An Intermediary Quaternion-based Control for Trajectory Following Using a Quadrotor

Jose Colmenares-Vazquez, Nicolas Marchand, Pedro Castillo Garcia, Jose-Ernesto Gomez-Balderas

An Intermediary Quaternion-based Control for Trajectory Following Using a Quadrotor

J. Colmenares-Vázquez, N. Marchand, P. Castillo, J.E. Gómez-Balderas

Abstract— **This work uses the intermediary quaternions in the design of a backstepping control technique with integral properties in order to perform an autonomous trajectory tracking using a quadcopter vehicle. Nowadays, in order to determine the orientation of a vehicle, most of the inertial systems of aircrafts can give directly the rotation matrix and taking advantage of this fact, the intermediary quaternions can be determined in a simple way from this matrix. Moreover, one specific orientation corresponds to only one intermediary quaternion and this helps to cope the unwinding phenomenon presented when working with the classical quaternions. The proposed control algorithm is validated numerically and experimentally when the quadrotor follows a circular trajectory. In addition, during the simulation part, some external perturbations and white noise were added in order to test the robustness of the algorithm.**

I. Introduction

In the domain of aerial vehicles, the Euler angles are commonly used for representing the orientation of a vehicle, but working with them involves problems of singularity for some specific angle values. Another possible parameterization could be the rotation matrix, but its use implies more computational cost because there are nine values that change as its rotation axis and angle changes. Also, during the design of an attitude control algorithm, it is necessary to work with vectors instead of matrices and this reason will make necessary to use another equivalent to the rotation matrix. One third option would be the use of quaternions because they do not have the singularity issue presented when working with the Euler angles and also they has a lower computational cost compared to that of rotation matrices. Therefore, the use of quaternions seems to be a good option but because the quaternions are not a complete representation as a rotation matrix, there are some aspects to have into consideration. For example, there are always two quaternions that can indicate one specific orientation and this fact leads to a drawback known as the unwinding phenomenon. This phenomenon consists in taking a longer attitude path instead of the shortest one and this leads to undesirable behaviors, particularly when the vehicle is in an environment with disturbances.

*This work was supported by the CONACYT (Consejo Nacional de Ciencia y Tecnología) of the Mexican government, LabEx PERSYVAL-Lab (ANR-11-LABX-0025), Equipex ROBOTEX (ANR-10-EQPX-44-01) in France

J. Colmenares Vázquez, N. Marchand, J.E. Gomez Balderas are with Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France. josue.colmenares-vazquez, nicolas.marchand, jose-ernesto.gomez-balderas, @gipsa-lab.fr

P. Castillo is with Sorbonne Universités, Université de Technologie de Compiègne, CNRS, France castillo@hds.utc.fr

To cope this problem, it is needed to choose the convenient quaternion at each instant during the phase of control design. Then, it will be significant to question if it is possible to use another parameterization similar to the quaternions but without their drawbacks. The intermediary quaternions described in [1], [2] could be used as an alternative to the classical ones. Nowadays, most of the inertial systems can provide the rotation matrix for determining the orientation of a body and taking advantage of this fact, the intermediary quaternions can be easily obtained from this rotation matrix. Moreover, to one specific orientation corresponds only one of theses quaternions. On the contrary, the computation of the classical quaternions from a rotation matrix have more than one expression and each one satisfies some criteria in order to be chosen. This motivates the use of the intermediary quaternions and the way to adapt them in the control design of a trajectory following.

In this work, it is considered a mathematical model of an ideal quadrotor obtained by the Euler-Newton approach, see [3], [4] for more details. This model does not take into account the asymmetries in the geometry of the drone, nor the fact the helices are flexible neither the dynamics of the motors. Thus, the model is simple and is similar to those in [5], [6] but not equal because these quaternions take into account the full rotation angle instead of the half angle considered in the classical quaternions. With regard to the control algorithms, there exist several ones in the literature, for instance, and just to mention a few, there are the ones based on saturations, sliding modes or on backstepping techniques and some of these are treated in [7]–[11].

The contribution of this work is the implementation of the intermediary quaternions in a backstepping technique with integral properties in order to design a control algorithm for performing an autonomous quadcopter navigation with circular path tracking. The mainly criterion for choosing the backstepping technique was its well-known robustness and, additionally, this is complemented by the integral property which helps to reduce the effects of the dynamics not taken into account in the ideal model and also the effects of the disturbances. The control law will be free of the unwinding phenomenon and will be validated by the tracking of a circular trajectory.

II. Preliminaries

A. Intermediate Quaternion

The rotation matrix $\mathbb{R}$ can also be expressed in function of its axis $\boldsymbol{\sigma}$ and angle ϑ of rotation. That is,

$$\mathbb{R} = \cos\vartheta\, I + (1-\cos\vartheta)\, \boldsymbol{\sigma}\boldsymbol{\sigma}^T + \sin\vartheta\, [\boldsymbol{\sigma}]^\times \qquad (1)$$

where $[\boldsymbol{\sigma}]^\times$ stands for the skew symmetric matrix and it is defined as:

$$[\boldsymbol{\sigma}]^\times = \begin{bmatrix} 0 & -\sigma_z & \sigma_y \\ \sigma_z & 0 & -\sigma_x \\ -\sigma_y & \sigma_x & 0 \end{bmatrix} \tag{2}$$

with $\boldsymbol{\sigma} = [\sigma_x, \sigma_y, \sigma_z]^T$. From this representation, a unitary quaternion can be formed as $\boldsymbol{q} = \cos\vartheta + \boldsymbol{\sigma}\sin\vartheta$. This quaternion is related to the classical unitary rotation quaternion. The difference resides on one takes the full angle and the classical one takes the half of the rotation angle. This quaternion is called intermediary because of its relation to the rotation one.

