

## TRAJECTORY PLANNING FOR AN UNMANNED QUADROTOR

Martín A. Pucheta<sup>a,b</sup>, Nicolás Alberto<sup>a</sup>, Claudio J. Paz<sup>a</sup> and Gonzalo Perez-Paina<sup>a</sup>

<sup>a</sup>*Centro de Investigación en Informática para la Ingeniería (CIII), Facultad Regional Córdoba de la Universidad Tecnológica Nacional, Maestro López esq. Cruz Roja Argentina, X5016ZAA Córdoba, Argentina, {mpucheta,cpaz}@frc.utn.edu.ar, <http://cii.frc.utn.edu.ar>*

<sup>b</sup>*Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)*

**Keywords:** Quadrotors, Newton-Euler dynamics, Euler angles, Quaternions, Nonlinear Control, Trajectory generation.

**Abstract.** The quadrotor is an unmanned aerial vehicle (UAV) popular for its low-cost and broad range of application fields, inspection in harvesting environments, surveillance, cinematographic filmation, among others. Most of these applications are easily achieved if a set of predefined programmed trajectories or tasks are selected from a database of automatic tasks safety performed inside the limit cycle and compatible with actuators bounds. This paper presents two controllers that simultaneously coordinates the position and orientation motions of a quadrotor. The first controller is based on Euler angles and the second on quaternions. The equations of motion for the quadrotor are computed and simulated using Euler angles by means of a first-order reduction of the nonlinear Newton-Euler equations. This simulator of the plant is used to supply the observed state to simulate both controllers. The generation of trajectories with bounded derivatives on the velocities is used to provide continuous actions on the actuators and also prevent their saturation, enabling the minimization of trajectory tracking errors and reducing or avoiding the overshooting. Finally, the simulation for three kinematic tasks, a circle, a 8-shape, and a square trajectory are shown to test the controller behaviors and adjust their gains.

## 1 INTRODUCTION

The quadrotor is an unmanned aerial vehicle (UAV) popular for its low-cost and broad range of application fields, inspection in harvesting environments, surveillance, cinematographic filmation, among others (Gupte et al., 2012; Lim et al., 2012; Mahony et al., 2012). One of the most relevant concerns about UAVs is the human security under their operation. Human security can be assured by secure human-machine interaction (operation) and UAVs with controllers robust to internal (noise and vibrations) and external (wind) perturbances.

Some of the mentioned applications can be more safety achieved if some predefined programmed trajectories or tasks are selected from a database. This set of automatic tasks are safety performed inside the limit cycle of the dynamic system with control actions compatible with actuators bounds. Using an automatic task the human visual monitoring can be minimized and the human can be located in a safe area. For instance, the application can be to inspect electric towers using a vertical movement up to the inspection plane followed by a planar circle pointing towards the tower axis. A similar task can be thought to inspect wind turbine towers.

The motion equations of aerial vehicles, their flight dynamics and details of their control systems are well-known for airplanes and spacecraft systems (Stevens and Lewis, 1992; Sidi, 1997). However, the literature on developments done on multi-rotor control belongs to the last decade (Corke, 2011). The dynamics simulation of the quadrotor is simple and was addressed starting from several formalisms. Here, the vectorial representation of position is combined with two attitude representation, a set of Euler angles and quaternions, which are a minimal and a redundant set, respectively, to represent orientation. Consequently two controllers, one for each formalism, will be described.

The quaternions representation has computational advantages with respect to Euler angles, so they are used for acrobatic tasks, in order to avoid the gimbal locking effect (Pucheta et al., 2014). Recently, Fresk and Nikolakopoulos (2013) proposed the use of quaternions as a robust kinematic representation of attitude and defined a controller which seems to be very simple in comparison to those based on Euler angles.

In this work, a feedback controller based on quaternions and a quadrotor simulated using Euler angles are used to follow smooth references generated by means of Splines and parabolic blendings in a similar fashion done for robotic manipulators (Sciavicco and Siciliano, 2000; Siciliano and Khatib, 2008; Corke, 2011). The final goal is to integrate the estimation (Paz et al., 2014) and control into an open-architecture autonomous quadrotor developed in the CIII by Gaydou et al. (2014), using the most convenient kinematic representation, or combinations of thereof, for the whole mechatronic system, including the programming of the embedded systems (Paz et al., 2014) and more complex real-time localization, mapping, planning, and intelligent decisions to reach an autonomous flight.

The dynamic model of the quadrotor is described in Section 2. The controllers in terms of Euler angles and in terms of quaternions are introduced in Section 3. Finally, Section 4 describes the generation of references for predefined tasks and shows the quaternion-based controller behavior and simulation results.

## 2 DYNAMIC MODEL OF THE QUADROTOR

The quadrotor is modelled as a body with inertia and a mass lumped in its center of mass. The second Newton law is valid in an inertial frame denoted as  $\{O, e_1, e_2, e_3\}$ , where  $e_1$  and  $e_2$  span a plane parallel to the ground and  $e_3$  points upwards; for this purpose, it is assumed as Earth fixed frame; see Fig. 1. A body frame of reference is located on the body  $\{B, b_1, b_2, b_3\}$

with vectors aligned with the principal axes of inertia,  $b_1$  points towards the front direction,  $b_2$  towards the left, and  $b_3$  points upwards the propellers plane (Mahony et al., 2012). The integration of the equations of motion for the rotational dynamics in body frame are simplified because the inertia is invariant with time in that axes.

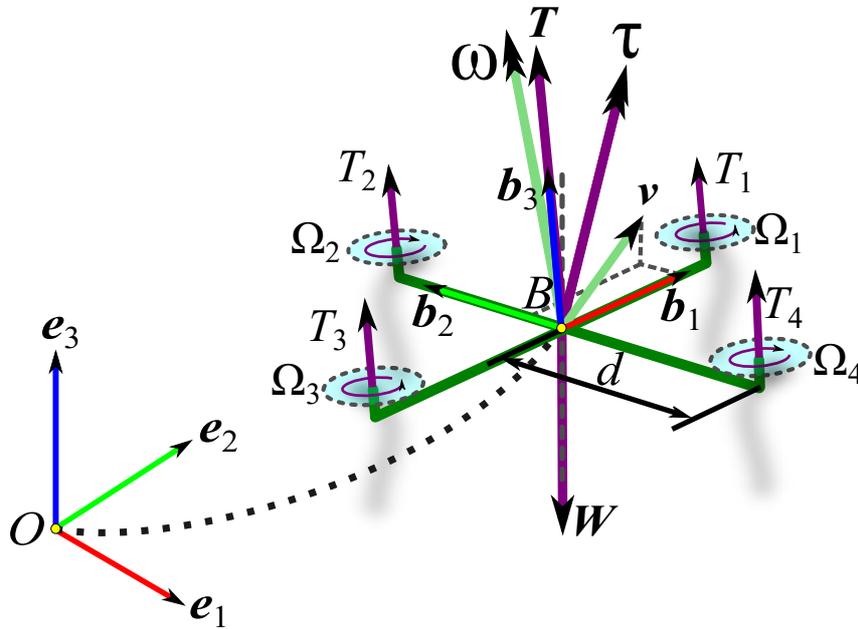


Figure 1: Definitions for the quadrotor dynamics

The vehicle has six degrees-of-freedom and is an under-actuated system because it has four inputs as degree of control, the four velocities of rotation of the propellers with axes parallel to  $b_3$ . Each propeller produces a thrust force always in direction  $b_3$ . Propellers in opposite sides of axis  $b_1$  have a right helix and produce a positive drag moment, while propellers in opposite sides of  $b_2$  have a left helix and produce a negative drag moment. In hovering, the resultant of the thrust forces must be equal to the weight and the drag moments exist but are cancelled by pairs.

