Explicit flatness-based MPC design for indoor nano-quadcopter position tracking

Huu-Thinh Do, Ionela Prodan

ORIGINAL ARTICLES

Explicit flatness-based MPC design for indoor nano-quadcopter position tracking

Huu-Thinh Do and Ionela Prodan

Univ. Grenoble Alpes, Grenoble INP†, LCIS, F-26000, Valence, France.
† Institute of Engineering and Management University Grenoble Alpes.

ARTICLE HISTORY
Compiled June 24, 2024

ABSTRACT

Due to the nonlinearities and operational constraints of quadcopters, Model Predictive Control (MPC) encounters the requirement of high computational power. This problem may prove impractical, especially for hardware-limited setups. By removing the need for online solvers, Explicit MPC (ExMPC) stands out as a strong candidate. Yet, the formulation was usually hindered by the system's nonlinearity and dimensionality. In this paper, we propose an ExMPC solution for quadcopter position stabilization. With the former issue, the system is exactly linearized into a concatenation of three double integrators. For the latter, with a suitable characterization of the new convoluted constraints, the stabilizing ExMPC can be computed for each double integrator separately. The controller is validated via simulations and experiments on a nanodrone platform. The proposed scheme provides similar performance and theoretical guarantees to the up-to-date nonlinear MPC solutions but with notably less computational effort, allowing scalability in a centralized manner.

KEYWORDS
Explicit MPC, feedback linearization, differential flatness, quadcopter position control.

1. Introduction

With an increasingly wide application in autonomous tasks like delivery or surveillance, quadcopters have proven essential thanks to their mobility and hoverability. Although being investigated for decades, optimally governing these thrust-propelled vehicles remains an interesting topic in the research community due to their trigonometric-induced nonlinearity and challenges to rigorously provide both constraint satisfaction and stability guarantees. Typically, to deal with the model complexity, the systems are decoupled into two layers: attitude and position control (Bertrand, Guénard, Hamel, Piet-Lahanier, & Eck, 2011; Cao & Lynch, 2015; Chen & Wang, 2013). The navigation, hence, is handled in a hierarchical scheme (a.k.a., inner–outer loop control).

Several studies have been conducted to deal with both problems by classical and modern approaches (e.g., PI/PID (Chen & Wang, 2016; Lei, Li, & Chen, 2018), back-

*Email: {huu-thinh.do, ionela.prodan}@lcis.grenoble-inp.fr,
This work is funded by La Région Auvergne-Rhône-Alpes, Pack Ambition Recherche 2021 - PlanMAV, RECPLAMALCIR, Ambition Internationale 2023, Horizon-TA C7H-REG24A10 and PERSYVAL-UGA C7H-LXP21A34.

stepping (Tran, Sam, He, Luu-Trung-Duong, & Truong, 2018) or data-driven control (Torrente, Kaufmann, Föhn, & Scaramuzza, 2021)). Distinguishable from the others, Model Predictive Control (MPC) provides a framework to ensure these requirements and achieves optimality thanks to its receding horizon mechanism. Indeed, the vehicle's navigation has been structurally integrated into the MPC design from various standpoints in the literature (see Table 1). For example, with the standard axioms of the terminal ingredients for Nonlinear MPC (NMPC) (Mayne, Rawlings, Rao, & Scokaert, 2000), the works in (Nguyen, Prodan, & Lefèvre, 2020b, 2020c) provide theoretical guarantees for the stabilization of both layers. Moreover, from a geometrical viewpoint, an NMPC design was proposed in (Pereira, Leite, & Raffo, 2021) to handle obstacle avoidance and aggressive trajectory tracking. A more computationally efficient NMPC for the position control is proposed in (Gomaa, De Silva, Mann, & Gosine, 2022) where feasibility and stability are ensured without terminal ingredients. Still, it is well known that the strategy requires high computational power and long sampling time when the nonlinear model is employed for prediction.

