Open access • Journal Article • DOI:10.1080/00207179.2018.1502474

Flat trajectory design and tracking with saturation guarantees: a nano-drone application — Source link

Ngoc Thinh Nguyen, Ionela Prodan, Laurent Lefèvre

Institutions: University of Grenoble

Published on: 02 Jun 2020 - International Journal of Control (Taylor & Francis)

Topics: Trajectory and Quadcopter

Flat trajectory design and tracking with saturation guarantees: a nano-drone application

Ngoc Thinh Nguyen, Ionela Prodan, Laurent Lefèvre

Flat trajectory design and tracking with saturation guarantees: a nano-drone application

Ngoc Thinh Nguyen, Ionela Prodan and Laurent Lefèvre

Univ. Grenoble Alpes, Grenoble INP, LCIS, F-26000, Valence, France

ARTICLE HISTORY
Compiled June 27, 2018

ABSTRACT
This paper deals with the problem of trajectory planning and tracking of a quadcopter system based on the property of differential flatness. Firstly, B-splines characterizations of the flat output allow for optimal trajectory generation subject to waypoints constraints, thrust and angles constraints while minimizing the trajectory length. Secondly, the proposed tracking control strategy combines feedback linearization and nested saturation control via flatness. The control strategy provides bounded inputs (thrust, roll, pitch angles) while ensuring the overall stability of the tracking error dynamics. The control parameters are chosen based on the information of the a priori given reference trajectory. Moreover, conditions for the existence of these parameters are presented. The effectiveness of the trajectory planning and the tracking control design are analyzed and validated through simulation and experiments results over a real nano quadcopter platform, the Crazyflie 2.0.

KEYWORDS
Trajectory planning; Trajectory tracking; Differential flatness; B-splines parametrization; Feedback linearization; Nested saturation control, Nano quadcopter platform

1. Introduction

UAVs (Unmanned Aerial Vehicles) are rapidly growing in popularity within the research and industrial communities (Do, 2015; Hassanalian & Abdelkefi, 2017; Prodan et al., 2013). Even if they are still in the very early stage of global usage, the miniature flying robots technology started to develop and prosper as more and more industries are realizing their potential and scope (Hassanalian & Abdelkefi, 2017; Kerma, Mokhtari, Abdelaziz, & Orlov, 2012). A particular type of drones, the quadcopter systems manifest highly coupled multivariable dynamics and underactuated configuration (Lu, Liu, Guo, & Chen, 2017; Nguyen, Prodan, & Lefèvre, 2017; Shi, Zhang, & Zhou, 2015), hence related problems of constrained trajectory generation and tracking mechanism still remain open. One feasible approach is to off-line generate a reference trajectory which respects certain specific objectives (e.g., waypoints tracking, state/input constraints satisfaction), and then, design an effective on-line tracking mechanism (Cao & Lynch, 2016; Lu et al., 2017).

In the literature, there are many control methods for quadcopter trajectory tracking which ultimately reduce to some variant of the feedback linearization control proce-

Email: {ngoc-thinh.nguyen,ionela.prodan,laurent.lefevre}@lcis.grenoble-inp.fr

dure (Aguilar-Ibáñez, Sira-Ramírez, Suárez-Castañón, Martínez-Navarro, & Moreno-Armendariz, 2012; Formentin & Lovera, 2011; Nguyen, Prodan, Stoican, & Lefèvre, 2017; Zhao & Go, 2014). A shortcomming of most of these methods is that they do not consider constraints on the inputs. Henceforth, bounding the inputs changes the the system's expected behavior and may lead to undesired consequences (see, e.g. the anti-windup issue (Wu & Lu, 2004)). Maggiore (2015) provides a feedback law for stabilizing the quadcopter at a fixed position, allowing the system to have a bounded thrust and the converging input bounded state stability property. Furthermore, Cao and Lynch (2016) design a feedback control scheme which provides bounded thrust, roll and pitch angles based on the BF (Body Frame) consideration. However, it requires a convoluted transformation of the reference from the IF (Inertial Frame) to the BF since the reference trajectory is usually designed based on the IF, e.g., passing through waypoints which are determined w.r.t. the IF. Moreover, when considering the IF representation of the system, Cao and Lynch (2016) state that it is impossible to impose bounds on these two angles separately due to the existence of the direction angle in the roll and pitch angles.

The framework followed here is based on previous results of the authors (Nguyen, Prodan, & Lefèvre, 2018; Nguyen, Prodan, Stoican, & Lefèvre, 2017) and in the rest of the paper, we present several contributions which, to the best of our knowledge, are new to the state of the art:

i) propose a feedback linearization controller facilitated by nested control design for quadcopter trajectory tracking based on the IF which provides bounded inputs (thrust and roll, pitch angles) (as opposed to Cao and Lynch (2016)) and allows the overall system to have the converging input bounded state stability.
ii) propose a modification on the original nested control design (Teel, 1992) which allows the system to have larger saturation limits which vary depending on the a priori given references, thus, exploits more capability of the system than the fixed saturation limits (Teel, 1992).
iii) propose a condition for choosing the angle bounds employed in the control design based on the a priori given trajectory and a condition for ensuring the existence of all the related parameters consisting of the reference trajectory, the control design and the limit of system. Thus, we create an unified design scheme for trajectory generation and tracking with bounded thrust and bounded angles while respecting the physical constraints of the system.
iv) validate the control method through simulation and experimental testing over the nano quadcopter Crazyflie 2.0 platform (Giernacki, Skwierczyński, Witwicki, Wroński, & Kozierski, 2017).

The outline of the paper is organized as follows. Section 2 briefly presents the translational dynamics of a quadcopter system and further provides information related to the built-in controller and the constraints of the Crazyflie (CF) quadcopter platform. Section 3 presents the constrained trajectory offline generation by using differential flatness and B-spline parametrization. Section 4 details a feedback linearization control design of a quadcopter system. Then, nested control design is employed for design trajectory tracking corrective terms with bounded inputs. Several conditions are proposed to ensure the existence of all the desired parameters. Extensive simulation and experimental results are provided in Section 5 over a Crazyflie quadcopter system. Section 6 presents the conclusions and future work.

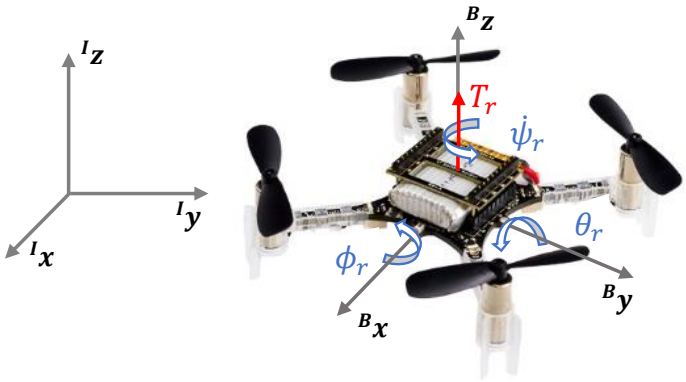

Figure 1. Inertial fram $\mathcal{I}$, body frame $\mathcal{B}$ and the four control inputs of the Crazyflie quadcopter platform. The built-in controller of the quadcopter controls the four rotors to track the quadruple of $\{T_r, \phi_r, \theta_r, \dot{\psi}_r\}$ given by user.

2. Dynamical model and its flatness-based characterization

This section briefly introduces the dynamical model that we employ to control the Crazyflie (CF) quadcopter system (shown in Fig.1) (Giernacki et al., 2017; Luis & Ny, 2016).

2.1. Dynamical model

As illustrated in Fig.1, the quadcopter will operate in two different coordinate systems: the *body frame* $\mathcal{B} = \{^{B}\boldsymbol{x}, ^{B}\boldsymbol{y}, ^{B}\boldsymbol{z}\}$[1] which is attached to the mass center and the *inertial frame* $\mathcal{I} = \{^{I}\boldsymbol{x}, ^{I}\boldsymbol{y}, ^{I}\boldsymbol{z}\}$ which is fixed to the ground. The attitude of the quadcopter is defined by the orientation of frame $\mathcal{B}$ w.r.t. frame $\mathcal{I}$. In general, this relation is described through a 3D rotation matrix which is the product of the sequence of three successive rotations. For the quadcopter we apply the roll–pitch–yaw XYZ (ϕ, θ, ψ) sequence which is usually used in aerospace research (Lu et al., 2017; Nguyen, Prodan, Stoican, & Lefèvre, 2017). I.e., firstly rotate the quadcopter a roll angle, ϕ, around $^{I}\boldsymbol{x}$ axis, next, a pitch angle, θ, around $^{I}\boldsymbol{y}$ axis and finally, a yaw angle, ψ, around $^{I}\boldsymbol{z}$ axis (see also Fig.1 for corresponding directions). As a result, the associated rotation matrix is given as [2]:

$$
{}^{\mathcal{I}}_{\mathcal{B}}R = \begin{bmatrix} \mathrm{c}\,\theta\,\mathrm{c}\,\psi & \mathrm{s}\,\phi\,\mathrm{s}\,\theta\,\mathrm{c}\,\psi - \mathrm{c}\,\phi\,\mathrm{s}\,\psi & \mathrm{c}\,\phi\,\mathrm{s}\,\theta\,\mathrm{c}\,\psi + \mathrm{s}\,\phi\,\mathrm{s}\,\psi \\ \mathrm{c}\,\theta\,\mathrm{s}\,\psi & \mathrm{s}\,\phi\,\mathrm{s}\,\theta\,\mathrm{s}\,\psi + \mathrm{c}\,\phi\,\mathrm{c}\,\psi & \mathrm{c}\,\phi\,\mathrm{s}\,\theta\,\mathrm{s}\,\psi - \mathrm{s}\,\phi\,\mathrm{c}\,\psi \\ -\mathrm{s}\,\theta & \mathrm{s}\,\phi\,\mathrm{c}\,\theta & \mathrm{c}\,\phi\,\mathrm{c}\,\theta \end{bmatrix}, \tag{1}
$$

Assuming the centrifugal force and external perturbation force (e.g., friction) are nullified, there are only gravitational force (along the negative $^{I}\boldsymbol{z}$ axis) and thrust force (along the positive $^{B}\boldsymbol{z}$ axis) which affect the translation dynamics of the quadcopter:

$$
\ddot{\xi} = -g\,{}^{I}\boldsymbol{z} + {}^{\mathcal{I}}_{\mathcal{B}}R\,{}^{B}\boldsymbol{z}T, \tag{2}
$$

[1] The left superscript notations used here are adopted from Craig (2005); Inaba and Corke (2016).

[2] Note that, in order to write in a more compact way we have used in this paper $'s'$, $'c'$ and $'t'$ to denote the $\sin(\cdot)$, $\cos(\cdot)$ and $\tan(\cdot)$ functions, respectively.

where g is the gravity. $\xi \triangleq \left[x\ y\ z\right]^\top$ represents the position w.r.t frame $\mathcal{I}$ and the thrust force has the normalized magnitude T, i.e., the thrust magnitude divided by the mass of the quadcopter, having the same unit with the gravity (m/s^2).
By using the built-in controllers, the CF quadcopter can control the four rotors to track the thrust input denoted as T_r, the roll, pitch angle inputs, denoted as $\phi_r,\ \theta_r$ and the yaw rate input denoted as $\dot{\psi}_r$. Since we do not modify these inner controllers, we also neglect the rotation dynamics and the rotors configuration of the CF quadcopter system which were detailed in Nguyen, Prodan, Stoican, and Lefèvre (2017) and Giernacki et al. (2017). The built-in controllers of CF contain two loops (Giernacki et al., 2017; Luis & Ny, 2016): *i)* an attitude PID controller which compares the angle inputs, $\phi_r,\ \theta_r$, and the real angles, $\phi,\ \theta$ received as the feedback from CF, then, provides the references of the roll, pitch angle rates; *ii)* a PID rate controller which compares the rate references included the foregoing yaw rate input $\dot{\psi}_r$, and the real angle rates obtained from CF quadcopter in order to calculate the torques. Finally, the torques and the thrust input T_r are transformed into the four rotor speeds by using the appropriate configuration (i.e., X configuration for the CF quadcopter (Luis & Ny, 2016)).

