2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Tube MPC via flatness for multicopter trajectory tracking

Huu-Thinh Do[1] and Ionela Prodan[1]

***Abstract*— Nonlinearity and dimensionality have always been computationally challenging problems when it comes to the online implementation of optimization-based control approaches, especially in the presence of disturbances. In this paper, we show how to alleviate the problem via a variable reformulation derived from differential flatness for a quadcopter vehicle. More specifically, we present a robust model predictive control design, to track a predefined trajectory of a quadcopter in the presence of disturbances. The synthesis procedure starts with a coordinate change mediated by the model's flatness property. In this new representation, the dynamics become linear in closed-loop at the price of more convoluted constraint expressions, which are usually disregarded in the literature or simple approximations are proposed. Subsequently, with a proper parameterization to portray the feasible domain, the trajectory tracking problem is transformed into the stabilization of a constrained linear time-invariant system under disturbances, which is then handled by a robust model predictive controller. Simulations and experimental results are presented to analyze and validate the proposed scheme.**

***Index Terms*— Quadcopter trajectory tracking, feedback linearization, tube-based MPC, differential flatness.**

I. Introduction

The problem of maneuvering the quadcopter vehicles with not only stability, but also optimality and robustness, still remains open for discussion in the literature. The primary issue lies in the vehicle's trigonometric complexity, which typically demands high computational power to find a feasible stabilizing control for the non-linear problem [1], [2]. To address these challenges, while many robust techniques have been offered, Model Predictive Control (MPC) is of special interest to us thanks to its optimality and constraint-handling capability. Theoretically, over a finite prediction horizon in the future, stability, feasibility, and robustness can be ensured with a proper formulation of the cost, constraints, and a local controller within the MPC design [3]. However, due to the problem's complex expressions, its solution is hard to find, at least in the limited sampling time of aerial vehicles. As a partial solution, linearization of the system dynamics around its operating points is typically exploited to achieve a linear representation (also referred to as the approximation with "small" valued angles/states), then robust or adaptive approaches can be applied to ensure performance and constraints. Although the strategy appears to be generic and computationally attractive, the controller's validity is only secured "near" a reference [4], [5], or partially assured for decoupled dynamics [6], [7].

[1]Univ. Grenoble Alpes, Grenoble INP†, LCIS, 26000 Valence, France. Email: {huu-thinh.do,ionela.prodan}@lcis.grenoble-inp.fr
†Institute of Engineering and Management Univ. Grenoble Alpes. This work is funded by La Région, Pack Ambition Recherche 2021 - PlanMAV.

In contrast, the notion of exact linearization has promised to solve the issue by reshaping the nonlinear dynamics into a linear, controllable system. Indeed, the quadcopter's model is known to be differentially flat [8], [9] (i.e., all states and inputs can be expressed algebraically with a special output called flat output and a finite number of its derivatives). This input-output representation implies that, via an input change and a dynamic feedback, the system can be converted into chains of integrators in a new coordinate, called *the flat output space* [10]. Ideally, this property is beneficial for both the vehicle's trajectory planning problem and control synthesis, as they are now reduced to the level of trivial integrators. However, owing to that reformulation, although the dynamics' nonlinearity is discarded, the original constraint set becomes convoluted. To handle this disadvantage, conservative linear estimations of such a new feasible domain have been exploited to design a controller for the new linear system [8], [11]. Even so, due to such conservatism, the robustness of the quadcopter under this strategy remains inadequately covered in the literature, especially for the trajectory tracking problem. Thus, in this paper, we exploit the well-known linearizing transformation for the quadcopter's translation dynamics and present a synthesis procedure for robust trajectory tracking based on MPC. Specifically, we:

- address the problem of the nonlinear convoluted constraints arising from the linearization in closed-loop in the flat output space by a surface parameterization;
- formulate an efficient synthesis procedure for the quadcopter's position tracking with tube-based MPC with ellipsoidal terminal ingredients;
- validate the design via simulation and experimental tests over the Crazyflie 2.1 nano-drone platform. Experiment video can be found at `https://youtu.be/aJHZyouXtwo`.

Notation: Bold capital letters are matrices with appropriate dimensions. $\text{diag}(\cdot)$ returns a diagonally arranged matrix with the components given within. $\boldsymbol{Q} \succ 0$ and $\boldsymbol{Q} \succeq 0$ imply $\boldsymbol{Q}$ is positive definite and semi-definite, respectively. For $\boldsymbol{Q} \succeq 0$, $\lambda_{max}(\boldsymbol{Q})$, $\lambda_{min}(\boldsymbol{Q})$ denote its largest and smallest eigenvalues, respectively. Bold letters denote vectors, while the subscript k denote the discrete step k. $\mathbf{0}_{n\times m}$ represents a matrix of dimension $n \times m$ of which all components are 0. For $\boldsymbol{x} \in \mathbb{R}^n$, ${}^i\boldsymbol{x}$ represents its i-th entry, while the superscript "ref" presents the reference for the signal to follow. $\|\boldsymbol{x}\|_{\boldsymbol{P}} \triangleq \sqrt{\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}}$ and $\|\boldsymbol{x}\|_2 \triangleq \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$. $\mathbf{N}^p(a,b)$ returns a set of p evenly spaced samples over the interval $[a,b]$. $\text{co}\{\cdot\}$ denote the convex hull. Finally, $\oplus$ and $\ominus$ denote the Minkowski sum and Pontryagin difference, respectively.

II. Model representation and MPC setup

First, we recapitulate the quadcopter model employed for position control design and its representation in the flat output space, which possesses linear dynamics and a non-linear constraint. The ellipsoidal terminal ingredient synthesis for constrained linear system is also presented.

A. Quadcopter position control

The quadcopter's discretized translation dynamics:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_d\boldsymbol{x}_k + \boldsymbol{B}_d\boldsymbol{h}_\psi(\boldsymbol{u}_k), \tag{1}$$

where the vectors and matrices are defined as:

$$\begin{aligned}
&\boldsymbol{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k, z_k, \dot{z}_k]^\top, \boldsymbol{u}_k = [T_k, \phi_k, \theta_k]^\top,\\
&\boldsymbol{A}_d = \text{diag}(A_1, A_1, A_1), \boldsymbol{B}_d = \text{diag}(B_1, B_1, B_1)\\
&\boldsymbol{h}_\psi(\boldsymbol{u}_k) = \begin{bmatrix} T_k(\cos\phi_k \sin\theta_k \cos\psi + \sin\phi_k \sin\psi) \\ T_k(\cos\phi_k \sin\theta_k \sin\psi - \sin\phi_k \cos\psi) \\ -g + T_k \cos\phi_k \cos\theta_k \end{bmatrix}
\end{aligned} \tag{2}$$

where, at time step k, $\boldsymbol{x}_k \in \mathbb{R}^6$ denotes the system's with x_k, y_k, z_k and $\dot{x}_k, \dot{y}_k, \dot{z}_k$ as the positions of the drone and their time derivative, respectively; $\boldsymbol{u}_k \in \mathbb{R}^3$ gathers the three input including the normalized thrust T_k, the roll (ϕ_k) and pitch (θ_k) angles; the yaw angle $\psi \in [-\pi; \pi]$ is assumed to be known by sensor measurement. t_s is the sampling time and $\boldsymbol{A}_1 = \left[\begin{smallmatrix} 1 & t_s \\ 0 & 1 \end{smallmatrix}\right], \boldsymbol{B}_1 = \begin{bmatrix} 0.5t_s^2 & t_s \end{bmatrix}^\top$.

