)$$

**Controllers for X and Y:** PD fuzzy controllers + I crisp controllers are implemented. The proportional error constants are  $K_{P_x} = K_{P_y} = 0.7$ , the derivative constants are unitary and the integral constants are  $K_{I_x} = K_{I_y} = 0.05$ . The limits of the integrators are  $\in [-0.08, 0.08]$  and they include anti windup.

The input membership functions have similar geometries to the ones in the Z controller.

On the other hand, the output membership functions in Fig. 6 are normalized and they represent an angle of attack to which the quadcopter should incline. The crisp output is then multiplied by 0.2269 rad (13°). The X controller gives the angle  $\theta$ , while the Y controller outputs the angle  $\phi$ . These two outputs then become references of the controllers  $\theta$  and  $\phi$ , respectively. In other words, they are cascade controllers.

*Controllers for  $\phi$  and  $\theta$ :* PD fuzzy controllers are implemented. The proportional error constants are unitary and the derivative constants are  $K_{d_\theta} = K_{d_\phi} = 0.8$ . There are no integrators because these states are really unstable and they change really fast. An integrator would only cause bigger instabilities. The input membership functions are in Figs. 7 and 8, while the output ones are in Fig. 9.

As it can be observed, the membership functions of these controllers do not have conventional forms. The input membership functions were constructed in this way, in order to generate a big fuzzy error, when the crisp error is not that big. This is why the membership functions N and P, are so close to the membership function Z, as well as why the Z function is so small in comparison to the other membership functions. Nevertheless, the most characteristic membership functions are the output ones. As Fig. 9 shows, the Z function is really small and it is far away from the other functions. This generates a much more efficient controller response. When the fuzzy rules determine a small output for N or P, the actual crisp output is big because the defuzzification is calculated by bisector method. When there are highly instable states, this kind of output shows a really good response in order to maintain those states within range. A small error generates a big push towards the zero error state.

The crisp output of these controllers go through an algorithm of states decoupling by algebraic restrictions. The  $\phi$  controller outputs an angular velocity “a”, which is added to the rotors of positive torque for  $\phi$  (1 and 2). Likewise, “a” goes into Eq.

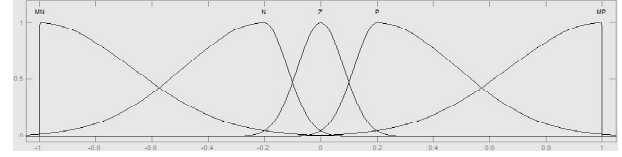


Fig. 6 Output membership functions for X and Y.

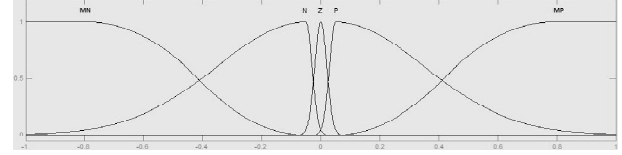


Fig. 7 Membership functions for the error of  $\phi$  and  $\theta$ .

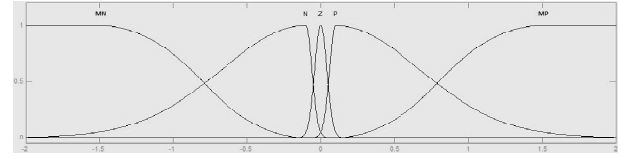


Fig. 8 Membership functions for the derivative of the error of  $\phi$  and  $\theta$ .

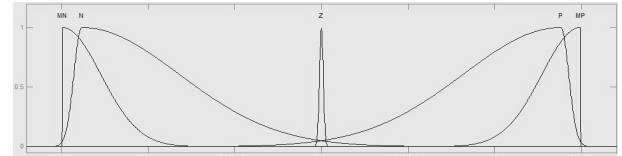


Fig. 9 Membership functions for the output of  $\phi$  and  $\theta$ .