Remark 1. In our work, the built-in controller is assumed to be capable of tracking the roll, pitch angle inputs, $\phi_r,\ \theta_r$, i.e.:

$$\phi(t) \to \phi_r(t) \text{ and } \theta(t) \to \theta_r(t) \text{ as } t \to \infty. \tag{3}$$

We also neglect the transient effect on tracking the thrust input T_r along the ${}^{\boldsymbol{B}}\boldsymbol{z}$ axis (i.e., the response is fast enough not to cause much delays in the system (Luis & Ny, 2016)). Hence, we consider only the saturation effect for the real normalized thrust T as follows:

$$T = \min\left(T_r, T_{limit}\right), \tag{4}$$

where the normalized thrust limit of the CF quadcopter is given as $T_{limit} = 20.18\, m/s^2$ (Luis & Ny, 2016). □

2.2. Flatness-based characterization

Differential flatness represents a generalization of the structural properties of the linear systems to nonlinear systems, which exhibit a state representation obtained via derivatives of the input and output signals (Lévine, 2011). It allows us to implicitly validate the dynamics taking into account constraints (with some difficulties) and even provide a feedback law to linearize the nonlinear system. According to Nguyen, Prodan, Stoican, and Lefèvre (2017), the quadcopter system is differentially flat with the associated flat output $\mathbf{z} = \left[z_1\ z_2\ z_3\ z_4\right]^\top$ defined as follows:

$$\mathbf{z} = \left[x\ y\ z\ \tan\left(\frac{\psi}{2}\right)\right]^\top. \tag{5}$$

Thus, the three angles, ϕ, θ and ψ, and the thrust, T, are expressed in terms of $\mathbf{z}$ as:

$$\phi = \arcsin\left(\frac{2z_4\ddot{z}_1 - (1-z_4^2)\ddot{z}_2}{(1+z_4^2)\sqrt{\ddot{z}_1{}^2 + \ddot{z}_2{}^2 + (\ddot{z}_3+g)^2}}\right), \tag{6}$$

$$\theta = \arctan\left(\frac{(1-z_4^2)\ddot{z}_1 + 2z_4\ddot{z}_2}{(1+z_4^2)(\ddot{z}_3+g)}\right), \tag{7}$$

$$\psi = 2\arctan(z_4), \tag{8}$$

$$T = \sqrt{\ddot{z}_1{}^2 + \ddot{z}_2{}^2 + (\ddot{z}_3+g)^2}. \tag{9}$$

Remark 2. Within this work, we exploit only (6)–(9) in order to constrain the roll, pitch angles and the thrust for both trajectory generation and tracking control design, while the full flatness-based representations of the quadcopter system corresponding to the flat output (5) can be found in Nguyen, Prodan, Stoican, and Lefèvre (2017). There also exist different flatness-based representations of the quadcopter system according to different choices of the flat output (Aguilar-Ibáñez et al., 2012; Engelhardt, Konrad, Schäfer, & Abel, 2016). However, our proposed flat output (5) eliminates the trigonometric terms of ψ as shown in (6)–(9), thus, providing more efficient calculations. □

For expressing with clarity and in a compact way the forthcoming results, we denote the acceleration vector by $k \triangleq [k_1\ k_2\ k_3]^\top$, where k_1, k_2 and k_3 are given as:

$$k_1 = \ddot{z}_1, \quad k_2 = \ddot{z}_2, \quad k_3 = \ddot{z}_3 + g. \tag{10}$$

By considering the flatness-based representation of the roll and pitch angles in (6)-(7) and then, denoting them shortly as: $\phi = \Phi(k_1, k_2, k_3, z_4)$ and $\theta = \Theta(k_1, k_2, k_3, z_4)$, we have the following proposition (Nguyen et al., 2018).

Proposition 2.1 (Nguyen et al. (2018))**.** *The roll, $\Phi(k_1, k_2, k_3, z_4)$, and pitch, $\Theta(k_1, k_2, k_3, z_4)$, angles are bounded by the same function of k_1, k_2, k_3 as follows:*

$$|\Phi(k_1, k_2, k_3, z_4)| \leq \epsilon(k_1, k_2, k_3), \quad |\Theta(k_1, k_2, k_3, z_4)| \leq \epsilon(k_1, k_2, k_3), \ \forall z_4 \in \mathbb{R}, \tag{11}$$

where the angle boundary function, $\epsilon(\cdot)$, is defined as:

$$\epsilon(k_1, k_2, k_3) = \arcsin\left(\sqrt{\frac{k_1^2 + k_2^2}{k_1^2 + k_2^2 + k_3^2}}\right). \tag{12}$$

Sketch of the proof. The two functions $|\Phi(k_1, k_2, k_3, z_4)|$ and $|\Theta(k_1, k_2, k_3, z_4)|$ are bounded as follows:

$$|\Phi(\cdot)| \leq \arcsin\left(\sqrt{\frac{k_1^2 + k_2^2}{k_1^2 + k_2^2 + k_3^2}}\right), \ |\Theta(\cdot)| \leq \arctan\left(\sqrt{\frac{k_1^2 + k_2^2}{k_3^2}}\right). \tag{13}$$

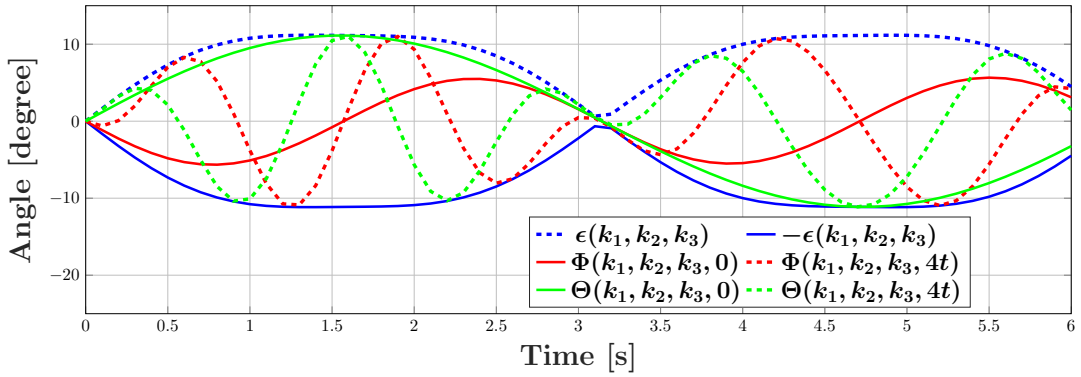

Figure 2. Illustrations of the roll, pitch angles bounded by the angle boundary ϵ.

Next, by considering ϵ defined in (12), we arrive to:

$$\tan\epsilon = \sqrt{\frac{k_1^2+k_2^2}{k_3^2}}. \tag{14}$$

Combining (13) and (14), we obtain the results of Proposition 2.1, thus, completing the proof. □

Illustrative example: Let us illustrate the Proposition 2.1 by using some simple trajectories of k_1, k_2 and k_3 given as follows:

$$k_1(t) = 2\sin(t), \quad k_2(t) = \sin(2t), \quad k_3 = 0.5\sin(0.5t) + g. \tag{15}$$

The evaluations of the roll, $\phi = \Phi(k_1,k_2,k_3,z_4)$, and pitch, $\Theta(k_1,k_2,k_3,z_4)$, angles under different trajectories of the yaw angle represented by $z_4(t)$ are illustrated in Figure 2. We can observe that the values of the roll (red lines) and pitch (green lines) angles of the specific 3D trajectory vary according to the yaw angle values but they are always bounded by $-\epsilon(t)$ and $\epsilon(t)$ (blue lines). □

Remark 3. There always exists real values of z_4, denoted by z_{4_ϕ} and z_{4_θ}, such that:

$$\left|\Phi(k_1,k_2,k_3,z_{4_\phi})\right| = \epsilon(k_1,k_2,k_3), \quad \left|\Theta(k_1,k_2,k_3,z_{4_\theta})\right| = \epsilon(k_1,k_2,k_3), \tag{16}$$

in which z_{4_ϕ} and z_{4_θ} are the solutions of the quadratic equations given by:

$$k_1 z_{4_\phi}^2 - 2k_2 z_{4_\phi} - k_1 = 0, \quad k_2 z_{4_\theta}^2 + 2k_1 z_{4_\theta} - k_2 = 0. \tag{17}$$

Thus, in the context of considering mismatches on yaw angle tracking presented in the following sections, the angle boundary function $\epsilon(k_1,k_2,k_3)$ is not conservative to be employed in the inequalities (11). □

This property is very useful and will be employed later for imposing constraints on the roll and pitch angles for the offline trajectory generation and online-tracking control design subject to saturation inputs.

3. Constrained trajectory offline generation

This section addresses the offline trajectory generation for a quadcopter system. First, defines briefly the B-splines curve, then employs B-splines for the flat output parametrization to generate a feasible 3D trajectory which respects the CF quadcopter dynamics (2) and constraints on states and inputs. In the literature, various states and inputs constraints of the aerial systems are usually imposed based on a predefined yaw angle trajectory, e.g., zero angle (Cowling, Yakimenko, Whidborne, & Cooke, 2007; Lu et al., 2017; Mueller & D'Andrea, 2013) or a spline with specific degree (Engelhardt et al., 2016). I.e., introducing a predefined yaw angle trajectory into the dynamics (2), then, imposing constraints on the resulted system. However, the built-in controller of the CF quadcopter controls only the yaw rate, $\dot{\psi}_r$, as detailed in Section 2 and it does not provide very good tracking results, even for maintaining a constant direction (i.e., $\dot{\psi}_r(t) = 0$). Therefore, the unavoidable change in the yaw angle trajectory may cause violation of the constraints if we follow these foregoing approaches.

In order to overcome this restriction, we decouple the position and the yaw angle trajectory generation procedure. Firstly, we define the yaw angle rate reference trajectory as $\dot{\psi}_r(t) = 0$ since it is straightforward to be implemented and further verified. However, we will not use this information for the position trajectory generation (hereinafter, we call it shortly, trajectory generation).

The trajectory generation will take into account various states and inputs constraints such as boundary constraints, waypoints constraints, constraints on thrust, and roll, pitch angles without requiring knowledge on a predefined yaw angle trajectory. Thus, possible tracking errors in yaw angle do not affect the validation of the above mentioned constraints. For further use, we denote the references of states and inputs related to the trajectory generation by adding a "bar" overhead, e.g., the position, $\bar{\xi}$, the roll, pitch angles, $\bar{\phi}, \bar{\theta}$, the thrust, $\bar{T}$.