The inputs' constraints are defined as:

$$\boldsymbol{u}_k \in \mathcal{U} = \{0 \le T_k \le T_{\max}\,; |\phi_k|\,, |\theta_k| \le \epsilon_{max}\} \tag{3}$$

where g is the gravitational acceleration and $T_{max} > 0$, $\epsilon_{max} \in (0; \pi/2)$ are constant bounds of the inputs.

Recall the variable reformulation (a.k.a. inverse kinematics) $\boldsymbol{u}_k = \boldsymbol{\varphi}_\psi(\boldsymbol{v}_k)$ where:

$$\begin{aligned}
\boldsymbol{\varphi}_\psi(\boldsymbol{v}_k) &\triangleq [{}^1\boldsymbol{\varphi}_\psi(\boldsymbol{v}_k),\, {}^2\boldsymbol{\varphi}_\psi(\boldsymbol{v}_k),\, {}^3\boldsymbol{\varphi}_\psi(\boldsymbol{v}_k)]^\top\\
&= \begin{bmatrix} \sqrt{v_{1k}^2 + v_{2k}^2 + (v_{3k}+g)^2} \\ \sin^{-1}\left((v_{1k}\sin\psi - v_{2k}\cos\psi)/\left({}^1\boldsymbol{\varphi}_\psi(\boldsymbol{v}_k)\right)\right) \\ \tan^{-1}((v_{1k}\cos\psi + v_{2k}\sin\psi)/(v_{3k}+g)) \end{bmatrix},
\end{aligned} \tag{4}$$

the system (1) becomes:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_d\boldsymbol{x}_k + \boldsymbol{B}_d\boldsymbol{v}_k, \tag{5}$$

under the condition $v_{3k} \ge -g$, with $\boldsymbol{v}_k \triangleq [v_{1k}, v_{2k}, v_{3k}]^\top$ gathering the new input. For a more detailed computation, we refer to the work [11] where both the flat representation and the linearizing law are derived.

Remark 1: It is important to note that the transformation (4) is deduced from the input-output relationship of the quadcopter's model, where the flat output is considered as $[x, y, z]^\top$ [9], [11]. Moreover, for this model, as a property of differentially flat systems, the system dynamics (5) describe three concatenated chains of integrators.

As discussed in our previous work [9], the feasible domain $\tilde{\mathcal{U}} \triangleq \{\boldsymbol{v}_k : \boldsymbol{\varphi}_\psi(\boldsymbol{v}_k) \in \mathcal{U}\}$ is non-convex and time varying (due to the dependence on ψ), which proves impractical for real-time control. To handle this problem, the convex time-invariant subset of the domain is adopted as:

$$\begin{aligned}
\mathcal{V}_c = \Big\{\boldsymbol{v} = &[v_1, v_2, v_3]^\top \in \mathbb{R}^3 :\\
&\begin{bmatrix} v_1^2 + v_2^2 + (v_3+g)^2 - T_{max}^2 \\ \sqrt{v_1^2 + v_2^2} - (v_3+g)\tan\epsilon_{max} \end{bmatrix} \le 0\Big\}.
\end{aligned} \tag{6}$$

For brevity, the proof for the inclusion $\mathcal{V}_c \subset \tilde{\mathcal{U}}, \forall\psi \in \mathbb{R}$ is omitted, the details can be found in [9].

Remark 2: It can be seen directly that, $\mathcal{V}_c$ contains the origin in its interior, thus, there exists $\boldsymbol{v}_\varepsilon > 0$ such that the set can be under-approximated by the saturation constraint: $|\boldsymbol{v}| \le \boldsymbol{v}_\varepsilon$. Then, anti-saturation control can be designed via an anti-windup PID or nested saturation control [11], [12] with global asymptotic stability. However, as shown in [9], such norm-based approximation truncates significantly the original set, conservatively limiting the control admissible solution. Thus, subsequently, we exploit the set's geometric representation to construct a less conservative approximation.

From (6), $\mathcal{V}_c$ contains a sphere of radius T_{max} and a convex cone characterized by ϵ_{max}. With this geometric interpretation and by parameterizing such bounding surfaces, an inner-approximation of $\mathcal{V}_c$ can be constructed as:

$$\mathcal{V}_c^{\text{in}} = \text{co}\left\{\begin{matrix}[0, 0, -g]^\top, [R^\star\cos\alpha_i, R^\star\sin\alpha_i, v_3^\star]^\top,\\ [r_j\cos\alpha_i, r_j\sin\alpha_i, \sqrt{T_{max}^2 - r_j^2} - g]^\top\end{matrix}\right\} \tag{7}$$

with $\alpha_i \in \mathbf{N}^{s_1}(0, 2\pi), r_j \in \mathbf{N}^{s_2}(0, R^\star)$ for some integer $s_1, s_2 > 2$ (the larger s_1 and s_2, the more closely $\mathcal{V}_c^{\text{in}}$ approaches $\mathcal{V}_c$) and $R^\star \triangleq T_{max}\sin\epsilon_{max}$, $v_3^\star \triangleq T_{max}\cos\epsilon_{max} - g$. An illustration of the parameterization is shown in Fig. 1. Hereinafter, as opposed to $\mathcal{V}_c$, $\mathcal{V}_c^{\text{in}}$ will be employed to take advantage of its linear representation in the control problem, which is discussed later.

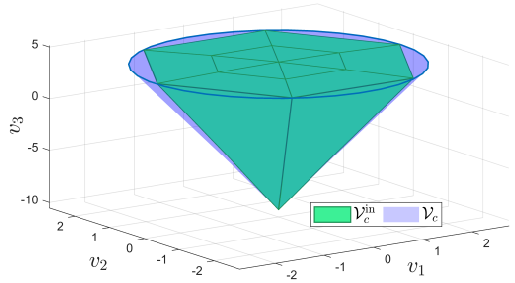

Fig. 1. $\mathcal{V}_c$ parameterization with $s_1 = 7, s_2 = 3$, $g = T_{max}/1.45 = 9.81m/s^2, \epsilon_{max} = 0.1745\ rad$.