The intermediary quaternion can be written also as the sum of a real part and a vectorial one, i.e., $\boldsymbol{q} = q_r + \boldsymbol{q_v}$, and that sense, a vector could be seen as a quaternion with a zero real part. From Eqn. (1) it can be deduced the following expressions for these parts can be deduced

$$q_r = \cos\vartheta = \frac{1}{2}(tr(\mathbb{R}) - 1) \tag{3}$$

$$[\boldsymbol{q_v}]^\times = \sin\vartheta\,[\boldsymbol{\sigma}]^\times = \frac{1}{2}(\mathbb{R} - \mathbb{R}^T) \tag{4}$$

where $[\boldsymbol{q_v}]^\times$ is the skew symmetric matrix of the vector $\boldsymbol{q_v}$ as defined for $\boldsymbol{\sigma}$ in (2). The previous expressions let us easily to determine the intermediary quaternion. This quaternion are also related to the angular velocity of the body and taking into account that this quaternion takes the full angle, it results $\dot{\boldsymbol{q}} = \boldsymbol{q} \circ \boldsymbol{\omega}$; where $\circ$ means the multiplication between quaternions, and developing this last expression, it follows

$$\dot{q}_r = -\boldsymbol{q_v}^T\boldsymbol{\omega} \tag{5}$$

$$\dot{\boldsymbol{q}}_{\boldsymbol{v}} = q_r\boldsymbol{\omega} + \boldsymbol{q_v} \times \boldsymbol{\omega} \tag{6}$$

Now, supposing that a desired rotation matrix $\mathbb{R}_d$ is given, then the desired angular velocity satisfies

$$\frac{d}{dt}\mathbb{R}_d = \mathbb{R}_d[\boldsymbol{\omega}_d]^\times \implies [\boldsymbol{\omega_d}]^\times = \mathbb{R}_d^T\frac{d}{dt}\mathbb{R}_d \tag{7}$$

where $[\boldsymbol{\omega}_d]^\times$ means the skew symmetric matrix of the desired angular velocity. Alternatively, $\boldsymbol{\omega_d}$ can be rewritten in terms of quaternions as follows $\dot{\boldsymbol{q}}_d = \boldsymbol{q}_d \circ \boldsymbol{\omega}_d \implies \boldsymbol{\omega}_d = \bar{\boldsymbol{q}}_d \circ \dot{\boldsymbol{q}}_d$; with $\bar{\boldsymbol{q}}$ as the conjugated quaternion of $\boldsymbol{q}$. This becomes useful because this lets compute $\boldsymbol{\omega}_d$ faster than the Eqn. (7). In the same way, the derivative of this desired angular velocity is given by

$$\dot{\boldsymbol{\omega}}_d = \|\boldsymbol{\omega}_d\|^2 + \bar{\boldsymbol{q}}_d \circ \ddot{\boldsymbol{q}}_d \tag{8}$$

with $\|\boldsymbol{\omega}_d\|$ stands for the magnitude of $\boldsymbol{\omega}_d$. More references see [1], [2].

B. Mathematical Model

Our mathematical model is based on the Newton Euler approach and it is given by

$$\begin{aligned} m\ddot{\boldsymbol{r}} &= \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g} \\ \dot{\boldsymbol{q}} &= \boldsymbol{q} \circ \boldsymbol{\omega} \\ \mathbb{J}\dot{\boldsymbol{\omega}} &= \boldsymbol{\tau} - [\boldsymbol{\omega}]^\times\mathbb{J}\boldsymbol{\omega} \end{aligned} \tag{9}$$

The bold letters represent vectors or quaternions. $\boldsymbol{F} = [0, 0, f]^T$ means the thrust generated by the helices where f represents the magnitude of the thrust, $\boldsymbol{F_g}$ indicates the gravity force. $\boldsymbol{q}$ stands for the intermediary quaternion, $\boldsymbol{\omega}$ represents the angular velocity in the body frame. m stands for the drone mass and $\boldsymbol{r}$ defines the position of the mass center in a inertial frame. $\mathbb{R}$ describes the rotation matrix generated in the zyx convention. $\mathbb{J}$ is the inertia matrix of the drone. $\boldsymbol{\tau}$ defines the torques applied to the vehicle. The parameters m and $\mathbb{J}$ are supposed to be known.

III. CONTROL ALGORITHMS

The control goal is to develop a control scheme based on the intermediary quaternion for path tracking.

A. Position Control

The position control law is based on the backstepping technique with integral properties which helps to compensate the uncertainties in the model and the asymmetries not considered in the model. Firstly, let us define the position error as $\boldsymbol{e_r} = \boldsymbol{r} - \boldsymbol{r}_d$, thus, $\dot{\boldsymbol{e}}_{\boldsymbol{r}} = \dot{\boldsymbol{r}} - \dot{\boldsymbol{r}}_d = \boldsymbol{v} - \dot{\boldsymbol{r}}_d$; where $\boldsymbol{r}_d$ means the reference position and $\boldsymbol{v}$ indicates the velocity. Next, let us propose a positive definite function in order to design a virtual velocity $\boldsymbol{v}^v$ that will ensure the convergence to the desired position

$$V_r = \frac{1}{2}\,\boldsymbol{\chi_1}^T\mathbb{K}_{ir}\boldsymbol{\chi_1} + \frac{1}{2}\,\boldsymbol{e_r}^T\boldsymbol{e_r} \tag{10}$$

where $\mathbb{K}_{ir}$ stands for a positive constant matrix and $\boldsymbol{\chi_1} = \int_0^t \boldsymbol{e_r}\; d\tau$ represents the integral property in the position model. Differentiating the expression for V_r, it results $\dot{V}_r = \boldsymbol{\chi_1}^T\mathbb{K}_{ir}\boldsymbol{e_r} + \boldsymbol{e_r}^T\dot{\boldsymbol{e}}_{\boldsymbol{r}}$ and taking the virtual velocity as:

$$\boldsymbol{v}^v = \dot{\boldsymbol{r}}_d - \mathbb{K}_{ir}\boldsymbol{\chi_1} - \mathbb{K}_r\boldsymbol{e_r} \tag{11}$$

