Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Trajectory Tracking for a Multicopter under a Quaternion Representation ⋆

Huu Thien Nguyen * **Ngoc Thinh Nguyen** ** **Ionela Prodan** ***
Fernando Lobo Pereira *

* *Research Unit SYSTEC - Research Center for Systems and Technologies, hosted at Faculty of Engineering, Porto University, and at Institute for Systems and Robotics, Porto. E-mail: huu-thien.nguyen@outlook.com, flp@fe.up.pt*
** *Univ. of Luebeck, Institute for Robotics and Cognitive Systems, Luebeck, Germany. Email: nguyen@rob.uni-luebeck.de*
*** *Univ. Grenoble Alpes, Grenoble INP †, LCIS, Valence, France. E-mail: ionela.prodan@lcis.grenoble-inp.fr.*
† *Institute of Engineering and Management Univ. Grenoble Alpes.*

Abstract: This paper proposes a two-layer hierarchical control scheme for trajectory tracking of a multicopter system using attitude quaternion. We first present the differential flatness properties of the system and then, exploit them to design the feedback linearization laws for the position controller at the high control level. Next, the computed-torque control method is employed for stabilizing the attitude quaternion. The whole control scheme is illustrated through simulations while the position controller is further tested under experiments over a nano-drone quadcopter platform.

Keywords: Quaternion; Trajectory tracking; Differential flatness; Feedback linearization; Computed torque control; Multicopter unmanned vehicle

1. INTRODUCTION

Multicopters (also referred to as multirotor aerial vehicles) are receiving a growing interest in both research and industry communities due to their broad range of aerial applications such as target tracking, package delivery, and structural inspection (Floreano and Wood, 2015). In most of the applications involving multicopter systems, trajectory tracking is an indispensable control task and hence, the number of related research works has substantially been increasing since the early 2000s (Nascimento and Saska, 2019). A popular solution is to decouple the multicopter dynamics by employing a hierarchical two-layer control scheme (Mellinger and Kumar, 2011; Formentin and Lovera, 2011), which not only guarantees stability by singular perturbation theory (Nascimento and Saska, 2019), but is also convenient for autonomous trajectory tracking task on real multicopters.

Regarding the control designs within each layer, we can classify the existing approaches by their attitude representations: i) Euler angles (Formentin and Lovera, 2011; Nguyen et al., 2017), ii) intrinsic rotation matrix (Mellinger and Kumar, 2011), and iii) quaternion (Carino et al., 2015; Johnson and Leang, 2014; Alaimo et al., 2013; Fresk and Nikolakopoulos, 2013). Between the three of them, using Euler angles (roll, pitch, yaw) is the most popular method due to its straightforward interpretation (e.g., the vehicle's direction is measured by the yaw angle). The Euler angles representation suits various control designs such as PID, LQR, Lyapunov-based control, feedback linearization, Model Predictive Control, etc., as pointed out by Nascimento and Saska (2019), and the references therein. However, some serious drawbacks of this representation are the presence of singularities that can occur in some specific circumstances (e.g., when the vehicle rotates up to a perpendicular position), and the "gimbal lock", which happens when two axes of rotation align together, and the quadcopter loses a degree of freedom (Diebel, 2006). Moreover, using Euler angles requires repeatedly calculating their trigonometric functions which are time-consuming and occupy significant computational resources (Fresk and Nikolakopoulos, 2013).

In order to avoid the aforementioned issues, a feasible solution is to employ the attitude quaternion. Various control methods are applied for the quaternion-based dynamics of the multicopters, such as PD, LQR, backstepping, and feedback linearization (Chovancová et al., 2016; Johnson and Leang, 2014; Alaimo et al., 2013; Fresk and Nikolakopoulos, 2013). Next, based on the unit norm property of the quaternion, Yang (2012) used his own proposed "reduced quaternion model" for a spacecraft by using only 3 quaternion components that always satisfy the norm constraint, in order to design an LQR controller. Carino et al. (2015) exploited the relation between quaternion and

⋆ This work has been done while the first author was preparing his Master project at Univ. Grenoble Alpes, Grenoble INP, LCIS, F-26000, Valence, France. Also, the work is financially supported by: SYSTEC R&D Unit UIDB+P/00147/2020 funded by FCT/MCTES (PIDDAC); Project STRIDE – NORTE-01-0145-FEDER-000033, funded by N2020, ERDF; and Project MAGIC PTDC/EEI-AUT/32485/2017 funded by FEDER-COMPETE2020 and POCI-FCT/MCTES.

axis-angle representation to derive a feedback linearization controller for the attitude, subsequently, they continued with a feedback linearization for the position controller.

Even though the foregoing works are technically sound, we believe that these contributions still lack several important considerations. E.g., the model is linearized around the hovering condition (Chovancová et al., 2016), and hence, under more challenging scenarios, the control performances cannot be guaranteed. Moreover, the unit quaternion norm constraint is not strictly examined (Johnson and Leang, 2014; Alaimo et al., 2013; Fresk and Nikolakopoulos, 2013), or is simply guaranteed by renormalization after each iteration (Carino et al., 2015), which probably introduces error into the system due to analytical drift through multiple integrations, and may eventually cause divergence and instability (Whitmore, 2000).