3.1. B-splines parametrization

B-splines basis functions are well-suited to flatness parametrization due to their ease of enforcing continuity and because their degree depends only up to which derivative is needed to ensure continuity (Prodan et al., 2013; Stoican, Prodan, & Popescu, 2015; Stoican, Prodan, Popescu, & Ichim, 2017).
According to Piegl and Tiller (1995), a B-spline of order d is characterized by a *knot-vector*:

$$\mathbb{T} = \{\tau_0, \tau_1 \dots \tau_m\}, \tag{18}$$

of non-decreasing time instants ($\tau_0 \leq \tau_1 \leq \cdots \leq \tau_m$) which parametrizes the associated basis functions $B_{i,d}(t)$ defined as:

$$B_{i,1}(t) = \begin{cases} 1, \text{ for } \tau_i \leq t < \tau_{i+1} \\ 0 \text{ otherwise} \end{cases}, \tag{19a}$$

$$B_{i,d}(t) = \frac{t - \tau_i}{\tau_{i+d-1} - \tau_i} B_{i,d-1}(t) + \frac{\tau_{i+d} - t}{\tau_{i+d} - \tau_{i+1}} B_{i+1,d-1}(t), \tag{19b}$$

for $d > 1$ and $i \in \{0, 1, \dots, n\}$ where $n = m - d$.
We consider a fixed knot-vector $\mathbb{T}$ with $\tau_i = t_0,\ i \in \{0, \dots, d\}$ and $\tau_q = t_f,\ q \in$

$\{n, \ldots, n+d\}$ (Suryawan, 2012) constructed as follows:

$$\mathbb{T} = \{\underbrace{t_0, ..., t_0}_{d+1}, \tau_{d+1}, ..., \tau_{n-1}, \underbrace{t_f, ..., t_f}_{d+1}\}, \tag{20}$$

with the intermediary points $\tau_d, \ldots, \tau_n$ are equally distributed along these extremes. Considering the row vector of *control points* in three dimensional space $\mathbb{R}^3$ given as:

$$\mathbf{P} = \begin{bmatrix} p_0, p_1, \ldots, p_n \end{bmatrix}, \tag{21}$$

we define a *B-splines curve* as a linear combination of the control points (21) and the B-spline functions (19a)–(19b):

$$\bar{\xi}(t) = \sum_{i=0}^{n} B_{i,d}(t) p_{x_i} = \mathbf{P}\mathbf{B}_d(t), \tag{22}$$

where $\bar{\xi} = \begin{bmatrix} \bar{x} & \bar{y} & \bar{z} \end{bmatrix}^\top$ and $\mathbf{B}_d(t) = \begin{bmatrix} B_{0,d}(t) \ldots B_{n,d}(t) \end{bmatrix}^\top$. This construction yields various interesting properties which are enumerated in Suryawan (2012) and Stoican et al. (2015), some of them are related to the forthcoming results, hence, given in detail:

(1) Endpoint interpolation: the first control point coincides with the initial point and the last control point coincides with the last point (Suryawan, 2012). E.g.:

$$p_0 = \xi(t_0), \quad p_n = \xi(t_f). \tag{23}$$

(2) The B-spline curve lies in the convex hull generated by the control points $\mathbf{P}$;
(3) The 'r' order derivatives of B-spline basis function can be expressed as linear combination of B-spline basis function as:

$$\mathbf{B}_d^{(r)}(t) = M_r \mathbf{B}_{d-r}(t) = M_r L_r \mathbf{B}_d(t), \tag{24}$$

with matrices $M_r,\ L_r$ of appropriate dimensions and content given in Stoican et al. (2015); Suryawan (2012).

3.2. State constraints

Assume that the CF quadcopter has a known initial state composed of the position, and its higher derivatives up to the acceleration, i.e., $\{\xi_0,\ \dot{\xi}_0,\ \ddot{\xi}_0\}$. The trajectory is considered to satisfy the initial state constraints and also drive the CF quadcopter to a defined final state consisting of position, ξ_f, velocity $\dot{\xi}_f$ and acceleration $\ddot{\xi}_f$. By employing the two properties of the B-splines curve detailed in (23) and (24), the boundary constraints are constructed as:

$$p_0 = \xi_0, \qquad \mathbf{P}M_1L_1\mathbf{B}_d(t=t_0) = \dot{\xi}_0, \qquad \mathbf{P}M_2L_2\mathbf{B}_d(t=t_0) = \ddot{\xi}_0, \tag{25}$$

$$p_n = \xi_f, \qquad \mathbf{P}M_1L_1\mathbf{B}_d(t=t_f) = \dot{\xi}_f, \qquad \mathbf{P}M_2L_2\mathbf{B}_d(t=t_f) = \ddot{\xi}_f. \tag{26}$$

Note that, $\mathbf{B}_d(t = t_0) = \left[1, 0, \ldots, 0\right]^\top$ and $\mathbf{B}_d(t = t_f) = \left[0, \ldots, 0, 1\right]^\top$.
Moreover, we consider a collection of $N + 1$ waypoints[3] and the time instances associated to them (there must be no conflict with the boundary conditions (25), (26)):

$$\mathbb{W} = \{w_k\} \text{ and } \mathbb{T}_\mathbb{W} = \{t_k\}, \quad k \in \{0, \ldots, N\}. \tag{27}$$

The trajectory has to pass through each waypoint w_k at the time instant t_k, i.e.:

$$\mathbf{P}\mathbf{B}_d(t_k) = w_k. \tag{28}$$

3.3. Input constraints

Since the built-in controller of the CF quadcopter is composed of linear PID controllers constructed around the hovering states (roll, pitch angles equal zero), the ideal operating conditions are small values of these two angles. Thus, the reference trajectory is subject to saturation constraints on roll and pitch angles with desired maximum value $\epsilon_d \in \left(0, \frac{\pi}{2}\right)$ given as follows:

$$\left|\bar{\phi}\right| \leq \epsilon_d, \ \left|\bar{\theta}\right| \leq \epsilon_d. \tag{29}$$

Moreover, since we do not want the CF to have an aggressive altitude variation, the normalized thrust reference, $\bar{T}$, is also limited by its lower bound, $g - \Delta_g > 0$ and upper bound, $g + \Delta_g$, given as follows:

$$g - \Delta_g \leq \bar{T} \leq g + \Delta_g, \tag{30}$$

where $\Delta_g > 0$ is a desired parameter. Note that, the desired parameter Δ_g only needs to satisfy $g - \Delta_g > 0$. Consequently, the upper bound of the thrust $g + \Delta_g$ is always smaller than $2g$, thus, the thrust reference, $\bar{T}$, always respects the real thrust limit, $T_{limit} = 20.18 > 2g$ from (4).

Proposition 3.1 (Nguyen et al. (2018))**.** *By imposing the constraint that* $\mathcal{K} \triangleq \left[\bar{k}_1^2 + \bar{k}_2^2 \quad \bar{k}_3^2\right]^\top$ (10) *lies inside a polytopic region defined as a convex sum of vertices as follows:*

$$\mathcal{K} \in \text{Conv}\Big\{ \left(0, (g - \Delta_g)^2\right), \left(\sin^2 \epsilon_d \ (g - \Delta_g)^2, \ \cos^2 \epsilon_d \ (g - \Delta_g)^2\right), \\ \left(\sin^2 \epsilon_d \ (g + \Delta_g)^2, \ \cos^2 \epsilon_d \ (g + \Delta_g)^2\right), \left(0, (g + \Delta_g)^2\right) \Big\}, \tag{31}$$

the constraints on roll, pitch angles in (29) *and the constraint on thrust in* (30) *are satisfied regardless of the yaw angle values.* □

Sketch of the proof. From Proposition 2.1, we have that $\left|\bar{\phi}\right| \leq \epsilon\left(\bar{k}_1, \bar{k}_2, \bar{k}_3\right)$ and $\left|\bar{\theta}\right| \leq \epsilon\left(\bar{k}_1, \bar{k}_2, \bar{k}_3\right)$. Thus, we constrain the angle boundary $\epsilon(\cdot)$ to be lower than the maximum angle value ϵ in order to obtain (29). Then, by combining the results with

[3]Note that, considering waypoints at the trajectory generation level is coherent with typical software-hardware UAV configurations which use waypoints in the communication protocol.

the constraints on thrust in (30), i.e., $g - \Delta_g \leq \sqrt{\bar{k}_1^2 + \bar{k}_2^2 + \bar{k}_3^2} \leq g + \Delta_g$, we arrive to condition (31). For more details the reader is referred to Nguyen et al. (2018). □

3.4. Constrained trajectory with minimal length

In our work, we choose to minimize the length of the trajectory along the time interval $[t_0, t_f]$ which is also subject to various states and inputs constraints described in Section 3.2 and 3.3. It results in an optimization problem with a quadratic cost function in terms of the control points $\mathbf{P}$ (Stoican et al., 2015) defined as:

$$\begin{aligned} \mathbf{P} = \arg\min_{\mathbf{P}} \int_{t_0}^{t_f} \left(\mathbf{P} M_1 \mathbf{B}_{d-1}\right)^\top \left(\mathbf{P} M_1 \mathbf{B}_{d-1}\right) dt, \\ \text{s.t. constraints (25), (26), (28) and (31) are verified,} \end{aligned} \tag{32}$$

For solving the optimization problem (32), the constraints (25), (26), (28) and (31) can be enforced at discrete moments (Prodan et al., 2013; Stoican et al., 2015) or even be guaranteed continuously (Stoican et al., 2017) along the time interval $[t_0, t_f]$. The first method is straightforward to implement but it does not guarantee the constraints fulfillment "in-between" the discrete moments. In case of constraints violation, increasing the sampling points can alleviate the problem. The inevitable drawback, high computation time, can be accepted since the trajectory generation is done offline before flight. The second method employs particular geometrical properties of the B-spline functions given in Section 3.1 in order to obtain a continuous constraints validation with fixed complexity (the number of constraints depends on the B-spline degree and on the number of control points but not on the number of waypoints) (Stoican et al., 2017).

Remark 4. Solving the nonlinear optimization problem (32) may provide a local minimum result instead of the expected globally optimal solution (i.e., the shortest curve). The solution is required to be manually verified after and the parameters (e.g., number of control points) may be changed until obtaining a good solution. Besides the minimal trajectory length, (32) can take into account various optimization objectives like input variation, magnitude, energy minimization and the like (Prodan et al., 2013; Stoican et al., 2017). □

After solving (32), we obtain the 3D reference trajectory, $\bar{\xi}$, which satisfies all the imposed constraints, i.e., boundary condition(25)-(26), waypoints (28) and bounded angles and thrust (31). We remind that we give a zero yaw angle rate input $\dot{\psi}_r = 0$ along the time interval $[t_0, t_f]$[4]. Recalling all the necessary tools and constructions for the offline constrained trajectory generation procedure, we provide in the following the proposed control design method for tracking the given quadcopter references.

4. Feedback linearization control design with saturating inputs

A typical control scheme for quadcopter trajectory tracking (and UAV systems in general) is illustrated in Figure 3. The preferred approach is to consider two control

[4]We denote the yaw angle rate input by $\dot{\psi}_r$ since we want to give the uniform format of the four inputs sent to the CF platform, i.e., $\{T_r, \phi_r, \theta_r, \dot{\psi}_r\}$.

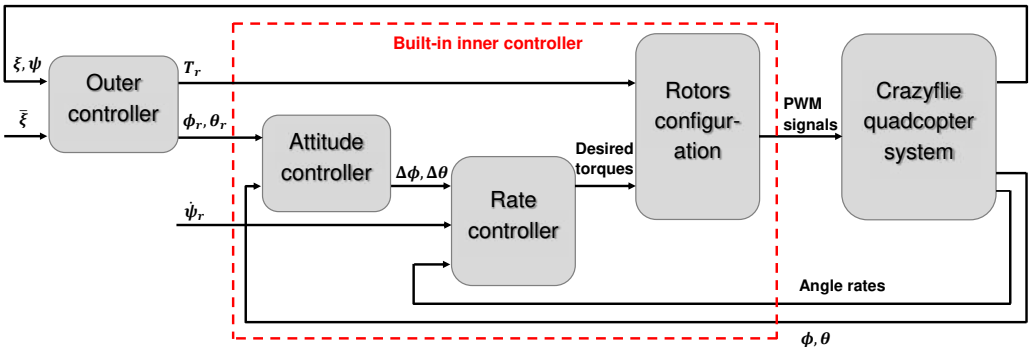

Figure 3. Control scheme for the Crazyflie quadcopter system.

layers in order to exploit the decoupling between the translational and rotational dynamics of the quadcopters (Cao & Lynch, 2016; Engelhardt et al., 2016; Lu et al., 2017). At the higher level, the *outer controller* compares an externally given reference position, $\bar{\xi}$, with the real position, ξ, and provides as outputs the reference angles η_r and thrust T_r. They are sent to the lower level *inner controller* which compares the reference angles η_r with the real angles η in order to provide the necessary angle torques. These torques are then combined with the thrust T_r in order to obtain the four rotor speeds by using the appropriate configuration of the quadcopter as detailed in Section 2.