it follows that $\dot{V}_r|_{\boldsymbol{v}=\boldsymbol{v}^v} = -\boldsymbol{e_r}^T\mathbb{K}_r\boldsymbol{e_r} \quad \leq 0 \quad \forall\, t \geq 0$ with $\mathbb{K}_r$ as a positive constant matrix. Now, let us define the velocity error $\boldsymbol{e_v} = \boldsymbol{v} - \boldsymbol{v}^v$, thus, $\dot{\boldsymbol{e}}_{\boldsymbol{v}} = \dot{\boldsymbol{v}} - \dot{\boldsymbol{v}}^v = \frac{1}{m}\boldsymbol{u} - \dot{\boldsymbol{v}}^v$ with

$$\boldsymbol{u} = \mathbb{R}\boldsymbol{F} + \boldsymbol{F_g} \tag{12}$$

Now, proposing the following positive definite function in order to make converge the position error to zero $V_L = V_r + \frac{1}{2}\boldsymbol{e_v}^T\boldsymbol{e_v}$ thus, $\dot{V}_L = \dot{V}_r + \boldsymbol{e_v}^T\dot{\boldsymbol{e}}_{\boldsymbol{v}}$ and taking into account that $\boldsymbol{v} = \boldsymbol{v}^v + \boldsymbol{e_v}$, the expression $\dot{V}_L$ can be written as $\dot{V}_L = -\boldsymbol{e_r}^T\mathbb{K}_r\boldsymbol{e_r} + \boldsymbol{e_r}^T\boldsymbol{e_v} + \boldsymbol{e_v}^T\dot{\boldsymbol{e}}_{\boldsymbol{v}}$, then by choosing,

$$\boldsymbol{u} = m\,(\dot{\boldsymbol{v}}^{\boldsymbol{v}} - \boldsymbol{e_r} - \mathbb{K}_v\boldsymbol{e_v}) \tag{13}$$

with $\mathbb{K}_v$ as a positive constant matrix, it yields

$$\dot{V}_L = -\boldsymbol{e_r}^T\mathbb{K}_r\boldsymbol{e_r} - \boldsymbol{e_v}^T\mathbb{K}_v\boldsymbol{e_v} \quad \leq 0 \quad \forall\, t \geq 0$$

Substituting Eqn. (11) into Eqn. (13), the expression for $\boldsymbol{u}$ becomes

$$\begin{aligned} \boldsymbol{u} = \quad & m(\,\ddot{\boldsymbol{r}}_d - (\mathbb{K}_r + \mathbb{K}_v)(\boldsymbol{v} - \dot{\boldsymbol{r}}_d) \\ & -(I + \mathbb{K}_{ir} + \mathbb{K}_v\mathbb{K}_r)\boldsymbol{e_r} - \mathbb{K}_v\mathbb{K}_{ir}\boldsymbol{\chi_1}\,) \end{aligned} \tag{14}$$

Note that $\boldsymbol{r_d}$ is the position reference and $\dot{\boldsymbol{r}}_{\boldsymbol{d}}$ and $\ddot{\boldsymbol{r}}_{\boldsymbol{d}}$ are the velocity and acceleration obtained from this reference. Besides, it is always possible to generate this $\boldsymbol{u}$ from a pair $(\mathbb{R}_d, \boldsymbol{F}_d)$, where $\mathbb{R}_d$ and $\boldsymbol{F_d}$ are the desired rotation matrix and desired force which will generate the control law defined by (14). Firstly, replacing $\boldsymbol{F}$ given in the body frame by the

desired thrust $\boldsymbol{F_d} = [0, 0, f_d]^T$ in Eqn. (12), it follows that $f_d = \|\boldsymbol{u} - \boldsymbol{F_g}\|$. Then, defining the desired rotation matrix as $\mathbb{R}_d = [\boldsymbol{R_{d1}}, \boldsymbol{R_{d2}}, \boldsymbol{R_{d3}}]$ where $\boldsymbol{R_{di}}$ is a column vector. From Eqn. (12) it can be deduced that

$$\boldsymbol{R_{d3}} = \begin{bmatrix} R_{d3,x} \\ R_{d3,y} \\ R_{d3,z} \end{bmatrix} = \frac{1}{f_d}(\boldsymbol{u} - \boldsymbol{F_g}) \tag{15}$$

Now, using the expression for the rotation matrix in the zyx convention, the expressions for $\boldsymbol{R_{d1}}$ and $\boldsymbol{R_{d2}}$ can be obtained

$$\boldsymbol{R_{d1}} = \begin{bmatrix} R_{d1,x} \\ R_{d1,y} \\ R_{d1,z} \end{bmatrix} = \begin{bmatrix} \cos\psi_d \cos\theta_d \\ \sin\psi_d \cos\theta_d \\ -\sin\theta_d \end{bmatrix} \tag{16}$$

$$\boldsymbol{R_{d2}} = \boldsymbol{R_{d3}} \times \boldsymbol{R_{d1}} \tag{17}$$

with

$$\cos\theta_d = \left(1 + \left[\frac{1}{R_{d3,z}}(R_{d3,x}\cos\psi_d + R_{d3,y}\sin\psi_d)\right]^2\right)^{-1} \tag{18}$$

$$\sin\theta_d = \cos\theta_d \left[\frac{1}{R_{d3,z}}(R_{d3,x}\cos\psi_d + R_{d3,y}\sin\psi_d)\right] \tag{19}$$

where ψ_d is the desired yaw angle. These equations are valid when $u_z \neq -mg$. In applications where $u_z > -mg$ the expressions for $\boldsymbol{R_{di}}$ always hold.