Therefore, we present in this paper, several contributions which can compensate for the drawbacks and provide significant advantages w.r.t. the aforementioned works:

- a complete flatness-based representation of the nonlinear multicopter dynamics using the attitude quaternion;
- a feedback linearization control law constructed based on the foregoing flatness property, able to linearize the multicopter translational dynamics without any simplifications;
- a computed-torque attitude controller that fully exploits the attitude quaternion and guarantees their unit norm property.

The remainder of the paper is organized as follows. Section 2 presents the mathematical model of a standard multicopter system using the attitude quaternion and its differential flatness property. Section 3 details the hierarchical control design for trajectory tracking. Next, the simulation and experimental results are given and discussed in Section 4. Finally, Section 5 provides conclusions and considers future works.

2. MULTICOPTER MODEL

In this section, we recapitulate the multicopter dynamics expressed by using the attitude quaternion as given in Carino et al. (2015); Chovancová et al. (2016). Next, the differential flatness properties of the system are derived from its mathematical model.

2.1 Rotation kinematics with quaternion

The attitude of the multicopter system is described by using a unit quaternion $q \triangleq \begin{bmatrix} q_0 & \boldsymbol{q}^\top \end{bmatrix}^\top \in \mathbb{R}^4$ in which, $q_0 \geq 0$ and $\boldsymbol{q} \triangleq [q_1 \ q_2 \ q_3]^\top \in \mathbb{R}^3$, all satisfy:

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1. \tag{1}$$

The orientation of the body frame attached to the vehicle w.r.t. the global frame fixed to the ground is given through the following rotation matrix (Chovancová et al., 2016):

$$R = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_0q_2+q_1q_3) \\ 2(q_1q_2+q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_0q_1+q_2q_3) & 1-2(q_1^2+q_2^2) \end{bmatrix}. \tag{2}$$

Also, the relation between the quaternion derivative and the multicopter angle rate is given by Carino et al. (2015):

$$\dot{q} = \frac{1}{2} q \otimes \widehat{\boldsymbol{\omega}}, \tag{3}$$

in which, $\widehat{\boldsymbol{\omega}} = \begin{bmatrix} 0 & \boldsymbol{\omega}^\top \end{bmatrix}^\top \in \mathbb{R}^4$ extends the angle rate vector $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^\top \in \mathbb{R}^3$ and the operation $\otimes$ is defined in Carino et al. (2015); Chovancová et al. (2016). Hereinafter, we provide explicitly the inverse formulation of (3) which will be employed in later analysis:

$$\begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = 2 \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}, \tag{4}$$

which can be more synthetically expressed by:

$$\boldsymbol{\omega} = Q(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{5}$$

with $\boldsymbol{q} = [q_1 \ q_2 \ q_3]^\top$ as in (1). Note that (5) is obtained by introducing the following relation to (4):

$$q_0 = \sqrt{1 - (q_1^2 + q_2^2 + q_3^2)}, \tag{6}$$

which follows from the unit norm (1).

2.2 Multicopter dynamics with the quaternion formulation

The translational dynamics of the multicopter system are obtained by using Newton's second law of motion (Formentin and Lovera, 2011; Nguyen et al., 2017):

$$m\ddot{\boldsymbol{\xi}} = m\boldsymbol{g} + R\boldsymbol{e}_z T, \tag{7}$$

with $\boldsymbol{\xi} \triangleq [x \ y \ z]^\top \in \mathbb{R}^3$, the position, $\boldsymbol{g} = [0 \ 0 \ -g]^\top$, the gravity pointing downward, m, the system mass, R the rotation matrix from (2), $\boldsymbol{e}_z \triangleq [0 \ 0 \ 1]^\top$ the unit vector of the z-axis, and $T \in \mathbb{R}_+$ the input thrust.
Next, the rotational dynamics of the multicopter are considered as those of a 3-dimensional rigid body and hence, are given by Carino et al. (2015); Nguyen et al. (2017):

$$J\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (J\boldsymbol{\omega}) = \boldsymbol{\tau} \tag{8}$$

with $J = \mathrm{diag}\{J_x, J_y, J_z\} \in \mathbb{R}^{3\times 3}$ the inertia tensor, $\boldsymbol{\omega}$ the angle rate as in (3), and $\boldsymbol{\tau} \triangleq [\tau_x \ \tau_y \ \tau_z]^\top \in \mathbb{R}^3$ gathering the input torques.
It will be more convenient to consider the matrix forms of the dynamics (7)–(8), which are explicitly given as follows:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{T}{m} \begin{bmatrix} 2(q_0q_2 + q_1q_3) \\ 2(q_2q_3 - q_0q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}, \tag{9}$$

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} (J_y - J_z)J_x^{-1}\omega_y\omega_z \\ (J_z - J_x)J_y^{-1}\omega_z\omega_x \\ (J_x - J_y)J_z^{-1}\omega_x\omega_y \end{bmatrix} + \begin{bmatrix} J_x^{-1}\tau_x \\ J_y^{-1}\tau_y \\ J_z^{-1}\tau_z \end{bmatrix}. \tag{10}$$