B. Tube-MPC synthesis

Next, we summarize the MPC synthesis for a linear system with stability and feasibility guaranteed, which later serves as the ingredients for the tube-based MPC design.

Proposition 1: (*Largest constrained ellipsoid [13]*) Consider a symmetric matrix $\boldsymbol{P} \succ 0$ and a polyhedron $\mathcal{H} = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}_i^\top\boldsymbol{x} \le b_i, i = 1, ..., p\}$, define an ellipsoid: $\mathcal{E} = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_{\boldsymbol{P}} \le \varepsilon\}$. Then, the maximum ε such that $\mathcal{E} \subset \mathcal{H}$ can be found by solving:

$$\begin{aligned}
&\varepsilon^* = \arg\max_\varepsilon\ \log\det\mathbf{B}(\varepsilon),\\
&\text{s.t } \mathbf{B}(\varepsilon) = (\boldsymbol{V}\boldsymbol{S}^{1/2}\boldsymbol{U}^\top)^{-1}\varepsilon \text{ and } \|\mathbf{B}(\varepsilon)\boldsymbol{a}_i\|_2 \le b_i.
\end{aligned} \tag{8}$$

where $\boldsymbol{V}, \boldsymbol{S}, \boldsymbol{U}$ come from the singular value decomposition of $\boldsymbol{P}$ with $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$. □

Proposition 2: (Stabilizing MPC with ellipsoidal terminal region [14]) Given the matrices $\boldsymbol{Q} \succ 0, \boldsymbol{R} \succeq 0$ and a constrained linear dynamics:

$$\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{A}_d\tilde{\boldsymbol{x}}_k + \boldsymbol{B}_d\tilde{\boldsymbol{v}}_k \text{ subject to } \tilde{\boldsymbol{v}}_k \in \mathcal{V}^\star, \tag{9}$$

where $\tilde{\boldsymbol{x}}_k \in \mathbb{R}^n, \tilde{\boldsymbol{v}}_k \in \mathbb{R}^m$, and $\mathcal{V}^\star$ is a polyhedral constraint set, consider the following procedure:

- Choose a feedback gain $\boldsymbol{K}$ such that $\boldsymbol{A}_{cl} \triangleq \boldsymbol{A}_d + \boldsymbol{B}_d\boldsymbol{K}$ is a Schur matrix;
- Choose $\boldsymbol{M}$ such that
$$\boldsymbol{M} \succeq \boldsymbol{Q} + \lambda_{max}(\boldsymbol{R})\boldsymbol{K}^\top\boldsymbol{K}; \tag{10}$$
- Solve the Lyapunov equation (11) to find $\boldsymbol{P}$:
$$\boldsymbol{A}_{cl}^\top\boldsymbol{P}\boldsymbol{A}_{cl} - \boldsymbol{P} + \boldsymbol{M} = 0; \tag{11}$$
- Following Proposition 1 to find the largest ellipsoid:
$$\mathcal{E}_f = \left\{\tilde{\boldsymbol{x}} : \tilde{\boldsymbol{x}}^\top\boldsymbol{P}\tilde{\boldsymbol{x}} \leq \varepsilon_f\right\}, \tag{12}$$
such that $\mathcal{E}_f \subset \tilde{\mathcal{X}} \triangleq \{\tilde{\boldsymbol{x}} : \boldsymbol{K}\tilde{\boldsymbol{x}} \in \mathcal{V}^\star\}$.

Then, we have:

1) The control law $\tilde{\boldsymbol{v}}_k = \boldsymbol{K}\tilde{\boldsymbol{x}}_k \in \mathcal{V}^\star, \forall\tilde{\boldsymbol{x}}_k \in \mathcal{E}_f$ and also renders the ellipsoid $\mathcal{E}_f$ forward invariant for the system (9) (i.e, $\tilde{\boldsymbol{x}}_k \in \mathcal{E}_f \Rightarrow \tilde{\boldsymbol{x}}_{k+1} \in \mathcal{E}_f, k \geq 0$ [15]);
2) $V_f(\tilde{\boldsymbol{x}}_{k+1}) + V_s(\tilde{\boldsymbol{x}}_k, \boldsymbol{K}\tilde{\boldsymbol{x}}_k) - V_f(\tilde{\boldsymbol{x}}_k) \leq 0 \,\forall\tilde{\boldsymbol{x}}_k \in \mathcal{E}_f$
$$\text{with } \begin{cases} V_f(\tilde{\boldsymbol{x}}_k) \triangleq \|\tilde{\boldsymbol{x}}_k\|_{\boldsymbol{P}}^2 \\ V_s(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{v}}_k) \triangleq \|\tilde{\boldsymbol{x}}_k\|_{\boldsymbol{Q}}^2 + \|\tilde{\boldsymbol{v}}_k\|_{\boldsymbol{R}}^2. \end{cases} \tag{13}$$

□

Proof: From the definition (12), $\tilde{\boldsymbol{v}}_k = \boldsymbol{K}\tilde{\boldsymbol{x}}_k \in \mathcal{V}^\star$, $\forall\tilde{\boldsymbol{x}}_k \in \mathcal{E}_f$ always holds. Next, with condition (11), we have:
$$\begin{aligned} &\|\tilde{\boldsymbol{x}}_{k+1}\|_{\boldsymbol{P}}^2 = \tilde{\boldsymbol{x}}_k^\top\boldsymbol{A}_{cl}^\top\boldsymbol{P}\boldsymbol{A}_{cl}\tilde{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}_k^\top(\boldsymbol{P} - \boldsymbol{M})\tilde{\boldsymbol{x}}_k \\ &\leq (1 - \lambda_{min}(\boldsymbol{M})/\lambda_{max}(\boldsymbol{P}))\tilde{\boldsymbol{x}}_k^\top\boldsymbol{P}\tilde{\boldsymbol{x}}_k \leq \tilde{\boldsymbol{x}}_k^\top\boldsymbol{P}\tilde{\boldsymbol{x}}_k \end{aligned} \tag{14}$$
Then, if $\tilde{\boldsymbol{x}}_k \in \mathcal{E}_f$, $\|\tilde{\boldsymbol{x}}_{k+1}\|_{\boldsymbol{P}}^2 \leq \|\tilde{\boldsymbol{x}}_k\|_{\boldsymbol{P}}^2 \leq \varepsilon_f$, or $\tilde{\boldsymbol{x}}_{k+1} \in \mathcal{E}_f$, namely $\mathcal{E}_f$ is forward invariant for the closed-loop.