Hereinafter, we concentrate on the design of the outer controller which is to provide the angle inputs ϕ_r, θ_r and the thrust input T_r allowing the CF quadcopter to track the a priori given position trajectory, $\bar{\xi}$. In order to avoid the unexpected effect of the thrust saturation detailed in (4), the thrust input T_r has to respect the thrust limit, T_{limit}, as follows [5]:

$$T_r \leq T_{limit}. \tag{33}$$

Furthermore, the built-in controllers (whose aim is to control the three angles) of the CF quadcopter are constructed around the hovering conditions, i.e., zero roll and pitch angles, thus, constraints on the angle inputs ϕ_r, θ_r are necessary:

$$|\phi_r| \leq \epsilon_c, \quad |\theta_r| \leq \epsilon_c, \tag{34}$$

with $\epsilon_c \in \left(\epsilon_d, \frac{\pi}{2}\right)$, the maximum value of the angle inputs sent to the CF quadcopter. We emphasize that ϵ_c must be larger than the desired maximum angle value ϵ_d from (29) employed for the trajectory generation. At first, this is due to the fact that the CF quadcopter may need to tilt more than the desired angle values in order to counteract the position tracking errors. However, we will provide an analysis on the value of ϵ_c in the forthcoming section. The control strategy is based on the concept of feedback linearization of quadcopter systems (Formentin & Lovera, 2011; Nguyen, Prodan, Stoican, & Lefèvre, 2017) and nested control design (Cao & Lynch, 2016; Teel, 1992).

[5] We do not consider $T_r \geq T_{min}$ as in (30) since from a practical point of view, the thrust can be equal to zero under real circumstances when counteracting an external perturbation.

4.1. Feedback linearization via flatness

In this section, we introduce a feedback law which will drive the translation dynamics (2) to simpler dynamical systems with appropriately designed corrective terms. The design of the controller[6] is facilitated by the flatness characterizations (6)–(9) given as follows:

$$T_r = \sqrt{u_x^2 + u_y^2 + (u_z + g)^2}, \tag{35a}$$

$$\phi_r = \arcsin\left(\frac{2z_4u_x - (1 - z_4^2)u_y}{(1 + z_4^2)\sqrt{u_x^2 + u_y^2 + (u_z + g)^2}}\right), \tag{35b}$$

$$\theta_r = \arctan\left(\frac{(1 - z_4^2)u_x + 2z_4u_y}{(1 + z_4^2)(u_z + g)}\right), \tag{35c}$$

where u_x, u_y and u_z are the virtual inputs of the dynamics (2). Note that, the feedback linearization law given in (35) is validated only for the translation dynamics (2) since we concentrate on designing the *outer controller* illustrated in Figure 3. The feedback linerization laws for the complete quadcopter system including both translation and rotation dynamics can be found in Aguilar-Ibáñez et al. (2012); Formentin and Lovera (2011)

Remark 5. In (35b)-(35c), z_4 stands for the real yaw angle provided by the CF platform. Possible tracking errors in the yaw angle do not propagate in the position tracking since the roll and pitch angle inputs use the actual value of the yaw angle (Nguyen, Prodan, Stoican, & Lefèvre, 2017). □

Under the assumption that the precise measurement of the yaw angle is available, the input thrust T_r satisfies the thrust limit T_{limit}, i.e., $T_r \leq T_{limit}$ for the real thrust to equal to the input thrust, $T = T_r$ as detailed in Remark 1 (which will be enforced by designing the appropriate virtual inputs u_x, u_y, u_z detailed in the forthcoming section), and the CF quadcopter does not turn upside down, i.e., the roll, pitch angles stay in the range $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$, we have the following proposition.

Proposition 4.1. *The feedback law* (35) *drives the dynamics* (2) *to one of the two following systems depending on the value of* u_z*:*

$$\begin{cases} \ddot{x} = u_x \\ \ddot{y} = u_y \\ \ddot{z} = u_z \end{cases}, \qquad if\ u_z \geq -g, \tag{36}$$

$$\begin{cases} \ddot{x} = -\cos(2\psi)u_x - \sin(2\psi)u_y \\ \ddot{y} = -\sin(2\psi)u_x + \cos(2\psi)u_y \\ \ddot{z} = -u_z - 2g \end{cases}, \qquad if\ u_z < -g. \tag{37}$$

Sketch of the proof. Without detailing all the steps, the reasoning is based on the altitude dynamics exploited from (2) given as:

$$\ddot{z} = -g + T\cos\phi\cos\theta. \tag{38}$$

[6]similar result can be found in our previous work (Nguyen, Prodan, Stoican, & Lefèvre, 2017).

Since $T\cos\phi\cos\theta \geq 0$, $\ddot{z}+g$ always has a non-negative value. Thus, introducing (35) into the dynamics (2) leads to $\ddot{z}+g=|u_z+g|$. Consequently, $\ddot{z}=u_z$ if $u_z \geq -g$ and $\ddot{z}=-u_z-2g$ if $u_z<-g$ which lead to the corresponding results of $\ddot{x}$ and $\ddot{y}$. □

The dynamical system (36) is usually the desired goal of the feedback law (35) with various different designs of the virtual inputs u_x, u_y, u_z as detailed in (Formentin & Lovera, 2011; Nguyen, Prodan, Stoican, & Lefèvre, 2017; Zhao & Go, 2014). However, if the condition of $u_z \geq -g$ is not noticed, the system (2) may become unstable as detailed in Appendix A. Thus, in order to guarantee the closed-loop stability, we will ensure $u_z \geq -g$ by designing the virtual input u_z based on nested control method in the following Section 4.2.

4.2. Virtual input design

This section details the design of the virtual inputs u_x, u_y and u_z from (35) which allows the dynamics (36) to track the apriori given trajectory consisting of $\{\bar{\ddot{\xi}}, \bar{\dot{\xi}}, \bar{\xi}\}$. In the literature, a simple PD (or PID instead) corrective term is usually employed (Formentin & Lovera, 2011; Nguyen, Prodan, Stoican, & Lefèvre, 2017; Zhao & Go, 2014). Another approach is to apply an LQR controller (Cowling et al., 2007). However, these control designs do not take into account the saturation, the constraints and even the unstable mode (37) of the quadcopter system. In order to satisfy the imposed constraints on thrust and angles (33)-(34), and to overcome the unstable mode (37), it is essential to bound the virtual control inputs:

$$|u_x| \leq U_x, \quad |u_y| \leq U_y, \quad |u_z| \leq U_z. \tag{39}$$

Proposition 4.2. *Considering the modified version of the nested control design method for trajectory tracking (Teel, 1992) defined as:*

$$u_q = \bar{\ddot{q}} + \lambda_q \sigma\left(\frac{K_{q_1}}{\lambda_q}\dot{e}_q + \frac{1}{2}\sigma\left(\frac{K_{q_2}}{\lambda_q}\dot{e}_q + \frac{K_{q_1}K_{q_2}}{\lambda_q}e_q\right)\right),\ q \in \{x,y,z\}, \tag{40}$$

where $e_q = \bar{q} - q$ represents the position tracking error and $K_{q_1}, K_{q_2} \in \mathbb{R}^+$ are the control parameters, the saturation limit λ_q is defined as:

$$\lambda_q = U_q - |\bar{\ddot{q}}|, \tag{41}$$

where the bound U_q is chosen such that $\lambda_q > 0$. The saturation function $\sigma : \mathbb{R} \to [-1,1]$ (Liu, Chitour, & Sontag, 1996) is defined as:

$$\sigma(s) = \text{sign}(s)\min(|s|, 1). \tag{42}$$

The nested control design defined in (40) *is bounded as required in* (39)*, i.e., $|u_q| \leq U_q$, $q \in \{x,y,z\}$. Furthermore, introducing u_q* (40) *to the dynamics* (36) *leads to globally asymptotically stable error dynamics in terms of e_q.*

Sketch of the proof. The bounds of u_q are derived from applying the Triangle inequality to the control action (40) as follows:

$$|u_q| \leq |\bar{\ddot{q}}| + |(U_q - |\bar{\ddot{q}}|)\,\sigma(\cdot)| \leq |\bar{\ddot{q}}| + U_q - |\bar{\ddot{q}}|. \tag{43}$$

Thus, $|u_q| \leq U_q$ given in (39) is satisfied. Moreover, substituting (40) into the dynamics (36) leads to:

$$\ddot{e}_q = -\lambda_q \sigma \left(\frac{K_{q_1}}{\lambda_q} \dot{e}_q + \frac{1}{2} \sigma \left(\frac{K_{q_2}}{\lambda_q} \dot{e}_q + \frac{K_{q_1} K_{q_2}}{\lambda_q} e_q \right) \right), \tag{44}$$

with $e_q = \bar{q} - q$ which is globally asymptotically stable for any positive parameters K_{q_1} and K_{q_2} as detailed in the Appendix B. □

Remark 6. We emphasize that the design (41) requires the smoothness of $\bar{\ddot{q}}$ in order to ensure the smoothness of u_q in (40). Indeed, we modify the nested control design introduced in Teel (1992) by employing the varied λ_q in (41), $q \in \{x, y, z\}$ in order to better exploit the saturating term. This is opposed to the fixed $\lambda_q = U_q - \max |\bar{\ddot{q}}|$ as considered in (Aguilar-Ibáñez et al., 2012; Teel, 1992). In Aguilar-Ibáñez et al. (2012), the nested control design of the quadcopter motion along the z-axis is employed to ensure only the condition of $u_z > g$ and not to guarantee the saturation of the control inputs. □

Proposition 4.3. *Consider the a priori given trajectory consisting of* $\{\bar{q}, \bar{\dot{q}}, \bar{\ddot{q}}\}$, $q \in \{x, y, z\}$. *The quadcopter system* (2) *tracks the trajectory by using the feedback law given in* (35) *with the associated virtual inputs defined in* (40). *By choosing the bounds* U_x, U_y, U_z *in* (40) *such that the following conditions are satisfied:*

$$U_z < g, \tag{45}$$

$$U_q > max\, |\bar{\ddot{q}}|\,,\ q \in \{x, y, z\}, \tag{46}$$

$$U_x^2 + U_y^2 \leq (-U_z + g)^2 \tan^2 \epsilon_c, \tag{47}$$

$$\sqrt{U_x^2 + U_y^2 + (U_z + g)^2} \leq T_{limit}, \tag{48}$$

the controller provides the angle inputs ϕ_r, θ_r *and the thrust input* T_r *which satisfy the conditions* (33)-(34) *and further, asymptotically stabilizes the error dynamics* (44) *which implies that the quadcopter system asymptotically tracks the trajectory.*

Sketch of the proof. Firstly, by considering (35b)-(35c), we have that: $\phi_r = \Phi(u_x, u_y, u_z + g, z_4)$ and $\theta_r = \Phi(u_x, u_y, u_z + g, z_4)$. Thus, by using Proposition 2.1, we arrive to the bounds on the roll, pitch angle inputs, i.e., $|\phi_r|\,, |\theta_r| \leq \epsilon(u_x, u_y, u_z + g)$ (12) where $\epsilon(\cdot)$ is also bounded in terms of U_x, U_y, U_z as follows:

$$\epsilon(u_x, u_y, u_z + g) = \arctan\left(\sqrt{\frac{u_x^2 + u_y^2}{(u_z + g)^2}} \right) \leq \arctan\left(\sqrt{\frac{U_x^2 + U_y^2}{(-U_z + g)^2}} \right). \tag{49}$$