Meanwhile, facing the issue of nonlinearity, it is customary to seek for the notions of linearization to simplify the control design. Indeed, as an intrinsic characteristic of the vehicle, the properties of differentially flat systems have been exploited as a bridge between nonlinear system analysis and linear control theory. The class contains the systems of which all the states and inputs can be algebraically expressed in terms of a special output, called the flat output, and a finite number of its derivatives (Fliess, Lévine, Martin, & Rouchon, 1993). The expression (i.e., the flat representation) provides different paths to remedy the system nonlinearity, especially in optimization-based solutions. On one hand, with the flat representation, a feasible trajectory can be generated via curve parameterization tools (Do, Prodan, & Stoican, 2021; Prodan, Stoican, & Louembet, 2019). Then, along such an integral curve, a tangent approximation can be obtained, which is then efficiently handled by scheduling the MPC (Prodan et al., 2013). While the tracking is simplified to a Quadratic Program (QP), one major theoretical drawback for this approach resides on the quantification of the approximation error and characterization of the stability for the original nonlinear system. On the other hand, with the input-output relationship provided by the flat representation, the model can be transformed to a linear equivalent system (Fliess, Lévine, Martin, & Rouchon, 1995) in a new coordinate system, called the flat output space. The navigation now can be handled by closing the loop for such a linear system. With this approach and the NMPC strategy, an insightful comparative study was provided in (Sun, Romero, Foehn, Kaufmann, & Scaramuzza, 2022) for two different flight scenarios. The result first shows that, for a feasible reference, the flatness-based controller's tracking performance is comparable with the NMPC but with significantly smaller computational footprint. However, when being pushed to saturation with physically infeasible flights, the NMPC provides more reliable performance with its capacity of constraint-handling. Indeed, when it comes to physical limitation in feedback linearization control, a critical difficulty rises from the complicated input constraints during the inversion between the original and the new coordinates. Generally, for flat systems, this direction still remains open for a constructive strategy guaranteeing stability under constrained actuation. Pursuing the idea of handling constraints with the linearized system, particular results on sketching the new feasible domain for quadcopters' thrust, attitude and angular rate can be found in (Nguyen, Prodan, & Lefèvre, 2018), which maps the original limits to nonlinear sets of the flat output and up to its third order derivatives. In (Do & Prodan, 2023; Mueller & D'Andrea, 2013), linear approximation of the input constraints was employed with MPC for the outer loop control problem, reducing the complexity of

the position tracking and embedding the system's stability analysis in a linear setting. With similar motivation, it was shown in (Nguyen, Prodan, & Lefèvre, 2020a; Xie & Veitch, 2020) as well that, for the outer-loop, the linearized model can achieve global asymptotic stability based on the nested saturation control setup (Teel, 1992).

Table 1. MPC-based control for quadcopter systems

	Reference
Position control	NMPC with feedback linearization local control, computationally complex (Nguyen, Prodan, & Lefevre, 2019).
	Computationally simplified NMPC setup with stability, feasibility guaranteed, strict tuning (Gomaa et al., 2022).
	QP-MPC in the flat output space, with constraint approximation (Do & Prodan, 2023; Mueller & D'Andrea, 2013).
	Explicit solution of Bézier curves, no stability analysis and constraint characterization (Liu, Lu, & Chen, 2015).
Attitude control	*Explicit MPC*; assumption of small roll, pitch angles (Jiajin, Rui, Yingjing, & Jianxiao, 2017).
	Stability and feasibility guaranteed, conservative constraint approximation with computed-torque local controller (Nguyen et al., 2020b).
	Low-cost NMPC, suboptimal formulation, hovering point approximation for system prediction (Zanelli, Horn, Frison, & Diehl, 2018).
Full dynamics	assumption of small roll, pitch angles; soft constraints (Wang, Pan, Shi, Hu, & Zhao, 2021).
	Learning NMPC with disturbance rejection, experimentally validated (Torrente et al., 2021).
	Finite horizon LQR for the approximated dynamics, constraints neglected (Cohen, Abdulrahim, & Forbes, 2020).