Finally, with the conditions in (10) and (11), we have:
$$\begin{aligned} &V_f(\tilde{\boldsymbol{x}}_{k+1}) + V_s(\tilde{\boldsymbol{x}}_k, \boldsymbol{K}\tilde{\boldsymbol{x}}_k) - V_f(\tilde{\boldsymbol{x}}_k) \\ &= \tilde{\boldsymbol{x}}_k^\top\boldsymbol{A}_{cl}^\top\boldsymbol{P}\boldsymbol{A}_{cl}\tilde{\boldsymbol{x}}_k + \tilde{\boldsymbol{x}}_k^\top(\boldsymbol{Q} + \boldsymbol{K}^\top\boldsymbol{R}\boldsymbol{K})\tilde{\boldsymbol{x}}_k - \tilde{\boldsymbol{x}}_k^\top\boldsymbol{P}\tilde{\boldsymbol{x}}_k \\ &= -\tilde{\boldsymbol{x}}_k^\top(\boldsymbol{M} - \boldsymbol{Q} - \boldsymbol{K}^\top\boldsymbol{R}\boldsymbol{K})\tilde{\boldsymbol{x}}_k \leq 0, \forall\tilde{\boldsymbol{x}}_k \in \mathbb{R}^n. \end{aligned} \tag{15}$$

■

Corollary 1: From the settings of Proposition 2, consider the standard optimization problem at time step k [3]:
$$V_{N_p}(\tilde{\boldsymbol{x}}_k) \triangleq \min \textstyle\sum_{i=0}^{N_p-1} V_s(\tilde{\boldsymbol{x}}_{k+i}, \tilde{\boldsymbol{v}}_{k+i}) + V_f(\tilde{\boldsymbol{x}}_{k+N_p}) \tag{16}$$
$$\text{s.t} \begin{cases} \tilde{\boldsymbol{x}}_{i+k+1} = \boldsymbol{A}_d\tilde{\boldsymbol{x}}_{i+k} + \boldsymbol{B}_d\tilde{\boldsymbol{v}}_{i+k};\ \tilde{\boldsymbol{x}}_{k+N_p} \in \mathcal{E}_f & \text{(17a)} \\ \tilde{\boldsymbol{v}}_{i+k} \in \mathcal{V}^\star,\ i \in \{0, 1, ..., N_p - 1\}, & \text{(17b)} \end{cases}$$
where N_p denotes the prediction horizon, $\mathcal{E}_f$ is the terminal region; the polyhedron $\tilde{\mathcal{V}}$ describes the input constraints; $\boldsymbol{Q}, \boldsymbol{P} \succ 0, \boldsymbol{R} \succeq 0$ are weighting matrices. Then, by finding $\boldsymbol{P}$ and ε_f as in Proposition 2, the problem (16)-(17) is recursively feasible, and the origin $\boldsymbol{0}_{n\times 1}$ is asymptotically stable under the closed-loop system given by repetitively applying the first control action found in (16)-(17).

Proof: The properties achieved in Proposition 2 satisfy the conditions for feasibility and stability presented in Section 3.3 in [3] for a standard MPC scheme. ■

Remark 3: The existence of $\mathcal{E}_f$ in (12) simply requires the set $\tilde{\mathcal{X}}$ to include the origin as its interior point. In such case, there always exists a sufficiently small value of ε_f so that $\mathcal{E}_f$ is inscribed in $\tilde{\mathcal{X}}$.

III. TUBE MPC FORMULATION FOR A QUADCOPTER UNDER DISTURBANCES

In this part, we establish and implement the robust MPC strategy for the quadcopter's tracking error dynamics by stabilizing the tracking error dynamics with robust MPC.

A. Trajectory tracking error dynamics

First, consider the disturbed quadcopter model (1):
$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_d\boldsymbol{x}_k + \boldsymbol{B}_d\boldsymbol{h}_\psi(\boldsymbol{u}_k) + \boldsymbol{w}, \tag{18}$$
where $\boldsymbol{w} \in \mathbb{R}^6$ denotes the system's disturbance which is assumed to be bounded, i.e.: $\boldsymbol{w} \in \mathcal{W}$. Then, by using the mapping (4), the system (18) becomes:
$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_d\boldsymbol{x}_k + \boldsymbol{B}_d\boldsymbol{v}_k + \boldsymbol{w}. \tag{19}$$
Next, we assume to have a well-defined reference trajectory of the quadcopter with all constraints satisfied, namely:
$$\begin{cases} \boldsymbol{x}_{k+1}^{\text{ref}} = \boldsymbol{A}_d\boldsymbol{x}_k^{\text{ref}} + \boldsymbol{B}_d\boldsymbol{v}_k^{\text{ref}}, & \text{(20a)} \\ \boldsymbol{v}_k^{\text{ref}} \in \mathcal{V}_c^{\text{ref}} \subset \mathcal{V}_c^{\text{in}} \text{ as in (7)}, & \text{(20b)} \end{cases}$$
where $\mathcal{V}_c^{\text{ref}}$ represents the polytope containing the input reference sequence $\boldsymbol{v}_k^{\text{ref}}$. This set $\mathcal{V}_c^{\text{ref}}$ is assumed to be included by the constraint set $\mathcal{V}_c^{\text{in}}$ of $\boldsymbol{v}_k$. Note that the dynamics expressed in (20a) simply represent the trajectory of three independent double integrators. Hence, effective time parameterization tools can be used to generate the reference signal satisfying the constraints (20b) [11]. Subsequently, from (20) and (19), we achieve the tracking error dynamics:
$$\begin{cases} \tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{A}_d\tilde{\boldsymbol{x}}_k + \boldsymbol{B}_d\tilde{\boldsymbol{v}}_k + \boldsymbol{w} & \text{(21a)} \\ \text{s.t } \tilde{\boldsymbol{v}}_k \in \tilde{\mathcal{V}}, & \text{(21b)} \end{cases}$$
where $\tilde{\boldsymbol{x}}_k \triangleq \boldsymbol{x}_k - \boldsymbol{x}_k^{\text{ref}}$ denotes the deviation of the state $\boldsymbol{x}_k$ with respect to the reference $\boldsymbol{x}_k^{\text{ref}}$; likewise, $\tilde{\boldsymbol{v}}_k \triangleq \boldsymbol{v}_k - \boldsymbol{v}_k^{\text{ref}}$ denotes the control action employed to compensate such deviation and disturbances while $\tilde{\mathcal{V}} \triangleq \mathcal{V}_c^{\text{in}} \ominus \mathcal{V}_c^{\text{ref}}$ is the corresponding constraint set for the input $\tilde{\boldsymbol{v}}_k$ of system (21a).

Prior to this point, the trajectory tracking problem has been reduced to the stabilization of the linear disturbed system (21a) towards the origin $\tilde{\boldsymbol{x}}_e = \boldsymbol{0}_{6\times 1}, \tilde{\boldsymbol{v}}_e = \boldsymbol{0}_{3\times 1}$. Therefore, in the next part, we synthesize the robust MPC [16] subject to the bounded disturbance $\boldsymbol{w}$.