Remark 1. By comparing the quaternion-based rotation matrix (2) with the one resulted from the Euler angles approach detailed in Formentin and Lovera (2011); Nguyen et al. (2017), the relations between quaternion and Euler angles are as follows:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan \dfrac{2(q_0q_1 + q_2q_3)}{1 - 2(q_1^2 + q_2^2)} \\ \arcsin[2(q_0q_2 - q_1q_3)] \\ \arctan \dfrac{2(q_0q_3 + q_1q_2)}{1 - 2(q_2^2 + q_3^2)} \end{bmatrix}, \tag{11}$$

being (ϕ, θ, ψ), the roll, pitch and yaw angles, respectively. Similar results can also be found in Diebel (2006). □

2.3 Flatness-based representation

Differential flatness definition: Consider the nonlinear system in general form:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \tag{12}$$

with the state $\mathbf{x}(t) \in \mathbb{R}^n$ and input vectors $\mathbf{u}(t) \in \mathbb{R}^m$. The system (12) is differentially flat if there exists a flat output $\mathbf{z}(t) \in \mathbb{R}^m$ (Fliess et al., 1995):

$$\mathbf{z}(t) = \Upsilon(\mathbf{x}(t), \mathbf{u}(t), \dot{\mathbf{u}}(t), \cdots, \mathbf{u}^{(s)}(t)) \tag{13}$$

such that the states and inputs can be algebraically expressed in terms of $\mathbf{z}(t)$ and a finite number of its higher-order derivatives [1] :

$$\mathbf{x}(t) = \Upsilon_x(\mathbf{z}(t), \dot{\mathbf{z}}(t), \cdots, \mathbf{z}^{(r)}(t)), \tag{14a}$$
$$\mathbf{u}(t) = \Upsilon_u(\mathbf{z}(t), \dot{\mathbf{z}}(t), \cdots, \mathbf{z}^{(r+1)}(t)). \tag{14b}$$

Flatness properties of the multicopter:
The multicopter system (5)–(10) is differentially flat with the flat output $\mathbf{z} \in \mathbb{R}^4$ defined as follows:

$$\mathbf{z} \triangleq [x \ y \ z \ q_3]^\top, \tag{15}$$

with (x, y, z) the 3D position from (9) and q_3 the fourth element of the quaternion q from (1).
Hereinafter, we will show the flatness-based representation of the multicopter's states and inputs. First, from (9), the first three elements of the quaternion q as in (1) are expressed as follows:

$$q_0 = \Gamma_{q_0}(\ddot{x}, \ddot{y}, \ddot{z}, q_3, g), \tag{16a}$$
$$q_1 = \Gamma_{q_1}(\ddot{x}, \ddot{y}, \ddot{z}, q_3, g), \tag{16b}$$
$$q_2 = \Gamma_{q_2}(\ddot{x}, \ddot{y}, \ddot{z}, q_3, g), \tag{16c}$$

and the thrust T is obtained as:

$$T = m\sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z} + g)^2}, \tag{17}$$

with $[x \ y \ z \ q_3]$ the flat output chosen as in (15) and g the gravity. The explicit expressions for (16)–(17) are derived in the Appendix A. For further use, we shorten the descriptions (16)–(17) to $q_i = \Gamma_{q_i}(\ddot{\boldsymbol{\xi}}, q_3)$ with $i \in \{0, 1, 2\}$ and $T = \Gamma_T(\ddot{\boldsymbol{\xi}})$.
Next, by introducing (5) into (8), we obtain the input torque $\boldsymbol{\tau}$ as follows:

$$\boldsymbol{\tau} = J\,[Q(\boldsymbol{q})\ddot{\boldsymbol{q}} + D_{\boldsymbol{q}}[Q(\boldsymbol{q})\dot{\boldsymbol{q}}]\dot{\boldsymbol{q}}] + [Q(\boldsymbol{q})\dot{\boldsymbol{q}}] \times [JQ(\boldsymbol{q})\dot{\boldsymbol{q}}], \tag{18}$$

with $D_{\boldsymbol{q}}(\cdot)$ the Jacobian w.r.t. $\boldsymbol{q}$. (18) can be easily interpreted in the flat output space by introducing (16b)–(16c).

The flatness representations of the multicopter system as in (16)–(18) can be used for various applications such as constrained trajectory generation and control designs (Nguyen et al., 2018). In the next section, we will exploit them to design feedback linearization control laws.