B. Attitude Control

The next step is to design an orientation control algorithm which follows the desired rotation matrix. In order to compensate the effect of the unknown dynamics in our simplified model, an integral part is added to the controller. First, the rotation error matrix is defined as

$$\mathbb{R}_d = \mathbb{R}\mathbb{R}_e \quad \Longrightarrow \quad \mathbb{R}_e = \mathbb{R}^T\mathbb{R}_d \tag{20}$$

This error matrix represents the orientation that the vehicle needs to rotate from its actual attitude in order to reach the desired attitude. From (20) and using (2) and (4), the error intermediate quaternion is computed as

$$q_{e,r} = \frac{1}{2}\left(\, tr(\mathbb{R}_e) - 1 \,\right) \tag{21}$$

$$\boldsymbol{q_{e,v}} = \frac{1}{2}\begin{bmatrix} \mathbb{R}_{e3,2} - \mathbb{R}_{e2,3} \\ \mathbb{R}_{e1,3} - \mathbb{R}_{e3,1} \\ \mathbb{R}_{e2,1} - \mathbb{R}_{e1,2} \end{bmatrix} \tag{22}$$

The control goal is to make converge $\mathbb{R}_e \to I$ or equivalently $\boldsymbol{q_e} \to 1 + \boldsymbol{0}$. Proposing the following positive function

$$\begin{aligned} V_q &= \tfrac{1}{2}(1 - \boldsymbol{q_e}) \circ (1 - \bar{\boldsymbol{q}}_{\boldsymbol{e}}) + \tfrac{1}{2}\,\boldsymbol{\chi_2}^T\mathbb{K}_{iq}\boldsymbol{\chi_2} \\ &= \tfrac{1}{2}(1 - q_{e,r})^2 + \tfrac{1}{2}\boldsymbol{q_{e,v}^T q_{e,v}} + \tfrac{1}{2}\,\boldsymbol{\chi_2}^T\mathbb{K}_{iq}\boldsymbol{\chi_2} \end{aligned} \tag{23}$$

with $\mathbb{K}_{iq}$ as a positive constant diagonal matrix and $\boldsymbol{\chi_2} = \int_0^t \boldsymbol{q_{e,v}}\, d\tau$, and differentiating (23), it follows

$$\dot{V}_q = (1 - q_{e,r})(-\dot{q}_{e,r}) + \boldsymbol{q_{e,v}^T}\dot{\boldsymbol{q}}_{\boldsymbol{e,v}} + \boldsymbol{\chi_2}^T\mathbb{K}_{iq}\boldsymbol{q_{e,v}} \tag{24}$$

then by using Eqns. (5) and (6), it yields $\dot{V}_q = \boldsymbol{q_{e,v}^T}\boldsymbol{\omega_e} + \boldsymbol{q_{e,v}^T}\mathbb{K}_{iq}\boldsymbol{\chi_2}$, where $\boldsymbol{\omega_e}$ is the error angular rate that satisfies

$$\frac{d}{dt}\mathbb{R}_e = \mathbb{R}_e[\boldsymbol{\omega_e}]^\times \tag{25}$$

In order to find the relationship between $\boldsymbol{\omega_e}$, $\boldsymbol{\omega}$ and $\boldsymbol{\omega_d}$, the Eqn. (20) is differentiated and by using $\dot{\mathbb{R}} = \mathbb{R}[\boldsymbol{\omega}]^\times$ and the Eqns. (7) and (25), it results

$$[\boldsymbol{\omega_e}]^\times = [\boldsymbol{\omega_d}]^\times - \mathbb{R}_e^T[\boldsymbol{\omega}]^\times\mathbb{R}_e \tag{26}$$

An alternative vectorial relationship can be deduced for these angular rates. For that purpose, considering Figure 1, it is observed that the angular rates belong to two different systems. Taking into account that $\mathbb{R}_e[\boldsymbol{\omega}]^\times\mathbb{R}_e$ is the equivalent matrix in the system S_d^B to the matrix $[\boldsymbol{\omega}]^\times$ in the system S^B, the following vectorial relationship is deduced

$$\boldsymbol{\omega_e} = \boldsymbol{w_d} - \mathbb{R}_e^T\boldsymbol{\omega} \tag{27}$$

Then, let us propose a virtual angular rate error which makes converge the vehicle to the desired attitude. Thus, $\boldsymbol{\omega_e^v} = -\mathbb{K}_q\boldsymbol{q_{e,v}} - \mathbb{K}_{iq}\boldsymbol{\chi_2}$ where $\mathbb{K}_q$ is a constant positive diagonal matrix used for tuning the algorithm. This virtual velocity leads to

$$\dot{V}_q|_{\boldsymbol{\omega_e} = \boldsymbol{\omega_e^v}} = -\boldsymbol{q_{e,v}^T}\mathbb{K}_q\boldsymbol{q_{e,v}} \leq 0 \tag{28}$$

Observe from Eqn. (28) that there are two possible equilibrium points, that is, $\boldsymbol{q_{e1}} = -1 + \boldsymbol{0}$ or $\boldsymbol{q_{e2}} = 1 + \boldsymbol{0}$. Suppose the dynamical system starts at $\boldsymbol{q_e} = -1 + \boldsymbol{0}$. Then, a small perturbation is added and the system gets out from its equilibrium. Also, note that (23) is a non-increasing function because of (28). This prevent the system to return to this equilibrium point. Therefore, the system will converge to the global minimum of the (23), that is, $\boldsymbol{q_e} \to \boldsymbol{q_{e2}} = 1 + \boldsymbol{0}$.

Now, it is necessary to make converge $\boldsymbol{\omega_e} \to \boldsymbol{\omega_e^v}$. From (27), it results

$$\dot{\boldsymbol{\omega}}_{\boldsymbol{e}} = \dot{\boldsymbol{\omega}}_{\boldsymbol{d}} - \mathbb{R}_e^T\dot{\boldsymbol{\omega}} + \boldsymbol{\omega_e} \times \mathbb{R}_e^T\boldsymbol{\omega} \tag{29}$$