B. Robust model predictive control

As introduced above, let us formulate the robust MPC presented in [16] for the constrained and disturbed system (21). The online optimization at step k is defined as follows:
$$V_{N_p}(\tilde{\boldsymbol{x}}_k) \triangleq \min \textstyle\sum_{i=0}^{N_p-1} V_s(\tilde{\boldsymbol{x}}_{k+i}^*, \tilde{\boldsymbol{v}}_{k+i}^*) + V_f(\tilde{\boldsymbol{x}}_{k+N_p}^*) \tag{22}$$
$$\text{s.t} \begin{cases} \tilde{\boldsymbol{x}}_{i+k+1}^* = \boldsymbol{A}_d\tilde{\boldsymbol{x}}_{i+k}^* + \boldsymbol{B}_d\tilde{\boldsymbol{v}}_{i+k}^*; & \text{(23a)} \\ \tilde{\boldsymbol{x}}_k^* \in \tilde{\boldsymbol{x}}_k \oplus \mathcal{Z};\ \tilde{\boldsymbol{x}}_{N_p}^* \in \mathcal{E}_f; & \text{(23b)} \\ \tilde{\boldsymbol{v}}_{i+k}^* \in \mathcal{V}^\star \triangleq \tilde{\mathcal{V}} \ominus \boldsymbol{K}\mathcal{Z},\ i \in \{0, 1, ..., N_p - 1\}, & \text{(23c)} \end{cases}$$
where $\boldsymbol{A}_d, \boldsymbol{B}_d$ is defined as in the model's parameters (2); $V_s(\cdot), V_f(\cdot)$ are the quadratic cost given in (13) with given weighting $\boldsymbol{Q} \succ 0, \boldsymbol{R} \succeq 0$ of choice; N_p is the prediction horizon; $\tilde{\boldsymbol{x}}_{i+k}^*$ and $\tilde{\boldsymbol{v}}_{i+k}^*$ denote the predicted state and control action at time step $i+k$, respectively, for the nominal version of system (19) (i.e., without the effect of the disturbance $\boldsymbol{w}$). The remaining ingredients in the constraints (23) are defined as follows. Firstly, given a state feedback gain $\boldsymbol{K}$ that renders the closed-loop matrix $\boldsymbol{A}_{cl} = \boldsymbol{A}_d + \boldsymbol{B}_d\boldsymbol{K}$ a Schur matrix, then $\mathcal{Z}$ is a robust positively invariant (RPI) set [17], [18] for the autonomous system: $\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{A}_{cl}\tilde{\boldsymbol{x}}_k + \boldsymbol{w}$. In this setting, we employ the outer-approximation of the minimal RPI proposed in [17]. Then, the set $\mathcal{Z}$ plays a role as the "origin" for our stabilizing problem [16]. Next, the

terminal ingredients ($\boldsymbol{P}$ and the set $\mathcal{E}_f$) for the nominal MPC settings are found exactly with the Proposition 2 with the gain $\boldsymbol{K}$ chosen as aforementioned, the dynamics (23a) and the constraint set $\mathcal{V}^\star \triangleq \tilde{\mathcal{V}} \ominus \boldsymbol{K}\mathcal{Z}$. It is also assumed that the disturbance's bound $\mathcal{W}$ is sufficiently manageable and hence results in non-empty feasible sets for the constraints in (23).

In short, the offline synthesis can be summarized as:

1) Choose $\boldsymbol{Q} \succ 0$ and $\boldsymbol{R} \succeq 0$ to formulate the stage and terminal cost value function of the form (13);
2) Choose the gain $\boldsymbol{K}$ stabilizing the pair $(\boldsymbol{A}_d, \boldsymbol{B}_d)$;
3) Compute the mRPI set $\mathcal{Z}$ as proposed in [17];
4) Compute the terminal ingredients subject to the constraint $\mathcal{V}^\star \triangleq \tilde{\mathcal{V}} \ominus \boldsymbol{K}\mathcal{Z}$ as in Proposition 2.

Remark 4: To avoid computational issues caused by the system's large dimensionality, we do not employ the computation of the maximal positively invariant set [19] for the closed-loop $\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{A}_{cl}\tilde{\boldsymbol{x}}_k$ to play the role of the terminal region $\mathcal{E}_f$. More specifically, although proven finite, the number of equalities portraying such a set for this 6 dimensional problem may exceed the memory capacity provided for the computation software. Yet, if the polyhedral set can be computed, the terminal constraints $\mathcal{E}_f$ will become linear, as opposed to being quadratic as in (12), and reduce the complexity of the online problem (22)-(23).

Nevertheless, with these setups, the control input applied to the system (21a) is defined as:

$$\tilde{\boldsymbol{v}}_k = \tilde{\boldsymbol{v}}^{\text{mpc}}_{N_p}(\tilde{\boldsymbol{x}}_k) \triangleq \tilde{\boldsymbol{v}}^*_k + \boldsymbol{K}(\tilde{\boldsymbol{x}}_k - \tilde{\boldsymbol{x}}^*_k). \tag{24}$$

Consequently, with this scheme, as proven in [16], the set $\mathcal{Z}$ is robustly exponentially stable for the closed-loop:

$$\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{A}_d\tilde{\boldsymbol{x}}_k + \boldsymbol{B}_d\tilde{\boldsymbol{v}}^{\text{mpc}}_{N_p}(\tilde{\boldsymbol{x}}_k) + \boldsymbol{w}. \tag{25}$$

Namely, $\tilde{\boldsymbol{x}}_k$ is maintained inside the "tube" $\tilde{\boldsymbol{x}}^*_k \oplus \mathcal{Z}$ which converges to $\mathcal{Z}$ when $k \to \infty$. In other words, one can state:

$$\boldsymbol{x}_k - \boldsymbol{x}^{\text{ref}}_k = \tilde{\boldsymbol{x}}_k \in \tilde{\boldsymbol{x}}^*_k \oplus \mathcal{Z} \Leftrightarrow \boldsymbol{x}_k \in \boldsymbol{x}^{\text{ref}}_k \oplus \tilde{\boldsymbol{x}}^*_k \oplus \mathcal{Z} \tag{26}$$

with $\tilde{\boldsymbol{x}}^*_k$ exponentially converges to $\boldsymbol{0}_{6\times 1}$. With the sequence of the reference for $\boldsymbol{x}_k$ and $\boldsymbol{v}_k$ denoted as $\boldsymbol{x}^{\text{ref}}, \boldsymbol{v}^{\text{ref}}$, respectively, the control implementation is given by Algorithm 1.