With the goal of following the latter and advancing toward a low-cost solution while providing both input and state constraints as well as stability guarantees, the Explicit MPC (Bemporad, Morari, Dua, & Pistikopoulos, 2002) (ExMPC) constitutes a promising candidate. The underlying idea is to convert the nonlinearity of the quadcopter's position control into a standard QP problem via the flatness-based exact linearization, with a stability guarantee via standard MPC synthesis (Mayne et al., 2000) and finally solve such problem within the multi-parametric quadratic programming framework. The online implementation includes locating the current state inside the look-up table computed offline as a set of critical regions, and evaluating the associated optimal control. Although there exists related works for this direction in the literature (Jiajin et al., 2017; Liu et al., 2015), the discussed methods rely on a near hovering point assumption or insufficiently connect the stability arguments. Furthermore, to the best of the authors' knowledge, although the simulation findings in these works affirms the theory, the effectiveness of the strategy has not been experimentally validated for the model under study. Therefore, exploiting the exactly linearized model in closed-loop of the quadcopter's translational dynamics, we:

- propose an original synthesis procedure for implementing the ExMPC to the outer loop of the quadcopter model via its exact linearization from flatness.
- test experimentally the proposed scheme over multiple nanodrones (the experiment video is available at `https://youtu.be/u7PsNDheIR4`) and provide the necessary steps and code for implementation.

While the flat representation was already discussed in the literature (Do & Prodan,

2023; Mueller & D'Andrea, 2013), we focus on the implementation of the ExMPC and its ancillary technical prospects. More specifically, in Section 2, the quadcopter position control problem in both the state/input state and the flat output space will be recalled. Therein, the linear approximation of the new constraint set is introduced. Section 3 presents the ExMPC formulation in the flat output space. Simulations are also provided to highlight the computational advantage of our study compared with the standard implicit MPC approach. Section 4 verified the control scheme via experimental tests with the Crazyflie platform. Finally, Section 5 concludes and provides future directions.

Notation: Matrices with appropriate dimension are denoted via bold upper-case letters. Bold lower-case letters represent vectors. $\|\boldsymbol{x}\|_{\boldsymbol{P}} \triangleq \sqrt{\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x}}$ denotes the weighted norm. The letter k represents the signal's value at the discrete step kt_s with the sampling time t_s. diag$(\cdot)$ returns a diagonal matrix formed by its arguments. $\ominus$ denotes the Pontryagin difference. conv$\{\cdot\}$ denotes the convex hull.

2. System description in the flat output space

2.1. The constrained outer loop

At time step k, the discretized Euler-Newton model for the outer loop control design is given as (Cao & Lynch, 2021; Nguyen et al., 2020c):

$$\boldsymbol{\xi}_{i,k+1} = \boldsymbol{A}\boldsymbol{\xi}_{i,k} + \boldsymbol{B}h_i(\boldsymbol{u}_k), \tag{1}$$

where $\boldsymbol{p}_k = [p_{1,k}\ p_{2,k}\ p_{3,k}]^\top \triangleq [x_k\, y_k\, z_k]^\top$ and x, y, z describe the position of the quadcopter in the inertial frame (m); while, for $i \in \{1, 2, 3\}$, $\boldsymbol{\xi}_{i,k} \triangleq [p_{i,k}\ \dot{p}_{i,k}]^\top \in \mathbb{R}^2$ denote the states vector collecting the position components and their derivative in each axis; $\boldsymbol{u}_k \triangleq [T_k\ \phi_k\ \theta_k]^\top \in \mathbb{R}^3$ collects inputs of the system including the normalized thrust (m/s^2), the roll and pitch angles (rad). $\boldsymbol{A} \triangleq \begin{bmatrix} 1 & t_s \\ 0 & 1 \end{bmatrix}, \boldsymbol{B} \triangleq \begin{bmatrix} 0.5t_s^2 \\ t_s \end{bmatrix}$. The remaining functions are defined as:

$$\begin{cases} h_1(\boldsymbol{u}_k) = T_k(\cos\phi_k \sin\theta_k \cos\psi + \sin\phi_k \sin\psi), \\ h_2(\boldsymbol{u}_k) = T_k(\cos\phi_k \sin\theta_k \sin\psi - \sin\phi_k \cos\psi), \\ h_3(\boldsymbol{u}_k) = -g + T_k \cos\phi_k \cos\theta_k, \end{cases} \tag{2}$$

where $g \approx 9.81 m/s^2$ is the gravitational acceleration, t_s is the sampling time and ψ denotes the yaw angle. Moreover, $\boldsymbol{u}_k$ is constrained inside a set $\mathcal{U}$ defined as:

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

with $T_{max} > 0, \epsilon_{max} \in (0; \pi/2)$ denoting the upper bounds of the thrust (T_k), the roll and the pitch angles (ϕ_k, θ_k).