(19), which gives the angular velocity to add to the rotors of negative torque for  $\phi$ .

$$b_{1,2} = -\frac{\omega_3 + \omega_4}{2} \pm \sqrt{\left(\frac{\omega_3 + \omega_4}{2}\right)^2 - a(a + \omega_1 + \omega_2)} \quad (19)$$

Considering that changing  $\psi$  changes which rotors generate positive and negative torque to  $\phi$  (do not forget that the modeling and control are referenced to the inertial frame, which explains why  $\psi$  determines which rotors generate positive and negative torque. See Fig. 2.), these equations must be used: Eqs. (20) and (21), where “a<sub>1</sub>”, “a<sub>2</sub>”, “b<sub>3</sub>” and “b<sub>4</sub>”, are the angular velocity variation to be added to the rotors 1, 2, 3 and 4, respectively.

$$a_1, b_3 = \begin{cases} a_1 = a, b_3 = b, & \psi \in \left[-\frac{\pi}{4} \pm 2\pi k, \frac{3\pi}{4} \pm 2\pi k\right] \\ a_1 = -a, b_3 = -b, & \psi \in \left(\frac{3\pi}{4} \pm 2\pi k, -\frac{7\pi}{4} \pm 2\pi k\right) \end{cases} \quad k = 1, 2, \dots, n \quad (20)$$



$$a_2, b_4 = \begin{cases} a_2 = a, b_4 = b, & \psi \in \left[ -\frac{3\pi}{4} \pm 2\pi k, \frac{\pi}{4} \pm 2\pi k \right] \\ a_2 = -a, b_4 = -b, & \psi \in \left( \frac{\pi}{4} \pm 2\pi k, \frac{5\pi}{4} \pm 2\pi k \right) \end{cases}$$

$$k = 1, 2, \dots, n \quad (21)$$

The same procedure is applied to  $\theta$ .

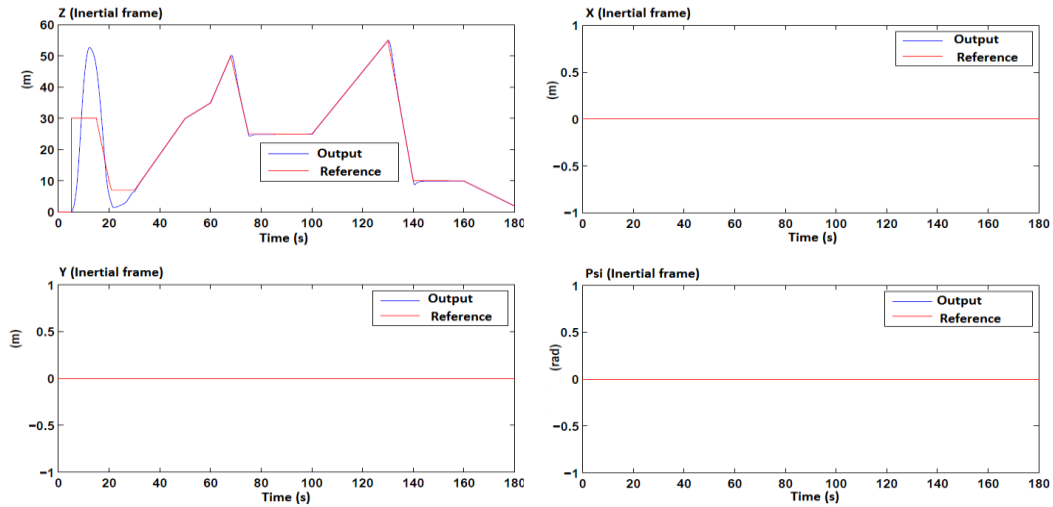
*Controller for  $\psi$ :* P fuzzy controller + I crisp controller is implemented. The proportional error constant is unitary and the integral constant is  $K_{I_\psi} = 0.06$ . The limits of the integrator are  $\in [-0.005, 0.005]$  and it includes anti windup.