Algorithm 1: Robust MPC for trajectory tracking

Input: Reference sequence $\boldsymbol{x}^{\text{ref}}, \boldsymbol{v}^{\text{ref}}$; the optimization setup as in (22)-(23).

for $(\boldsymbol{x}^{\text{ref}}_k, \boldsymbol{v}^{\text{ref}}_k)$ **in** $\boldsymbol{x}^{\text{ref}} \times \boldsymbol{v}^{\text{ref}}$ **do**
- Measure $\boldsymbol{x}_k$ and ψ at time step k;
- Compute $\tilde{\boldsymbol{x}}^*_k$ and $\tilde{\boldsymbol{v}}^*_k$ by solving (22)-(23);
- $\tilde{\boldsymbol{v}}^{\text{mpc}}_{N_p} \leftarrow \tilde{\boldsymbol{v}}^*_k + \boldsymbol{K}(\tilde{\boldsymbol{x}}_k - \tilde{\boldsymbol{x}}^*_k)$ as in (24);
- $\boldsymbol{u}_k \leftarrow \boldsymbol{\varphi}_\psi(\boldsymbol{v}^{\text{ref}}_k + \tilde{\boldsymbol{v}}^{\text{mpc}}_{N_p})$ as in (4);
- Apply $\boldsymbol{u}_k$ to the vehicle;

end

IV. EXPERIMENTAL VALIDATION

The experiments are done over the Crazyflie 2.1 quadcopter platform where the drone's inputs are the desired thrust, roll and pitch angles, gathered in the vector $\boldsymbol{u}_k$, together with the desired yaw angle, assigned to zero.

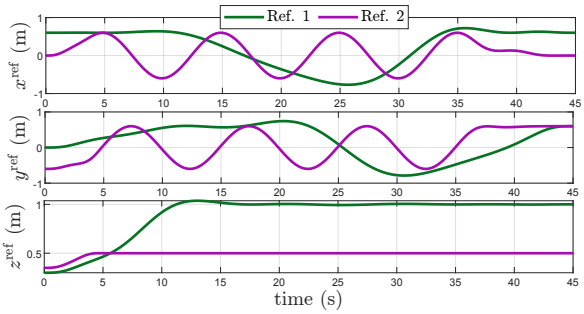

Fig. 2. Two references employed for the experimental validation.

Furthermore, the position and velocity of the quadcopter are acquired via a system of 8 *Qualisys* motion capture cameras. Note that the platform expects the normalized thrust $T\,(m/s^2)$ to be converted into the quadcopter's unit which varies from 0 to 65535. Then, these signals will be tracked via a built-in inner-loop containing two cascade PID and a transformation from the required torque to PWM signals designed for the drone's X-shape configuration [20].

A. Parameter setup

Firstly, the reference is required to be at least two time differentiable to guarantee the continuity and boundedness of $\boldsymbol{v}^{\text{ref}}$ as in (20b). Herein, two references are used as follows:

- *Ref. 1*: The position of the drone will be parameterized in time with the B-spline method [11]. More specifically, at timestamp $t_r\,(s)$, we imposed the position of the drone to pass through a waypoint $\boldsymbol{p}_r$ as:

$$\boldsymbol{p}_r^\top \in \left\{ \begin{matrix} [6,0,3];[6,3,5];[6,6,10]; \\ [0,6,10];[-6,6,10];[-6,-6,10]; \\ [6,-6,10];[6,0,10];[6,6,10] \end{matrix} \right\} \times 10\,(\text{cm}), \tag{27}$$

  with $t_r \in \mathbf{N}^9(0,45)$ (s). Then, by minimizing the curve's length and restraining their second derivatives (which is the mathematical interpretation of $\boldsymbol{v}^{\text{ref}}_k$) inside $\mathcal{V}^{\text{in}}_c$, we obtain the green solid curve depicted in Fig. 2.
- *Ref. 2*: The circular motion is parameterized as:

$$x^{\text{ref}}(t) = 0.6\cos\omega t, y^{\text{ref}}(t) = \sin 0.6\omega t, \\ z(t) = 0.5\ (\text{m}), \omega = 0.2\pi\ \text{rad/s}, \tag{28}$$

  and illustrated as the purple curve in Fig. 2. Note that at the beginning and the end of the reference, we introduce smooth curves to reduce the overshoot effect caused by introducing a sudden reference.

Consequently, $\mathcal{V}^{\text{ref}}_c$ as in (20b) can be obtained by bounding the $\boldsymbol{v}^{\text{ref}}$ with a box-type constraint (See Fig. 3).

The parameter of constraint set $\mathcal{U}$ as in (3) is provided in Table I while the corresponding constraint $\mathcal{V}_c$ is approximated by $\mathcal{V}^{\text{in}}_c$ as in (7) with $s_1 = 10, s_2 = 3$ which accounts for 91.9% of $\mathcal{V}_c$ as in (6). Finally, the disturbance set $\mathcal{W}$ as in (18) is considered to be bounded as:

$$\mathcal{W} = \{\boldsymbol{w} : |^l\boldsymbol{w}| \leq 0.02, l = 1, ..., 6\}, \tag{29}$$

which is estimated from our data of the difference between the system predicted and actual behavior after the same control action [6]. The set was intentionally chosen to be symmetric so that the heavy computation of the mRPI set $\mathcal{Z}$

in 6 dimensional space is reduced to that of the three double integrators as in (19) and their Cartesian product.

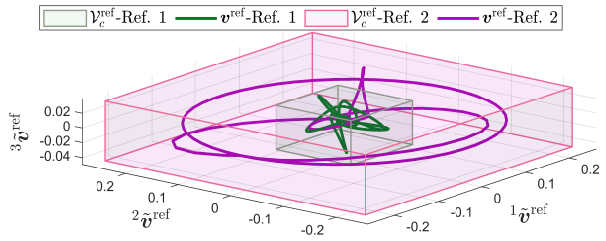

Fig. 3. Two input $\boldsymbol{v}^{\text{ref}}$ of the two references and their bounding boxes.

