Quadrotor Aggressive Deployment, Using a Quaternion-based Spherical Chattering-free Sliding-mode Controller

Hernan Abaunza, Pedro Castillo Garcia

▶ To cite this version:

HAL Id: hal-02283712
https://hal.archives-ouvertes.fr/hal-02283712

Submitted on 4 Oct 2019

Quadrotor Aggressive Deployment, Using a Quaternion-based Spherical Chattering-free Sliding-mode Controller

Hernán Abaunza and Pedro Castillo

Abstract—This paper introduces a non-conventional approach for autonomous multi-rotor UAV deployment, in which a quadrotor is aggressively launched through the air with its motors turned off. A continuous quaternion attitude trajectory is proposed to safely recover the vehicle into hover mode. Then, an operator then could take the command or continue a desired mission in autonomous mode. The controller is a chattering-free sliding mode algorithm based on the geometrical properties of quaternions and axis-angle rotations. Lyapunov theory is used to analyze the system stability. The proposed methodology is validated in real world indoor and outdoor experiments.

Index Terms—Quadrotor, Sliding Mode, Hand-launching, Deployment, Launching, Quaternion

I. INTRODUCTION

IN the near future, UAVs will be capable of performing impressive tasks that involve very complex maneuvers. Looking towards such future, some progress has already been made by proposing techniques for performing agile flights. For instance [32] proposed an approach to navigate a quadrotor while avoiding collisions using stereo vision. [33] addressed the problem of identifying obstacle shapes using a real-time guidance algorithm. [22] introduced a solution for tracking aggressive trajectories for obstacle avoidance. Agile flight for fixed-wing autonomous aircraft has also been studied, including autonomus navigation in [6] and [20], and collision avoidance as presented in [48] and [30].

In [2], an approach for landing a quadrotor on a moving platform in the presence of disturbances was proposed by means of an adaptive dynamic surface control technique. A related approach was presented by [29], where a controller based on invariant ellipsoids was proposed for landing a quadrotor on a ship deck. In [26], researchers performed flights that involve high accelerations and angular velocities by employing visual-inertial sensors. [28] and [11] proposed algorithms for estimating the vehicle states and generating trajectories for flying through narrow gaps at different inclinations.

In many real-world applications, quickly deploying rotorcraft can sometimes be difficult and time-consuming, often requiring clear and flat spaces to initialize the vehicles. For instance, in firefighting, security, or rescue scenarios where

Hernán Abaunza and Pedro Castillo are with Sorbonne Universités, Université de Technologie de Compiègne, CNRS UMR 7253 Heudiasyc, Compiègne, France. e-mail: (castillo,habaunza)@hds.utc.fr.

fast response is essential, objects around the deployment area (like tall grass or debris) could hinder the initial steps of any mission. Other activities like outdoor sports such as hiking, climbing, or biking could lack of an appropriate surface to place the vehicle.

An intuitive and fast deployment could be to just hand-launch the vehicle. Some expert hobbyists do it manually with advanced piloting skills; nevertheless, they usually do so with a close to zero attitude and/or with the motors turned on. However spinning propellers could be hazardous for the launcher, and the vehicle itself. For safety reasons, therefore, quadrotor hand-launching should be performed with the motors switched off. Launching a drone in this manner and autonomously recovering while in the air is a real problem that has not been totally solved.

Only a few works have addressed related approaches. For instance, [17] employs reinforced learning to stabilize a quadrotor, among their experiments the authors throw the vehicle by hand. However, aggressive hand-launch analysis, detection and recovery are not addressed in their work, instead, 2150 iterations of their learning algorithm are required to stabilize the vehicle.

Similarly, in [10] the authors propose an approach to recover a quadrotor from free-fall conditions detected by an acceleration threshold. Their approach uses quaternions to compute recovery rotations, once recovered, visual odometry is used to stabilize the vehicle position. In this work the motors are already rotating before launching, and accelarated aggressively after the condition is validated for attitude stabilization.

In both works, [17] and [10], no analysis of the drone launching is proposed. In addition, the experiments were done with motors turned on, reducing the control problem challenge and allowing a simpler attitude regulation. From an aerodynamic analysis, when launching a rotorcraft with motors turned on, some aerodynamic phenomena appear; for example, the air body resistance is increased (greater drag force), reducing the translational launching velocity. Moreover, as each rotor is turning (even in slow angular velocities), a passive trim of the angular speed is introduced into the system, imposing an aerodynamic attitude rate compensation.

In this paper, a solution for this problem is proposed by introducing recovery and control algorithms formulated on quaternion and sliding mode methodologies. This technique provides on one hand, an analysis of the launching condition. On one hand, an aggressive attitude trajectory to autonomously recover the quadrotor rotational stability is proposed. On the

other hand, a nonlinear controller capable of quickly tracking this trajectory is developed. The recovery and launching detection algorithms are computed by only employing inertial sensors allowing the system to be used either in indoors, outdoors, or GPS-denied environments. Continuous functions are defined instead of state machines to activate the motors and switch between recovery and mission-specific attitude references, giving a more precise description of the system and simplifying its implementation.

Sliding mode control (SMC) has been studied in the last decades by some eminent authors. Some of the most relevant works that established the basis of sliding mode control theory are [42]-[24]. Works like [8] and [46] provide theoretical and practical grounds for SMC. As many authors have remarked, SMC approaches use discontinuous functions (such as the sign function) to ensure convergence of the system states to a desired manifold in finite time. This fact induces an aggressive switching response on the control input, known as chattering effect, which in the real world can damage actuators. For this reason, researchers have been studying alternative approaches such as fractional order systems [45] and [34], high-order sliding mode controllers [25]-[35], or by ensuring asymptotic convergence around the sliding manifold by means of continuous functions such as sigmoids [3]-[9].

Each of the previous approaches have advantages and drawbacks as remarked by [43]. Fractional order control is complex to tune, and is not straightforwardly free from chattering. High-order SMC is more difficult to implement with little advantages. While continuous functions reduce the robustness of the closed-loop system against external disturbances.

Quaternion techniques have been addressed in other works. For instance, an introduction to unit quaternions and some applications to aerial systems are presented in [1]-[21]. [41] presented an autonomous attitude recovery controller for spacecraft based on a quaternion feedback linearization technique. [16] designed a robust quaternion-based sliding mode algorithm for tracking attitude spacecraft trajectories. [14] and [13] proposed optimal control techniques for quadrotors based on quaternion attitude error equations. Some of the first works that have combined sliding mode theory with quaternion-based models were [5], [27] and [4], where attitude sliding mode controllers based on the kinematic quaternion equations of a general rigid aircraft were introduced. However, the previous works have applied quaternions to other techniques, not exploring the advantages of sliding mode approaches.

Until very recently, most sliding mode control algorithms for quadrotors were based on the classical dynamic equations based on Euler angles, which drag some complications such as nonlinear equations, singularities, and gimbal lock effects. To deal with these problems, some works like [44] have proposed model-free sliding mode controllers that partially ignore the vehicle behavior in order to simplify the control design procedure. Other works like [12] and [19] employed sliding mode algorithms to compensate external disturbances, but consider conservative assumptions such as small angle rotations and slow movements to design the control equations.

More recent contributions like [18] and [31] proposed controllers based on fractional sliding modes applied to quadrotors using quaternion-based models. [15] developed a quaternion-based bounded attitude controller using saturation functions. [37], [49], and [47] presented modified algorithms which diminish the chattering effect of the sliding mode, with applications to quadrotor vehicles.

Nevertheless, most works found in the literature have applied sliding mode controllers on quadrotors to classical flight missions. In this work, we propose a new and challenging scenario, where a good performance of the sliding mode controller is needed to successfully recover a quadrotor from an aggressive launch. Also, this work proposes spherical sliding mode technique, which instead of using the typical sign function, a unitary sphere is used to control a sliding manifold on a 3-D space using direction axes. Sigmoid functions are also employed such that the algorithm becomes chattering-free. In our best knowledge, this is the first work that specifically conceives an aggressive handlaunch deployment approach for a quadrotor when motors are turned off.

The rest of the paper is organized as follows: The quadrotor rotational and vertical dynamic equations are presented in Section III. A quaternion-based spherical chattering-free sliding mode controller is developed in Section IV, while the recovery attitude trajectory is detailed in Section V. A continuous automatic detection and switching technique is proposed in Section VI for the aggressive launching scenario. Real world experiments are presented in Section VII.

II. PROBLEM STATEMENT

Consider a scenario in which a quadrotor is aggressively launched through the air with its motors turned off. Since the vehicle starts to fall by the effects of gravity, it describes a pseudo-parabolic trajectory with an unknown orientation and initial velocity. Under these conditions, a recovery strategy must be proposed to stabilize and recover the vehicle such that a user can easily operate it after deployment. Such deployment technique should include the following characteristics:

- The quadrotor starts from the ground, with the motors turned off.
- An operator manually launches the vehicle.
- The initial attitude and velocity conditions are unknown.
- Inertial sensors detect the launching acceleration.
- The drone is recovered autonomously into a stabilized hover mode.
- Once the vehicle is recovered, the operator can use a joystick or any other input signal to execute a mission.