The input and output membership functions of this controller have similar geometries than  $\phi$  and  $\theta$  controllers. The main difference is that there is no

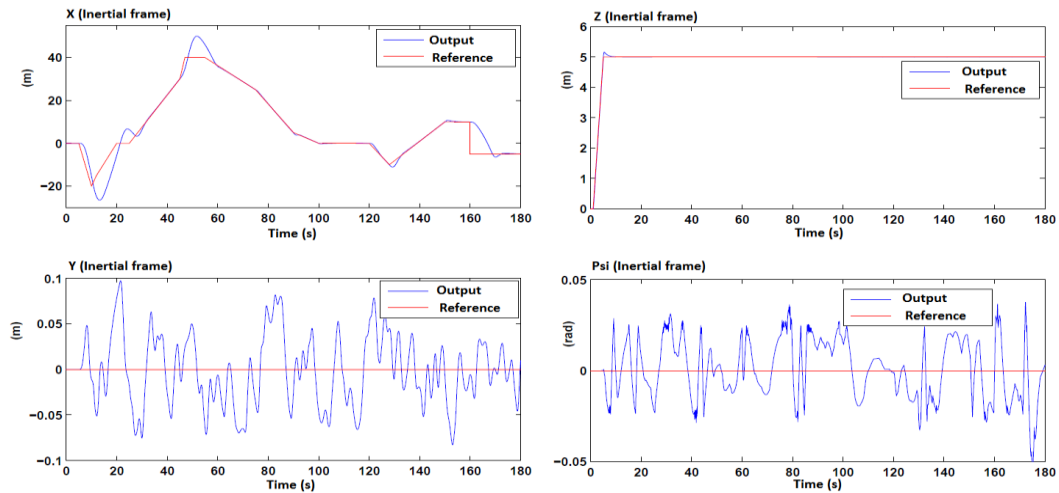
derivative effect. Finally, as well as for the other angles' controllers, the crisp output of this controller goes through an algorithm of states decoupling through algebraic restrictions.

### 3. Results

In order to analyze states decoupling, simulation was run with the values proposed by Ref. [6]. First, tests were done with variable Z reference with abrupt changes, while the other states references (X, Y and  $\psi$ ) stayed at zero, Fig. 10. Then, the next test was run with variable X reference, after elevating the quadcopter 5 m from the ground, Fig. 11. The same was done for Y, Fig. 12. Finally, the last test was for variable  $\psi$  reference, Fig. 13.



**Fig. 10** Response with variable Z (original in colors).



**Fig. 11** Response with variable X (original in colors).

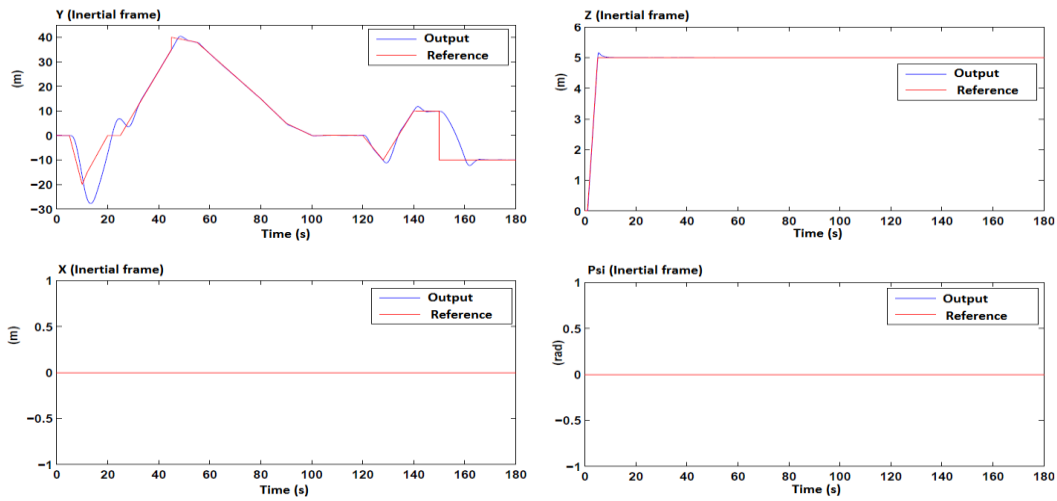


Fig. 12 Response with variable Y (original in colors).