Thus, (47) leads to $\epsilon(\cdot) \leq \epsilon_c$, hence, (34) is validated due to the monotony property of the function $\arctan(\cdot)$ in $\mathbb{R}^+$. Secondly, by introducing (39) into (35a), we obtain the bound on the desired thrust, T_r, given as:

$$T_r \leq \sqrt{U_x^2 + U_y^2 + (U_z + g)^2}. \tag{50}$$

By employing (48), from (50), we arrive to $T \leq T_{limit}$ (33). Note that, $T \leq T_{limit}$

is a prerequisite of Proposition 4.1. Thus, Proposition 4.1 is validated. Furthermore, introducing (45) into (39) leads to $u_z > -g$. According to Proposition 4.1, the feedback law (35) leads to the dynamics (36). Finally, (46) validates $\lambda_q > 0$ in (41), $q \in \{x, y, z\}$, this representing the conditions of the nested control design in (40). It ensures the stability of the error dynamics (44), i.e., the tracking capability of the overall controller, thus, completing the proof. □

In order to efficiently choose the virtual input vector $U = \left[U_x, U_y, U_z\right]^\top$ which is subject to various constraints (45)–(48), we solve an optimization problem[7] expressed as:

$$\begin{aligned} U = &\arg\max_{U} \|U\|^2, \\ &\text{s.t. constraints (45)–(48) are verified,} \end{aligned} \tag{51}$$

where $\|U\|^2 = U_x^2 + U_y^2 + U_z^2$. Thus, we maximize the saturation limit λ_q in (41), $q \in \{x, y, z\}$ and consequentially, provide a better capability to counteract the tracking error for the virtual input design (40).

Proposition 4.4. *A sufficient condition for the existence of U_x, U_y, U_z satisfying (45)–(48) is that the maximum value of the roll, pitch angle inputs, ϵ_c satisfies:*

$$\tan\epsilon_c > \sqrt{2}\,\frac{g + \Delta_g}{g - \Delta_g}\tan\epsilon_d, \tag{52}$$

and that the thrust limit T_{limit} verifies:

$$T_{limit} > \sqrt{\left(2g - (g - \Delta_g)\cos\epsilon_d\right)^2 + 2\sin^2\epsilon_d\,(g + \Delta_g)^2}, \tag{53}$$

with ϵ_d and Δ_g in (29)–(30), are the desired parameters of the reference trajectory.

Sketch of the proof. As detailed in Appendix C, we have that:

$$\max|\ddot{\bar{x}}| \leq (g + \Delta_g)\sin\epsilon_d, \tag{54}$$
$$\max|\ddot{\bar{y}}| \leq (g + \Delta_g)\sin\epsilon_d, \tag{55}$$
$$\max|\ddot{\bar{z}}| \leq g - (g - \Delta_g)\cos\epsilon_d. \tag{56}$$

Thus, in order to ensure the validation of (45) and (46), we arrive to:

$$U_x > (g + \Delta_g)\sin\epsilon_d, \tag{57}$$
$$U_y > (g + \Delta_g)\sin\epsilon_d, \tag{58}$$
$$g > U_z > g - (g - \Delta_g)\cos\epsilon_d. \tag{59}$$

By using $U_z < g$, condition (47) leads to:

$$U_z \leq g - \cot\epsilon_c\sqrt{U_x^2 + U_y^2}. \tag{60}$$

[7]The cost function can be modified in order to give more flexibility to the corrective term of a particular axis. Here, we consider the unweighted ℓ_2 norm since we provide equal importance to all axes.

Furthermore, from (48), we have that:

$$U_z \leq \sqrt{T_{limit}^2 - U_x^2 - U_y^2} - g. \tag{61}$$

Introducing (57)-(58) into (60) and (61), respectively, leads to:

$$U_z < g - \sqrt{2}(g + \Delta_g)\cot\epsilon_c \sin\epsilon_d, \tag{62}$$

$$U_z < \sqrt{T_{limit}^2 - 2(g + \Delta_g)^2 \sin^2\epsilon_d} - g. \tag{63}$$

Thus, in order to ensure the existence of U_z constrained by (59), (62) and (63), we arrive to:

$$g - (g - \Delta_g)\cos\epsilon_d < g - \sqrt{2}(g + \Delta_g)\cot\epsilon_c \sin\epsilon_d, \tag{64}$$

$$g - (g - \Delta_g)\cos\epsilon_d < \sqrt{T_{limit}^2 - 2(g + \Delta_g)^2 \sin^2\epsilon_d} - g. \tag{65}$$

From (64)–(65), we obtain (52) and (53), hence, completing the proof. □

Remark 7. Indeed, the condition (53) should be considered as a criteria for choosing the parameters ϵ_d and Δ_g employed in (29) and (30). This ensures a feasible reference trajectory for the feedback control law (35) with the known thrust limit T_{limit}. □

4.3. Overall stability analysis

This section investigates the effect of the built-in controller detailed in Remark 1 on the overall stability of the position tracking capability. Since the thrust input T_r is constrained to be under the thrust limit T_{limit} from (33), the thrust T applied to the CF quadcopter system equals the thrust input, i.e., $T = T_r$, as detailed in (4). Thus, introducing the feedback law (35) into the dynamics (2) explicitly leads to:

$$\ddot{q} = u_q + d_q, \tag{66}$$

where $q \in \{x, y, z\}$ and d_x, d_y, d_z are given as:

$$d_x = T_r(\mathrm{c}\,\phi\,\mathrm{s}\,\theta\,\mathrm{c}\,\psi + \mathrm{s}\,\phi\,\mathrm{s}\,\psi) - T_r(\mathrm{c}\,\phi_r\,\mathrm{s}\,\theta_r\,\mathrm{c}\,\psi + \mathrm{s}\,\phi_r\,\mathrm{s}\,\psi), \tag{67}$$

$$d_y = T_r(\mathrm{c}\,\phi\,\mathrm{s}\,\theta\,\mathrm{s}\,\psi - \mathrm{s}\,\phi\,\mathrm{c}\,\psi) - T_r(\mathrm{c}\,\phi_r\,\mathrm{s}\,\theta_r\,\mathrm{s}\,\psi - \mathrm{s}\,\phi_r\,\mathrm{c}\,\psi), \tag{68}$$

$$d_z = T_r\,\mathrm{c}\,\phi\,\mathrm{c}\,\theta - T_r\,\mathrm{c}\,\phi_r\,\mathrm{c}\,\theta_r, \tag{69}$$

where the angle inputs, ϕ_r, θ_r, detailed in (35b)-(35c) are the references for the actual roll, pitch angles, ϕ, θ, to follow.

Proposition 4.5. *The terms d_x, d_y, d_z given in* (67)-(69) *are bounded and $d_q \to 0$, $q \in \{x, y, z\}$ as $t \to \infty$.*

Sketch of the proof. Applying the Triangular inequality to d_x, d_y, d_z given in (67)-(69), we obtain that:

$$|d_x| \leq 4T_r, \quad |d_y| \leq 4T_r, \quad |d_z| \leq 2T_r. \tag{70}$$

Since T_r is bounded as constrained in (33), (70) leads to d_x, d_y, d_z bounded. Furthermore, from (3), we have that the built-in controller is capable of tracking the roll, pitch angle inputs (35b)-(35c), hence, $d_q \to 0$, $q \in \{x, y, z\}$ as $t \to \infty$, thus, completing the proof. □

Introducing the virtual inputs u_q, $q \in \{x, y, z\}$ detailed in (40) into the overall dynamics (66) leads to:

$$\ddot{e}_q = -\lambda_q \sigma \left(\frac{K_{q_1}}{\lambda_q} \dot{e}_q + \frac{1}{2} \sigma \left(\frac{K_{q_2}}{\lambda_q} \dot{e}_q + \frac{K_{q_1} K_{q_2}}{\lambda_q} e_q \right) \right) + d_q, \tag{71}$$

where $e_q = \bar{q} - q$ as given in (40). Since the perturbation d_q is bounded and vanishes asymptotically as detailed in Proposition 4.5, all solutions of (71) are bounded (Teel, 1992). This property of the nested saturation control design (40) is a form of converging input bounded state stability. It guarantees that small vanishing input disturbances do not produce unbounded solutions (Maggiore, 2015).

Moreover, the reference acceleration terms $\ddot{\bar{q}}$ in (40) are bounded as detailed in (54)-(56) which implies that all solution of the closed-loop system (66) are bounded. As a result, according to Maggiore (2015), the closed-loop system (66) obtains almost globally asymptotic tracking for the a priori given trajectory consisting of $\{\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}}\}$, $q \in \{x, y, z\}$.

Remark 8. Maggiore (2015) introduces a hierarchical control design for stabilizing a thrust-propelled system at a fixed position which can be seen as a particular case of our tracking control law given in (35). Thus, the foregoing overall stability analysis is also validated in Maggiore (2015) with similar ideas on bounded perturbation–bounded solutions. □

5. Simulation and experiment results

This section presents first the trajectory generation results as introduced in Section 3. Then, the tracking controller proposed in Section 4 are tested by using a simulation model of a Crazyflie (CF) quadcopter (Giernacki et al., 2017) and lastly a real platform at the Laboratory of Conception and Integration of System (LCIS), France. For the simulation, the complete CF model given in Luis and Ny (2016) is employed including translation, rotation dynamics and the built-in controller (further details are given in Section 2). The experimental platform at laboratory LCIS shown in Figure 4 includes an indoor CF nano quadcopter equipped with a 10-DOF IMU (accelerometer, gyro, magnetometer, and high precision pressure), a Loco positioning system[8] including a deck attached to the CF quadcopter and six nodes fixed around the experimental room with known positions. The Loco positioning system is functioning in Two Way Ranging (TWR) mode in which, the deck pings the nodes in sequence. This allows the deck to obtain the distance between itself and the six nodes, then, the deck can calculate its position compared to the six nodes. After gathering all the necessary feedback information such as the position and the angles, the CF quadcopter communicates with the ground station computer through a 2.4Ghz low-latency/long-range radio messages by using the Crazyradio PA USB radio dongle. The computer sends an input message

[8]More information of the Crazyflie quadcopter and the Loco positioning system can be found in `https://www.bitcraze.io`

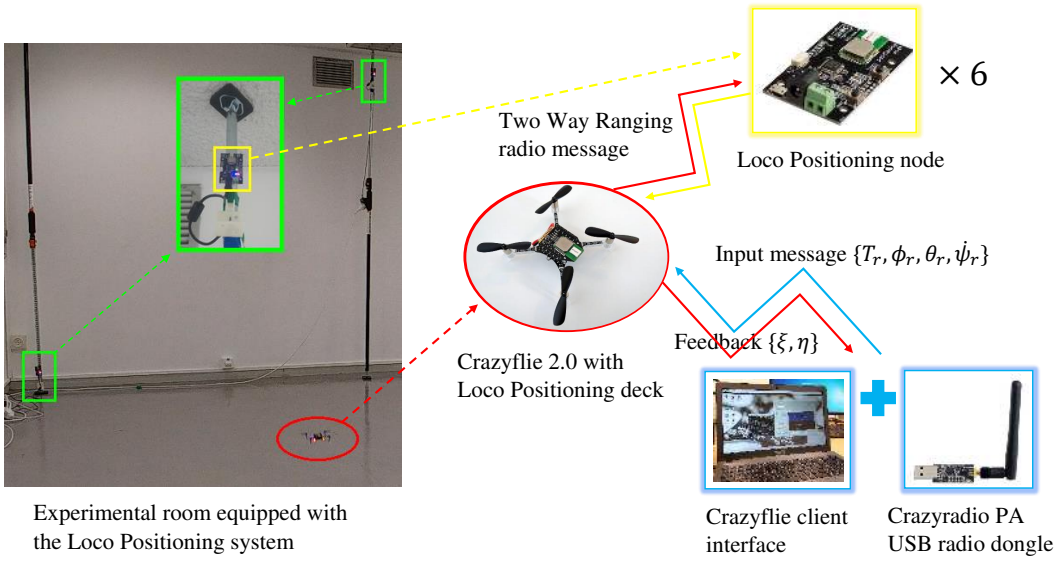

Figure 4. Operational control system setup of the Crazyflie quadcopter.