To deploy a quadrotor with such technique, a stabilization and control strategy will be developed. Recovery and control algorithms for the attitude and altitude dynamics will be designed as illustrated in Figure 1. This proposal will be detailed in the following sections.

III. QUADROTOR QUATERNION MODELING

The conception of the deployment strategy starts with the quadrotor dynamic model which will consider the rotational and translational equations as separate systems. Euler's theorem of rotation states that any 3-dimensional rotation can be defined by its magnitude ϑ in radians, and its axis of rotation symbolized by a unit vector $\vec{n} \in \mathbb{R}^3$.
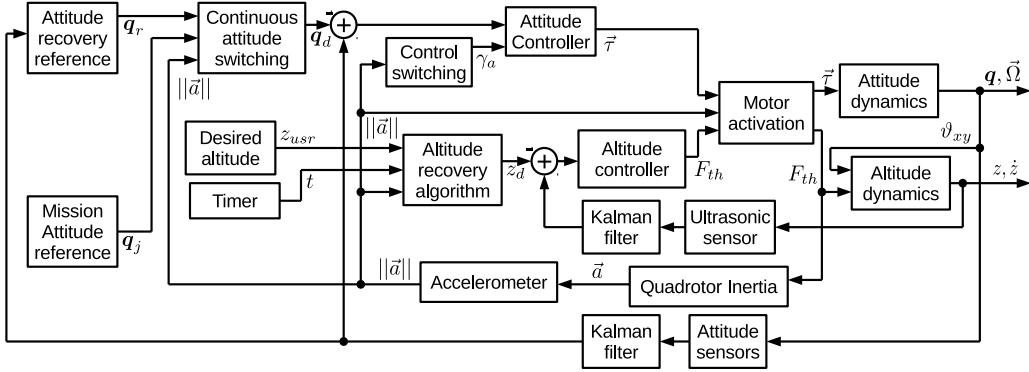
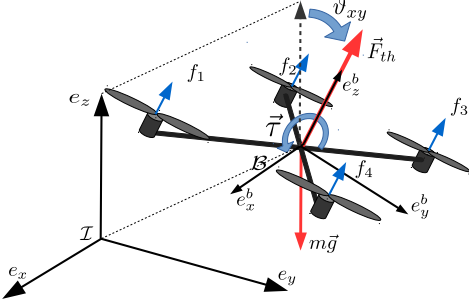Figure 1: Recovery and control diagram, the proposed algorithms will be developed in the following sections.

Figure 2: Quadrotor free body diagram. Considering the propellers produce a total thrust force in the body's vertical axis, and a torque vector in $\mathbb{R}^3$.

Considering the quadrotor free-body diagram from Figure 2, the rotation quaternion from the inertial $\mathcal{I}$ to the vehicle's local frame $\mathcal{B}$ is defined as

$$\boldsymbol{q} := e^{\vec{n}\,\vartheta/2} = \cos(\vartheta/2) + \vec{n}\sin(\vartheta/2) = q_0 + \vec{q}, \qquad (1)$$

where $q_0 \in \mathbb{R}$ and $\vec{q} = [q_1 \; q_2 \; q_3]^T$ represent the quaternion components. Inversely, the axis-angle representation of a rotation can be computed using a quaternion-logarithm as

$$\vec{\vartheta} = \vec{n}\vartheta = 2\ln \boldsymbol{q} \in \mathbb{R}^3, \quad \dot{\vec{\vartheta}} = \vec{\Omega}, \qquad (2)$$

where $\ln(\cdot)$ is given by

$$\ln \boldsymbol{q} = \begin{cases} \frac{\vec{q}}{||\vec{q}||}\arccos(q_0), & ||\vec{q}|| \neq 0, \\ [0 \; 0 \; 0]^T, & ||\vec{q}|| = 0. \end{cases} \qquad (3)$$

Introducing Newton's equations of motion, the rotational and vertical dynamics of a quadrotor are given by

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t) = \begin{bmatrix} f_1(\boldsymbol{q}, \vec{\Omega}, \vec{\tau}, t) \\ f_2(z, \dot{z}, F_{th}, t) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\boldsymbol{q} \otimes \vec{\Omega} \\ J^{-1}(\vec{\tau} - \vec{\Omega} \times (J\vec{\Omega})) \\ \dot{z} \\ F_{th}\cos(\vartheta_{xy})/m + \vec{g} \end{bmatrix}, (4)$$

where the system state $\mathbf{x} = [\boldsymbol{q}, \vec{\Omega}, z, \dot{z}]^T$ is given by the vehicle attitude quaternion $\boldsymbol{q}$, its angular velocity $\vec{\Omega} = [\Omega_x, \Omega_y, \Omega_z]^T$, the altitude z, and its vertical speed $\dot{z}$. The control inputs $\mathbf{u} = [\vec{\tau}, F_{th}]^T$ are defined by the total thrust force $F_{th} = \sum_{i=1}^4 f_i$, $F_{th} \in \mathbb{R}$ and the total torque $\vec{\tau} \in \mathbb{R}^3$.

The quaternion product is symbolized by $\otimes$, the body mass and inertia matrix are denoted by m and $J = \text{diag}([J_{xx}, J_{yy}, J_{zz}])$ respectively. The gravitational acceleration is symbolized as $\vec{g} := [0 \; 0 \; -g]^T$ with $g \approx 9.81[m/s^2]$, m defines the vehicle mass, and t represents the time, see [1]. ϑ_{xy} denotes the angle between the vertical axes of $\mathcal{I}$ and $\mathcal{B}$, and its cosine is computed as

$$\cos(\vartheta_{xy}) = \left(\boldsymbol{q} \otimes [0 \; 0 \; 1]^T \otimes \boldsymbol{q}^*\right) \cdot [0 \; 0 \; 1]^T. \qquad (5)$$

Remark 1: Let $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ be two quaternions such that

$$\boldsymbol{q}_1 := \cos(\vartheta/2) + \vec{n}\sin(\vartheta/2),$$
$$\boldsymbol{q}_2 := \cos((\vartheta + 2\pi)/2) + \vec{n}\sin((\vartheta + 2\pi)/2),$$

then, $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ represent the same orientation, with the difference that $\boldsymbol{q}_2$ takes an additional 2π rotation around $\vec{n}$, furthermore, $\boldsymbol{q}_1 = -\boldsymbol{q}_2$.

In order to design independent attitude and altitude controllers, (4) is divided into two separate subsystems as

$$f_1(\boldsymbol{q}, \vec{\Omega}, \vec{\tau}, t) = \frac{d}{dt}\begin{bmatrix} \boldsymbol{q} \\ \vec{\Omega} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\boldsymbol{q} \otimes \vec{\Omega} \\ J^{-1}(\vec{\tau} - \vec{\Omega} \times (J\vec{\Omega})) \end{bmatrix}, \qquad (6)$$

and

$$f_2(z, \dot{z}, F_{th}, t) = \frac{d}{dt}\begin{bmatrix} z \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \dot{z} \\ F_{th}\cos(\vartheta_{xy})/m + \vec{g} \end{bmatrix}. \qquad (7)$$

IV. CONTROL STRATEGY

Consider that (7) is straightforwardly stabilized by defining a proportional-derivative controller for the net thrust force as

$$F_{th} := \begin{cases} \dfrac{-k_{pz}(z - z_d) - k_{dz}\dot{z} + mg}{\cos(\vartheta_{xy})} & \text{if } \cos(\vartheta_{xy}) > 0, \\ F_{\min} & \text{otherwise} \end{cases} \qquad (8)$$

where z_d denotes the desired altitude, $k_{pz}, k_{dz} \in \mathbb{R}^+$ are positive gains, and $F_{\min} > 0$ defines a minimum constant thrust. Note from (5) and (8) that if an attitude controller takes the vehicle thrust vector to a vertical orientation, then $\cos(\vartheta_{xy}) = \left(\boldsymbol{q} \otimes [0 \; 0 \; 1]^T \otimes \boldsymbol{q}^*\right) \cdot [0 \; 0 \; 1]^T \to 1$.

Therefore, if the vehicle inclination ϑ_{xy} is equal or greater than $90°$, F_{th} will be left at a minimal value to reduce the risk of accelerating downwards. Once the rotational controller recovers, the thrust force will close the altitude dynamics loop.

A. Attitude Control Formulation

An attitude controller will be now developed to track any desired orientation for system (6).

Define a function $\Gamma : \mathbb{R}^3 \to \mathbb{R}^3$, which bounds any vector into a sphere (see Figure 3), respecting its original direction and modifying its magnitude in terms of a continuous arctangent function as

$$\Gamma(\vec{k}) := \hat{\boldsymbol{k}}\,\frac{2}{\pi}\tan^{-1}\left(||\vec{k}||\right). \qquad (9)$$
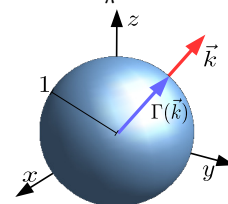

Figure 3: Spherical bounding of a 3-D vector, its bounded magnitude is defined by an arctangent function.

Lemma 1: Given any vector $\vec{k}$, $\Gamma(\vec{k})$ is bounded by a unitary sphere and it holds

$$\text{sign}(\Gamma(\vec{k})_i) = \text{sign}(\vec{k}_i), \tag{10}$$

for every component $i = x, y, z$ of both vectors.