## 2.1 Euler equations

The equations of motion of the vehicle is described as a ordinary second order differential equation obtained from Newton law or by means of Lagrange derivation.

The translational dynamics is expressed starting from the position of the center of mass in expressed in the inertial frame. The quadrotor is subjected to its own weight, the conservative force  $W$ , and the non-conservative forces  $T_i$  imparted by the propellers.

$$\Sigma F = F_\Sigma + W = R \Sigma_{i=1}^4 T_i - m g e_3 = \frac{d(mv)}{dt} = m\dot{v} \quad (1)$$

where,  $m$  is the vehicle mass,  $v$  is the linear velocity in the inertial frame,  $g$  is the gravitational acceleration, and  $R$  is the orthogonal matrix that change the coordinates from the body frame to the inertial frame (Pucheta et al., 2014); its columns are the vectors  $b_i$  expressed in the inertial frame,

$$R = [b_1 \ b_2 \ b_3] \in SO(3). \quad (2)$$

By considering the external non-conservative moments applied by the propellers, the rotational dynamics can be obtained from the second Newton's law, which is valid in an inertial frame. The Euler's equation in the inertial frame expresses that the external torque  $\tau_I$  acting on the body is equal to the rate of change of the angular momentum  $\mathbb{I}_I \omega_I$

$$\begin{aligned} \Sigma M &= \tau_I = \frac{d(\mathbb{I}_I \omega_I)}{dt} \\ &= \widetilde{\omega}_I \mathbb{I}_I \omega_I + \mathbb{I}_I \dot{\omega}_I \end{aligned} \quad (3)$$

where,  $\mathbb{I}_I$  is the inertia matrix and  $\omega_I$  the angular velocity. The tilde operator ( $\widetilde{\bullet}$ ) forms a skew symmetric matrix of the vector  $\bullet$  such that the cross product can be expressed as a matrix-vector operation,  $\widetilde{v}u = v \times u, \forall u$ . To write this rotational dynamics in the body frame, firstly it is analyzed that each column in  $R$  has a rate  $\dot{b}_i = \omega_I \times b_i = \widetilde{\omega}_I b_i$  as the body frame rotates with angular velocity  $\omega_I$ , then, these rates are collected by columns in an additional equation

$$\dot{R} = \widetilde{\omega}_I R \quad (4)$$

secondly, the relationship between the moment  $\tau$ , angular velocity  $\omega$ , and inertia tensor  $\mathbb{I}$  in the body frame are related to the inertial frame by the changes of coordinates

$$\begin{aligned} \tau_I &= R \tau \quad \therefore \quad \tau = R^T \tau_I \\ \omega_I &= R \omega \quad \therefore \quad \omega = R^T \omega_I \\ \mathbb{I} &= R^T \mathbb{I}_I R \quad \therefore \quad \mathbb{I}_I = R \mathbb{I} R^T \end{aligned} \quad (5)$$

Using these relationships and replacing the inertial frame dynamics (3), and  $\widetilde{\omega}_I$  from (4), and considering that  $\dot{R}\omega = 0$  leads to

$$\begin{aligned} \tau &= R^T \tau_I = R^T \left( \widetilde{\omega}_I \mathbb{I}_I \omega_I + \mathbb{I}_I \frac{d(\omega_I)}{dt} \right) \\ &= R^T \left( \dot{R} R^T R \mathbb{I} R^T R \omega \right) + R^T \left( R \mathbb{I} R^T (\dot{R} \omega + R \dot{\omega}) \right) \end{aligned} \quad (6)$$

where, in the second term  $\dot{R}\omega = (\widetilde{\omega}_I R)\omega = \widetilde{\omega}_I \omega_I = \omega_I \times \omega_I = \mathbf{o}$ . Then, by simplifying the identity matrices that result from  $R^T R$ , the expression in the body frame is the set of equations

$$\begin{cases} \tau = \widetilde{\omega} \mathbb{I} \omega + \mathbb{I} \dot{\omega} \\ \dot{R} = R \widetilde{\omega} \end{cases} \quad (7)$$

where the inertia matrix is diagonal for a symmetric quadrotor if and only if the body frame is taken to be aligned with the principal axes of inertia; the arrangements of components on the quadrotor must ensure that the center of mass contains the origin of this frame and that mass distribution is symmetric about the planes formed by these axes. Then, a diagonal inertia matrix is easily invertible as required for the integration of the equation of motion. Finally, the dynamics of the quadrotor can be written as

$$\begin{cases} \dot{r} = v_I \\ \dot{v}_I = \frac{1}{m} R \Sigma_{i=1}^n T_i - g e_3 \\ \dot{R} = R \widetilde{\omega} \\ \dot{\omega} = \mathbb{I}^{-1} [\Sigma_{i=1}^n (T_i \times d_i + Q_i) - \widetilde{\omega} \mathbb{I} \omega] \end{cases} \quad (8)$$

where,  $n$  is the number of rotors,  $\mathbf{T}_i = T_i \mathbf{b}_3$  are the thrust forces in body frame, and  $\mathbf{d}_i$  are the moment arms of the rotors measuring the distances from the rotor axes to the origin  $B$  of the body frame. This equation is also valid for multirotors taking into account that a even number of rotors is required to easy fulfill the cancellation of drag moments.

By using simplified aero-dynamical models, the thrusts and drag moments of the propellers can be considered as linear functions of the squared velocity of rotation,  $\Omega_i$ .

$$\begin{cases} T_i = k \Omega_i^2 \\ Q_i = q \Omega_i^2 \end{cases} \quad (9)$$

where, the trust coefficient  $k > 0$  is determined by test in bench and mainly depends on the rotor geometry (area, radius), helix profile, and, an assumed constant, density of the air; the same considerations are made for the drag coefficient  $q > 0$ . These coefficients are the same for all the rotors.

For the quadrotor, if the angular velocity vectors  $\boldsymbol{\Omega}_i, i = \{1, 3\}$  are positive because these propellers rotate around  $\mathbf{b}_3$  in positive sense, and propellers  $\boldsymbol{\Omega}_i, i = \{2, 4\}$  rotate in negative sense, the drag moments of the rotors are opposite to them, respectively,  $\mathbf{Q}_i = -Q_i \mathbf{b}_3, i = \{1, 3\}$  and  $\mathbf{Q}_i = Q_i \mathbf{b}_3, i = \{2, 4\}$ . The arrangement of the type of helices (left and right types) ensures that the four air vortices are generated downwards. The assumption that thrusts and drag moments are aligned with  $\mathbf{b}_3$  neglects the effects of flexibility in the airframe and in the helices.

The relationships between the applied forces (in  $z$ -direction of the body frame) and torques (around each axis of the body frame) can now be written, in body frame, in terms of the rotor velocities as

$$\begin{aligned} \begin{bmatrix} F_{\Sigma z} \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} &= \begin{bmatrix} \sum_{i=1}^4 T_i \\ T_2 d - T_4 d \\ T_3 d - T_1 d \\ -Q_1 + Q_2 - Q_3 + Q_4 \end{bmatrix} = \begin{bmatrix} k \sum_{i=1}^4 \Omega_i^2 \\ k d (\Omega_2^2 - \Omega_4^2) \\ k d (\Omega_3^2 - \Omega_1^2) \\ q (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix} \\ &= \begin{bmatrix} k & k & k & k \\ 0 & kd & 0 & -kd \\ -kd & 0 & kd & 0 \\ -q & q & -q & q \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \end{aligned} \quad (10)$$

where matrix  $\mathbf{A}$  is invertible because  $k, d, q > 0$ , and then, the thrust and moments can be produced by determined and unique values of the squared angular velocities of the rotors computed as

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} F_{\Sigma} \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} F_{\Sigma}/(4k) - \tau_z/(4q) - \tau_y/(2kd) \\ F_{\Sigma}/(4k) + \tau_z/(4q) + \tau_x/(2kd) \\ F_{\Sigma}/(4k) - \tau_z/(4q) + \tau_y/(2kd) \\ F_{\Sigma}/(4k) + \tau_z/(4q) - \tau_x/(2kd) \end{bmatrix} \quad (11)$$

Other arrangements of multirotors with 6, 8 and 16 propellers can be used to provide redundancy against failure and to increase the payload capability.