In the literature, system (1) is known to be differentially flat, i.e., there exists a coordinate change and an endogenous dynamic feedback law that linearizes the model in closed-loop (Levine, 2009). Indeed, consider the variable transformation (a.k.a., the

inverse kinematics):

$$\begin{aligned} T_k &= \sqrt{\tilde{u}_{1,k}^2 + \tilde{u}_{2,k}^2 + (\tilde{u}_{3,k} + g)^2}, \\ \phi_k &= \arcsin\left((\tilde{u}_{1,k}\sin\psi - \tilde{u}_{2,k}\cos\psi)/T_k\right), \\ \theta_k &= \arctan\left((\tilde{u}_{1,k}\cos\psi + \tilde{u}_{2,k}\sin\psi)/(\tilde{u}_{3,k} + g)\right). \end{aligned} \tag{4}$$

Then, with $\tilde{u}_{3,k} \geq -g$, system (1) yields:

$$\boldsymbol{\xi}_{i,k+1} = \boldsymbol{A}\boldsymbol{\xi}_{i,k} + \boldsymbol{B}\tilde{u}_{i,k}, i \in \{1, 2, 3\}, \tag{5}$$

where $\tilde{\boldsymbol{u}}_k = [\tilde{u}_{1,k}\ \tilde{u}_{2,k}\ \tilde{u}_{3,k}]^\top \in \mathbb{R}^3$ is the new input vector of the system. Note that the dynamics (5) in the new coordinates (called the flat output space) is now simply composed of three double integrators. However, the constraints described as $\mathcal{U}$ given in (3) now are complicated. Indeed, the admissible set of $\tilde{\boldsymbol{u}}_k$, such that $\boldsymbol{u}_k \in \mathcal{U}$, is ψ-dependent, hence practically time-varying, and non-convex (Do & Prodan, 2023). For these impractical drawbacks, we introduce a convex subset of the feasible domain as follows. First, from (4), one can state:

$$\sin|\phi_k| = |(\tilde{u}_{1,k}\sin\psi - \tilde{u}_{2,k}\cos\psi)/T_k| \leq \sqrt{(\tilde{u}_{1,k}^2 + \tilde{u}_{2,k}^2)/(\tilde{u}_{1,k}^2 + \tilde{u}_{2,k}^2 + (\tilde{u}_{3,k} + g)^2)}, \tag{6}$$

$$\tan|\theta_k| = |(\tilde{u}_{1,k}\cos\psi + \tilde{u}_{2,k}\sin\psi)/(\tilde{u}_{3,k} + g)| \leq \sqrt{(\tilde{u}_{1,k}^2 + \tilde{u}_{2,k}^2)/(\tilde{u}_{3,k} + g)^2}. \tag{7}$$

Then, with $|\phi_k|, |\theta_k| \leq \epsilon_{max} < \pi/2$, by bounding the right-hand side of (6) and (7) with $\sin\epsilon_{max}$ and $\tan\epsilon_{max}$, respectively, we arrive to the condition:

$$\tilde{u}_{1,k}^2 + \tilde{u}_{2,k}^2 \leq (\tilde{u}_{3,k} + g)^2 \tan^2\epsilon_{max}. \tag{8}$$

Thus, collecting (8), constraint $0 \leq T_k \leq T_{max}$ and the linearizing condition in (5), we obtain a new constraint set for the virtual input $\tilde{\boldsymbol{u}}$:

$$\begin{aligned} \mathcal{V}_c = \big\{\tilde{\boldsymbol{u}}_k \in \mathbb{R}^3 : \tilde{u}_{1,k}^2 + \tilde{u}_{2,k}^2 + (\tilde{u}_{3,k} + g)^2 \leq T_{max}^2, \\ \tilde{u}_{1,k}^2 + \tilde{u}_{2,k}^2 \leq (\tilde{u}_{3,k} + g)^2 \tan^2\epsilon_{max}, \tilde{u}_{3,k} \geq -g\big\}. \end{aligned} \tag{9}$$

Consequently, the problem has now been converted to the governing of a system of three double integrators in (5) with their inputs $\tilde{u}_{i,k}$ intricately restrained in $\mathcal{V}_c$ in (9).