Proof 1: Computing the normalized vector of $\Gamma(\vec{k})$, it yields

$$\frac{\Gamma(\vec{k})}{||\Gamma(\vec{k})||} = \frac{\hat{k}\, \frac{2}{\pi}\tan^{-1}\left(||\vec{k}||\right)}{\frac{2}{\pi}\tan^{-1}\left(||\vec{k}||\right)} = \hat{k}, \tag{11}$$

therefore, $\vec{k}$, and $\Gamma(\vec{k})$ share the same direction, and since the magnitude of the former is defined by an arctangent function with a positive definite argument, it holds

$$0 < \frac{2}{\pi}\tan^{-1}\left(||\vec{k}||\right) < 1. \quad \blacksquare \tag{12}$$

Theorem 1: *The attitude subsystem* (6) *of a quadrotor converges asymptotically to the quaternion origin* $\boldsymbol{q}_{\mathcal{O}} = 1 + [0\ 0\ 0]^T$ *by means of* $\vec{\tau} = [\tau_x, \tau_y, \tau_z]$, *with*

$$\tau_x = -\frac{2}{\pi}(k_{px}\vartheta_x + k_{d2x}\Omega_x)\tan^{-1}(||\vec{\sigma}||)/||\vec{\sigma}|| \tag{13}$$
$$- J_{zz}\Omega_y\Omega_z - J_{yy}\Omega_y\Omega_z - k_{d1x}\Omega_x,$$

$$\tau_y = -\frac{2}{\pi}(k_{py}\vartheta_y + k_{d2y}\Omega_y)\tan^{-1}(||\vec{\sigma}||)/||\vec{\sigma}|| \tag{14}$$
$$- J_{xx}\Omega_x\Omega_z - J_{zz}\Omega_x\Omega_z - k_{d1y}\Omega_y,$$

$$\tau_z = -\frac{2}{\pi}(k_{pz}\vartheta_z + k_{d2z}\Omega_z)\tan^{-1}(||\vec{\sigma}||)/||\vec{\sigma}|| \tag{15}$$
$$- J_{yy}\Omega_x\Omega_y - J_{xx}\Omega_x\Omega_y - k_{d1z}\Omega_z,$$

where $k_{px}, k_{py}, k_{pz}, k_{d1x}, k_{d1y}, k_{d1z}, k_{d2x}, k_{d2y}, k_{d2z} \in \mathbb{R}$ represent positive control gains, $\vec{\vartheta} = [\vartheta_x, \vartheta_y, \vartheta_z]^T = 2\ln(q)$, and

$$\vec{\sigma} = \begin{bmatrix} k_{px}\vartheta_x + k_{d2x}\Omega_x \\ k_{py}\vartheta_y + k_{d2y}\Omega_y \\ k_{pz}\vartheta_z + k_{d2z}\Omega_z \end{bmatrix}. \tag{16}$$

Proof 2: Define a Lyapunov candidate function as

$$V_1 = \frac{1}{2}\vec{\sigma} \cdot \vec{\sigma} \;\rightarrow\; \dot{V}_1 = \vec{\sigma} \cdot \dot{\vec{\sigma}}, \tag{17}$$

where $\vec{\sigma} \in \mathbb{R}^3$ is a sliding manifold such that

$$\begin{aligned} \vec{\sigma} &= K_1\vec{\Omega} + K_2\vec{\vartheta}, \\ \dot{\vec{\sigma}} &= K_1(J^{-1}\vec{\tau} - J^{-1}\vec{\Omega} \times J\vec{\Omega}) + K_2\vec{\Omega}, \end{aligned} \tag{18}$$

where $K_j \in \mathbb{R}^{3\times 3}, j = 1, 2, ...$, are positive diagonal matrices. Note from Lemma 1 that asymptotic stability for (17) is reached if

$$\dot{\vec{\sigma}} = -K_3\Gamma\left(K_4\vec{\sigma}\right). \tag{19}$$

Introducing (19) into (18), yields

$$\begin{aligned} \vec{\tau} &= -JK_1^{-1}K_3\Gamma\left(K_4K_1\vec{\Omega} + K_4K_2\vec{\vartheta}\right) \\ &\quad -JK_1^{-1}K_2\vec{\Omega} + \vec{\Omega} \times J\vec{\Omega}, \end{aligned} \tag{20}$$

propose $K_3 = K_1J^{-1}$, $K_p = K_4K_2$, $K_{d1} = K_3^{-1}K_2$, and $K_{d2} = K_4K_1$, (20) is finally expressed as

$$\vec{\tau} = -K_{d1}\vec{\Omega} - \Gamma\left(K_p\vec{\vartheta} + K_{d2}\vec{\Omega}\right) + \vec{\Omega} \times J\vec{\Omega}. \tag{21}$$

Finally, the derivative of (17) is expressed as

$$\dot{V}_1 = (K_1\vec{\Omega} + K_2\vec{\vartheta}) \cdot (K_1(J^{-1}\vec{\tau} - J^{-1}\vec{\Omega} \times J\vec{\Omega}) + K_2\vec{\Omega}). \tag{22}$$

Introducing (21), it yields

$$\dot{V}_1 = -\left(K_1\vec{\Omega} + K_2\vec{\vartheta}\right) \cdot \Gamma\left(K_4\left(K_1\vec{\Omega} + K_2\vec{\vartheta}\right)\right). \tag{23}$$

From Lemma 1, it is clear that (23) is negative definite for all $K_1\vec{\Omega} + K_2\vec{\vartheta} \neq 0$, therefore forcing the system to asymptotically converge to the manifold $\vec{\sigma} = K_1\vec{\Omega} + K_2\vec{\vartheta} \rightarrow 0$.

Note that since the system converges asymptotically towards $\vec{\sigma} \rightarrow 0$, it is safe to say that an arbitrary $\vec{\epsilon} = [\epsilon_x\ \epsilon_y\ \epsilon_z]^T$ with $\epsilon_x, \epsilon_y, \epsilon_z > 0$, can be defined such that there exists a time $t_\epsilon > 0$ where for each vector component $i = x, y, z$ the system dynamics reach $\sigma(t_\epsilon)_i \leq \epsilon_i$ such that for every $t > t_\epsilon$, it holds

$$\sigma(t)_i = \left(K_1\vec{\Omega}(t) + K_2\vec{\vartheta}(t)\right)_i < \epsilon_i \tag{24}$$

Propose a positive-definite function and its derivative as

$$V_2 = \frac{1}{2}\vec{\vartheta} \cdot \vec{\vartheta} \;\rightarrow\; \dot{V}_1 = \vec{\vartheta} \cdot \vec{\Omega}. \tag{25}$$

Introducing the behavior of $\vec{\Omega}$ relative to the sliding manifold from (18) into (25) yields

$$\dot{V}_2 = \vec{\vartheta} \cdot K_1^{-1}\left(\vec{\sigma} - K_2\vec{\vartheta}\right). \tag{26}$$

from (23) and (24), it can be inferred that t_ϵ can be defined as the instant when the system states will be bounded by $\vec{\epsilon} = [\epsilon_x\ \epsilon_y\ \epsilon_z]^T$, such that

$$\dot{V}_2 = \vec{\vartheta} \cdot K_1^{-1}\left(\vec{\sigma} - K_2\vec{\vartheta}\right) < \vec{\vartheta} \cdot K_1^{-1}\left(\vec{\epsilon} - K_2\vec{\vartheta}\right). \tag{27}$$

if matrices K_1 and K_2 are selected such that $K_1^{-1}K_2 > 0$, and $\vec{\epsilon} < K_2\vec{\vartheta}$, then it is ensured that $\dot{V}_2 < 0$ for all $\vec{\vartheta}, \vec{\Omega} \neq \vec{0}$ after $t > t_\epsilon$.

Ensuring that that $\vec{\Omega}$ and $\vec{\vartheta}$ asymptotically converge to zero and following (1), then $q \rightarrow q_{\mathcal{O}}$. Expanding the components of every axis from (21), it yields (13), (14) and (15). $\blacksquare$

Since the controller is based on (9), which employs a continuous arctangent function to define its magnitude, the algorithm yields a chattering-free behavior on the system. This makes it smooth for small attitude errors, yet strong for achieving large rotations, also reducing actuators fatigue.

V. ATTITUDE TRAJECTORY FORMULATION

An aggressive deployment implies the vehicle could be launched at any possible initial attitude. Thus it must be recovered from such conditions to a hovering state before handing over the baton to the user. To address this problem, a combination of two attitude references is proposed.