3. TRACKING CONTROL DESIGN FOR THE MULTICOPTER

This section addresses the tracking control design for the multicopter system (9)–(10). We make use of the standard hierarchical two-layer control scheme illustrated in Fig. 1 which is usually employed in the literature (Carino et al., 2015; Nguyen et al., 2017). The position controller at the high control level compares the reference position $\boldsymbol{\xi}_r$ with the actual position $\boldsymbol{\xi}$ received as feedback from the system, in order to calculate the thrust T and the reference quaternion $q_r = \begin{bmatrix} q_{0_r} & \boldsymbol{q}_r{}^\top \end{bmatrix}^\top = [q_{0_r} \ q_{1_r} \ q_{2_r} \ q_{3_r}]^\top$. Then, the last three components of the latter term, $\boldsymbol{q}_r = [q_{1_r} \ q_{2_r} \ q_{3_r}]^\top$, are sent to the attitude controller at the low control level to track. Note that, the low control level is required to run at a higher frequency than the high-level to establish the closed-loop stability for the whole scheme (Nascimento and Saska, 2019).

3.1 Feedback linearization position controller

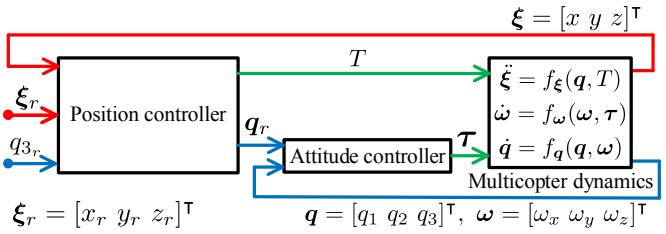

Fig. 1. Hierarchical control scheme of the multicopter.

For the position controller at the high-level, we employ the flat representations (16)–(17) to design the feedback linearization controller as follows:

$$q_{i_r} = \Gamma_{q_i}(\ddot{\boldsymbol{\xi}}^*, q_{3_r}), \tag{19a}$$
$$T = \Gamma_T(\ddot{\boldsymbol{\xi}}^*), \tag{19b}$$

with $i \in \{0, 1, 2\}$, q_{3_r} the reference of q_3 defined by the users and the corrective term $\ddot{\boldsymbol{\xi}}^* \triangleq [\ddot{x}^* \ \ddot{y}^* \ \ddot{z}^*]^\top$ given by:

$$\ddot{\boldsymbol{\xi}}^* = \ddot{\boldsymbol{\xi}}_r + K_{p_\xi}\boldsymbol{\epsilon}_\xi + K_{d_\xi}\dot{\boldsymbol{\epsilon}}_\xi + K_{i_\xi}\int \boldsymbol{\epsilon}_\xi dt, \tag{20}$$

with $\boldsymbol{\epsilon}_\xi = \boldsymbol{\xi}_r - \boldsymbol{\xi}$ and K_{p_ξ}, K_{d_ξ}, K_{i_ξ} are diagonal positive definite matrices from $\mathbb{R}^3$.

Proposition 2. By applying the controller (19a)–(19b) to (9), it follows that the closed loop linear system (21) is stable.

$$\ddot{\boldsymbol{\epsilon}}_\xi + K_{p_\xi}\boldsymbol{\epsilon}_\xi + K_{d_\xi}\dot{\boldsymbol{\epsilon}}_\xi + K_{i_\xi}\int \boldsymbol{\epsilon}_\xi dt = 0. \tag{21}$$

Proof. From (19), with $\ddot{z}^* \geq -g$, we obtain the following relations:

$$2(q_{0_r}q_{2_r} + q_{1_r}q_{3_r})T/m = \ddot{x}^*, \tag{22a}$$
$$2(q_{2_r}q_{3_r} - q_{0_r}q_{1_r})T/m = \ddot{y}^*, \tag{22b}$$
$$(q_{0_r}^2 - q_{1_r}^2 - q_{2_r}^2 + q_{3_r}^2)T/m - g = \ddot{z}^*, \tag{22c}$$
$$q_{0_r}^2 + q_{1_r}^2 + q_{2_r}^2 + q_{3_r}^2 = 1. \tag{22d}$$

From (22a)–(22c), the control law (19) drives the translational dynamics (9) to:

$$[\ddot{x} \ \ddot{y} \ \ddot{z}]^\top = [\ddot{x}^* \ \ddot{y}^* \ \ddot{z}^* \ ]^\top, \tag{23}$$

or,

$$\dot{\mathbf{x}} = A_{\mathbf{x}}\mathbf{x} + B_{\mathbf{x}}\ddot{\boldsymbol{\xi}}^*, \tag{24}$$

with $\mathbf{x} \triangleq \begin{bmatrix} \boldsymbol{\xi}^\top & \dot{\boldsymbol{\xi}}^\top \end{bmatrix}^\top \triangleq [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^\top$ being the translational position and velocity, $\ddot{\boldsymbol{\xi}}^* \triangleq [\ddot{x}^* \ \ddot{y}^* \ \ddot{z}^*]^\top$ denoting virtual control inputs designed as in (20). The two matrices $A_{\mathbf{x}} \in \mathbb{R}^{6\times6}$ and $B_{\mathbf{x}} \in \mathbb{R}^{6\times3}$ are defined as:

$$A_{\mathbf{x}} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}, \quad B_{\mathbf{x}} = \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3} \end{bmatrix}. \tag{25}$$

[1] Any system admitting a flat description has the number of flat outputs $\mathbf{z}(t)$ equal to the number of inputs $\mathbf{u}(t)$ (Fliess et al., 1995).