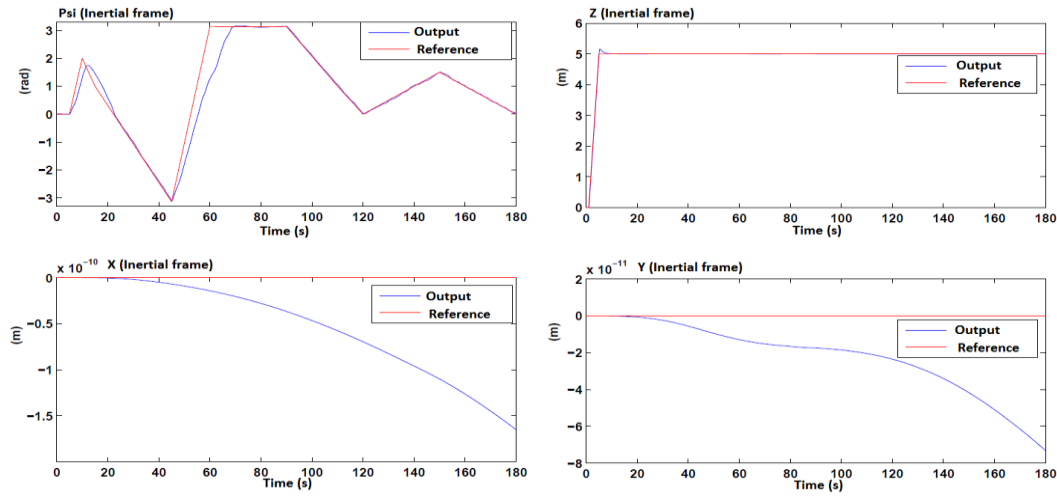


Fig. 13 Response with variable  $\psi$  (original in colors).

$\phi$  and  $\theta$  graphs are not of great interest, because they are not actually relevant for a user. What an actual user might actually look for, are the three Cartesian coordinates and then, its orientation  $\psi$ . As it was already explained,  $\phi$  and  $\theta$  are only used to reach X and Y references, as cascade controllers.

For a good analysis, not only states' outputs are of relevance, there is a need to analyze the output that is demanded to the rotors. This is the one that shows if these controllers are viable for the physical world or not. In addition, it helps analyzing the efficiency of the controller. During a simulation that varies every state of the quadcopter, with trajectories in each state that includes steps, ramps and steady states, the rotors output was obtained (Fig. 14). This is something that

normally is not analyzed during simulation, and most papers do not talk about it. Even when a simulation appears to have a great response on the system's states, like what is shown in Ref. [8], which looks like a promising control, there is no way of knowing what is happening on the control law output. One of the biggest complications of fuzzy control, is that it tends to present big chattering when error approaches zero. That is why the motors output is shown here, so it can be properly studied.

### 3.1 Results Analysis

As it can be seen in Fig. 10, while Z changes, the other states remain constant. Even during the first 20s, where the overshoots of Z are present, the other states



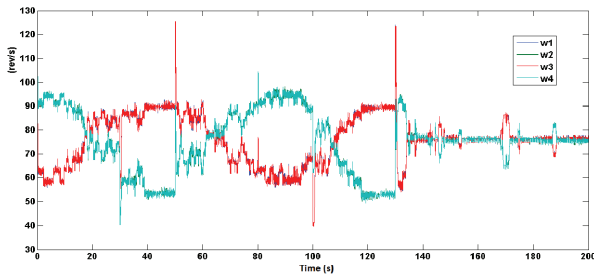


Fig. 14 Rotors output (original in colors).

do not change. It is worth mentioning that this state is the most stable one.