A. Recovery Trajectory

Firstly, to stop the vehicle from falling, its thrust vectors must compensate the gravitational acceleration as fast as possible. The representation of the vertical axis of the inertial frame $\mathcal{I}$ with respect to the body's coordinate system $\mathcal{B}$ can be defined by a unit vector $n_z^\beta \in \mathbb{R}^3$ as

$$n_z^\beta = \boldsymbol{q}^* \otimes n_z \otimes \boldsymbol{q}, \tag{28}$$

where $n_z := [0, 0, 1]^T$

Introducing the definitions of *dot* and *cross* products between two vectors, and widely known trigonometric expressions into (1) the shortest rotation between n_z^β and the vertical axis of the quadrotor can be computed as

$$\boldsymbol{q}_b := \left(\sqrt{\frac{1 + n_z^\beta \cdot n_z}{2}} + \frac{n_z^\beta \times n_z}{||n_z^\beta \times n_z||}\sqrt{\frac{1 - n_z^\beta \cdot n_z}{2}}\right), \tag{29}$$

since $\boldsymbol{q}_b$ is computed using vectors expressed in $\mathcal{B}$, the recovery trajectory quaternion is completed by adding the vehicle's rotation with respect to $\mathcal{I}$ as

$$\boldsymbol{q}_r = \boldsymbol{q} \otimes \boldsymbol{q}_b^*. \tag{30}$$

see Figure 4.
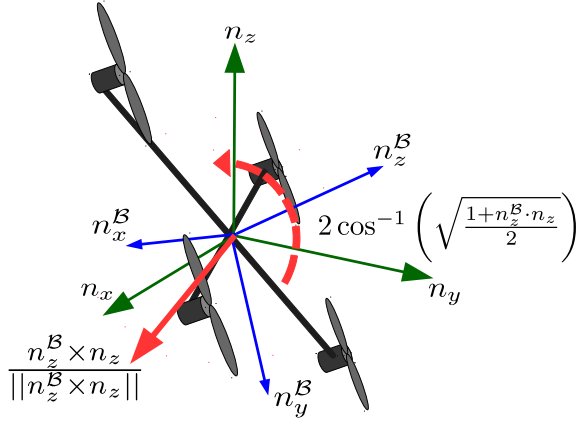
Figure 4: Shortest quaternion recovery rotation

VI. CONTINUOUS SWITCHING STRATEGY

When the quadrotor is launched aggressively without any thrust from the propellers, the accelerometers display near-zero value. This condition does not provide sufficient information to accurately estimate the vehicle attitude. Nevertheless, the acceleration signal can be used to detect the moment in which the quadrotor is deployed as Figure 5 illustrates.
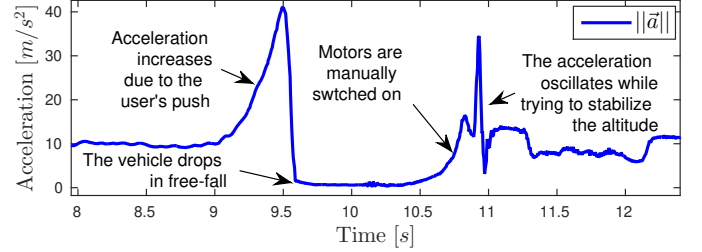

Figure 5: Acceleration profile example during manual recovery

B. Mission Specific Reference

Once the vehicle is recovered, the desired orientation might follow any reference according to the mission requirements. Some tasks like position holding or translational trajectory tracking, would define it in terms of a high-layer position control. Other applications like emergency response or outdoors video capture could require a human operator, whatever the case, the mission attitude reference is independent from the aggressive hand-launching deployment challenge. For repeatability and ease of performing experiments, in this work a human is considered to pilot the quadrotor in semi-autonomous mode by defining a desired orientation $\boldsymbol{q}_{usr}$ provided by a joystick.

Following Remark 1, the vehicle initial attitude might have either a positive or negative representation, one of them with an additional 2π rotation. This phenomenon might cause undesired behavior on the vehicle. To tackle this problem, the user input is complemented by changing its sign by defining

$$\boldsymbol{q}_j := \text{sign}(\pi - 2\ln(\boldsymbol{q}_{je}))\boldsymbol{q}_{usr}, \qquad (31)$$

where $\boldsymbol{q}_{je} = \boldsymbol{q}_{usr}^* \otimes \boldsymbol{q}$ symbolizes the rotation difference between the vehicle attitude and the user input.

C. Attitude Reference Combination

Once the recovery and user attitude references are defined, they are combined by the following function

$$\boldsymbol{q}_d := \boldsymbol{q}_r \otimes \boldsymbol{q}_{\xi r} \otimes \boldsymbol{q}_j \otimes \boldsymbol{q}_{\xi j}, \qquad (32)$$

where $\boldsymbol{q}_{\xi r}$ and $\boldsymbol{q}_{\xi j}$ are quaternions that enable and disable $\boldsymbol{q}_r$ and $\boldsymbol{q}_j$ as needed, and are derived from (1) as

$$\boldsymbol{q}_{\xi r} := e^{-\gamma_R \ln(\boldsymbol{q}_r)}, \qquad (33)$$

and

$$\boldsymbol{q}_{\xi j} := e^{-(1-\gamma_R)\ln(\boldsymbol{q}_j)}, \qquad (34)$$

where $0 < \gamma_R < 1$ is a scalar number which changes its value as needed to activate or deactivate each reference.

Remark 2: If $\gamma_R \to 0$, then $\boldsymbol{q}_{\xi r} \to 1 + [0\ 0\ 0]^T$, while $\boldsymbol{q}_{\xi j} \to \boldsymbol{q}_j^*$, and in consequence $\boldsymbol{q}_d \to \boldsymbol{q}_r$, inversely, if $\gamma_R \to 1$, then $\boldsymbol{q}_{\xi j} \to 1 + [0\ 0\ 0]^T$, while $\boldsymbol{q}_{\xi r} \to \boldsymbol{q}_r^*$, thus $\boldsymbol{q}_d \to \boldsymbol{q}_j$.

γ_R will be defined as a function of the vehicle's acceleration in the following section.

A. Control Terms Switching

A continuous switching technique is proposed to activate and deactivate the attitude terms from (21) when the accelerometer measurements drop by defining

$$\gamma_a := \frac{1}{2}\left(\tanh\left(\beta_a(||\vec{a}|| - \alpha_a)\right) + 1\right), \qquad (35)$$

where $\vec{a}$ represents the acceleration vector, $\beta_a > 0$ is a tuning gain, and α_a denotes the acceleration threshold from which free-fall condition is considered.

Note
$$\begin{aligned}\gamma_a \to 1 \ \ \text{if} \ \ ||\vec{a}|| > \alpha_a\,, \ \text{and} \\ \gamma_a \to 0 \ \ \text{if} \ \ ||\vec{a}|| < \alpha_a\,,\end{aligned} \qquad (36)$$

such that the attitude controller is rewritten as

$$\vec{\tau} = -K_{d1}\vec{\Omega} - \Gamma\left(\gamma_a K_p 2\ln\left(\boldsymbol{q}_d^* \otimes \boldsymbol{q}\right) + K_{d2}\vec{\Omega}\right) + \vec{\Omega} \times J\vec{\Omega}, \qquad (37)$$

notice when the vehicle is in free-fall, the attitude values are ignored and only the angular speed is regulated. Otherwise, the controller will track the desired rotation defined by $\boldsymbol{q}_d$.

B. Attitude Reference switching

Following a similar path, two scalar variables dependent on the vehicle attitude and angular velocity are proposed as

$$\gamma_\vartheta := \frac{1}{2}\left(\tanh\left(\beta_\vartheta(||2\ln(\boldsymbol{q}_{re})|| - \alpha_\vartheta)\right) + 1\right), \qquad (38)$$

and

$$\gamma_\Omega := \frac{1}{2}\left(\tanh\left(\beta_\Omega(||\vec{\Omega}|| - \alpha_\Omega)\right) + 1\right), \qquad (39)$$

where $\boldsymbol{q}_{re} := \boldsymbol{q}_r^* \otimes \boldsymbol{q}$ represents the rotation difference between the quadrotor attitude and the recovery quaternion from (30). $\beta_\vartheta, \beta_\Omega > 0$ are tuning parameters, and $\alpha_\vartheta, \alpha_\Omega$ symbolize the angle error and angular velocity thresholds to consider the vehicle as stabilized in hover mode.

Then, both terms from (38) and (39) are combined as

$$\gamma_R(t) := \tanh\left(\beta_R \int_{t_0}^{t} \gamma_\vartheta \gamma_\Omega dt\right), \quad \gamma_R(t_0) = 0, \qquad (40)$$

where t_0 defines the time when the quadrotor starts waiting to be launched and $\beta_R > 0$ is a tuning parameter.

Since (37) will stabilize the vehicle's attitude to the recovery values, then $\exists t$ such that $\forall t > t_h$ when the quadrotor attitude follows $||2\ln(\boldsymbol{q}_{re})|| < \alpha_\vartheta$ and $||\vec{\Omega}|| < \alpha_\Omega$. Therefore, (40) behaves as

$$\begin{aligned}\gamma_R \approx 0 \quad \text{for} \quad t_0 < t < t_h\,, \\ \gamma_R \to 1 \quad \text{for} \quad t > t_h\,.\end{aligned} \qquad (41)$$

Remark 3: (32), (40), and (41) imply that the attitude reference will be defined by the recovery trajectory before t_h, and by the user-defined input after t_h.