## 2.2 Quadrotor simulation by reduction from second to a first order ODE

The orthogonal matrix that change of coordinates from body to inertial frame (it is the transpose of the active rotation orthogonal matrix) can be parameterized in terms of  $xyz$  sequence of

Euler angles and arranged in vector form as  $\boldsymbol{\gamma} := [\phi \ \theta \ \psi]^T$ , the vector form of its time derivatives is  $\dot{\boldsymbol{\gamma}} := [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ . Then, the system (8) can be written as a first order ODE in the state space form

$$\begin{cases} \mathbf{x}' = \mathcal{F}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} = \mathcal{H}(\mathbf{x}) \end{cases} \tag{12}$$

where the state vector is arranged as  $\mathbf{x} = [\mathbf{r} \ \boldsymbol{\gamma} \ \dot{\mathbf{r}} \ \dot{\boldsymbol{\gamma}}]^T$ ,  $\mathbf{r}$  is the position of the quadrotor center of mass, the origin  $B$  of the body frame,  $\mathbf{v}_I = \dot{\mathbf{r}}$  its linear velocity vector in the inertial frame, and the control vector is  $\mathbf{u} := [F_\Sigma \ \tau_x \ \tau_y \ \tau_z]^T$ , containing the four inputs. The observation and measured values are simulated in the second equation  $\mathbf{y} = \mathcal{H}(\bullet)$ . Then, the orthogonal matrix that changes the coordinates from inertial to body frame in terms of the Euler's angles is computed as

$$\begin{aligned} \mathbf{R}_I^B(\boldsymbol{\gamma}) &:= \mathbf{R}_I^B(\phi, \theta, \psi) = \mathbf{R}_x(\phi) \mathbf{R}_y(\theta) \mathbf{R}_z(\psi) \\ &:= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &:= \begin{bmatrix} \cos \psi \cos \theta & \cos \theta \sin \psi & -\sin \theta \\ \cos \psi \sin \phi \sin \theta - \cos \phi \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \psi \sin \theta & \cos \theta \sin \phi \\ \sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta & \cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi & \cos \phi \cos \theta \end{bmatrix} \end{aligned} \tag{13}$$

This matrix is the transpose of the coordinate transformation from the body to the inertial frame presented above, such that  $\mathbf{R}^T := \mathbf{R}_I^B(\boldsymbol{\gamma})$  and  $\mathbf{R} = (\mathbf{R}_I^B)^T = \mathbf{R}_B^I$ .

The time derivatives of the Euler parameters can be arranged as a matrix/vector product to express their relation with the angular velocity in body frame

$$\begin{aligned} \boldsymbol{\omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= [\mathbf{e}_1; \mathbf{R}_x(\phi) \mathbf{e}_2; \mathbf{R}_x(\phi) \mathbf{R}_y(\theta) \mathbf{e}_3] \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \\ &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_x(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_x(\phi) \mathbf{R}_y(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ &= \begin{bmatrix} \dot{\phi} - \dot{\psi} \sin \theta \\ \dot{\theta} \cos \phi + \dot{\psi} \cos \theta \sin \phi \\ \dot{\psi} \cos \phi \cos \theta - \dot{\theta} \sin \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \\ &= \mathbf{L}(\boldsymbol{\gamma}) \dot{\boldsymbol{\gamma}} \end{aligned} \tag{14}$$

Then, the state form (12) takes the form

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v}_I \\ \dot{\mathbf{v}}_I = \frac{1}{m} \mathbf{R}(\boldsymbol{\gamma}) \sum_{i=1}^n \mathbf{T}_i - g \mathbf{e}_3 \\ \dot{\boldsymbol{\gamma}} = \mathbf{L}^{-1}(\boldsymbol{\gamma}) \boldsymbol{\omega} \\ \dot{\boldsymbol{\omega}} = \mathbb{I}^{-1}[\sum_{i=1}^n (\mathbf{T}_i \times \mathbf{d}_i + \mathbf{Q}_i) - \tilde{\boldsymbol{\omega}} \mathbb{I} \boldsymbol{\omega}] \end{cases} \tag{15}$$

The translational dynamics can be integrated in the inertial frame and the rotational dynamics integrated in the body frame. The quadrotor can works in very different flight modes and this is

application dependent. The quadrotor controller can receive the linear velocity in body frame by modifying the output of the solution  $\mathbf{x}$  of (15) in its third vector by  $\mathbf{v} = \mathbf{R}(\boldsymbol{\gamma})^T \mathbf{v}_I$ . This transformation is used to return the observed vector taking the form

$$\begin{aligned} \mathbf{y} &= [\mathbf{r}, \boldsymbol{\gamma}, \mathbf{R}(\boldsymbol{\gamma})^T \mathbf{v}_I, \dot{\boldsymbol{\gamma}}]^T \\ &= [\mathbf{r}, \boldsymbol{\gamma}, \mathbf{v}, \dot{\boldsymbol{\gamma}}]^T. \end{aligned} \tag{16}$$

### 3 CONTROLLER

Basic controllers for quadrotors are based on feedback control where feedback errors between measured and reference states are computed to take decisions in terms of the error itself, its derivatives and its integral inside a time interval. Although non-linear controllers are simpler, we here focus on feedback controllers.

#### 3.1 An Euler angles approach

The controller described by Corke (2011), available as a set of Matlab Simulink® blocks and routines, is reviewed and adapted to the proposed body frame convention and the QA3 quadrotor parameters. An intermediate frame  $V$  with axes parallel to the projections of the axes  $\mathbf{b}_1$  and  $\mathbf{b}_2$  on the ground plane, is used to approximate the error in the  $x, y$  velocity as shown in Fig. 2.

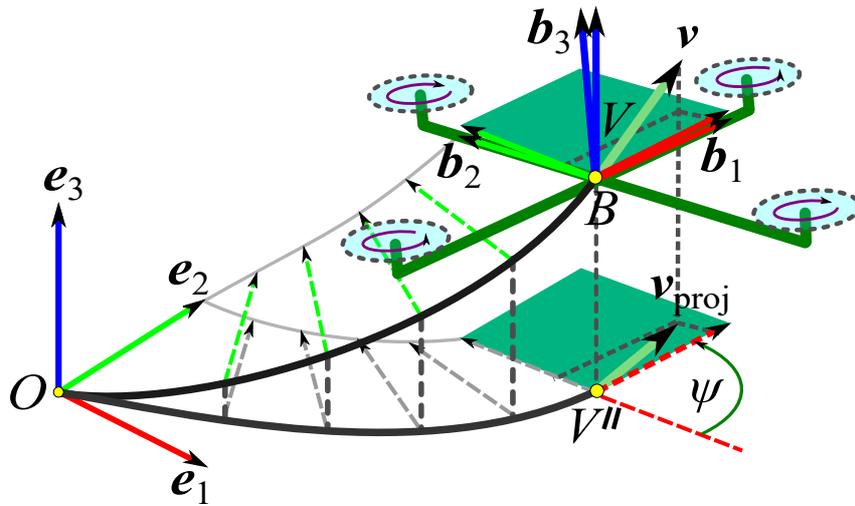


Figure 2: The velocity of the quadrotor projected on a plane parallel to the ground plane is approximated to the body frame velocity for smooth maneuvers.