Remark 1. *Noteworthy, in model* (1)*, the inclusion* $\tilde{\boldsymbol{u}}_k \in \mathcal{V}_c$ *strictly implies* $\boldsymbol{u}_k \in \mathcal{U}$ *regardless the value of the feedback signal* ψ *thanks to the upper bounds given in* (6) *and* (7)*. Moreover, the linearizing mapping* (4) *is, indeed, a homeomorphism which is known to map the interior and boundary of a set to those of its image, respectively. For these reasons, as in the following part, inner approximation of the nonlinear set* (9) *can be exploited without risking constraint violation.*

2.2. Linear inner approximation for the input constraints

It is noticeable that, while represented by quadratic inequalities, the set $\mathcal{V}_c$ in (9) can be also regarded as a set bounded by two surfaces: a disk of radius T_{max} centered at

$[0, 0, -g]^\top$, and a convex cone characterized by g and ϵ_{max}. Hence, by parameterizing the intersection ring between the two surfaces, the boundary point of the disk and the vertex $[0, 0, -g]^\top$, the set $\mathcal{V}_c$ can be approximated as:

$$\tilde{\mathcal{V}}_c = \text{conv} \left\{ \begin{matrix} [0, 0, -g]^\top, [R^\star \cos\alpha, R^\star \sin\alpha, \tilde{u}_3^\star]^\top \\ [r\cos\alpha, r\sin\alpha, \sqrt{T_{max}^2 - r^2} - g]^\top \end{matrix} \right\}, \tag{10}$$

with $\alpha \in \mathcal{N}(\ell_1, 0, 2\pi), r \in \mathcal{N}(\ell_2, 0, R^\star)$ for some large integers ℓ_1, ℓ_2, $R^\star \triangleq T_{max} \sin\epsilon_{max}$, $\tilde{u}_3^\star \triangleq T_{max}\cos\epsilon_{max} - g$ and $\mathcal{N}(n, a, b)$ denotes a finite collection of n evenly sampled real numbers over the interval $[a, b] \subset \mathbb{R}$.

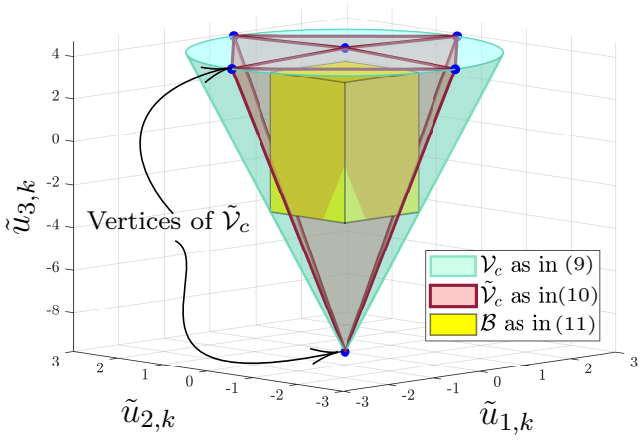

Figure 1. The inner approximation $\tilde{\mathcal{V}}_c$ (red) of $\mathcal{V}_c$ (cyan) with $\ell_1 = 5, \ell_2 = 2$, $\epsilon_{max} = 0.1745$ (rad), $g = T_{max}/1.45 = 9.81$ (m/s^2).

In this fashion, the constraint set $\mathcal{V}_c$ now can be approximated with the polytope $\tilde{\mathcal{V}}_c$ in (10) with some sufficiently large ℓ_1, ℓ_2.

In (Do & Prodan, 2023), it has been shown that the migration to the flat output space with the approximated input constraints provides a computationally attractive MPC formulation compared to NMPC solutions in the literature for the same model (Nguyen et al., 2019, 2020c), since the online optimization problem is now a QP for the linear dynamics and linear constraints. Accordingly, the stability and feasibility guarantees become more accessible within the linear MPC design (Mayne et al., 2000). With those advantages, subsequently, we extend our study to the implementation of the ExMPC, with a view to opening possible integration in a low-cost embedded architecture.

3. ExMPC for a quadcopter position control

Previously, the problem of nonlinearity has been tackled thanks to the mapping (4) and the approximation (10). With the complexity of a QP, the framework of linear ExMPC now can be applied to compute explicitly the optimal control for the six-dimensional (6D) system (5) with the constraint $\tilde{\boldsymbol{u}}_k \in \tilde{\mathcal{V}}_c$ as in (10). However, as notoriously known for the implementation of ExMPC, the dimensionality burden is significant for systems of high dimension (Lee, 2011). In other words, the explicit solution is exponentially

costly in the offline construction time, data storage capacity required and the online look-up time, with respect to the dimension and prediction horizon size.