C. Motor Activation

Considering a symmetrical quadrotor, the force f_i for each motor can be computed using (8) and (37) as

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \frac{1}{4\,l} \begin{bmatrix} 2 & 2 & l & l \\ -2 & -2 & l & l \\ -2 & 2 & -l & l \\ 2 & -2 & -l & l \end{bmatrix} \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \\ F_{th} \end{bmatrix}. \quad (42)$$

The relation between the thrust force and angular speed ω_i of each motor (see Figure 2) can be computed using blade element theory as

$$f_i = C_T \rho \pi r^3 \omega_i^2, \quad (43)$$

where C_T denotes the aerodynamic thrust parameter, ρ represents the air density, and r symbolizes the propeller radius, see [23]. Inversely, the angular speed can be computed as

$$\omega_i = \sqrt{f_i/(C_T \rho \pi r^3)}. \quad (44)$$

It is a common practice to consider the aerodynamic parameters in (44) as a constant $k_i \approx \frac{1}{C_T \rho \pi r^3}$. Simplifying the relation between the propeller force and speed as

$$\omega_i \approx \sqrt{f_i/(k_i)}. \quad (45)$$

Since the motors are activated with a continuous function, damage from current peaks and sudden angular accelerations can be avoided. An expression to automatically activate the motors when the quadrotor is being tossed is proposed as

$$\gamma_\mu(t) := \tanh\left(\zeta_\mu \int_{t_0}^{t} \left(\tanh\left(\beta_\mu(||\vec{a}|| - \alpha_\mu)\right) + 1\right) dt\right), \quad (46)$$

where $\zeta_\mu, \beta_\mu > 0$ are tuning parameters, and α_μ denotes the acceleration threshold from which aggressive tossing is considered. Therefore (45) can be rewritten as.

$$\omega_i \approx \gamma_\mu(t) \sqrt{\frac{f_i}{k_i}}. \quad (47)$$

Defining time t_l as when the quadrotor is lauched such that $||\vec{a}|| > a_\mu$. Considering $\gamma_\mu(t_0) = 0$, then (46) follows

$$\begin{aligned} \gamma_\mu &\approx 0 \quad \text{for} \quad t_0 < t < t_l, \\ \gamma_\mu &\to 1 \quad \text{for} \quad t > t_l. \end{aligned} \quad (48)$$

D. Altitude recovery

Since the launching conditions are initially unknown, the vehicle might be far from the desired altitude when the attitude is being recovered. To smoothen the vehicle behavior, an altitude recovery trajectory has to be designed.

Define t_{z_0} as the time when the quadrotor is recovered in hover mode, which is determined according to

$$\begin{aligned} \gamma_R(t) &< 0.5 \quad \text{for} \quad t \in [t_0, t_{z_0}], \\ \gamma_R(t) &\geq 0.5 \quad \text{for} \quad t > t_{z_0}. \end{aligned} \quad (49)$$

t_{z_f} defines the time when the altitude reference reaches a value defined by the pilot z_{usr}. If that $z_d(t) > z_{\text{usr}}$ for all $t_{z_0} < t < t_{z_f}$, and z_0 denotes the initial altitude reference, determined as

$$z_0 = \max\{z(t_{z_0}), z_{\min}, z_{\text{usr}}\}, \quad (50)$$

where $z_{\min}$ and $z(t_{z_0})$ symbolize the minimum safety altitude, and the measured height at t_{z_0} respectively, then the altitude recovery trajectory is computed as

$$z_d(t) := \begin{cases} z_0 - \int_{t_{z_0}}^{t} v_{zr}(\varsigma) d\varsigma & \forall \quad t_{z_0} \leq t \leq t_{z_f}, \\ z_{\text{usr}} & \text{otherwise}, \end{cases} \quad (51)$$

where $v_{zr}(\varsigma) > 0$ is the desired descent speed.

VII. EXPERIMENTAL VALIDATION

Multiple tests have been performed using the proposed strategy on a Parrot ARDrone2 quadrotor. The algorithm was coded using the FL-AIR framework [38]. In some experiments, the vehicle was launched with the propellers pointing downwards, others while rotating over its three x (roll), y (pitch), and z (yaw) axes. Due to paper length limitations, only two scenarios are illustrated in this manuscript. However the reader is encouraged to watch more tests on-line at the following video link: https://youtu.be/b52e7K9BHYs

The controller parameters are manually adjusted, and the final values are introduced in the following table:

$\alpha_a = g/2$	$\beta_a = 10$	$\alpha_\theta = \pi/4$
$\beta_\theta = 10$	$\alpha_\omega = \pi/4$	$\beta_\omega = 10$
$\alpha_\mu = 2g$	$\beta_\mu = 10$	$\zeta_\mu = 10$
$\beta_R = 10$	$k_{pz} = 0.6$	$k_{dz} = 0.25$
$K_p = \text{diag}\left([3.5 \ \ 3.5 \ \ 2]\right)$	$K_{d1} = 0.065\,I_{3\times3}$	$K_{d2} = 0.5\,K_{d1}$

Table I: Controller parameters for real world tests.

A. First launching test: Single axis rotation

The first test consisted on launching the vehicle horizontally, smoothly rotating over its z (yaw) axis, for illustrating how the activation and recovery scheme stabilizes the quadrotor.

The quadrotor was set in the ground with its motors turned off, all the initial conditions were set to zero. The algorithm begins computing the controller from $t = 0s$. At $t \approx 4s$, an operator grabs the vehicle from the ground, then proceeds to launch it at $t \approx 7s$.

The switching terms computed using equations (35), (40), and (46) are depicted in Figure 6. Note that the system detects a launching, activates the motors with γ_μ, then between $7.3s < t < 8.1s$ free-fall condition is detected by γ_a. At $t \approx 8.5s$ the vehicle is stabilized horizontally, this is detected by γ_R and the command is passed to the pilot. Finally, the quadrotor is landed, and all the variables are reset to the initial conditions, preparing for a new deployment.
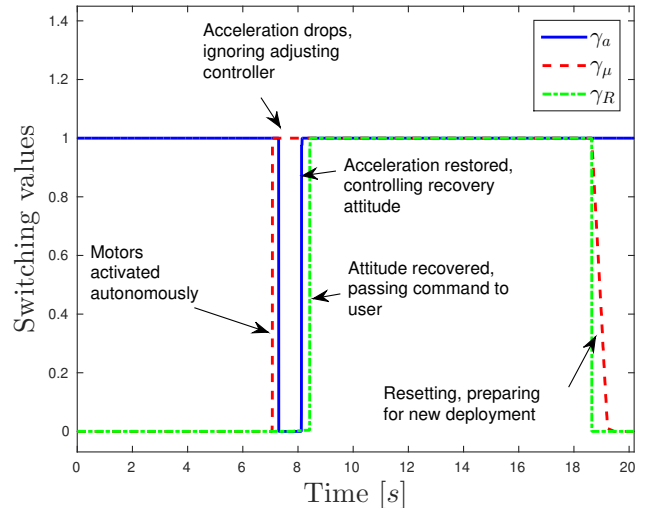

Figure 6: Automatic detection values during launching, recovery and landing for motor activation, enabling control parameters, and passing the command to the operator.

Figure 7 illustrates the recovery quaternion computed by (30), while the user-defined reference from (31) is depicted in Figure 8. Note the sign of the quaternion changes around $t \cong 8s$, because of the vehicle rotation over its yaw axis, the definition of q_j ensures the shortest rotation is tracked.
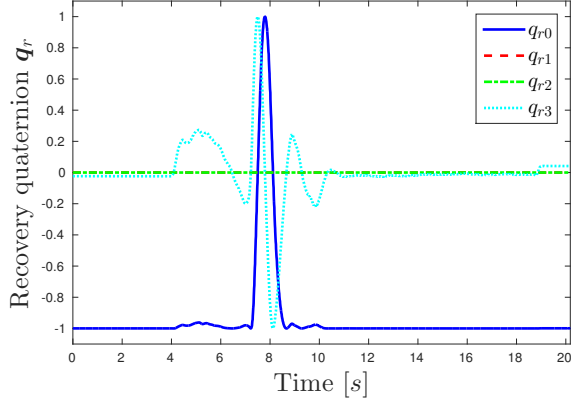

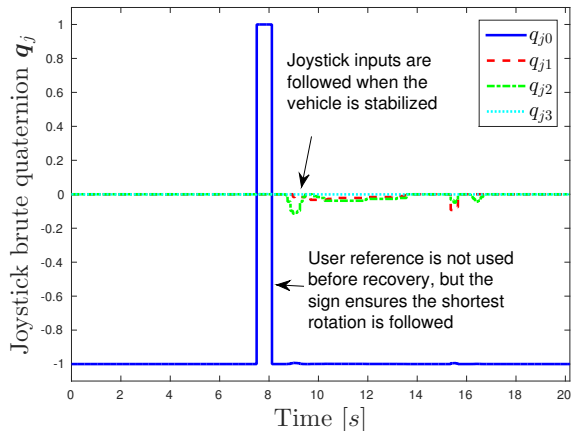Figure 7: Autonomous recovery quaternion, computing the shortest rotation required to stabilize the vehicle.

Figure 8: The user-defined reference is ignored during the launching, but enabled once stable hovering is achieved.