Given the desired references  $\{\mathbf{r}_{\text{ref}}(t), \psi_{\text{ref}}(t)\}$  and an observed state  $\mathbf{y}$ , the controller uses four main loops:

1. A first external loop controls the altitude  $z$ , using a proportional derivative (PD) controller.

$$F_{\Sigma} = K_{pz}(z_{\text{ref}} - z) + K_{dz}(-\dot{z}) + T_0, \tag{17}$$

where  $T_0 = \sqrt{mg/4k}$  is the total thrust required in hovering and it is the unique time independent feed-forward term.

2. In parallel to the first loop, a second external loop controls the yaw attitude  $\psi$ , using the PD controller

$$\tau_z \approx \tau_{\psi} = K_{p\psi}(\psi_{\text{ref}} - \psi) + K_{d\psi}(-\dot{\psi}) \tag{18}$$

where the yaw observed  $\psi$  is also used to change of frame the coordinates of the  $x, y$  error.

3. The most outer loop, compute the error vector of the  $x, y$ -position of the quadrotor, which is the difference between the the reference and the updated state for the  $x, y$ -vector changed in coordinates by applying a passive rotation around  $e_3$  by the actual yaw state  $\psi$ , so it is coupled with the yaw controller, which is also of PD type

$$\begin{bmatrix} x_{\text{err}}^V \\ y_{\text{err}}^V \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} x_{\text{ref}} - x \\ y_{\text{ref}} - y \end{bmatrix} \quad (19)$$

$$\begin{aligned} \theta_{\text{ref}} &= K_{pxy} x_{\text{err}}^V + K_{dxy}(-\dot{x}) \\ \phi_{\text{ref}} &= -[K_{pxy} y_{\text{err}}^V + K_{dxy}(-\dot{y})] \end{aligned} \quad (20)$$

where a positive pitch,  $\theta$ , moves the vehicle towards a positive  $x$  motion and a positive roll,  $\phi$ , moves the vehicle towards a negative  $y$  motion, therefore here a minus sign affects the value computed for  $\phi_{\text{ref}}$ .

4. The output of the  $x, y$ -position controller is the input reference for the roll and pitch PD controller.

$$\begin{aligned} \tau_x &\approx \tau_\phi = K_{p\phi}(\phi_{\text{ref}} - \phi) + K_{d\phi}(-\dot{\phi}) \\ \tau_y &\approx \tau_\theta = K_{p\theta}(\theta_{\text{ref}} - \theta) + K_{d\theta}(-\dot{\theta}) \end{aligned} \quad (21)$$

Therefore, the output of the roll, pitch, and yaw angle controllers gives as a result the moments  $\tau_\phi = \tau_{\text{roll}}$ ,  $\tau_\theta = \tau_{\text{pitch}}$ , and  $\tau_\psi = \tau_{\text{yaw}}$  around the axes used sequentially in the Euler angles definition that only under small angular displacements (or high update rate for the controller loop, which is the case in the real quadrotor) are a good approximation for the moments  $\tau_x$ ,  $\tau_y$ , and  $\tau_z$  applied to the quadrotor aircraft around Cartesian body-frame axes. The  $z$  controller provides a thrust  $F_\Sigma$  as output. Then, based on equation (11) the controller of the electric motors is supplied with the following rotor velocities with inverted signs for the second and four rotors

$$\begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \approx \begin{bmatrix} F_\Sigma/(4k) - \tau_z/(4q) - \tau_y/(2kd) \\ (-1)[F_\Sigma/(4k) + \tau_z/(4q) + \tau_x/(2kd)] \\ F_\Sigma/(4k) - \tau_z/(4q) + \tau_y/(2kd) \\ (-1)[F_\Sigma/(4k) + \tau_z/(4q) - \tau_x/(2kd)] \end{bmatrix} \approx \begin{bmatrix} F_\Sigma - \tau_z - \tau_y \\ (-1)[F_\Sigma + \tau_z + \tau_x] \\ F_\Sigma - \tau_z + \tau_y \\ (-1)[F_\Sigma + \tau_z - \tau_x] \end{bmatrix} \quad (22)$$

These four signals are combined and transformed into velocities of the rotors  $\omega_i$  but must be conditioned to have a lower and upper bound. The lower bound is related to the minimal RPM to have a smooth landing movement, e.g. the thrust should not be less than 15% of the rotation  $\Omega_0$  necessary to get hovering, using  $\Omega_{\text{min}} = 0.85\Omega_0$ .

$$\Omega_i = \begin{cases} 0.85\Omega_0 & \text{if } \Omega_i < 0.85\Omega_0 \\ \Omega_{\text{max}} & \text{if } \Omega_i > \omega_{\text{max}} \end{cases} \quad (23)$$

The upper bound limits the angular velocity of the rotors to avoids its overload and destruction. There are several techniques to scale these outputs, but here they are simply truncated to a maximum value; the simulation is useful to identify these values for the most aggressive desired maneuvers that avoid the saturation of actuators, e.g. in terms of maximum translational velocities.

Note that the controller of the (identical) electrical motors is not explained here, but its input signal is converted into pulse width modulated (PWM) signals inside a range of 0 – 100% to provide the voltage of reference to the three-phase brush-less electrical motors.

One disadvantage in using Euler angles is the singularity (“gimbal locking”) in  $\mathbf{L}$  when the pitch angle exceeds the  $\pi/2$ . Based on this controller it was easy to setup a controller that uses a singularity-free kinematic representation for the orientation, the quaternions.

### 3.2 A quaternion approach

The quaternion is an orientation representation which is free of singularities and was used for satellite controllers in the last decades. Because the simplicity of the controllers its application to quadrotor guidance has gained popularity; [Fresk and Nikolakopoulos \(2013\)](#) presented a quaternion based controller for orientation only. In this work, the quaternion-based controller is embedded in the Euler angles controller shown in the previous section to control both, position and orientation.

Quaternions are advantageous for several computational issues: (i) once built they can be operated algebraically and faster than using transcendental functions, (ii) they can be efficiently used in the state estimation of the orientation ([Paz et al., 2014](#)), so here we assume that the orientation quaternion is already build and available as part of the observed state.

In order to reuse the plant simulation in Euler angles, and only for simulation purposes, the observed state in Euler angles is used as data to simulate the state as

$$\mathbf{y} = [\mathbf{r}, \boldsymbol{\gamma}, \mathbf{v}, \dot{\boldsymbol{\gamma}}]^T \rightarrow \mathbf{y}_q = [\mathbf{r}, \mathbf{q}(\boldsymbol{\gamma}), \mathbf{v}, \boldsymbol{\omega}(\boldsymbol{\gamma})]^T \quad (24)$$

where,  $\boldsymbol{\omega}$  is the angular velocity in body frame computed from the Euler angles as  $\mathbf{L}(\boldsymbol{\gamma})\dot{\boldsymbol{\gamma}}$ , and  $\mathbf{q}(\boldsymbol{\gamma})$  is the vectorial form of four components of the unit quaternion

$$\hat{\mathbf{q}} = q_0 + q_1i + q_2j + q_3k, \quad \|\hat{\mathbf{q}}\| = 1 \quad (25)$$

representing the observed orientation, and computed using the Euler angles  $\boldsymbol{\gamma} = [\phi, \theta, \psi]$  as

$$\begin{aligned} cR &= \cos(\phi/2); & sR &= \sin(\phi/2); \\ cP &= \cos(\theta/2); & sP &= \sin(\theta/2); \\ cY &= \cos(\psi/2); & sY &= \sin(\psi/2); \end{aligned} \quad (26)$$

and

$$\begin{aligned} q_0 &= cR.cP.cY + sR.sP.sY \\ q_1 &= -cR.sP.sY + sR.cP.cY \\ q_2 &= cR.sP.cY + sR.cP.sY \\ q_3 &= cR.cP.sY - sR.sP.cY \end{aligned} \quad (27)$$

This conversion formulas are also used to build the reference quaternion from the reference Euler angles as  $\mathbf{q}_{\text{ref}}(\boldsymbol{\gamma}_{\text{ref}})$ .