Remark 1. In Carino et al. (2015), the authors introduce different feedback linearization designs for the position controller in more compact forms. However, their controllers implicitly constrain the fourth element of the reference quaternion, q_{3_r} as in (19), to be zero, which does not hold for applications requiring yawing such as aerial photography and filming. Therefore, the flatness-based feedback linearization controller proposed in (19), which takes into account a general value of q_{3_r}, provides more flexibility for the users. □

3.2 Computed-torque attitude controller

For the low control level, we apply the CTC (Computed Torque Control) method (Tzafestas, 2013), a special feedback linearization design for nonlinear systems admitting Lagrangian dynamics, to design the attitude controller. Let us consider the rotational dynamics written in the form of (18). Then the CTC method provides the input torque as follows:

$$\boldsymbol{\tau} = JQ(\boldsymbol{q})\boldsymbol{\tau}' + JD_{\boldsymbol{q}}[Q(\boldsymbol{q})\dot{\boldsymbol{q}}]\dot{\boldsymbol{q}} + [Q(\boldsymbol{q})\dot{\boldsymbol{q}}] \times [JQ(\boldsymbol{q})\dot{\boldsymbol{q}}], \quad (26)$$

in which, J is the inertial tensor from (8), $Q(\boldsymbol{q})$ is as in (5), $D_{\boldsymbol{q}}(\cdot)$ is the Jacobian w.r.t. $\boldsymbol{q}$, and $\boldsymbol{\tau}'$ is a corrective term given by:

$$\boldsymbol{\tau}' = \ddot{\boldsymbol{q}}_r + K_{p_q}\boldsymbol{\epsilon}_q + K_{d_q}\dot{\boldsymbol{\epsilon}}_q + K_{i_q}\int \boldsymbol{\epsilon}_q dt, \quad (27)$$

with $\boldsymbol{\epsilon}_q = \boldsymbol{q}_r - \boldsymbol{q}$, $\boldsymbol{q}_r$ the reference quaternion from the position controller, and K_{p_q}, K_{d_q}, K_{i_q} are diagonal positive definite matrices from $\mathbb{R}^3$.

Remark 2. Even though the attitude controller (26) tracks only three components $q_{1_r}, q_{2_r}, q_{3_r}$ of the quaternion, the first component q_0 still converges to q_{0_r}. This is proved by the following assertions obtained from (6) and (22d):

$$\begin{aligned} q_0 &= \sqrt{1-(q_1^2+q_2^2+q_3^2)}, \\ q_{0_r} &= \sqrt{1-(q_{1_r}^2+q_{2_r}^2+q_{3_r}^2)}, \end{aligned} \quad (28)$$

in which, $[q_1\ q_2\ q_3]^\top$ converges to $[q_{1_r}\ q_{2_r}\ q_{3_r}]^\top$, and hence, the attitude tracking capability is always guaranteed. Furthermore, it is important to choose the gains of both attitude and position controllers such that $\boldsymbol{\epsilon}_{\boldsymbol{q}}$ as in (27) converges to 0 faster than $\boldsymbol{\epsilon}_\xi$ from (21) in order to ensure good tracking capability (Nguyen et al., 2017). □

4. SIMULATION AND EXPERIMENTAL RESULTS

This section provides first the simulation results of the feedback linearization controllers proposed in Section 3. Subsequently, the high-level position controller (19) is further illustrated through real experiments over a nano-drone platform.

4.1 Simulation model

For the simulation model to be realistic, we add the drag force (Fox and McDonald, 1994) to dynamics (7):

$$m\ddot{\boldsymbol{\xi}} = m\boldsymbol{g} + R\boldsymbol{e}_z T + \boldsymbol{F_D}, \quad (29)$$

in which, the drag $\boldsymbol{F_D}$ is calculated by:

$$\boldsymbol{F_D} = -\frac{1}{2}\rho C_D A|\dot{\boldsymbol{\xi}}|\dot{\boldsymbol{\xi}}, \quad (30)$$

with C_D the drag coefficient, ρ the fluid density, $\dot{\boldsymbol{\xi}}$ the velocity and A the cross-section of the multicopter as used in Nguyen et al. (2017).

The parameters of the quadcopter used for simulation are chosen as:

- $m = 0.025\ [kg]$, $J_x = J_y = 4.856 \times 10^{-3}\ [kgm^2]$, $J_z = 8.801 \times 10^{-3}\ [kgm^2]$, $C_D = 0.8$, $\rho = 1.225\ [kg/m^3]$, $A = 0.01425\ [m^2]$.

4.2 Simulation results

We first generate a reference trajectory starting from the initial position $[1.8\ 2.27\ 0.6]^\top$ at $t = 0s$, which hovers at the position $[2.9\ 3\ 1.2]^\top$ during the time period $6s \leq t \leq 14s$, and finally, arrives to the final position $[3.6\ 2\ 0.6]^\top$ at $t = 20s$. The reference q_{3_r} as employed in (19) is fixed at zero. The simulation is performed in Matlab/Simulink 2017a with Yalmip (Löfberg, 2004) and MPT Toolboxes (Herceg et al., 2013) using the solver ode4 with sampling time $0.01s$.