Following (32), the recovery and user-defined references are combined, resulting in the total desired orientation q_d, which is illustrated in Figure 9. Note by comparing Figures 7 to 9 that q_d is similar to q_r that before $t \cong 9s$, but then it follows the reference given by q_j with a smooth switch.
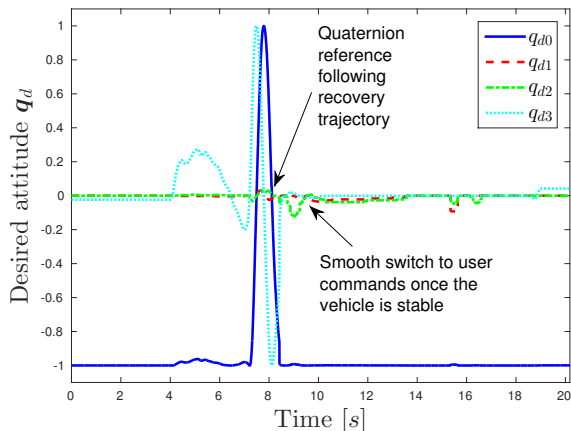
Figure 9: The recovery quaternion and user-defined references are combined by (32) which smoothly switches from one to the other according to the launching and recovery conditions.

Figures 10 to 13 depict the attitude quaternion response during the first experiment. Figure 10 illustrates the real part of q and q_d, symbolizing the cosines of the total rotation.
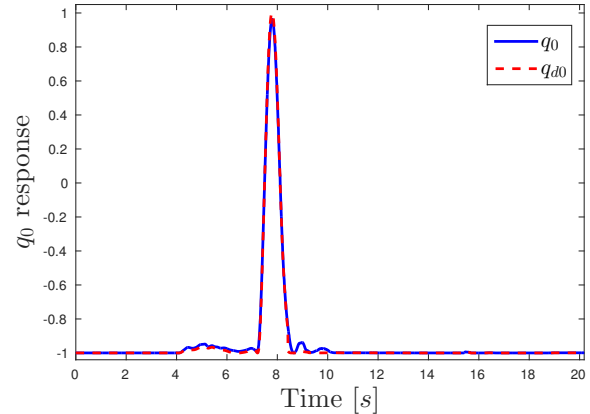

Figure 10: q_0 component response, the peak indicates two complete rotations are performed during the deployment, accomplishing stabilization afterwards.

Figures 11 to 13 represent the rotation around the x, y, and z axes. Note the vehicle starts its launching far from the reference in the x and y axes but recovers in-flight. The rotation around the z axis is tracked by the recovery quaternion and then smoothly switched to the user reference.
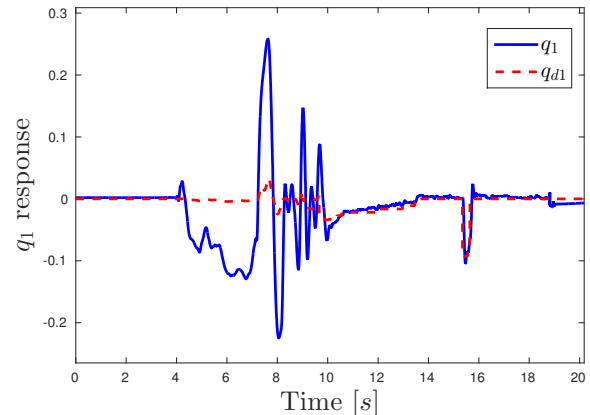

Figure 11: q_1 (x axis), the attitude values oscillate during the deployment, indicating an adjustment of the orientation to achieve stable hovering.
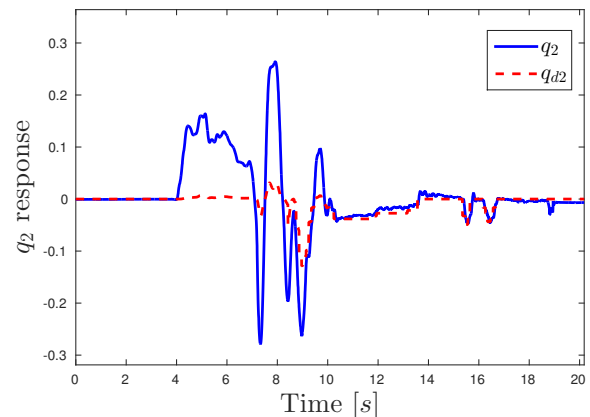

Figure 12: q_2 (y axis), displaying an aggressive rotation when recovering the drone, then a smooth attitude tracking.
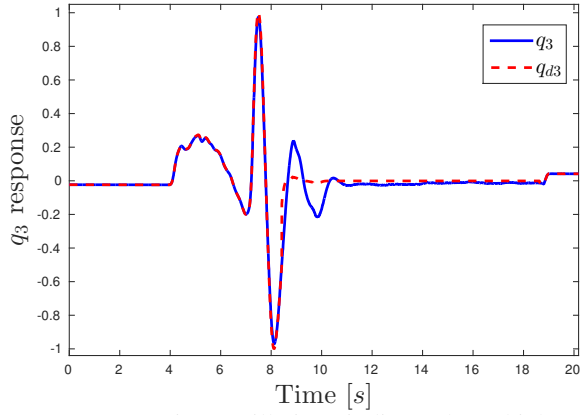
Figure 13: q_3 (z axis), oscillations indicate the vehicle rotates twice over this axis before achieving recovery.

Although the control algorithms are based on a quaternion approach, Figure 14 depicts the equivalent Euler yaw angle to better illustrate the vehicle's behavior. Note its rotation reveals the passing through two discontinuities during the test. This is avoided by the algorithm's quaternion formulation.
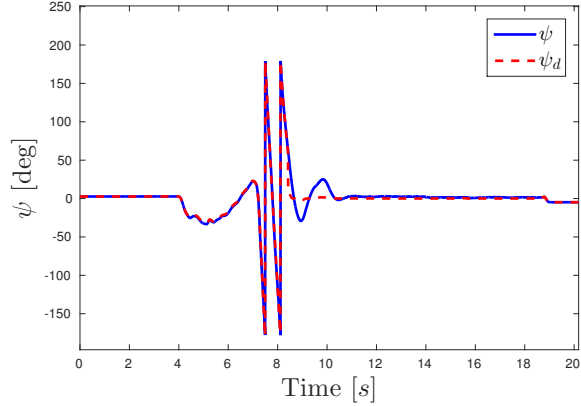

Figure 14: Equivalent yaw angle, revealing discontinuities when two 360° rotations are completed.

The altitude and vertical velocity responses are depicted in Figure 15. The altitude is measured using an ultrasonic sensor and a Kalman filter is implemented to improve the position signal and estimate the vertical speed.
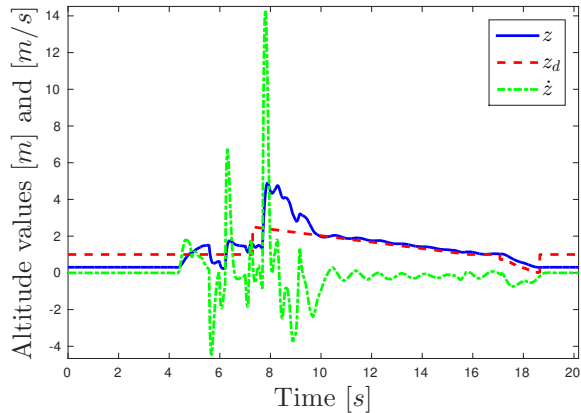

Figure 15: Altitude and vertical speed response, the quadrotor smoothly descends to the desired height after recovery.

Note the position reference follows a ramp descend the vehicle to the user-defined altitude (in this case $z_{usr} = 0.5m$) once it is recovered in hover mode. At $t \approx 17s$ the vertical reference lands the vehicle, preparing for a new deployment.

B. Second launching test: Multiple axis rotation

The second experiment was performed to prove the recovery algorithm under extreme deployment conditions. Here, the vehicle was launched while rotating multiple times over all of its three axes, passing through vertical inclinations, and even pointing its propellers downwards at some times. This kind of deployment has been tested multiple times, successfully recovering the vehicle.