and introducing the dynamics of $\boldsymbol{\omega}$ from (9) into (29), it yields $\dot{\boldsymbol{\omega}}_{\boldsymbol{e}} = \dot{\boldsymbol{\omega}}_{\boldsymbol{d}} - \mathbb{R}_e^T\mathbb{J}^{-1}(\,\boldsymbol{\tau} - [\boldsymbol{\omega}]^\times\mathbb{J}\boldsymbol{\omega}\,) + \boldsymbol{\omega_e} \times \mathbb{R}_e^T\boldsymbol{\omega}$. Then, proposing an extended positive function $V_{q\omega} = V_q + \frac{1}{2}\boldsymbol{e_{\omega_e}^T e_{\omega_e}}$, with $\boldsymbol{e_{\omega_e}} = \boldsymbol{\omega_e} - \boldsymbol{\omega_e^v}$. Therefore, $\dot{V}_{q\omega} = -\boldsymbol{q_{e,v}^T}\mathbb{K}_q\boldsymbol{q_{e,v}} + \boldsymbol{q_{e,v}^T e_{\omega_e}} + \boldsymbol{e_{\omega_e}^T}\dot{\boldsymbol{e}}_{\boldsymbol{\omega_e}}$. Thus, if $\boldsymbol{\tau}$ is taken as

$$\boldsymbol{\tau} = [\boldsymbol{\omega}]^\times\mathbb{J}\boldsymbol{\omega} + \mathbb{J}\mathbb{R}_e\left(\boldsymbol{\omega_e} \times \mathbb{R}_e^T\boldsymbol{\omega} + \dot{\boldsymbol{\omega}}_{\boldsymbol{d}} - \tilde{\boldsymbol{\tau}}\right) \tag{30}$$

it yields $\dot{\boldsymbol{e}}_{\boldsymbol{\omega_e}} = \dot{\boldsymbol{\omega}}_{\boldsymbol{e}} - \dot{\boldsymbol{\omega}}_{\boldsymbol{e}}^{\boldsymbol{v}} = \tilde{\boldsymbol{\tau}} - \dot{\boldsymbol{\omega}}_{\boldsymbol{e}}^{\boldsymbol{v}}$ and by choosing $\tilde{\boldsymbol{\tau}} = \dot{\boldsymbol{\omega}}_{\boldsymbol{e}}^{\boldsymbol{v}} - \boldsymbol{q_{e,v}} - \mathbb{K}_{\omega_e}\boldsymbol{e_{\omega_e}}$, then $\dot{V}_{q\omega}$ becomes

$$\dot{V}_{q\omega} = -\boldsymbol{q_{e,v}^T}\mathbb{K}_q\boldsymbol{q_{e,v}} - \boldsymbol{e_{\omega_e}^T}\mathbb{K}_{\omega_e}\boldsymbol{e_{\omega_e}} \leq 0 \tag{31}$$

with $\mathbb{K}_{\omega_e}$ as constant positive diagonal matrix. Eqn. (31) implies the system attitude is stable in the Lyapunov sense.

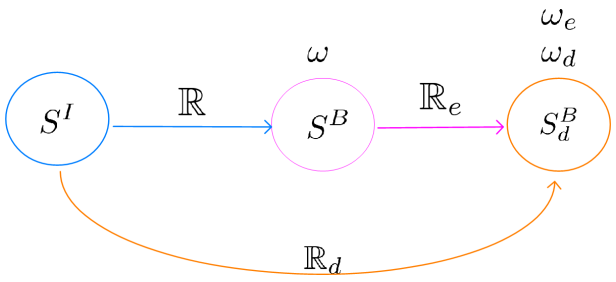


Fig. 1: Systems generated by the rotation matrices. S^I is the inertial system, S^B represents the body system and S_d^B is generated by the desired body orientation.

Therefore, $\boldsymbol{\omega_e} \to 0$ and $q_e \to 1 + \mathbf{0}$. This as well implies $\mathbb{R}_e \to I$ and $\boldsymbol{\omega} \to \mathbb{R}_e \boldsymbol{\omega_d}$ which in turn leads to $\boldsymbol{\omega} \to \boldsymbol{\omega_d}$. After some substitutions, the complete expression for $\boldsymbol{\tau}$ becomes

$$\begin{aligned}\boldsymbol{\tau} = \quad & [\boldsymbol{\omega}]^{\times}\mathbb{J}\boldsymbol{\omega} + \mathbb{J}\mathbb{R}_e\{\, \boldsymbol{\omega_e} \times \mathbb{R}_e^T\boldsymbol{\omega} + \dot{\boldsymbol{\omega}}_{\boldsymbol{d}} \\ & +(q_{e,r}\mathbb{K}_q + \mathbb{K}_{\omega_e})\boldsymbol{\omega_e} + (I + \mathbb{K}_{iq} + \mathbb{K}_{\omega_e}\mathbb{K}_q)\boldsymbol{q_{e,v}} \\ & \mathbb{K}_q(\boldsymbol{q_{e,v}} \times \boldsymbol{\omega_e}) + \mathbb{K}_{iq}\mathbb{K}_{\omega_e}\boldsymbol{\chi_2}\,\}\end{aligned} \tag{32}$$

C. The unwinding phenomenon

The intermediate quaternion $\boldsymbol{q_e} = -1 + \mathbf{0}$ corresponds to the maximum orientation error. This happens when the maximum rotation angle associated to $\mathbb{R}_e$ is $180°$. This means that the vehicle does not rotate more than $180°$ between two given orientations which implies the UAV always takes the shorter path. Now, supposing a perturbation is applied to the vehicle in order to get out from the attitude equilibrium point, the orientation control algorithm will force the vehicle to recover its equilibrium $\boldsymbol{q_e} = 1 + \mathbf{0}$. Thus, with this attitude algorithm the UAV will not undergo the unwinding phenomenon.