The control gains of the two controllers introduced in Section 3 are gathered in Table 1.

Table 1. Control gains for the simulation.

Controller	Control gains	Unit
Position controller	$K_{p_\xi} = diag(60, 60, 60)$	$[s^{-2}]$
	$K_{d_\xi} = diag(6, 6, 6)$	$[s^{-1}]$
	$K_{i_\xi} = diag(0.1, 0.1, 0.1)$	$[s^{-3}]$
Attitude controller	$K_{p_q} = diag(2000, 2000, 2000)$	$[s^{-2}]$
	$K_{d_q} = diag(10, 10, 10)$	$[s^{-1}]$
	$K_{i_q} = diag(10, 10, 5)$	$[s^{-3}]$

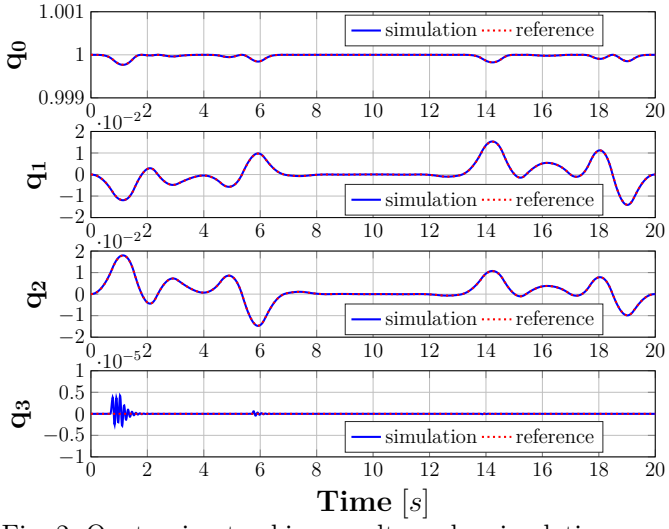

Fig. 2. Quaternion tracking results under simulation.

At first, we provide the quaternion tracking results of the attitude controller (26)–(27) in Fig. 2. We observe that besides some chattering behavior in q_3 around $t = 1s$ due to the instability inherent in the numerical differentiation, the three components q_1, q_2, q_3 generally track well their references. Thus, even though q_0 is not explicitly controlled by the CTC controller (26)–(27), it still follows the reference q_{0_r} as shown in the first plot of Fig. 2.

Next, the position references (plotted in red dashed lines) and their tracking results under simulation (plotted in blue

solid lines) are given in Fig. 3. Without any noticeable tracking errors, the position controller (19)–(20) shows its robustness, albeit the drag in (30) is not taken into account during the controller design procedure. In the next section, we will apply this position controller for controlling a real quadcopter platform under the same tracking scenario.

4.3 Experimental results

The experimental video could be viewed online at: https://youtu.be/NIdCI6FJXHM. The experiment is done indoor, with no wind disturbance.

The experimental platform includes a quadcopter Crazyflie (CF) and a Loco Positioning System (LPS) (Bitcraze, 2019; Nguyen et al., 2018), which comprises of 6 fixed, known position anchors in the room and a tag attached to the Crazyflie. The platform provides feedback on the quadcopter position, by continuously measuring the distances between the tag and the anchors. The data transmission between the ground station PC and the Crazyflie is accomplished by 2.4GHz low-latency/long-range radio messages between the tag and the Crazyradio PA radio USB dongle connected to the PC.

By decoupling the translational and rotational dynamics of the Crazyflie with the proposed hierarchical control approach, the autonomous trajectory tracking task could be accomplished by utilizing the high-level controller on the PC. The CF has its built-in controller which can only track the references of the thrust and the roll, pitch, yaw angles. Hence, the reference quaternion from the position controller (19) needs to be transformed into these Euler angles by using relation (11) before being sent to the CF.

From Fig. 3, we see that under experiment (actual results plotted in green solid lines), the CF tracks well the reference trajectory (red dashed lines), especially along the x and the y axes. For the altitude results, there are small tracking errors due to the low precision of the LPS when measuring the height (a similar problem is mentioned in Nguyen et al. (2018)). Furthermore, we apply an external disturbance (we push) the CF three times at $t = 8.11s$, $t = 8.57s$, and $t = 14.19s$ in order to test the robustness of the position controller (19)–(20). As can be seen from Fig. 3, the controller can compensate rapidly the errors. Fig. 5 provides the 3D view of the whole simulation and experimental scenarios in which the altitude tracking errors can be clearly observed.

Fig. 4 shows the quaternion tracking performances of the CF's built-in controller. In general, the actual quaternion results (plotted in green solid lines) follow their references (plotted in red dotted lines). The chattering behavior in the references can be diminished by choosing more appropriate control gains. The first push applied at $t = 8.11s$ has no significant effect on the system behavior. However, at $t = 8.57s$ and $t = 14.19s$, the heavier pushes do show visible effect: the references q_{1_r} and q_{2_r} immediately reverse their signs, which indicates the control effort of the position controller (19)–(20) to compensate the errors.