B. Controller tuning

First, $\boldsymbol{Q}, \boldsymbol{R}$ in (22) are chosen in the following form:

$$\begin{cases} \boldsymbol{Q} = \text{diag}(\boldsymbol{Q}^*, \boldsymbol{Q}^*, \boldsymbol{Q}^*); \boldsymbol{Q}^* \triangleq \text{diag}(\mu_p, \mu_v); \\ \boldsymbol{R} = \text{diag}(\mu_r, \mu_r, \mu_r), \end{cases} \tag{30}$$

where μ_p, μ_v, μ_r can be interpreted as the penalties for the deviation in position, velocity and input from the reference. The gain $\boldsymbol{K}$ in (23c) is found by placing the poles $p_i, i \in \{1, ..., 6\}$ in continuous time and computing the corresponding poles via the Z-transformation (i.e., $z_i = e^{p_i t_s}$) for the closed-loop $\boldsymbol{A}_{cl} = \boldsymbol{A}_d + \boldsymbol{B}_d \boldsymbol{K}$. We proceed to choose the poles in continuous time as:

$$p_1 = p_2 = p_3 = -4\mu_K \text{ and } p_4 = p_5 = p_6 = -5\mu_K, \tag{31}$$

with the intention of analyzing the effect of choosing poles with different rise times or convergence rates. Then we provide some criteria for choosing the gain and matrices appropriately. Illustrative examples for the tracking problem of Ref. 1 are given to analyze the closed-loop characteristics.

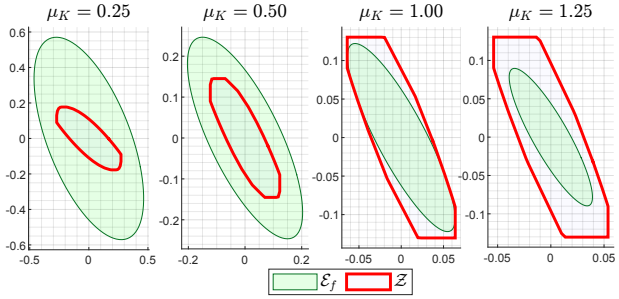

Fig. 4. Effect of changing the gain K towards fast closed-loop convergence, projected into the first two axes of $\tilde{\boldsymbol{x}}_k$ ($^1\tilde{\boldsymbol{x}}_k$-horizontal, $^2\tilde{\boldsymbol{x}}_k$-vertical).

On one hand, as mentioned in Remark 3 and Proposition 2, for the nominal system, the stabilizing gain $\boldsymbol{K}$ also needs to be chosen so that $\boldsymbol{0}_{6\times 1} \in \tilde{\mathcal{X}} \triangleq \{\tilde{\boldsymbol{x}} : \boldsymbol{K}\tilde{\boldsymbol{x}} \in \mathcal{V}^\star\}$. Meanwhile, under the presence of $\mathcal{W}$, $\boldsymbol{K}$ needs to stabilize the closed-loop robustly or fast enough so that the corresponding mRPI set $\mathcal{Z}$ in (23b) becomes satisfactory in terms of size. More specifically, as $\mathcal{Z}$ represents the bound of the tracking error, it reflects the controller's performance guarantee. With these observations, firstly, we would like to choose the gain $\boldsymbol{K}$ as "aggressive" as possible, or in our settings, μ_K as large as possible. Secondly, the choice of gain $\boldsymbol{K}$ is also tangled with the size of $\mathcal{E}_f$ which necessarily requires including $\mathcal{Z}$ (i.e., $\mathcal{Z} \subset \mathcal{E}_f$) so that the input constraints $\boldsymbol{K}\tilde{\boldsymbol{x}} \in \mathcal{V}^\star = \tilde{\mathcal{V}} \ominus \boldsymbol{K}\mathcal{Z}$ are also satisfied inside $\mathcal{Z}$, and so is the closed-loop performance. Therefore, as illustrated in Fig. 4, the two last choices of $\mu_K = 1.0$ and 1.25 violate the latter, while the first choice $\mu_K = 0.25$ results in a poor guarantee for the closed-loop (with the weighting fixed as $\mu_q = 50, \mu_v = 15$ and $\mu_r = 25$) as stated in the former principle.

TABLE I

CONTROLLER'S PARAMETERS AND SYSTEM'S SPECIFICATION

Parameters	Values
Sampling time t_s (s)	0.2
Prediction horizon N_p	10 steps
T_{max}, ϵ_{max} in (3)	$1.45g \approx 14.22\, m/s^2$; $0.1745\, rad$
$(\mu_K, \mu_p, \mu_v, \mu_r)$ in (30)-(31)	(0.5, 50, 15, 25)

On the other hand, after the suitable choice of $\boldsymbol{K}$ fixing the set $\mathcal{Z}$, it is also possible to modify the set $\mathcal{E}_f$ with $\boldsymbol{Q}$ and $\boldsymbol{R}$. The motivation for this adjustment comes from the fact that, for the stabilization of system (21), $\mathcal{E}_f \oplus \mathcal{Z}$ plays the role of the new terminal region for the system's tube trajectory [16]. Therefore, for a given initial tracking error point, the set can be modified to be "closer" or even to include the point, hence, resulting in a shorter prediction interval N_p. Depending on how the penalty weights are chosen, the set $\mathcal{E}_f$ can be characterized to some extent, while the condition $\mathcal{Z} \subset \mathcal{E}_f$ also needs to be ensured. For the experiments, the weighting matrices and gain are given in Table I.

C. Experimental results and discussion

Fig. 5 and Fig. 6 show the position tracking performance and how the tracking error of the quadcopter is restrained inside the tube $\mathcal{Z}$, respectively. Numerical details are also given in Table II where, the matrix $\boldsymbol{P}$ is represented as:

$$\boldsymbol{P} \triangleq \text{diag}(\boldsymbol{P}^*, \boldsymbol{P}^*, \boldsymbol{P}^*). \tag{32}$$

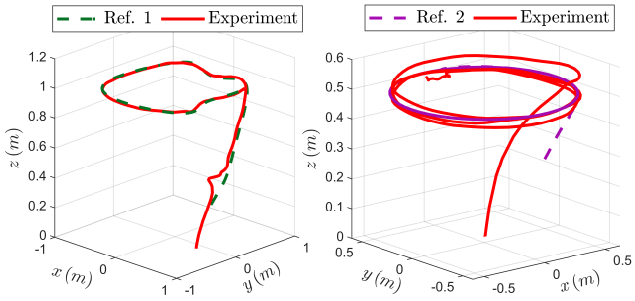

Fig. 5. Position tracking with the two references.

Overall, it can be seen that the controller is successfully implemented and verified within the flight tests with around 3-5 cm in root-mean-square (RMS) of tracking error[1] and under 40ms for the average computation time at each control loop (see Fig. 7). These results show that not only is the performance guaranteed, but also that the method is computationally efficient thanks to the linear model prediction obtained via the coordinate change.

It is also noticeable that, although later adequately concealed in the tube, there are significant overshoots of errors at the beginning of both flights. This problem comes from

[1]RMS= $\frac{1}{3}\sum_{q=1,3,5}\sqrt{N_s^{-1}\sum_{\ell=1}^{N_s}({}^q\boldsymbol{x}_k - {}^q\boldsymbol{x}^{\text{ref}})^2}$, N_s is the number of simulation steps.