Note that a quaternion-based state estimation as proposed by [Paz et al. \(2014\)](#) already supplies an state vector containing  $\mathbf{q}$  and  $\boldsymbol{\omega}$ , thus, they do not depends on the Euler angles.

Given the observed quaternion  $\mathbf{q}$  and the reference quaternion  $\mathbf{q}_{\text{ref}}$ , the quaternion error can be computed from the relative rotation computed as the quaternion product

$$\mathbf{q}_{\text{err}} = \mathbf{q}_{\text{ref}} \cdot \mathbf{q}^* \quad (28)$$

Then the axis error or relative rotation vector is computed using the components of  $\mathbf{q}_{\text{err}}$  as

$$\mathbf{e}_R = \begin{bmatrix} 2q_{2\text{err}}q_{0\text{err}} \\ 2q_{3\text{err}}q_{0\text{err}} \\ 2q_{4\text{err}}q_{0\text{err}} \end{bmatrix} \quad (29)$$

The quaternion-based position and orientation controller is described as follows. Given the desired references  $\{\mathbf{r}_{\text{ref}}(t), \psi_{\text{ref}}(t)\}$  and an observed state  $\mathbf{y}_q$ , the controller uses three main loops:

1. A first external loop controls the altitude  $z$ , using a proportional derivative (PD) controller.

$$F_\Sigma = K_{pz}(z_{\text{ref}} - z) + K_{dz}(-\dot{z}) + T_0, \quad (30)$$

where  $T_0 = \sqrt{mg/4k}$  is the total thrust required in hovering.

2. The most outer loop, compute the error vector of the  $x, y$ -position of the quadrotor, which is the difference between the reference and the updated state for the  $x, y$ -vector changed in coordinates by applying a passive rotation around  $\mathbf{e}_3$  by the actual yaw state  $\psi$ , so it is coupled with the yaw controller, which is also of PD type. From the observed quaternion  $\mathbf{q}$ , the yaw is computed as

$$\psi = \text{atan2}(2q_1q_2 + 2q_0q_3, q_1^2 + q_0^2 - q_3^2 - q_2^2). \quad (31)$$

Then, it is used as

$$\begin{bmatrix} x_{\text{err}}^V \\ y_{\text{err}}^V \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} x_{\text{ref}} - x \\ y_{\text{ref}} - y \end{bmatrix} \quad (32)$$

$$\begin{aligned} \theta_{\text{ref}} &= K_{pxy} x_{\text{err}}^V + K_{dxy}(-\dot{x}) \\ \phi_{\text{ref}} &= -[K_{pxy} y_{\text{err}}^V + K_{dxy}(-\dot{y})] \end{aligned} \quad (33)$$

where a positive pitch,  $\theta$ , moves the vehicle towards a positive  $x$  motion and a positive roll,  $\phi$ , moves the vehicle towards a negative  $y$  motion, therefore here a minus sign affects the value computed for  $\phi_{\text{ref}}$ .

3. The output of the  $x, y$ -position controller is part of the input reference for attitude PD controller because it is used with the yaw reference  $\psi_{\text{ref}}$  to build the quaternion reference  $\mathbf{q}_{\text{ref}}(\phi_{\text{ref}}, \theta_{\text{ref}}, \psi_{\text{ref}})$  by means of Eq. (27). Then, using Eqns. (28) and (29) the rotation error  $\mathbf{e}_R$  is computed and then used with  $\boldsymbol{\omega} = [p, q, r]$  in the PD attitude controller

$$\begin{aligned} \tau_x &= K_{pRx} e_{Rx} + K_{dRx}(-p) \\ \tau_y &= K_{pRy} e_{Ry} + K_{dRy}(-q) \\ \tau_z &= K_{pRz} e_{Rz} + K_{dRz}(-r) \end{aligned} \quad (34)$$

Note that the height and  $x - y$  position controller are identical in the two controllers. The main difference in the attitude loop is that the quaternion-based controller is robust to kinematic singularities and support large displacements.

## 4 SIMULATION RESULTS

The physical quadrotor properties required for the simulation of the plant are shown in Table 1. The values for the Earth gravity and air density are respectively  $g = 9.81 \text{ m/s}^2$  and  $\rho = 1.184 \text{ Kg/m}^3$ . For the rotor with an helix of type “APC E 9x6” and radius  $r$ , the value  $b$  was obtained from bench experiments by computing the lift force in a precise weighing balance for a wide range of angular rotation speeds. Then,  $b$  was found to be the slope of the linear regression of that experimental data, where  $b$  units are in  $\text{N} \cdot \text{s}^2/\text{rad}^2 = \text{Kg} \cdot \text{m}$ .

Aircraft	Actuators	Derived quantities
$m = 1.3 \text{ Kg}$	$r = 0.1145 \text{ m}$	$A = \pi \cdot r^2 \text{ m}^2$
$d = 0.3055 \text{ m}$	$b = 1.68 \cdot 10^{-5} \text{ N} \cdot \text{s}^2/\text{rad}^2$	
$I_{xx} = 0.0274 \text{ Kg} \cdot \text{m}^2$		$C_t = b/(\rho \cdot A \cdot r^2)$
$I_{yy} = 0.0274 \text{ Kg} \cdot \text{m}^2$		$C_q = C_t \cdot \sqrt{(C_t/2)}$
$I_{zz} = 0.0520 \text{ Kg} \cdot \text{m}^2$		$k = C_q \cdot \rho \cdot A \cdot r^3 \text{ N} \cdot \text{s}^2/\text{rad}^2$

Table 1: Physical parameters in SI units for the QAAA quadrotor developed in the CIII.

For these properties and a set of kinematic tasks, described in the following subsections, the gains of the two controllers were obtained by trial and error and are shown in Table 2. In future work several automated synthesis algorithms for tuning the PD controllers will be used. Up to now, a PD controller is used as the reference flight mode that can be switched to on-line generated trajectories based on optimal control and other non-linear control strategies for real-time avoidance of obstacles.

Euler angles	$z$ -loop	$x - y$ loop	roll	pitch	yaw
$K_p$	100	0.2	400	400	100
$K_d$	100	0.2	0.01	0.01	1
Quaternion	$z$ -loop	$x - y$ loop	roll	pitch	yaw
$K_p$	100	0.2	<b>500</b>	<b>500</b>	100
$K_d$	100	0.2	<b>50</b>	<b>50</b>	1

Table 2: Gains for Euler angles-based and quaternion-based full pose controllers for the QAAA quadrotor.

Three path following tasks off-line generated were used to compare the two controllers. Because no difference were appreciated between the tracking errors of the Euler angles-based controller and the quaternion-based controller, only the results for the quaternion-based one are shown.