5. CONCLUSIONS

This paper presented a hierarchical control design for multicopter trajectory tracking under the quaternion representation. We first provide the differential flatness properties of the system which are further employed to design a feedback linearization position controller at the high control level. Next, at the low-level, the standard computed-torque control method is applied to stabilize the rotational sub-system around the desired quaternion sent from the position controller. The proposed approaches are illustrated through simulations and experiments over a real quadcopter platform. Future works consist in seeking a

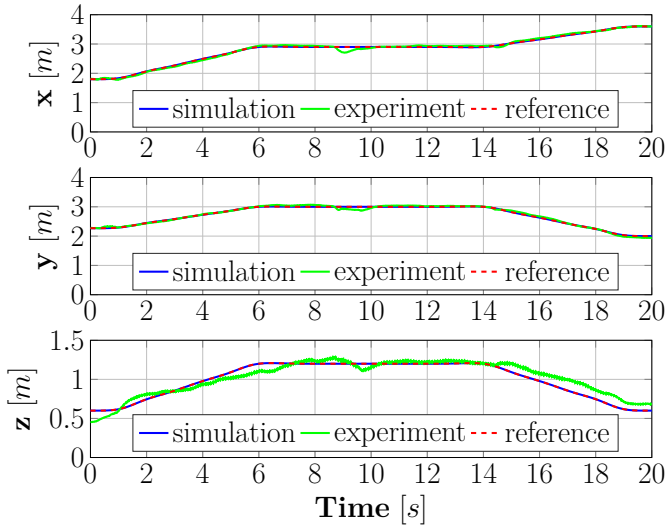

Fig. 3. Position with disturbances.

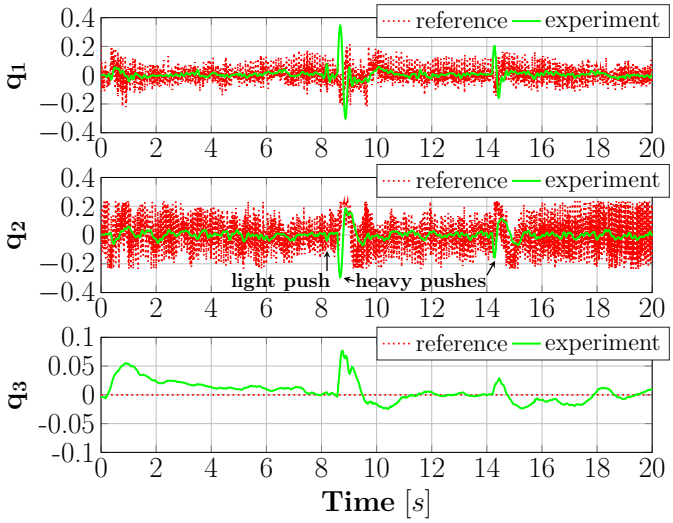

Fig. 4. Quaternion with disturbances.

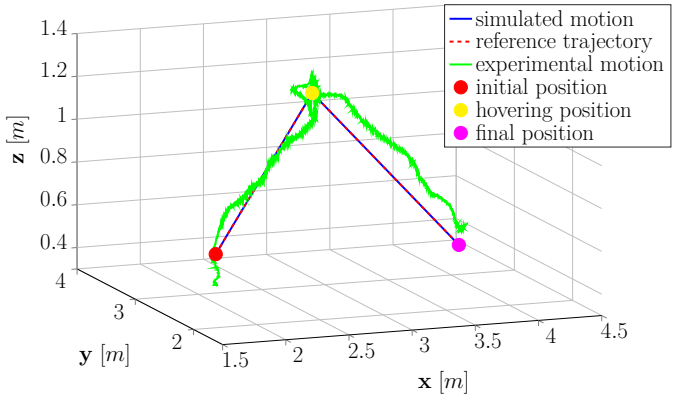

Fig. 5. 3D trajectory with disturbances.

new flat output which may result in a less convoluted flat representation, analyzing the robustness of the controlled system under saturation inputs and disturbances, as well as testing with more aggressive trajectories.

REFERENCES

Alaimo, A., Artale, V., Milazzo, C., Ricciardello, A., and Trefiletti, L. (2013). Mathematical modeling and control of a hexacopter. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 1043–1050. IEEE.

Bitcraze (2019). Crazyflie 2.0. `https://www.bitcraze.io/crazyflie-2/`. [Online; accessed July 19, 2019].

Carino, J., Abaunza, H., and Castillo, P. (2015). Quadrotor quaternion control. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 825–831. IEEE.

Chovancová, A., Fico, T., Hubinskỳ, P., and Duchoň, F. (2016). Comparison of various quaternion-based control methods applied to quadrotor with disturbance observer and position estimator. *Robotics and Autonomous Systems*, 79, 87–98.

Diebel, J. (2006). Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58(15-16), 1–35.

Fliess, M., Lévine, J., Martin, P., and Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. *International journal of control*, 61(6), 1327–1361.