IV. SIMULATIONS RESULTS

In order to provide a physical sense, the orientation is presented in Euler angles, but the simulations are carried out using the intermediary quaternions. Two simulation results are presented in this section, the first one is in order to show the vehicle takes the shortest path between two given orientations, for example if the vehicle needs to go from $(\psi_0, \theta_0, \phi_0) = (180, 0, 0)°$ to $(\psi_d, \theta_d, \phi_d) = (-170, -40, 40)°$ then the vehicle will arrive to $(\psi_{f1}, \theta_{f1}, \phi_{f1}) = (190, -40, 40)°$, another possibility could be a longer path arriving to $(\psi_{f2}, \theta_{f2}, \phi_{f2}) = (190, -40, -320)°$. In practice, the two final angular positions represent the same desired orientation, but the difference resides in the length of the followed path. The second simulation validates the control algorithm when the vehicle follows a circular trajectory. The parameters used in the attitude algorithm are

$$\begin{aligned} \mathbb{K}_q &= \text{diag}(32, 32, 32) & \mathbb{J}_{xx} &= 0.008\ kg \cdot m^2 \\ \mathbb{K}_{\omega_e} &= \text{diag}(128, 128, 128) & \mathbb{J}_{yy} &= \mathbb{J}_{xx} \\ \mathbb{K}_{iq} &= \text{diag}(32, 32, 32) & \mathbb{J}_{zz} &= 0.006\ kg \cdot m^2 \end{aligned}$$

For the first simulation, the initial condition was $(\psi_0, \theta_0, \phi_0) = (180°, 0°, 0°)$ while the desired orientations were $(\psi_d, \theta_d, \phi_d) = (\pm 170°, -40°, 40°)$. These orientations are rewritten as rotation matrices in order to apply the proposed algorithm. Figure 2 shows the trajectories followed when two attitude control laws are applied, one law is based on the intermediate quaternion and the other one on the classical quaternion. Furthermore, the two algorithms use the backstepping technique. Figure 3 confirms the intermediate quaternion algorithm takes the shorter path while the other one takes a longer one.

Note that in Fig. 3 the quadcopter attains different Euler angles, but in practice, these angles represent the same desired orientation. For these two previous figures, the rotation error matrices converge to the identity matrix in both cases. In the second simulation, our goal is to validate the proposed algorithm when a quadcopter performs a circular path tracking. The parameters used for the position control

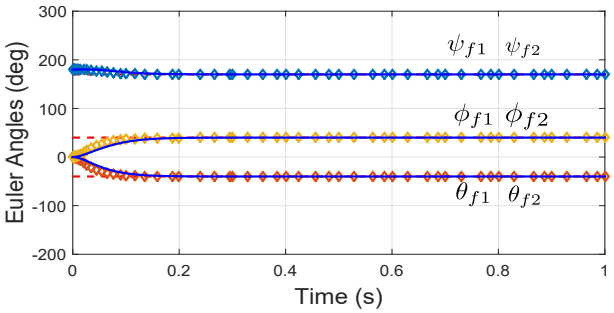


Fig. 2: Orientation given by Euler angles in xyz convention. Dotted lines represent the reference, diamond lines the responses when using the intermediate quaternion algorithm while solid lines are for the classical quaternion algorithm.

law are $\mathbb{K}_r = \text{diag}(16, 16, 12), \mathbb{K}_v = \text{diag}(4, 4, 2), \mathbb{K}_{ir} = \text{diag}(2, 2, 0.05), m = 0.260\ kg$.

The initial position is placed at $(x_0, y_0, z_0) = (0.7, 0, 0)\ m$ and the initial orientation is $(\psi_0, \theta_0, \phi_0) = (180°, 0°, 0°)$. The desired path is generated by a circular trajectory at a rate of $1.6\ rad/s$ with the vehicle front pointing to the circle center. The center is placed at $(0, 0, 0.5)\ m$ and the radius is $0.7\ m$. In order to test the robustness of this algorithm, a constant torque $(0.125, 0.075, -0.125)N \cdot m$ and a constant force $(0.05, 0.03, -0.05)N$ were added as external perturbations. In addition, a white noise was added to the position and orientation. The results are shown in Fig. 6 when the vehicle follows the desired trajectory with its front always pointing to the circle center. The desired orientation generated by the position control law is followed by the vehicle and the quaternion error $\boldsymbol{q_e} \to 1 + \mathbf{0}$. Note that the pitch angle takes a constant value in order to counteract the centrifugal force when it is turning.

V. PLATFORM, PROTOTYPE AND EXPERIMENTAL RESULTS

A. Platform and prototype

The experiments were carried out in the MOCA room, c.f. [12]. This room has a VICON vision system, the software VICON Tracker lets obtain the position and the orientation of the vehicle at a rate of 100 hz. This capture is made by means of markers placed on the vehicle. The data are used in the position algorithm and this will compute the desired orientation and the convenient thrust. The orientation and thrust are sent to

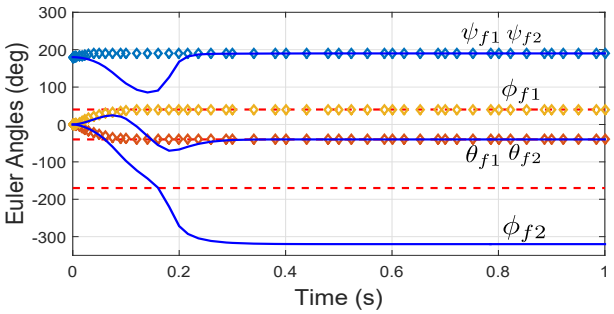


Fig. 3: Orientation given by Euler angles. Dotted lines are the references, diamond lines the intermediate quaternions and solid lines the classical quaternions.

the vehicle via DSMX radio. The vehicle executes the attitude algorithm based on intermediate quaternions and computes the required speed in each motor in order to move to the desired position. This process is described in Fig. 4.

The prototype used in the flight test is shown in Fig. 5 and its parameters are: mass = 0.260 kg; payload = 0.120 kg; length = 0.30 m; height = 0.15 m; helix diam = 0.125 m; battery = 1200 mAh 2S 7.4V 30C Li-Po and motor Brushless 28000 kv 13 gr. It uses a Crius One Pro V2 as a flight controller and its structure was designed and printed in the Gipsa Lab workshop.