### 4.1 Circle trajectory

This task described by Eq. (35) is similar to that proposed by Corke (2011), but parameterized with the total simulation time to test several speeds; see Figure 3. The quadrotor develops two turns while moving in yaw linearly with time. The total simulation time is  $T_{\text{sim}} = 20\text{s}$  and the initial position is  $r_0 = [0.0, 0.0, 0.15]$ . The results for the simulated states are shown in

Figure 4.

$$\begin{cases} x_{\text{ref}}(t) = \sin(\omega_x t), & \omega_x = 4\pi/T_{\text{sim}} \\ y_{\text{ref}}(t) = \cos(\omega_y t), & \omega_y = 4\pi/T_{\text{sim}} \\ y_{\text{ref}}(t) = 1 \\ \psi_{\text{ref}}(t) = \frac{2\pi}{T_{\text{sim}}} t; \end{cases} \quad (35)$$

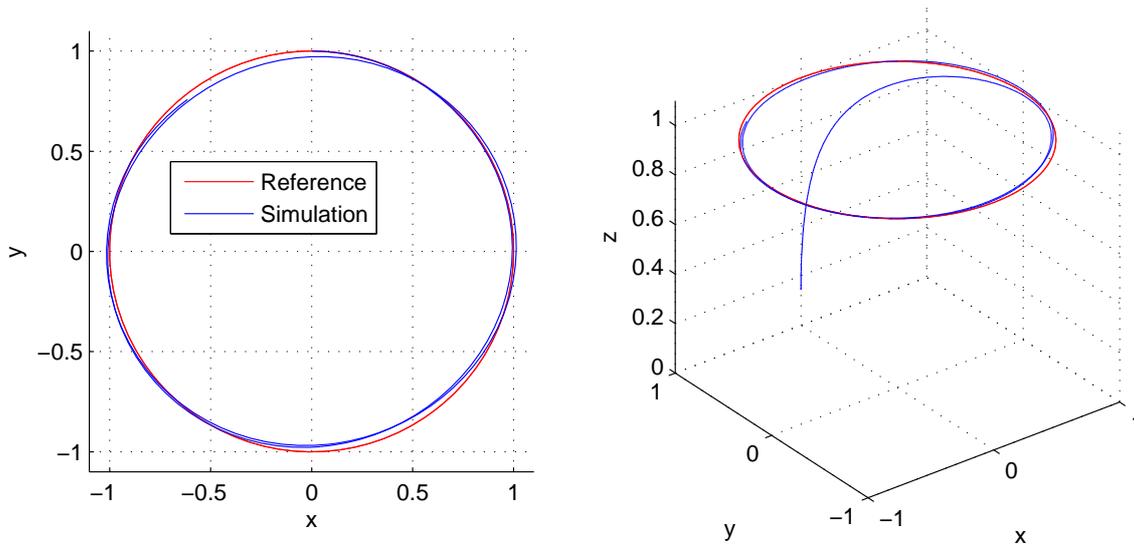


Figure 3: Circle trajectory

## 4.2 8-Shape trajectory

The 8-shape trajectory shown in Figure 5 is defined in a similar fashion that in the circle as

$$\begin{cases} x_{\text{ref}}(t) = \sin(\omega_x t), & \omega_x = 4\pi/T_{\text{sim}} \\ y_{\text{ref}}(t) = \sin(\omega_y t), & \omega_y = 8\pi/T_{\text{sim}} \\ y_{\text{ref}}(t) = 1 \\ \psi_{\text{ref}}(t) = \frac{2\pi}{T_{\text{sim}}} t; \end{cases} \quad (36)$$

The total simulation time is  $T_{\text{sim}} = 40\text{s}$  and the initial position is also  $\mathbf{r}_0 = [0.0, 0.0, 0.15]$ .

The results for the simulated states are shown in Figure 6.

## 4.3 Square-Shape trajectory

The trajectory is defined by multi-segments references approaching the following way-points (or via points); see Figure 8

$$\begin{aligned} \mathbf{p}_0 &= [0.0, 0.0, 0.15]; & \mathbf{p}_1 &= [0.0, 0.0, 1.0]; & \mathbf{p}_2 &= [1.0, 0.0, 1.0]; \\ \mathbf{p}_3 &= [1.0, 1.0, 1.0]; & \mathbf{p}_4 &= [0.0, 1.0, 1.0]; & \mathbf{p}_5 &= [0.0, 0.0, 1.0]; \\ \mathbf{p}_6 &= [0.0, 0.0, 0.15] = \mathbf{p}_0; \end{aligned} \quad (37)$$

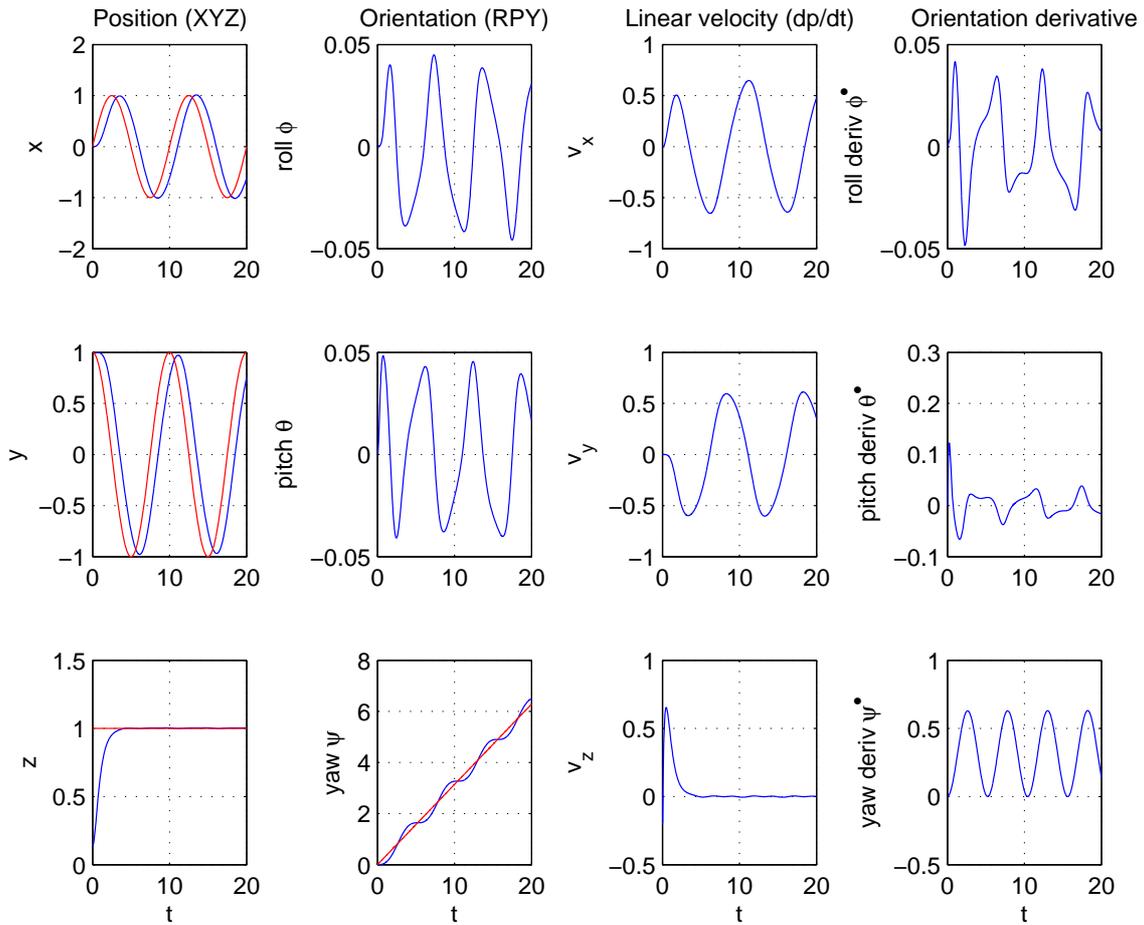


Figure 4: States for a circle trajectory: the references  $x_{ref}(t)$ ,  $y_{ref}(t)$ ,  $z_{ref}(t)$ , and  $\psi_{ref}(t)$  are shown in red color lines and the simulated states in blue color lines