Analyzing Fig. 11, when  $X$  has abrupt changes,  $Z$  does not change.  $Y$  has a small chattering that goes from  $-0.1$  m to  $0.1$  m. Likewise, there is a small vibration on  $\psi$ , which goes from  $-0.05$  rad to  $0.05$  rad. These variations on  $Y$  and  $\psi$  are small and continue showing good states decoupling.

On the other hand, Fig. 12 shows that abrupt changes on  $Y$ , do not alter the other states.

Looking at Fig. 13, it is able to determine that abrupt changes on  $\psi$  do not generate important movements on the other states.  $Z$  shows almost nonexistent movements, while  $X$  and  $Y$  have changes in the order of  $10^{-10}$  and  $10^{-11}$ , respectively. These last variations might be caused by Matlab<sup>®</sup> approximation errors or actual movements, but insignificant ones.

Finally, analyzing the rotors output, Fig. 14, it can be seen that most of the time there are no big spikes. Rotors chattering is approximately  $2-5 \frac{rev}{s}$ , during

one sampling period. This is easily obtained by a brushless DC motor. The only important spikes, like the one at 50 s, are generated by abrupt changes on the quadcopter. If the physical rotors do not reach these spikes, which are up to  $35 \frac{rev}{s}$  during one sampling period, the only thing that would happen is a bit slower response during those abrupt changes.

#### 4. Conclusions

Euler-Lagrange equations are a really good method

to model numerous physical systems. In addition, it is important to choose the correct rotation matrices to implement, being Tait-Bryan rotations the most practical ones, at least for aeronautic purposes.

Similarly, it is concluded that the implemented hybrid controller gave good results. References are effectively reached, while keeping a high level of states decoupling. Nonetheless, it should be noted that  $\phi$  and  $\theta$  are still completely coupled to  $X$  and  $Y$ , respectively. This is not something that can be changed, due to the quadcopter geometry that impedes this.

Finally, analyzing the rotors output, it can be deducted that this controller is suitable for a physical system.

#### References

- [1] Nathan, P. T., and Almurib, H. A. F. 2011. "Optimization of Nonlinearities through Control Techniques of the Quadrotor Aerial Vehicle." In *Proceedings of the 4th IEEE International Conference on Modeling, Simulation and Applied Optimization*. doi: 10.1109/ICMSAO.2011.5775511.
- [2] Esparza, et al. 2012. "Modelado, Diseño y Control Borroso de un Quadrotor." Eng. thesis, Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación, Spain.
- [3] Schaub, H., and Junkins, J. 2002. *Analytical Mechanics of Aerospace Systems*. Reston, Va: AIAA.
- [4] Patel, A. R., Patel, M. A., and Vyas, D. R. 2012. "Modeling and Analysis of Quadrotor Using Sliding Mode Control." In *44th IEEE Southeastern Symposium on System Theory*, 111-4.
- [5] Wells, D. A. 1967. *Schaum's Outline of Theory and Problems of Lagrangian Dynamics*. New York: McGraw-Hill.
- [6] Luukkonen, T. 2011. "Modelling and Control of Quadcopter." Aalto University. [http://sal.aalto.fi/publications/pdf-files/eluu11\\_public.pdf](http://sal.aalto.fi/publications/pdf-files/eluu11_public.pdf).
- [7] MathWorks. Defuzzification Methods. Accessed April 06, 2015. <http://www.mathworks.com/help/fuzzy/examples/defuzzification-methods.html>.
- [8] Santos, M., Santos, M., and Morata, F. 2010. "Intelligent Fuzzy Controller of a Quadrotor." In *Proceedings of the IEEE International Conference on Intelligent Systems and Knowledge Engineering*, 141-6.