B. Experiment results

The experiment consists in testing the developed algorithm in the tracking of a circular path at a rate of $1.6\ rad/s$ using a quadcopter vehicle. It is desired the vehicle front points to the circle center while it makes the following. The circle center is placed at $(0, 0, 0.5)\ m$ and its radius is $0.7\ m$. First, the UAV will take-off slowly and at the same time it will follow the desired circular path. The code for the position control algorithm is generated by Simulink of MATLAB® with a sample time of $0.01\ s$. This algorithm generates the desired orientation that is sent to the drone via a DSMX transmitter. Then, the embedded card will execute the attitude algorithm. The results of this experiment are shown in the Fig. 7. Observe that the results correspond to the expected values obtained in simulation. The Fig. 7(a) shows the vehicle follows the desired circular trajectory. Also, Fig. 7(b) demonstrates the orientation error quaternion converges to $\boldsymbol{q_e} = 1 + \mathbf{0}$. Fig. 7(c) confirms the pitch and roll angles agrees with their corresponding ones observed in simulation and Fig. 7(d) describes the yaw angle evolution when the vehicle is pointing to the circular center.

VI. CONCLUSIONS

The experimental and simulation results have validated the control algorithm developed in this work when a quadcopter vehicle performs a circular path tracking. Moreover, the robustness of this algorithm was proven in simulation by adding external perturbations and noise to the system dynamics. The convergence of the vehicle attitude to those produced by the attitude control law was confirmed in the graphic corresponding to $\boldsymbol{q_e}$. Also, this algorithm based on the intermediary quaternion has eliminated the concern of the unwinding phenomenon.

As a part of the future work, in the experimental part, it is envisaged to attain a greater turning speed and to reduce numerical computation by using vectors for finding the quadcopter orientation.

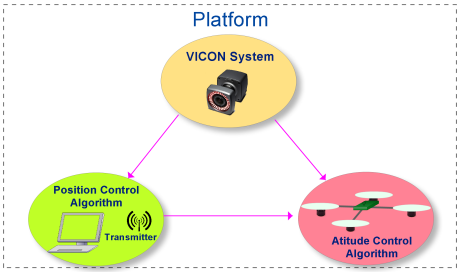


Fig. 4: Platform used in the experiments.

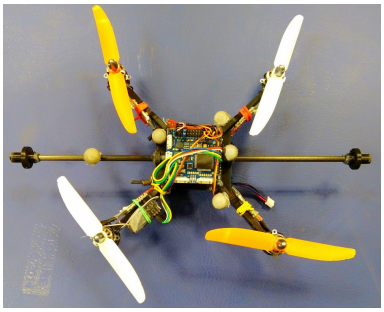


Fig. 5: Prototype used in the experiments.

ACKNOWLEDGMENT

The authors would like to thank J.J. Téllez Guzmán, J.U. Alvarez Muñoz and J. Dumon for their help during the experiments in the MOCA room.

REFERENCES

[1] Y. Song and W. Cai, "New intermediate quaternion based control of spacecraft: Part i - almost global attitude tracking," in *International Journal of Innovative Computing Information and Control*, vol. 8, Oct 2012, pp. 7307–7319.

[2] W. Cai and Y. Song, "New intermediate quaternion based control of spacecraft: Part ii - global attitude tracking," in *International Journal of Innovative Computing Information and Control*, vol. 8, Nov 2012, pp. 7853–7864.

[3] P. Castillo, R. Lozano and A. E. Dzul, *Modelling and Control of Mini-Flying Machines*. London: Springer-Verlag, 2005, ch. 2,3.

[4] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 20–32, Sept 2012.

[5] J. Cariño, H. Abaunza, and P. Castillo, "Quadrotor quaternion control," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, June 2015, pp. 825–831.

[6] H. Abaunza, P. Castillo, R. Lozano, and A. Victorino, "Quadrotor aerial manipulator based on dual quaternions," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2016, pp. 152–161.

[7] A. Hably and N. Marchand, "Global stabilization of a four rotor helicopter with bounded inputs," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, Oct 2007, pp. 129–134.

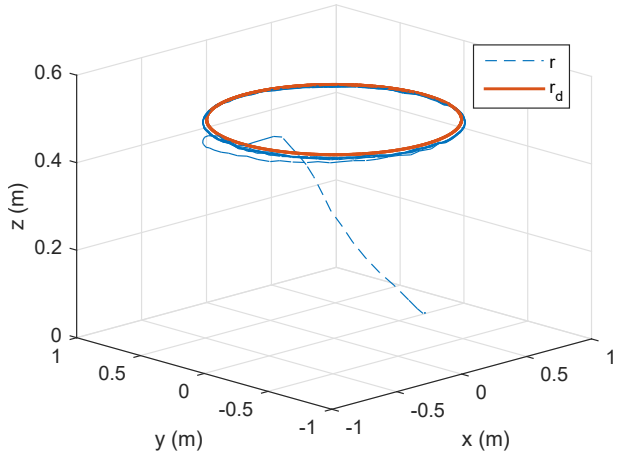
[8] J. U. Alvarez-Muñoz, N. Marchand, J. F. Guerrero-Castellanos, A. E. López-Luna, J. J. Téllez-Guzmán, J. Colmenares-Vázquez, S. Durand, J. Dumon, and G. Hasan, "Nonlinear control of a nano-hexacopter carrying a manipulator arm," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 4016–4021.

[9] H. Liu and X. Wang, "Quaternion-based robust attitude control for quadrotors," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, June 2015, pp. 920–925.

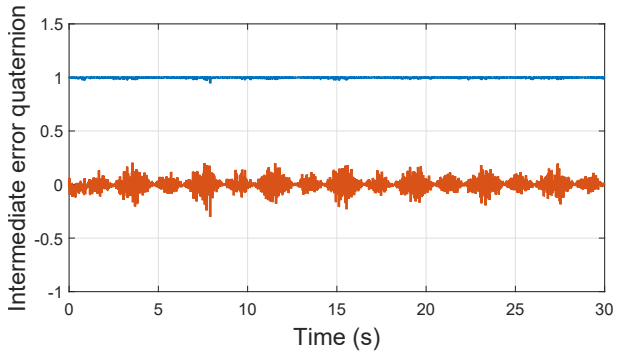
[10] S. D. Lucia, G. D. Tipaldi, and W. Burgard, "Attitude stabilization control of an aerial manipulator using a quaternion-based backstepping approach," in *Mobile Robots (ECMR), 2015 European Conference on*, Sept 2015, pp. 1–6.