Table 1. Constraints imposed on the optimal trajectory generation

	Constraints
Position, $\bar{\xi}(t)$	$\bar{\xi}(0) = [0 \quad 0 \quad 0]^\top$, $\bar{\xi}(5) = [0.5 \quad 2 \quad 1]^\top$, $\bar{\xi}(10) = [1.5 \quad 2 \quad 1]^\top$, $\bar{\xi}(15) = [2 \quad 0 \quad 0]^\top$
Velocity, $\dot{\bar{\xi}}(t)$	$\dot{\bar{\xi}}(0) = [0 \quad 0 \quad 0]^\top$, $\dot{\bar{\xi}}(15) = [0 \quad 0 \quad 0]^\top$
Acceleration, $\ddot{\bar{\xi}}(t)$	$\ddot{\bar{\xi}}(0) = [0 \quad 0 \quad 0]^\top$, $\ddot{\bar{\xi}}(15) = [0 \quad 0 \quad 0]^\top$
Angles, $\bar{\phi}, \bar{\theta}$	$\lvert\bar{\phi}\rvert \leq \epsilon_d, \lvert\bar{\theta}\rvert \leq \epsilon_d$ with $\epsilon_d = 5°$
Thrust, $\bar{T}$	$g - \Delta_g \leq \bar{T} \leq g + \Delta_g$ with $\Delta_g = 0.05g$

including the four inputs $\{T_r, \phi_r, \theta_r, \dot{\psi}_r\}$ to the CF quadcopter. Note that the thrust input T_r after being calculated by (35a) is required to be converted into the thrust unit of the CF quadcopter system, i.e., 16-bit integer valued from 0 to 65535.

5.1. Offline trajectory generation results

In this section, the results of the trajectory generation procedure introduced in Section 3 are presented. Since the experimental platform, i.e., CF quadcopter, works well only with small roll and pitch angles (i.e., smaller than 10°) (Giernacki et al., 2017), we imposed the desired maximum angle $\epsilon_d = 5°$ to generate the a priori feasible trajectory. This is to obtain the limit of the angles input ϵ_c under 10° for the tracking control design. Note that, for different platforms, the parameters ϵ_d and ϵ_c are easily modified according to particular requirements and capabilities. Furthermore, the normalized thrust $\bar{T}$ is also bounded by $g - \Delta_g$ and $g + \Delta_g$ with $\Delta_g = 0.05g$ by employing (31). We emphasize that the desired parameters $\epsilon_d = 5°$ and $\Delta_g = 0.05g$ are validated by the constraint (53) with the thrust limit, $T_{limit} = 20.18m/s^2$ as detailed Remark 7, thus ensuring the existence of the control parameters presented later. In addition, the trajectory constraints include passing through four given waypoints $\mathbb{W} = \left\{ [0 \quad 0 \quad 0]^\top, [0.5 \quad 2 \quad 1]^\top, [1.5 \quad 2 \quad 1]^\top, [2 \quad 0 \quad 0]^\top \right\}$ with the associated time instants $\{0, 5, 10, 15\}$ seconds. The trajectory possess all the derivatives of $\{x, y, z\}$ up

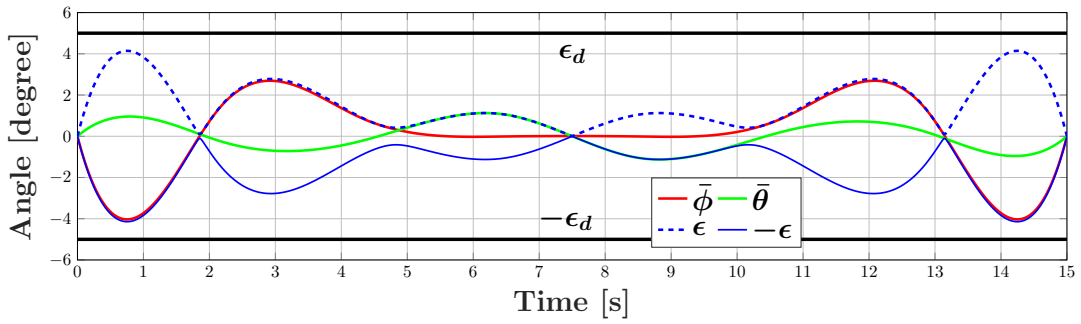

Figure 5. Roll, pitch angles references $\bar{\phi}, \bar{\theta}$, and the angle boundary ϵ as in (12) compared with the desired angle limit ϵ_d.

Table 2. Parameters of the virtual inputs u_q, $q \in \{x, y, z\}$ given in (40).

Virtual input	U_q	K_{q_1}	K_{q_2}
u_x	0.7168	1	2
u_y	0.7168	1	2
u_z	2.5970	2	3

to 2^{nd} order equal 0 at the initial and final time instants since we aim to perform the trajectory tracking task between the two hovering periods. All of the imposed constraints are gathered in Table 1.
The trajectory generation algorithms are implemented using Yalmip (Löfberg, 2004) in Matlab 2015a with a total processing time of 7.81 sec. In Figure 5, we provide results of the roll, pitch angles $\bar{\phi}$ and $\bar{\theta}$ (plotted in red and green lines) which are actually bounded by the angle boundary ϵ (12) (plotted in blue lines). Moreover, the angle boundary ϵ does not exceed the desired angle limit ϵ_d. Furthermore, in Figure 10, the thrust reference $\bar{T}$ (plotted in dashed red line) stays within the desired thrust limits, $[g - \Delta_g, g + \Delta_g]$. The 3D reference trajectory with waypoints will be presented later with their tracking results in Figure 6.

5.2. Controller design

After having the reference trajectory, the tracking controller is constructed by using the feedback law (35) facilitated by the nested control design (40). The limit of the angle inputs, ϵ_c in (34), must be chosen to satisfy the constraint (52) with ϵ_d and Δ_g given in Table 1, i.e, $\epsilon_c \geq 7.79°$. Thus, we take $\epsilon_c = 8°$. The bounds U_q, with $q \in \{x, y, z\}$ are chosen by solving the optimization problem (51) by using Yalmip (Löfberg, 2004) with a computation time of 0.81 sec and are delineated in Table 2 with the tuning parameters K_{q_1}, K_{q_2} in (40).

5.3. Trajectory tracking results

The quadcopter will track the foregoing trajectory reference which lasts 15 seconds and then keeps hovering at the final position for 5 seconds. We provide illustrations and tracking results for the simulation model and real experiment. For comparison, in each case we take the Integral of Absolute magnitude of the Error (IAE) over the position: $IAE = \int_{t_0=0}^{t_f=20} ||\bar{\xi} - \xi|| dt$ and gather them in Table 3. From Figures 6-7, we

Table 3. Tracking errors under two scenarios.

Tracking error	Simulation	Experiment
IAE	1.6228	3.8353

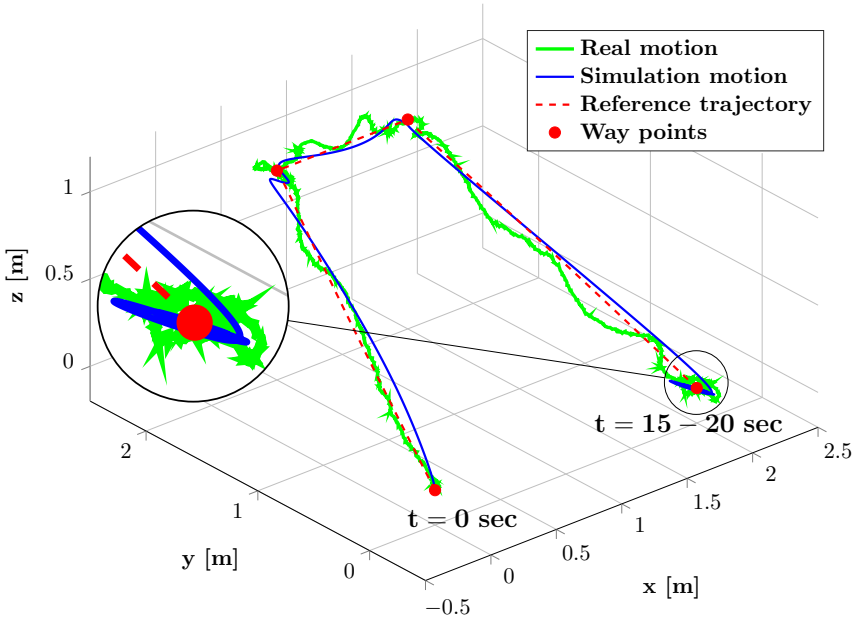

Figure 6. Quadcopter motions under simulation and experiment.

observe that both the simulation results (plotted in blue lines) and the real experiment results (in green lines) track well their reference (in dashed red lines). Furthermore, as observed in the inset from Figure 6 and also in Figure 7 from $t = [15, 20]$ seconds, the proposed controller also works for hovering at the fixed position. Another evidence for the effectiveness of our contributions is that the two IAE results given in Table 3 are not far from the other. These results indicate a good match between our simulation model and real experimental platform and further, validate the tracking capabilities of our proposed control approach.

For the saturation constraint on the angle inputs given in (34), we provide the results of the three angles (plotted in solid lines) compared with their associated inputs ϕ_r and θ_r (plotted in dashed lines in corresponding colors) in Figure 8 for the simulation and in Figure 9 for the real experiment. Due to the control design (35) and (40) with the appropriately chosen parameters given in Table 2, the angle inputs actually respect their limitation ϵ_c as constrained in equation (34). Note that we do not give the references for the yaw angle but their rate, i.e., $\dot{\psi}_r = 0$, hence, the yaw angle value (in solid blue line) varies significantly as can be seen in Figure 9. However, the tracking results of the yaw motion are not important for our control design as long as the yaw angle feedback is introduced to the controller (35) as detailed in Remark 5. Furthermore, as illustrated in Figure 8-9, the roll, pitch angles, ϕ, θ, track their associated inputs, ϕ_r and θ_r with noticeable delays under both simulation and real experiment which cause errors on the position tracking results as observed in Figure 7, especially for the x and y axes. The delays are due to the linear PID compensator employed in the built-in controller detailed in Section 2 while the rotation dynamics