Figure 16 illustrates the launching detection timing by the switching parameters from (35) and (46). The quadrotor rotational response is represented in Figures 17 to 20.
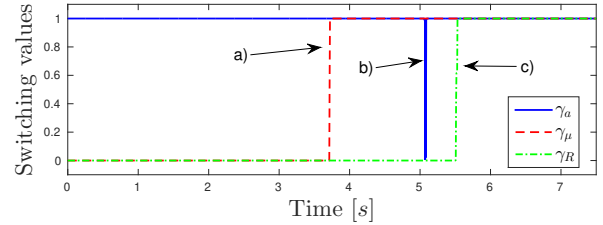

Figure 16: Detection terms performance, a) Tossing detected, starting motors. b) Free fall detected. c) Recovery achieved.
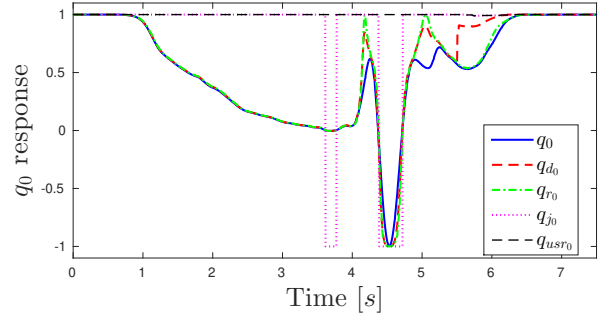

Figure 17: Quaternion q_0 component response, each change of sign indicates a 360^0 rotation during deployment.
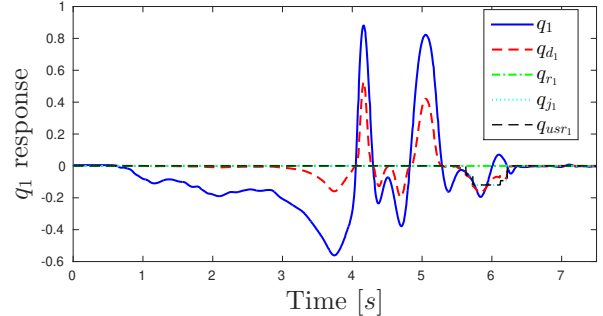

Figure 18: Quaternion q_1 component (over the x axis), the vehicle is first inclined, then rotates aggressively during the recovery stage.
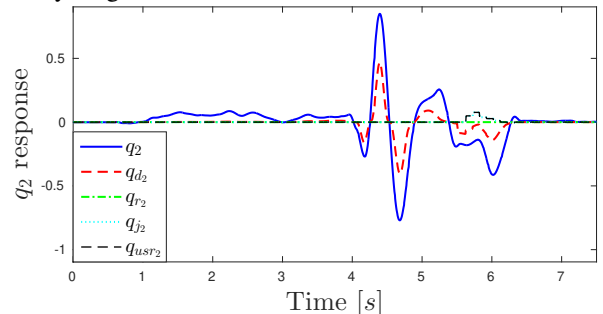

Figure 19: Quaternion q_2 component (response over the y axis), note how the combined rotation $\boldsymbol{q}_d$ pulls the vehicle attitude towards zero.
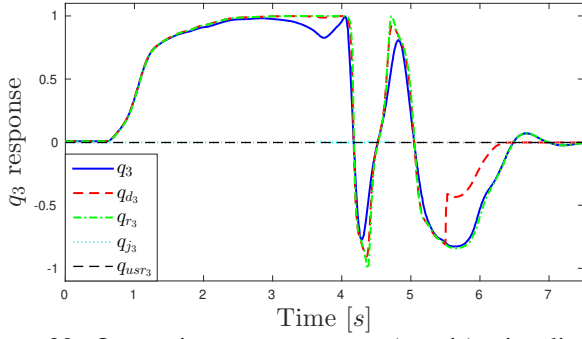
Figure 20: Quaternion q_3 component (z axis), signaling the vehicle also spins three times before stabilizing successfully.

For illustration purposes, Figures 21 to 23 depict the equivalent Euler angles computed from the previous quaternions. Due to the nature of this type of attitude representation, some undesired effects are revealed such as discontinuities and sudden changes in the signals.

Notice how the magnitude of the simultaneous pitch and roll rotations, along with their oscillations indicate the vehicle is aggressively rotated such that it passes through vertical, horizontal, and upside-down poses. Nevertheless, at the end, the proposed algorithm successfully recovers the quadrotor.
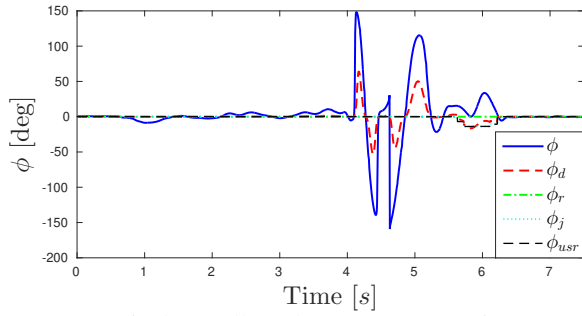

Figure 21: Equivalent roll angle response (rotation over the x axis), reaching almost $150°$ in multiple occasions.
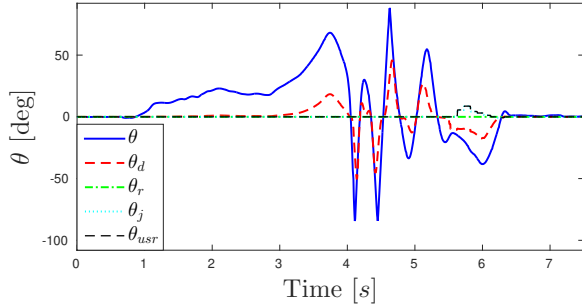

Figure 22: Equivalent pitch angle response (rotation over the y axis), approaching vertical poses of $90°$ several times.
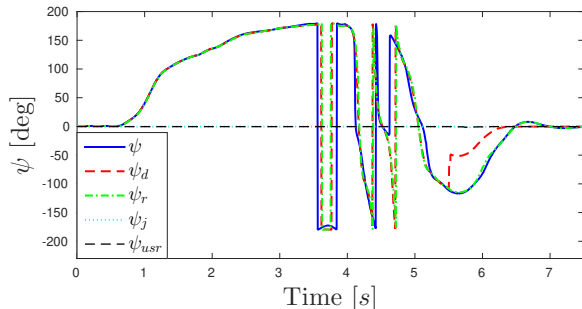

Figure 23: Equivalent yaw angle (rotation over the z axis), revealing multiple $360°$ turns before stabilizing.

VIII. Conclusion

An aggressive deployment strategy for a quadrotor was proposed in which the vehicle is launched with its motors turned off. In order to activate the vehicle in-flight, a continuous recovery trajectory was proposed employing unit quaternions. The vehicle thrust vector is aligned from any attitude to the vertical axis using the shortest rotation.

The proposal considers that once the vehicle is recovered, the desired orientation might follow any reference according to the mission requirements. This reference is introduced in a continuous switching equation. The control the control terms are also smoothly switched according to the vehicle sensors, seamlessly combining the recovery attitude with the reference required by the mission.

A quaternion-based chattering-free spherical sliding mode controller was proposed using Lyapunov theory to robustly track the aforementioned trajectory. Real-world experiments were executed for its validation, displaying a very good repeatable performance.

The proposed strategy can be implemented in real-world applications where a quadrotor (or another kind of multicopter) could be launched with its motors switched off, without needing a ground surface, and reducing its starting deployment time, which could be very valuable in urgency scenarios.

References

[1] ABAUNZA, H., CARIÑO, J., AND CASTILLO, P. Chapter 2 modeling approaches, book: Indoor navigation strategies for aerial autonomous systems,p. castillo, l. e. munoz, p. garcia, 2017.

[2] BOROWCZYK, A., NGUYEN, D.-T., NGUYEN, A. P.-V., NGUYEN, D. Q., SAUSSIÉ, D., AND LE NY, J. Autonomous landing of a quadcopter on a high-speed ground vehicle. Journal of Guidance, Control, and Dynamics 40, 9 (2017), 2378–2385.

[3] CHUNG, S. C.-Y., AND LIN, C.-L. A transformed lure problem for sliding mode control and chattering reduction. IEEE Transactions on Automatic Control 44, 3 (1999), 563–568.

[4] CRASSIDIS, J. L., AND MARKLEY, F. L. Sliding mode control using modified rodrigues parameters. Journal of Guidance, Control, and Dynamics 19, 6 (1996), 1381–1383.

[5] CRISTI, R., BURL, J., AND RUSSO, N. Adaptive quaternion feedback regulation for eigenaxis rotations. Journal of Guidance, Control, and Dynamics 17, 6 (1994), 1287–1291.

[6] DE LELLIS, E., MORANI, G., CORRARO, F., AND DI VITO, V. On-line trajectory generation for autonomous unmanned vehicles in the presence of no-fly zones. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 227, 2 (2013), 381–393.

[7] DRAKUNOV, S. V., AND UTKIN, V. I. Sliding mode control in dynamic systems. International Journal of Control 55, 4 (1992), 1029–1037.

[8] EDWARDS, C., AND SPURGEON, S. Sliding mode control: theory and applications. Crc Press, 1998.

[9] ERTUGRUL, M., KAYNAK, O., SABANOVIC, A., AND OHNISHI, K. A generalized approach for lyapunov design of sliding mode controllers for motion control applications. In Proceedings of 4th IEEE International Workshop on Advanced Motion Control-AMC'96-MIE (1996), vol. 1, IEEE, pp. 407–412.

[10] FAESSLER, M., FONTANA, F., FORSTER, C., AND SCARAMUZZA, D. Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor. In Robotics and Automation (ICRA), 2015 IEEE International Conference on (2015), IEEE, pp. 1722–1729.

[11] FALANGA, D., MUEGGLER, E., FAESSLER, M., AND SCARAMUZZA, D. Aggressive quadrotor flight through narrow gaps with on-board sensing and computing using active vision. In Robotics and Automation (ICRA), 2017 IEEE International Conference on (2017), IEEE, pp. 5774–5781.

[12] GALZI, D., AND SHTESSEL, Y. Unmanned rotorcraft tight formation flight control using sliding mode control driven by sliding mode disturbance observers. In AIAA Guidance, Navigation and Control Conference and Exhibit (2008), p. 7171.

[13] GHIGLINO, P., AND FORSHAW, J. L. Hex2oqtal: Translational optimal control exploiting quaternion error dynamics. IEEE Transactions on Aerospace and Electronic Systems 53, 3 (2017), 1181–1195.