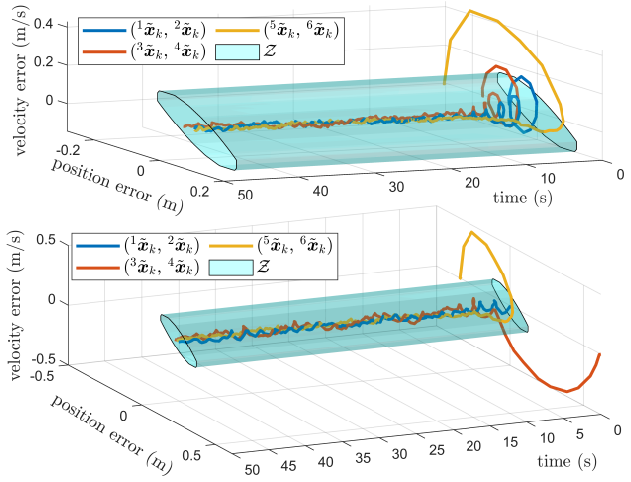

Fig. 6. Tracking error $\tilde{\boldsymbol{x}}_k$ for Ref. 1 (top) and Ref. 2 (bottom).

TABLE II

EXPERIMENTAL RESULTS AND CONTROL PARAMETERS

	Ref. 1	Ref. 2
RMS of tracking error	3.25 cm	5.46 cm
Average computation time	39.40 ms	38.11 ms
$\boldsymbol{P}^*$ as in (32)	$\begin{bmatrix}513.54, 328.59\\ 328.59, 356.89\end{bmatrix}$	$\begin{bmatrix}542.18, 329.80\\ 329.80, 368.92\end{bmatrix}$
ε_f as in (12)	8.7104	6.0724

the fact that the drone starts from a static state with zero angular velocity of the propellers, which later entails a sudden acceleration of the vehicle. However, all three inputs are not saturated and remain constrained in their bounds, as can be seen from Fig. 8.

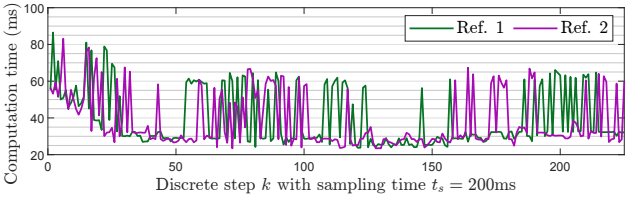

Fig. 7. Computation time for the proposed MPC scheme.

V. CONCLUSION

This paper presented a procedure to implement the robust trajectory tracking for a quadcopter based on tube MPC. The main advantage came from the problem reformulation through differential flatness, which transformed the nonlinear system into chains of integrators in closed-loop. Furthermore, an approximation of the feasible domain was given to convert the original tracking problem into a constrained stabilization problem under disturbances. Then, with a robust MPC setup for linear systems, the tracking performance was guaranteed along the trajectory without any approximation. The controller was successfully applied and validated via experimental tests. Further work will concentrate on multi-quadcopters formation stability and robustness guarantees.

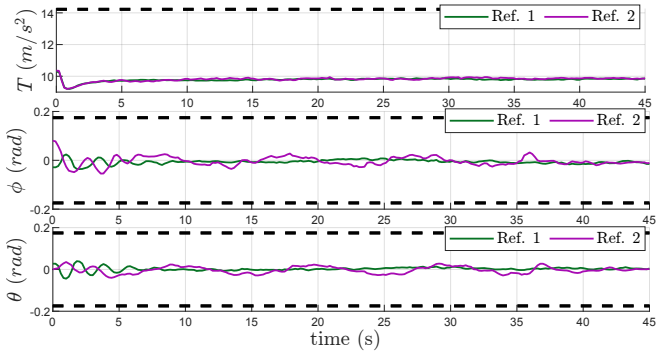

Fig. 8. Quadcopter's inputs for the tracking of Ref.1 and 2.

REFERENCES

[1] W. Xie, G. Yu, D. Cabecinhas, R. Cunha, and C. Silvestre, "Global saturated tracking control of a quadcopter with experimental validation," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 169–174, 2020.

[2] N. T. Nguyen and I. Prodan, "Stabilizing a multicopter using an nmpc design with a relaxed terminal region," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 126–132, 2021.

[3] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[4] D. Wang, Q. Pan, Y. Shi, J. Hu, and C. Zhao, "Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment," *IEEE Trans. on Cybernetics*, 2021.

[5] M. A. Santos, A. Ferramosca, and G. V. Raffo, "Tube-based MPC with nonlinear control for load transportation using a UAV," in *9th IFAC Symposium on Robust Control Design*, pp. 459–465, 2018.

[6] N. Michel, S. Bertrand, S. Olaru, G. Valmorbida, and D. Dumur, "Design and flight experiments of a tube-based model predictive controller for the ar. drone 2.0 quadrotor," *IFAC-PapersOnLine*, vol. 52, no. 22, pp. 112–117, 2019.

[7] A. Didier, A. Parsi, J. Coulson, and R. S. Smith, "Robust adaptive model predictive control of quadrotors," in *European Cont. Conf.*, IEEE, 2021.

[8] M. W. Mueller and R. D'Andrea, "A model predictive controller for quadrocopter state interception," in *2013 European Control Conference (ECC)*, pp. 1383–1389, IEEE, 2013.

[9] H.-T. Do and I. Prodan, "Indoor experimental validation of mpc-based trajectory tracking for a quadcopter via a flat mapping approach," in *2023 European Control Conference (ECC)*, 2023.

[10] J. Levine, *Analysis and control of nonlinear systems: A flatness-based approach*. Springer Science & Business Media, 2009.

[11] N. T. Nguyen, I. Prodan, and L. Lefèvre, "Flat trajectory design and tracking with saturation guarantees: a nano-drone application," *International Journal of Control*, pp. 1–14, 2018.

[12] S. Amini, B. Ahi, and M. Haeri, "Control of high order integrator chain systems subjected to disturbance and saturated control: A new adaptive scheme," *Automatica*, vol. 100, pp. 108–113, 2019.

[13] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[14] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[15] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer, 2008.

[16] D. Q. Mayne, M. M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[17] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne, "Invariant approximations of the minimal robust positively invariant set," *IEEE Trans. on Auto. Control*, vol. 50, no. 3, pp. 406–410, 2005.

[18] I. Kolmanovsky and E. G. Gilbert, "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Mathematical problems in engineering*, vol. 4, no. 4, pp. 317–367, 1998.

[19] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: The theory and application of maximal output admissible sets," *IEEE Trans. on Auto. Cont.*, vol. 36, no. 9, pp. 1008–1020, 1991.

[20] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski, "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering," in *2017 22nd International Conference on Methods and Models in Automation and Robotics*, pp. 37–42, IEEE, 2017.