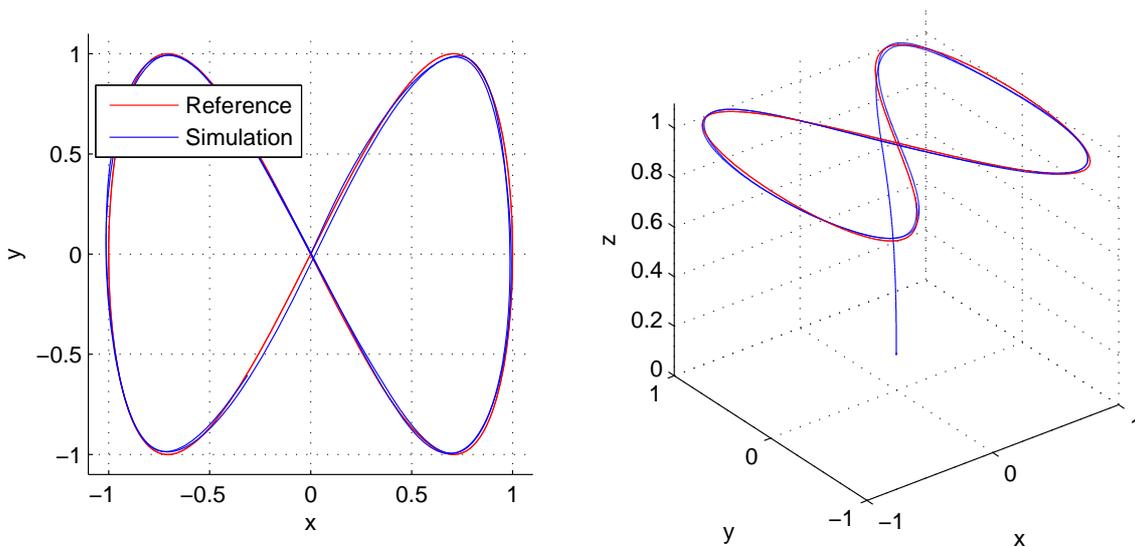


Figure 5: 8-shape trajectory

These points are interpolated using multi-segments trajectories to get a continuous smooth motion as proposed by Corke (2011, Ch.3) for planar mobile robots and arm manipulators, but here

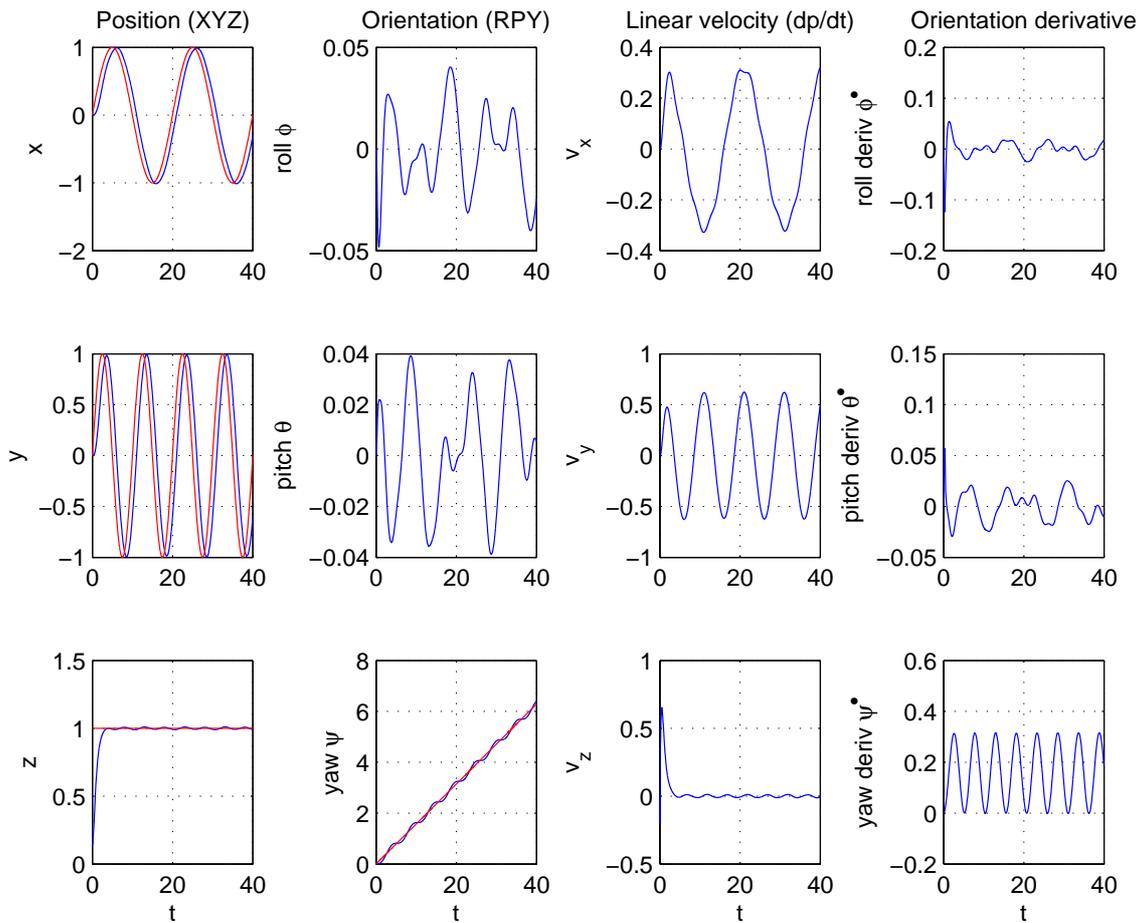


Figure 6: States for a 8-shape trajectory: the references  $x_{ref}(t)$ ,  $y_{ref}(t)$ ,  $z_{ref}(t)$ , and  $\psi_{ref}(t)$  are shown in red color lines and the simulated states in blue color lines

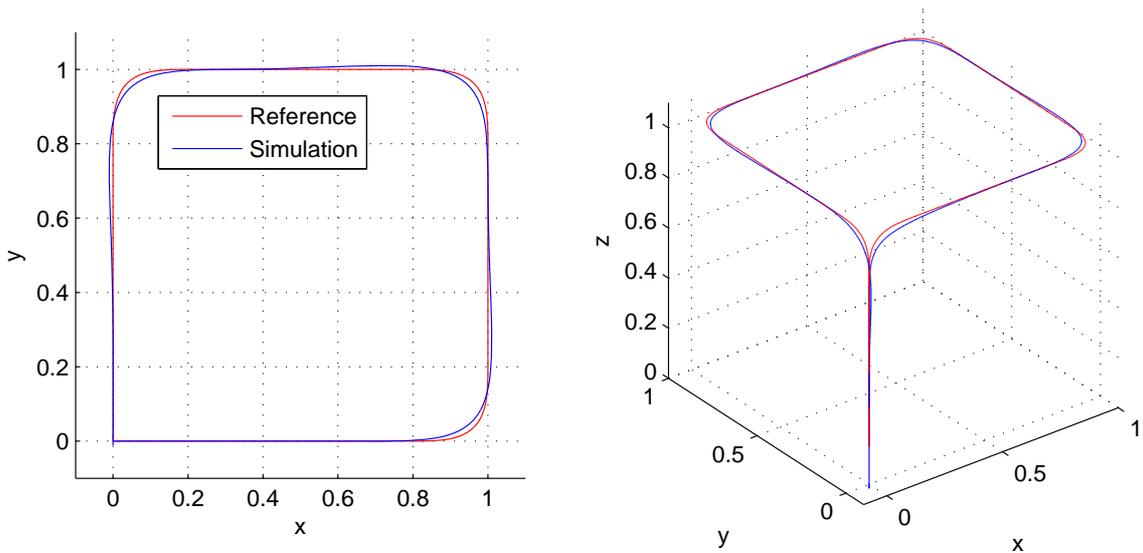


Figure 7: Square-shape trajectory

it is presented for quadrotors for the first time.

In this task, the quadrotor must move from one way-point to another one at constant velocity

without stopping, so that a polynomial blends (of degree five) near way points enforces a smooth transition between two straight directions of the trajectory. This transition can be constrained to be performed in a given time and also developed with a maximum velocity to avoid the saturation of actuators, in this case, to avoid that any electric rotors reaches its maximum angular velocity. The results for the simulated states are shown in Figure 7.