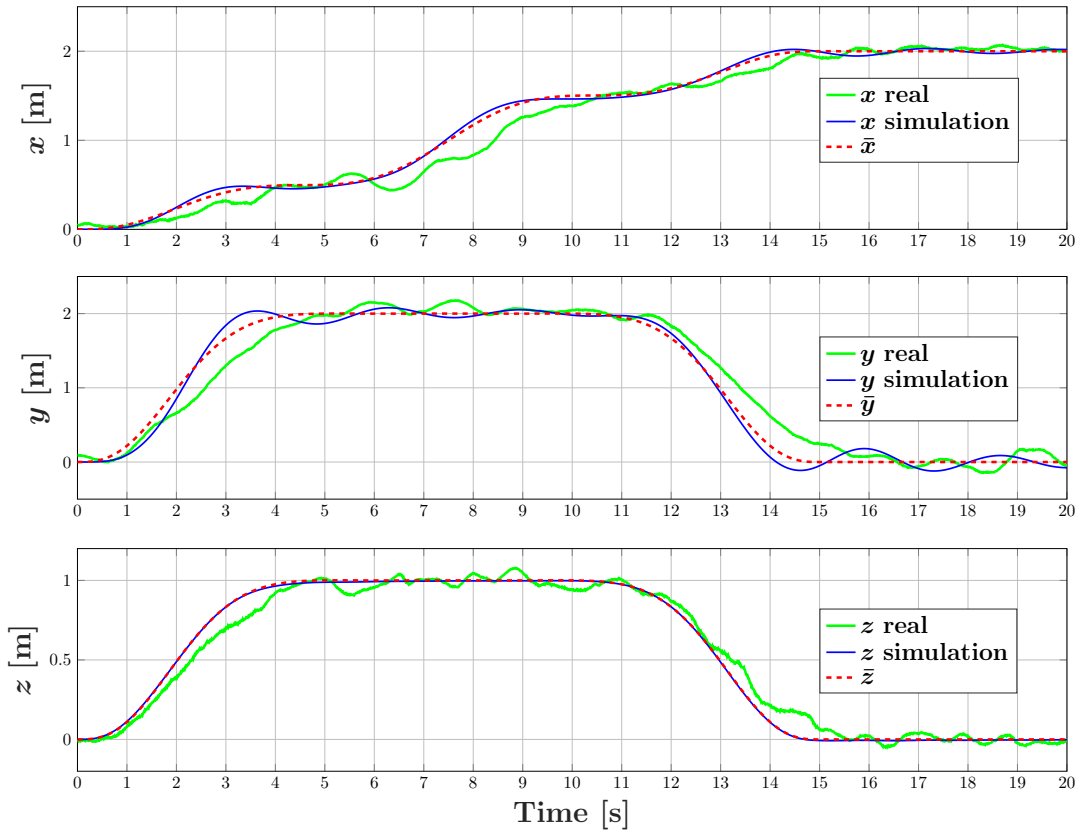

Figure 7. Tracking results of the three axes.

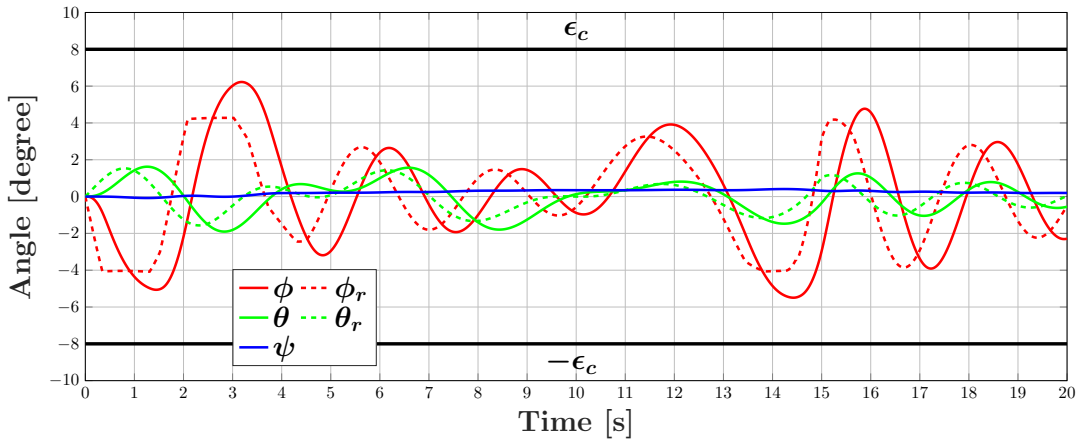

Figure 8. Roll, pitch and yaw angles, ϕ, θ, ψ, with respect to the angle inputs, ϕ_r, θ_r, under simulation.

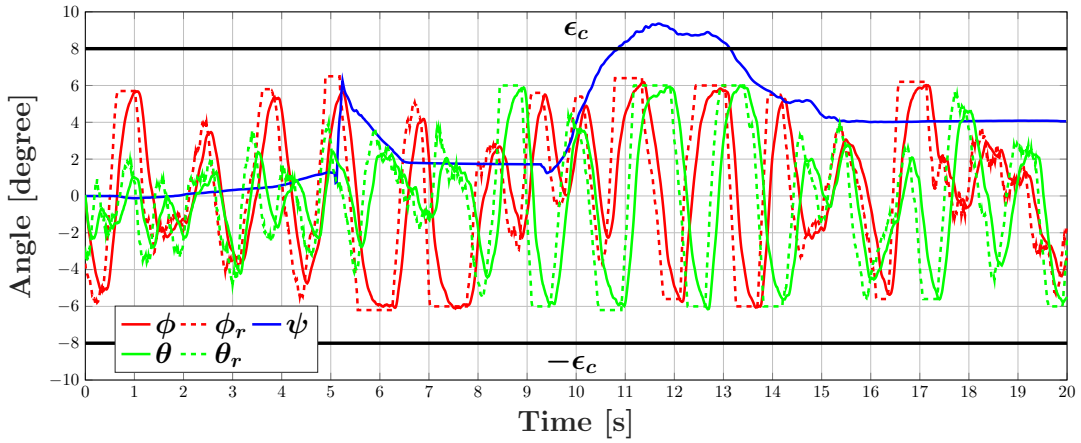

Figure 9. Roll, pitch and yaw angles, ϕ, θ, ψ, with respect to the angle inputs, ϕ_r, θ_r, under experiment.

of a quadcopter have significant nonlinearities (Nguyen, Prodan, Stoican, & Lefèvre, 2017). These phenomena can be reduced by choosing more appropriate PID control parameters or by replacing the built-in PID controller with another more appropriate control scheme (Landry, Deits, Florence, & Tedrake, 2016).

In Figure 10, we provide the results of the thrust input, T_r, given in (35a) under experiment and simulation (plotted in solid green and blue lines), respectively, compared with the thrust reference, $\bar{T}$ (plotted in dashed red line). They all clearly respect the thrust limit, $T_{limit} = 20.18\ m/s^2$, which is too large to appear in the plot. The thrust input under experiment (in green) oscillates around the reference with large amplitude which is mainly due to the precision of the Loco positioning system (around 10 cm), thus, even when the quadcopter is hovering from $t = 15$ seconds, the position feedback is still varying as observed from Figure 7. The other reasons are unknown disturbances affecting the CF quadcopter.

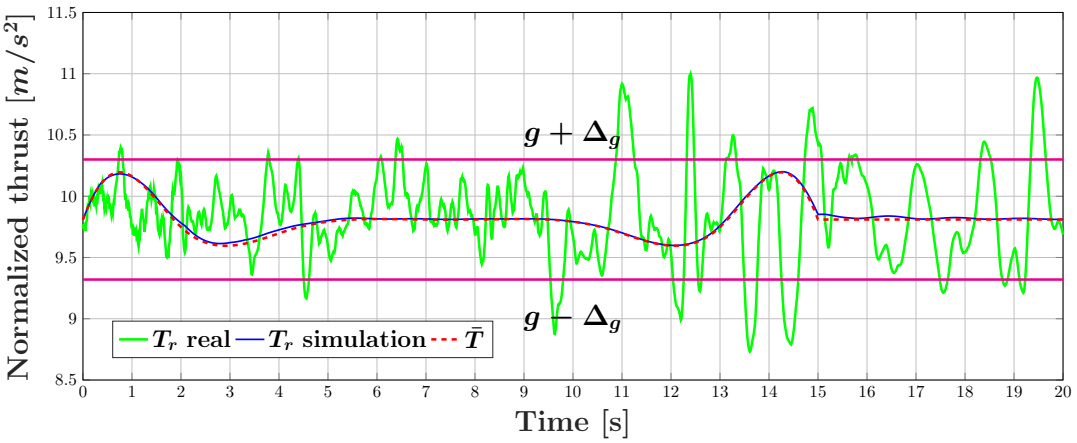

Figure 10. The thrust input T_r under simulation and experiment compared with the thrust reference $\bar{T}$.

6. Conclusions

This paper firstly addressed the optimal constrained motion planning problem for the quadcopter systems by using differential flatness properties and B-spline parametrization. Then, a tracking control design was introduced as a combination of feedback linearization and the nested control method. The control scheme successfully overcomes an unstable mode, usually ignored in the literature, and provides bounded thrust and roll, pitch angles which are designed to respect the reference trajectory and the real system constraints. Thus, the paper creates an unified trajectory generation and tracking control design procedure. Theoretical contributions are successfully applied to the control of a Crazyflie nano-quadcopter for hovering and trajectory tracking.

Future works will concentrate on the introduction of bounded/stochastic disturbances and unexpected events such as faults. Improving the precision of the position observing system for better experimental validation is also a future direction.

References

Aguilar-Ibáñez, C., Sira-Ramírez, H., Suárez-Castañón, M. S., Martínez-Navarro, E., & Moreno-Armendariz, M. A. (2012). The trajectory tracking problem for an unmanned four-rotor system: flatness-based approach. *International Journal of Control*, *85*(1), 69–77.

Cao, N., & Lynch, A. F. (2016). Inner–outer loop control for quadrotor uavs with input and state constraints. *IEEE Transactions on Control Systems Technology*, *24*(5), 1797–1804.

Cowling, I. D., Yakimenko, O. A., Whidborne, J. F., & Cooke, A. K. (2007). A prototype of an autonomous controller for a quadrotor uav. In *Proceedings of the IEEE European Control Conference (ECC)* (pp. 4001–4008).

Craig, J. J. (2005). *Introduction to robotics: mechanics and control* (Vol. 3). Pearson/Prentice Hall Upper Saddle River, NJ, USA.

Do, K. D. (2015). Coordination control of quadrotor vtol aircraft in three-dimensional space. *International Journal of Control*, *88*(3), 543–558.

Engelhardt, T., Konrad, T., Schäfer, B., & Abel, D. (2016). Flatness-based control for a quadrotor camera helicopter using model predictive control trajectory generation. In *Proceedings of the IEEE 24th Mediterranean conference on Control and Automation (MED'16)* (pp. 852–859).

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

Giernacki, W., Skwierczyński, M., Witwicki, W., Wroński, P., & Kozierski, P. (2017). Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering. In *Proceedings of the 22nd IEEE international conference on Methods and Models in Automation and Robotics (MMAR'22)* (pp. 37–42).

Hassanalian, M., & Abdelkefi, A. (2017). Classifications, applications, and design challenges of drones: a review. *Progress in Aerospace Sciences*, *91*, 99–131.

Inaba, M., & Corke, P. (2016). *Robotics research: The 16th international symposium isrr* (Vol. 114). Springer.

Kerma, M., Mokhtari, A., Abdelaziz, B., & Orlov, Y. (2012). Nonlinear h control of a quadrotor (uav), using high order sliding mode disturbance estimator. *International Journal of Control*, *85*(12), 1876–1885.

Landry, B., Deits, R., Florence, P. R., & Tedrake, R. (2016). Aggressive quadrotor flight through cluttered environments using mixed integer programming. In *International conference on robotics and automation (icra)* (pp. 1469–1475).

Lévine, J. (2011). On necessary and sufficient conditions for differential flatness. *Applicable Algebra in Engineering, Communication and Computing*, *22*(1), 47–90.

Liu, W., Chitour, Y., & Sontag, E. (1996). On finite-gain stabilizability of linear systems subject to input saturation. *SIAM Journal on Control and Optimization*, *34*(4), 1190–1219.

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

Lu, H., Liu, C., Guo, L., & Chen, W.-H. (2017). Constrained anti-disturbance control for a quadrotor based on differential flatness. *International Journal of Systems Science*, *48*(6), 1182–1193.