[11] E. Reyes-Valeria, R. Enriquez-Caldera, S. Camacho-Lara, and J. Guichard, "Lqr control for a quadrotor using unit quaternions: Modeling and simulation," in *Electronics, Communications and Computing (CONIELECOMP), 2013 International Conference on*, March 2013, pp. 172–178.

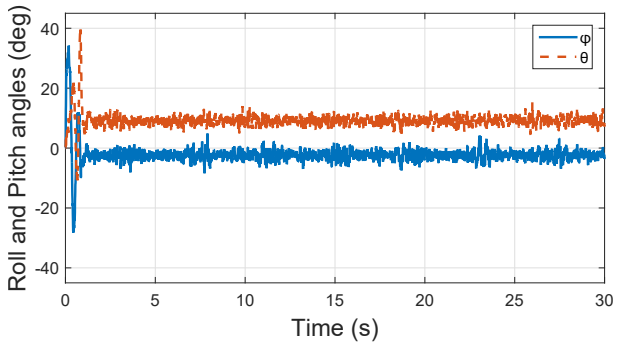
[12] Moca. (2016) MOCA Platform at Gipsa-Lab. [Online]. Available: "http://www.gipsa-lab.grenoble-inp.fr/recherche /plates-formes.php?id_plateforme=79"



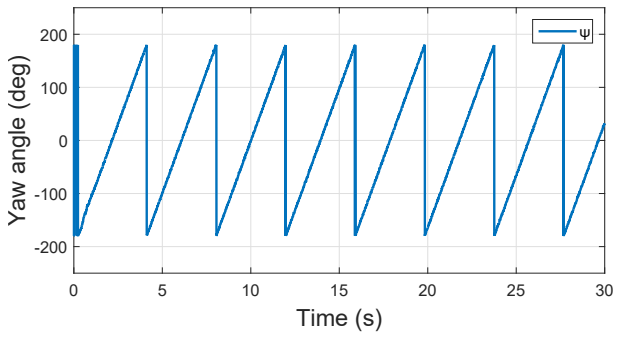
(a) Path described by the UAV in simulation when following the desired trajectory. .



(b) Error intermediate quaternion. The convergence $\boldsymbol{q_e} \to 1+\mathbf{0}$ ensures the vehicle takes the desired orientation.

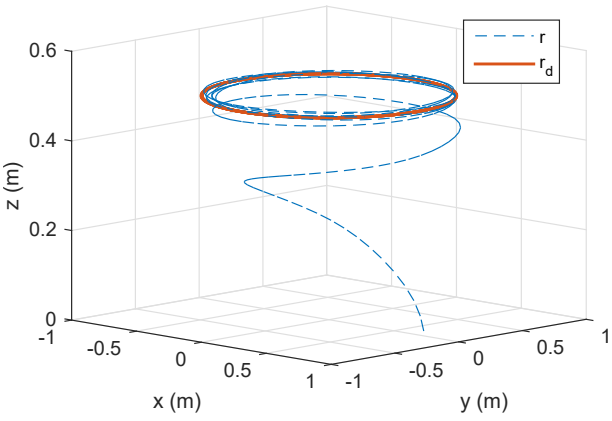


(c) Behavior of roll and pitch.

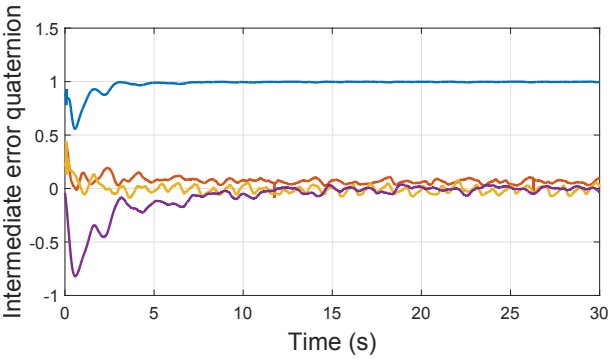


(d) The yaw angle is varying according to the vehicle position.

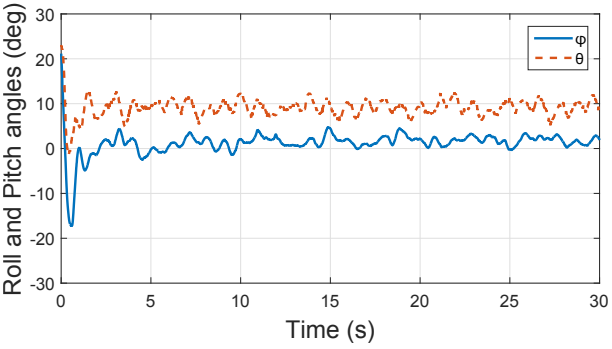
Fig. 6: Simulation results. The vehicle follows a circular trajectory at a angular speed of $1.6\ rad/s$ with its front pointing to the center.



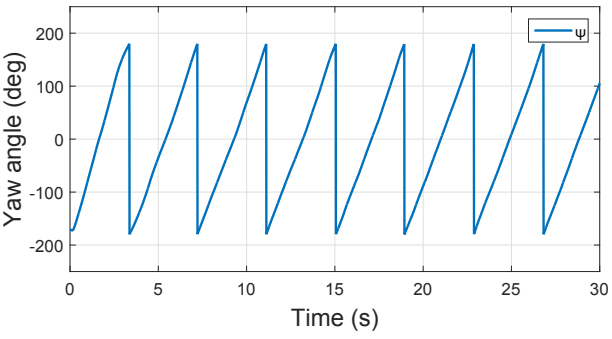
(a) Path described by the UAV during the experiment.



(b) Error intermediate quaternion. The quaternion is close to $\boldsymbol{q_e} \to 1+\mathbf{0}$



(c) Pitch and roll angles agrees with the behavior observed in simulation.



(d) Yaw angle when the vehicle is turning around the circle center.

Fig. 7: Experimental results. Followed paths when the vehicle follows a circular trajectory at $1.6\ rad/s$ centered at $(0, 0, 0.5)\ m$ and a radius of $0.7\ m$