Floreano, D. and Wood, R.J. (2015). Science, technology and the future of small autonomous drones. *Nature*, 521(7553), 460–466.

Formentin, S. and Lovera, M. (2011). Flatness-based control of a quadrotor helicopter via feedforward linearization. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 6171–6176.

Fox, R.W. and McDonald, A.T. (1994). *Introduction to fluid mechanics*. John Wiley.

Fresk, E. and Nikolakopoulos, G. (2013). Full quaternion based attitude control for a quadrotor. In *2013 European Control Conference (ECC)*, 3864–3869. IEEE.

Herceg, M., Kvasnica, M., Jones, C., and Morari, M. (2013). Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, 502–510. Zürich, Switzerland. `http://control.ee.ethz.ch/ mpt`.

Johnson, N.L. and Leang, K.K. (2014). Enhanced proportional-derivative control of a micro quadcopter. In *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers Digital Collection.

Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*. Taipei, Taiwan. URL `http://users.isy.liu.se/johanl/yalmip`.

Mellinger, D. and Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2520–2525.

Nascimento, T.P. and Saska, M. (2019). Position and attitude control of multi-rotor aerial vehicles: A survey. *Annual Reviews in Control*.

Nguyen, N.T., Prodan, I., and Lefèvre, L. (2018). Flat trajectory design and tracking with saturation guarantees: a nano-drone application. *International Journal of Control*, 1–14, Article in Press.

Nguyen, N.T., Prodan, I., Stoican, F., and Lefevre, L. (2017). Reliable nonlinear control for quadcopter trajectory tracking through differential flatness. *IFAC-PapersOnLine*, 50(1), 6971–6976.

Tzafestas, S.G. (2013). *Introduction to mobile robot control*. Elsevier.

Whitmore, S.A. (2000). Closed-form integrator for the quaternion (euler angle) kinematics equations. US Patent 6,061,611.

Yang, Y. (2012). Spacecraft attitude determination and control: Quaternion based method. *Annual Reviews in Control*, 36(2), 198–219.

Appendix A. PROOF FOR THE FLAT REPRESENTATIONS OF THE QUADCOPTER DYNAMICS

Gathering the translational dynamics (9) and the norm constraint (1), we have the system of equations:

$$m\ddot{x}/T = 2(q_0q_2 + q_1q_3), \tag{A.1a}$$
$$m\ddot{y}/T = 2(q_2q_3 - q_0q_1), \tag{A.1b}$$
$$m(\ddot{z}+g)/T = q_0^2 - q_1^2 - q_2^2 + q_3^2, \tag{A.1c}$$
$$1 = q_0^2 + q_1^2 + q_2^2 + q_3^2. \tag{A.1d}$$

Squaring both sides of (A.1a)–(A.1c), then taking their sum leads to:

$$\left(q_0^2 + q_1^2 + q_2^2 + q_3^2\right)^2 = \left[\ddot{x}^2 + \ddot{y}^2 + (\ddot{z}+g)^2\right] m^2/T^2. \tag{A.2}$$

Replacing (A.1d) to the left hand side of (A.2), and taking the square root of both sides, the thrust can be calculated as follows:

$$T = m\sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z}+g)^2}. \tag{A.3}$$

Taking the sum of both sides of (A.1c) and (A.1d), we have:

$$q_0^2 + q_3^2 = \left[1 + (\ddot{z}+g)m/T\right]/2. \tag{A.4}$$

Replacing (A.3) to (A.4), we arrive to [2]:

$$q_0 = \frac{1}{\sqrt{2}}\sqrt{\frac{\ddot{z}+g}{\sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z}+g)^2}} - 2q_3^2 + 1}. \tag{A.5}$$

Rewriting (A.1a) and (A.1b) in the matrix form leads to:

$$\begin{bmatrix} q_3 & q_0 \\ -q_0 & q_3 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \frac{m}{2T} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix}. \tag{A.6}$$

Next, we have that:

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \frac{m}{2T} \times \frac{1}{q_0^2 + q_3^2} \begin{bmatrix} q_3 & -q_0 \\ q_0 & q_3 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix}. \tag{A.7}$$

Replacing (A.3) and (A.5) to (A.7) leads to:

$$q_1 = \frac{\ddot{x}q_3 - \frac{1}{\sqrt{2}}\ddot{y}\sqrt{\frac{\ddot{z}+g}{\sqrt{\ddot{x}^2+\ddot{y}^2+(\ddot{z}+g)^2}} - 2q_3^2 + 1}}{(\ddot{z}+g) + \sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z}+g)^2}}, \tag{A.8a}$$

$$q_2 = \frac{\ddot{y}q_3 + \frac{1}{\sqrt{2}}\ddot{x}\sqrt{\frac{\ddot{z}+g}{\sqrt{\ddot{x}^2+\ddot{y}^2+(\ddot{z}+g)^2}} - 2q_3^2 + 1}}{(\ddot{z}+g) + \sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z}+g)^2}}. \tag{A.8b}$$

Thus, completing the proof.

[2] We only consider $\ddot{z} \geq -g$, which is always true for nominal operation.