Luis, C., & Ny, J. L. (2016, August). *Design of a trajectory tracking controller for a nanoquadcopter* (Tech. Rep.). Mobile Robotics and Autonomous System Laboratory, Polytechnique Montreal.

Maggiore, M. (2015). Reduction principles for hierarchical control design.

Mueller, M. W., & D'Andrea, R. (2013). A model predictive controller for quadrocopter state interception. In *Proceedings of the IEEE European Control Conference (ECC'13)* (pp. 1383–1389).

Nguyen, N. T., Prodan, I., & Lefèvre, L. (2017). Multi-layer optimization-based control design for quadcopter trajectory tracking. In *Proceedings of the 25th IEEE Mediterranean Conference on Control and Automation (MED'17)* (p. 601-606).

Nguyen, N. T., Prodan, I., & Lefèvre, L. (2018). Effective angular constrained trajectory generation for thrust-propelled vehicles. In *Proceedings of the 16th European Control Conference (ECC'18)* (p. 1833-1838).

Nguyen, N. T., Prodan, I., Stoican, F., & Lefèvre, L. (2017). Reliable nonlinear control for quadcopter trajectory tracking through differential flatness. *Proceeding of the 20th IFAC World Congress. IFAC-PapersOnLine*, *50*(1), 6971-6976.

Piegl, L., & Tiller, W. (1995). B-spline curves and surfaces. In *The nurbs book* (pp. 81–116). Springer.

Prodan, I., Olaru, S., Bencatel, R., Sousa, J., Stoica, C., & Niculescu, S. (2013). Receding horizon flight control for trajectory tracking of autonomous aerial vehicles. *Control Engineering Practice*, *21*(10), 1334–1349.

Shi, X.-N., Zhang, Y.-A., & Zhou, D. (2015). A geometric approach for quadrotor trajectory tracking control. *International Journal of Control*, *88*(11), 2217–2227.

Stoican, F., Prodan, I., & Popescu, D. (2015). Flat trajectory generation for way-points relaxations and obstacle avoidance. In *Proceedings of the 23th IEEE Mediterranean Conference on Control and Automation (MED'15)* (pp. 695–700).

Stoican, F., Prodan, I., Popescu, D., & Ichim, L. (2017). Constrained trajectory generation for uav systems using a b-spline parametrization. In *Proceedings of the 25th IEEE Mediterranean Conference on Control and Automation (MED'17)* (pp. 613–618).

Suryawan, F. (2012). *Constrained trajectory generation and fault tolerant control based on differential flatness and b-splines* (Unpublished doctoral dissertation). School of Electrical Engineering and Computer Science, The University of Newcastle, Australia.

Teel, A. R. (1992). Global stabilization and restricted tracking for multiple integrators with bounded controls. *Systems & control letters*, *18*(3), 165–171.

Wu, F., & Lu, B. (2004). Anti-windup control design for exponentially unstable lti systems with actuator saturation. *Systems & Control Letters*, *52*(3), 305–322.

Zhao, W., & Go, T. H. (2014). Quadcopter formation flight control combining mpc and robust feedback linearization. *Journal of the Franklin Institute*, *351*(3), 1335–1355.

Appendix A. The unstable mode (37) of the feedback linearization law

Proposition A.1. *Assuming that the virtual input u_z is designed for stabilizing the quadcopter altitude dynamics* (38) *at the origin $z = 0$ with the normal PD corrective term given as:*

$$u_z = -K_D\dot{z} - K_P z, \tag{A1}$$

where K_P, K_D are positive control parameters. According to Proposition 4.1, we have that $z = 0$ is the asymptotically stable equilibrium point for the dynamics $\ddot{z} = u_z$ from (36) *if $u_z \geq -g$. While if $u_z < -g$, the dynamics $\ddot{z} = -u_z - 2g$ is obtained from* (37) *and the associated equilibrium point is $z = 2g/K_P$ which is unstable.*

Proof. It is straightforward to have that: i) the origin $z = 0$ is asymptotically stable for the dynamics $\ddot{z} = -K_D\dot{z} - K_P z$ obtained by introducing (A1) into (36); ii) substituting (A1) into (37) leads to $\ddot{z} = K_D\dot{z} + K_P\,(z - 2g/K_P)$ which has the unstable equilibrium point $z = 2g/K_P$ with correct values of the K_P and K_D parameters of course. □

Note that, there is no pair of parameters K_P, K_D such that both closed -loop dynamics (36)–(37) are stabilized. On the other hand, there exist circumstances where the unstable dynamics (37) will lead (via the positive K_P, K_D parameter) to a value $u_z > -g$, thus allowing to switch to the stable dynamics (36).

Appendix B. Proof for the stability of the error dynamics (44)

This section proves the stability of the error dynamics (44) with the control parameters $K_{q_1}, K_{q_2} \in \mathbb{R}^+$. The forthcoming result is a generalization of the proof for the particular case of $K_{q_1} = K_{q_2} = 1$ which was introduced in Teel (1992). Considering the new variables e_1, e_2 defined as:

$$e_1 = K_{q_2} e_q + \dot{e}_q, \quad e_2 = \dot{e}_q. \tag{B1}$$

Introducing (B1) into (44) leads to:

$$\dot{e}_1 = \lambda_q \left[\frac{e_2}{2R_2} - \sigma \left(\frac{1}{2}\sigma \left(\frac{e_1}{R_1} \right) + \frac{e_2}{2R_2} \right) \right], \tag{B2}$$

$$\dot{e}_2 = -\lambda_q \sigma \left(\frac{1}{2}\sigma \left(\frac{e_1}{R_1} \right) + \frac{e_2}{2R_2} \right), \tag{B3}$$

where $R_1 = \lambda_q / K_{q_1}$ and $R_2 = \lambda_q / (2K_{q_2})$. Hereinafter, we prove that the equilibrium $(e_1 = 0, e_2 = 0)$ of the system (B2)-(B3) is globally asymptotically stable.
Firstly, we consider the "big" value of e_2, such that:

$$|e_2| > R_2, \tag{B4}$$

By defining the Lyapunov function $V_2 = e_2^2$, we have that:

$$\dot{V}_2 = -2\lambda_q e_2 \sigma \left(\frac{1}{2}\sigma \left(\frac{e_1}{R_1} \right) + \frac{e_2}{2R_2} \right). \tag{B5}$$

Case 1: $e_2 > 0$, (B4) leads to:

$$\frac{1}{2}\sigma \left(\frac{e_1}{R_1} \right) + \frac{e_2}{2R_2} > -\frac{1}{2} + \frac{1}{2} = 0. \tag{B6}$$

Introducing condition (B6) and $e_2 > 0$ into (B5) strictly leads to $\dot{V}_2 < 0$.
Case 2: $e_2 < 0$, (B4) leads to:

$$\frac{1}{2}\sigma \left(\frac{e_1}{R_1} \right) + \frac{e_2}{2R_2} < \frac{1}{2} - \frac{1}{2} = 0. \tag{B7}$$

Introducing condition (B7) and $e_2 < 0$ into (B5) strictly leads to $\dot{V}_2 < 0$.
Thus, if $e_2 \notin E_2 = \{e_2 : |e_2| \leq R_2\}$, $V_2 = e_2^2$ strictly decreases. Consequentially, e_2 enters E_2 in a finite time and remains in E_2.
Secondly, we consider that $e_2 \in E_2$, i.e., $|e_2| \leq R_2$, we arrive to:

$$\left| \frac{1}{2}\sigma \left(\frac{e_1}{R_1} \right) + \frac{e_2}{2R_2} \right| \leq \frac{1}{2} + \frac{1}{2} = 1. \tag{B8}$$

Thus, by considering the definition of the saturation function $\sigma(\cdot)$ given in (42), we have that:

$$\sigma \left(\frac{1}{2}\sigma \left(\frac{e_1}{R_1} \right) + \frac{e_2}{2R_2} \right) = \frac{1}{2}\sigma \left(\frac{e_1}{R_1} \right) + \frac{e_2}{2R_2}. \tag{B9}$$

Introducing (B9) into (B2) leads to:

$$\dot{e}_1 = -\frac{\lambda_q}{2}\sigma \left(\frac{e_1}{R_1} \right). \tag{B10}$$

By considering the Lyapunov function $V_1 = e_1^2$, we have that:

$$\dot{V}_1 = -\lambda_q e_1 \sigma\left(\frac{e_1}{R_1}\right) < 0. \tag{B11}$$

Thus, V_1 is decreasing which indicates that e_1 enters $E_1 = \{e_1, |e_1| \leq R_1\}$ in a finite time. When $e_1 \in E_1$, the dynamics (B2)-(B3) become:

$$\dot{e}_1 = -\frac{\lambda_q}{2R_1} e_1, \tag{B12}$$

$$\dot{e}_2 = -\frac{\lambda_q}{2R_1} e_1 - \frac{\lambda_q}{2R_2} e_2, \tag{B13}$$

which is exponentially stable around the origin ($e_1 = 0, e_2 = 0$). Thus, ($e_q = 0, \dot{e}_q = 0$) is also globally asymptotically stable for the error dynamics (44), hence, completing the proof.

Appendix C. Proof for the upper bound of max $|\bar{\bar{q}}|$, $q \in \{x, y, z\}$

This section introduces the proof for the upper bound of $\max |\bar{\bar{q}}|$, $q \in \{x, y, z\}$ employed in (54)-(56) which are rewritten hereinafter for keep tracking easily:

$$\max |\bar{\bar{x}}| \leq (g + \Delta_g) \sin \epsilon_d, \tag{C1}$$

$$\max |\bar{\bar{y}}| \leq (g + \Delta_g) \sin \epsilon_d, \tag{C2}$$

$$\max |\bar{\bar{z}}| \leq g - (g - \Delta_g) \cos \epsilon_d. \tag{C3}$$

From (31) and (10), with $\bar{\bar{q}}$, $q \in \{x, y, z\}$ are the results of the optimal trajectory generation, i.e., $\begin{bmatrix} \bar{\bar{x}}^2 + \bar{\bar{y}}^2 & (\bar{\bar{z}} + g)^2 \end{bmatrix}^\top$ stays inside the polytopic set P (31), we obtain that:

$$\bar{\bar{x}}^2 + \bar{\bar{y}}^2 \leq (g + \Delta g)^2 \sin^2 \epsilon_d, \tag{C4}$$

$$(g - \Delta g)^2 \cos^2 \epsilon_d \leq (\bar{\bar{z}} + g)^2 \leq (g + \Delta g)^2. \tag{C5}$$

From (C4), we arrive to:

$$|\bar{\bar{x}}| \leq (g + \Delta_g) \sin \epsilon_d, \ |\bar{\bar{y}}| \leq (g + \Delta_g) \sin \epsilon_d. \tag{C6}$$

Thus, (C6) leads to (C1) and (C2). Furthermore, from (C5), we have that:

$$(g - \Delta g) \cos \epsilon_d - g \leq \bar{\bar{z}} \leq \Delta g. \tag{C7}$$

As a result, we arrive to:

$$\max |\bar{\bar{z}}| \leq \max \left(|(g - \Delta g) \cos \epsilon_d - g|, \Delta g \right). \tag{C8}$$

Next, we have that:

$$|(g - \Delta g) \cos \epsilon_d - g| = g(1 - \cos \epsilon_d) + \Delta g \cos \epsilon_d. \tag{C9}$$

By employing the condition $\Delta g < g$ from (30), we obtain that:

$$g(1 - \cos \epsilon_d) + \Delta g \cos \epsilon_d \geq \Delta g. \tag{C10}$$

Combining (C9)-(C10) leads to:

$$\max \left(\left| (g - \Delta g) \cos \epsilon_d - g \right|, \Delta g \right) = g(1 - \cos \epsilon_d) + \Delta g \cos \epsilon_d. \tag{C11}$$

Introducing (C11) into (C8) leads to (C3), thus, completing the proof.