[14] GHIGLINO, P., FORSHAW, J. L., AND LAPPAS, V. J. Oqtal: Optimal quaternion tracking using attitude error linearization. IEEE Transactions on Aerospace and Electronic Systems 51, 4 (2015), 2715–2731.

[15] GUERRERO-CASTELLANOS, J., MARCHAND, N., HABLY, A., LESECQ, S., AND DELAMARE, J. Bounded attitude control of rigid bodies: Real-time experimentation to a quadrotor mini-helicopter. Control Engineering Practice 19, 8 (2011), 790–797.

[16] HU, Q., SHAO, X., AND CHEN, W.-H. Robust fault-tolerant tracking control for spacecraft proximity operations using time-varying sliding mode. IEEE Transactions on Aerospace and Electronic Systems 54, 1 (2018), 2–17.

[17] HWANGBO, J., SA, I., SIEGWART, R., AND HUTTER, M. Control of a quadrotor with reinforcement learning. IEEE Robotics and Automation Letters 2, 4 (2017), 2096–2103.

[18] IZAGUIRRE-ESPINOSA, C., MUÑOZ-VÁZQUEZ, A., SÁNCHEZ-ORTA, A., PARRA-VEGA, V., AND SANAHUJA, G. Fractional attitude-reactive control for robust quadrotor position stabilization without resolving underactuation. Control Engineering Practice 53 (2016), 47–56.

[19] JOOS, A., HAEFNER, A., WEIMER, F., AND FICHTER, W. Quadrocopter ground effect compensation with sliding mode control. In AIAA Guidance, Navigation, and Control Conference (2010), p. 7871.

[20] KOTHARI, M., POSTLETHWAITE, I., AND GU, D.-W. Multi-uav path planning in obstacle rich environments using rapidly-exploring random trees. In Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference (2009), IEEE, pp. 3069–3074.

[21] KUIPERS, J. B., ET AL. Quaternions and rotation sequences, vol. 66. Princeton university press Princeton, 1999.

[22] LANDRY, B., DEITS, R., FLORENCE, P. R., AND TEDRAKE, R. Aggressive quadrotor flight through cluttered environments using mixed integer programming. In Robotics and Automation (ICRA), 2016 IEEE International Conference on (2016), IEEE, pp. 1469–1475.

[23] LEISHMAN, G. J. Principles of helicopter aerodynamics with CD extra. Cambridge university press, 2006.

[24] LEVANT, A. Sliding order and sliding accuracy in sliding mode control. International journal of control 58, 6 (1993), 1247–1263.

[25] LEVANT, A. Quasi-continuous high-order sliding-mode controllers. In 42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475) (2003), vol. 5, IEEE, pp. 4605–4610.

[26] LING, Y., LIU, T., AND SHEN, S. Aggressive quadrotor flight using dense visual-inertial fusion. In Robotics and Automation (ICRA), 2016 IEEE International Conference on (2016), IEEE, pp. 1499–1506.

[27] LO, S.-C., AND CHEN, Y.-P. Smooth sliding-mode control for spacecraft attitude tracking maneuvers. Journal of Guidance, Control, and Dynamics 18, 6 (1995), 1345–1349.

[28] LOIANNO, G., BRUNNER, C., MCGRATH, G., AND KUMAR, V. Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu. IEEE Robotics and Automation Letters 2, 2 (2017), 404–411.

[29] LU, Q., REN, B., AND PARAMESWARAN, S. Shipboard landing control enabled by an uncertainty and disturbance estimator. Journal of Guidance, Control, and Dynamics 41, 7 (2018), 1502–1520.

[30] LUONGO, S., DI VITO, V., FASANO, G., ACCARDO, D., FORLENZA, L., AND MOCCIA, A. Automatic collision avoidance system: design, development and flight tests. In 2011 IEEE/AIAA 30th Digital Avionics Systems Conference (2011), IEEE, pp. 5C1–1.

[31] OLIVA-PALOMO, F., MUÑOZ-VÁZQUEZ, A., SÁNCHEZ-ORTA, A., PARRA-VEGA, V., IZAGUIRRE-ESPINOSA, C., AND CASTILLO, P. A fractional nonlinear pi-structure control for robust attitude tracking of quadrotors. IEEE Transactions on Aerospace and Electronic Systems (2019).

[32] PARK, J., AND KIM, Y. Collision avoidance for quadrotor using stereo vision depth maps. IEEE Transactions on Aerospace and Electronic Systems 51, 4 (2015), 3226–3241.

[33] PARK, J., AND KIM, Y. Horizontal-vertical guidance of quadrotor for obstacle shape mapping. IEEE Transactions on Aerospace and Electronic Systems 52, 6 (2016), 3024–3035.

[34] PISANO, A., RAPAIĆ, M. R., JELIČIĆ, Z. D., AND USAI, E. Sliding mode control approaches to the robust regulation of linear multivariable fractional-order dynamics. International Journal of Robust and Nonlinear Control 20, 18 (2010), 2045–2056.

[35] POLYAKOV, A., AND POZNYAK, A. Lyapunov function design for finite-time convergence analysis:"twisting" controller for second-order sliding mode realization. Automatica 45, 2 (2009), 444–448.

[36] POLYAKOV, A., AND POZNYAK, A. Reaching time estimation for "super-twisting" second order sliding mode controller via lyapunov function designing. IEEE Transactions on Automatic Control 54, 8 (2009), 1951–1955.

[37] PUKDEBOON, C., ZINOBER, A. S., AND THEIN, M.-W. L. Quasi-continuous higher order sliding-mode controllers for spacecraft-attitude-tracking maneuvers. IEEE Transactions on Industrial Electronics 57, 4 (2010), 1436–1444.

[38] SANAHUJA, G., AND ET AL. Fl-AIR - Framework Libre AIR. https://uav.hds.utc.fr/software-flair/. Accessed: 2019-01-11.

[39] SLOTINE, J.-J. E., LI, W., ET AL. Applied nonlinear control, vol. 199. Prentice hall Englewood Cliffs, NJ, 1991.

[40] SPRING, K. W. Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: a review. Mechanism and machine theory 21, 5 (1986), 365–373.

[41] TAFAZOLI, S., AND KHORASANI, K. Nonlinear control and stability analysis of spacecraft attitude recovery. IEEE Transactions on Aerospace and Electronic Systems 42, 3 (2006), 825–845.

[42] UTKIN, V. Variable structure systems with sliding modes. IEEE Transactions on Automatic control 22, 2 (1977), 212–222.

[43] UTKIN, V. Discussion aspects of high-order sliding mode control. IEEE Transactions on Automatic Control 61, 3 (2015), 829–833.

[44] WANG, H., YE, X., TIAN, Y., ZHENG, G., AND CHRISTOV, N. Model-free–based terminal smc of quadrotor attitude and position. IEEE Transactions on Aerospace and Electronic Systems 52, 5 (2016), 2519–2528.

[45] YIN, C., CHEN, Y., AND ZHONG, S.-M. Fractional-order sliding mode based extremum seeking control of a class of nonlinear systems. Automatica 50, 12 (2014), 3173–3181.

[46] YOUNG, K. D., UTKIN, V. I., AND OZGUNER, U. A control engineer's guide to sliding mode control. In Proceedings. 1996 IEEE International Workshop on Variable Structure Systems.-VSS'96- (1996), IEEE, pp. 1–14.

[47] ZHAO, B., XIAN, B., ZHANG, Y., AND ZHANG, X. Nonlinear robust sliding mode control of a quadrotor unmanned aerial vehicle based on immersion and invariance method. International Journal of Robust and Nonlinear Control 25, 18 (2015), 3714–3731.

[48] ZHU, L., CHENG, X., AND YUAN, F.-G. A 3d collision avoidance strategy for uav with physical constraints. Measurement 77 (2016), 40–49.

[49] ZHU, Z., XIA, Y., FU, M., ET AL. Adaptive sliding mode control for attitude stabilization with actuator saturation. IEEE Transactions on Industrial Electronics 58, 10 (2011), 4898–4907.

Hernan Abaunza was born in Morelos, Mexico, on October 1, 1990. He received the B.S. degree in mechatronic engineering from the ITESM (Mexico) in 2012, the M. Sc. degree in autonomous systems for aerial and submarine navigation from the Centro de Investigación y de Estudios Avanzados (CINVESTAV) (Mexico) in 2014, and his Ph.D. degree in Automatic Control from the University of Technology of Compiègne (France) in 2019. His research topics include real-time control applications, quaternion-based modeling and control, and autonomous navigation of unmanned aerial and ground vehicles.

Pedro Castillo was born in Morelos, Mexico, on January 8, 1975. He received the B. S. degree in electromechanic engineering from the Instituto Tecnologico de Zacatepec (Mexico) in 1997, the M. Sc. degree in electrical engineering from the Centro de Investigación y de Estudios Avanzados (CINVESTAV) (Mexico) in 2000, and the Ph.D. degree in automatic control from the University of Technology of Compiègne (France) in 2004. His research topics include real-time control applications, nonlinear dynamics and control, aerospace vehicles, vision, and underactuated mechanical systems.