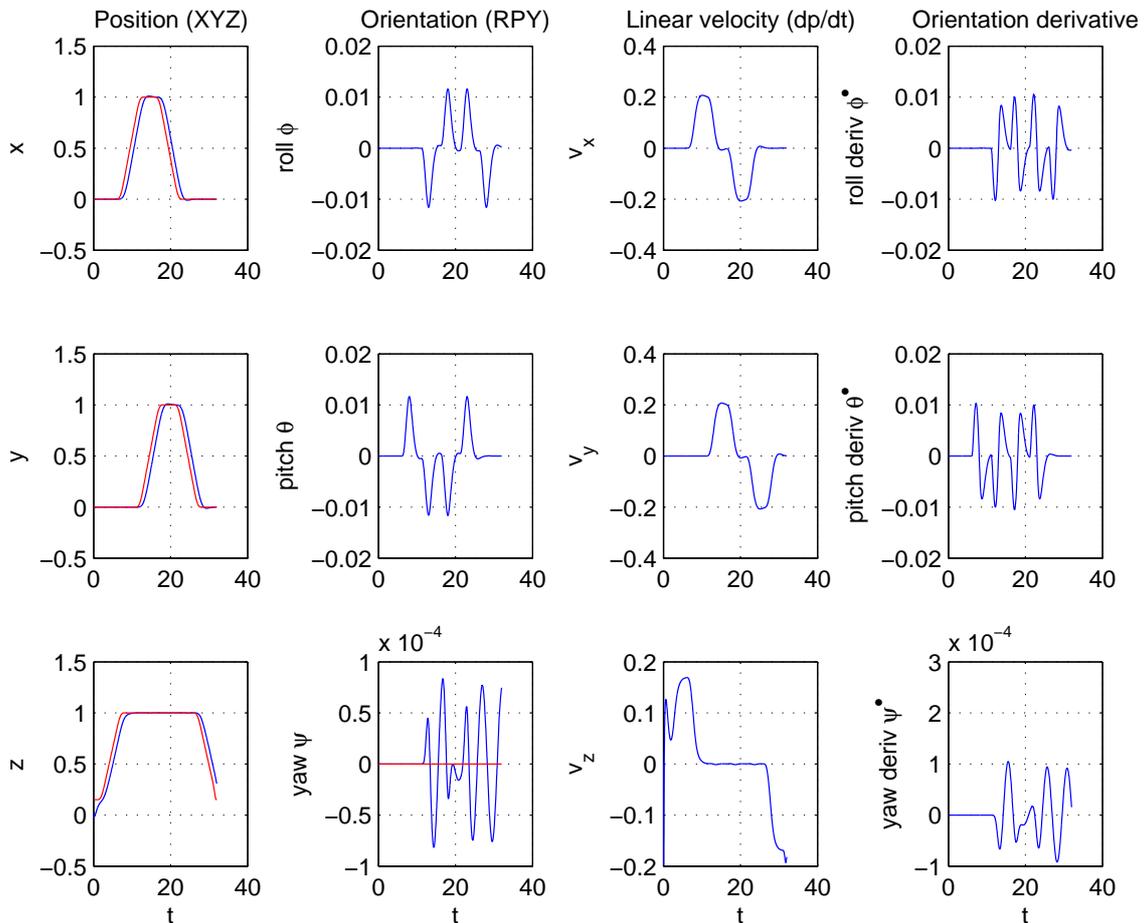


Figure 8: States for a square-shape trajectory: the references  $x_{\text{ref}}(t)$ ,  $y_{\text{ref}}(t)$ ,  $z_{\text{ref}}(t)$ , and  $\psi_{\text{ref}}(t)$  are shown in red color lines and the simulated states in blue color lines

## 5 CONCLUSIONS

A quadrotor controller based on Euler angles was reviewed and was used to introduce a new controller based on the quaternion representation for the orientation. Both controllers were tested and successfully tuned with three off-line generated path following tasks. Two of these smooth references were generated by transcendental functions and the third one was generated by means of discrete way-points with transitions smoothed through polynomials and parabolic blendings in the velocities, in a similar fashion done for references of robotic manipulators and CAD/CAM systems. The latter gives freedom to (off-line) generate very complex trajectories and opens a broad range of applications. The tracking errors for the two controller were similar. Because of the computational advantages of quaternion algebra and its singularity-free property, the quaternion-based controller will be chosen for all future quadrotor developments. These will include, (i) the development of automated strategies for tuning the PD controller;

(ii) the programming of the controller on the quadrotor embedded system (Paz et al., 2014) and its experimental test; (iii) the integration of the presented quaternion based controller with quaternion-based estimation, and (iv) the real-time path generation using simpler controllers based on optimal control strategies and several kinematic formalisms in SE(3).

### Acknowledgements

All the authors acknowledge the financial from the *Universidad Tecnológica Nacional (UTN)* through projects PID-UTN 3935, 2173, 2155, and UTI3832TC; the second author acknowledges the BINID fellowship during years 2014 and 2015; the third author acknowledges the UTN PhD grants program. The first author acknowledges the financial support from the *Agencia Nacional de Promoción Científica y Tecnológica PICT-2013-2894* and from *CONICET*. We gratefully thank to the anonymous reviewer for his corrections, suggested changes, and helpful comments.

### REFERENCES

- Corke P. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Tracts in Advanced Robotics, Volume 73. Springer, Berlin, Germany, 2011.
- Fresk E. and Nikolakopoulos G. Full quaternion based attitude control for a quadrotor. In *European Control Conference (ECC 2013)*, pages 3864–3869. 2013.
- Gaydou D., Suarez G., Paz C., Perez Paina G., and Araguás G. Robot volador no tripulado QA3: Diseño y construcción de un cuatrirrotor para experimentación. In *VIII Jornadas Argentinas de Robótica (JAR 2014)*. Buenos Aires, Argentina, 2014.
- Gupte S., Mohandas P., and Conrad J. A survey of quadrotor unmanned aerial vehicles. *South-eastcon, 2012 Proceedings of IEEE*, pages 1–6, 2012. ISSN 1091-0050.
- Lim H., Park J., Lee D., and Kim H. Build your own quadrotor: Open-source projects on unmanned aerial vehicles. *Robotics Automation Magazine, IEEE*, 19(3):33–45, 2012.
- Mahony R., Kumar V., and Corke P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *Robotics Automation Magazine, IEEE*, 19(3):20–32, 2012.
- Paz C., Infante G., Báez Carballo J., Díaz Báez F., and Cavenio C. Implementación de un filtro extendido de kalman para la estimación de la orientación de un UAV utilizando el estándar CMSIS. In *Proc. V Congreso de Microelectrónica Aplicada - UEA2014*. 2014. Instituto Universitario Aeronáutico, Córdoba, Argentina.
- Pucheta M., Paz C., and Pereyra M.E. Representaciones cinemáticas de orientación y ecuaciones de estimación. In G. Bertolino, M. Cantero, M. Storti, and F. Teruel, editors, *Mecánica Computacional Vol. XIXIII, Actas del XXI Congreso sobre Métodos Numéricos y sus Aplicaciones - ENIEF 2014*, pages 2303–2324. Bariloche, Argentina, 2014. (Artículo completo).
- Sciavicco L. and Siciliano B. *Modelling and Control of Robot Manipulators*. Advanced Textbooks in Control and Signal Processing. McGraw Hill, London, 2 edition, 2000.
- Siciliano B. and Khatib O. *Springer Handbook of Robotics*. Advanced Textbooks in Control and Signal Processing. Springer-Verlag Berlin Heidelberg, Würzburg, 2008.
- Sidi M.J. *Spacecraft Dynamics and Control: A Practical Engineering Approach*. Cambridge Aerospace Series Vol. 7. Cambridge University Press, New York, USA, 1997.
- Stevens B.L. and Lewis F.L. *Aircraft Control and Simulation*. John Wiley & Sons, New York, USA, 1992.