To overcome this challenge, the representation of three independent double integrators will be exploited. More specifically, the input constraints for $\tilde{\boldsymbol{u}}_k$ will be approximated by a box-type constraint, giving rise to the explicit MPC formulation for each double integrator in (5). Comparison and simulation with the solution of the classical 6D model will be provided. A summary of the control scheme is given in Figure 2.

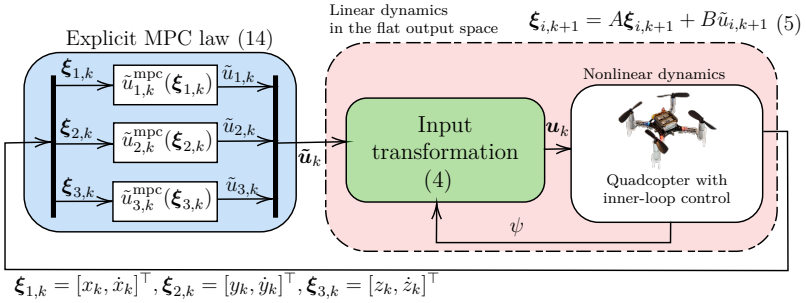

Figure 2. Explicit MPC scheme for the quadcopter position regulation.

3.1. ExMPC for the quadcopter position stabilization

Consider a box in $\mathbb{R}^3$ described as:

$$\mathcal{B} = \{\tilde{\boldsymbol{u}}_k : |\tilde{u}_{i,k}| \leq \bar{\tilde{u}}_i, i \in \{1,2,3\}\} \subset \mathcal{V}_c \text{ as in (9).} \tag{11}$$

Then, without loss of generality, we decouple the position tracking problem derived from (5) into the three following MPC regulation problems:

$$\min \sum_{j=0}^{N_p-1} V_s(\boldsymbol{\xi}_{i,k+j}, \tilde{u}_{i,k+j}) + V_f(\boldsymbol{\xi}_{i,k+N_p}) \tag{12a}$$

$$\text{s.t:} \begin{cases} \boldsymbol{\xi}_{i,k+j+1} = \boldsymbol{A}\boldsymbol{\xi}_{i,k+j} + \boldsymbol{B}\tilde{u}_{i,k+j}, \\ \boldsymbol{\xi}_{i,k+j} \in \mathcal{X}_i,\ j \in \{0, ..., N_p - 1\}, \\ |\tilde{u}_{i,k+j}| \leq \bar{\tilde{u}}_i, \boldsymbol{\xi}_{i,k+N_p} \in \mathcal{X}_{f,i},\ i \in \{1,2,3\}, \end{cases} \tag{12b}$$

where, for each $i \in \{1,2,3\}$, $\boldsymbol{\xi}_{i,k} \in \mathbb{R}^2$ is the state collecting the position and the velocity of the drone projected to the three axis $\{x, y, z\}$ established as in (1). $\tilde{u}_{i,k} \in \mathbb{R}$ is i-th component of the system's input in the flat output space described in (5); $\mathcal{X}_i \triangleq \{\boldsymbol{\xi}_i \in \mathbb{R}^2 : |\boldsymbol{\xi}_i| \leq [\bar{\mathrm{p}}_i, \bar{\dot{\mathrm{p}}}_i]^\top\}$ is the rectangular state constraint defining the workspace of the vehicle; $\mathcal{X}_{f,i} \subseteq \mathcal{X}_i$ denotes the terminal constraint set which is associated with the terminal cost $V_f(\cdot)$ to ensure both stability and feasibility (Mayne et al., 2000). In details, the state cost $V_s(\cdot)$ and terminal cost $V_f(\cdot)$ are chosen as:

$$V_s(\boldsymbol{\xi}, \tilde{u}) = \|\boldsymbol{\xi}\|^2_{\boldsymbol{Q}} + \|\tilde{u}\|^2_{\boldsymbol{R}};\ \ V_f(\boldsymbol{\xi}) = \|\boldsymbol{\xi}\|^2_{\boldsymbol{P}}, \tag{13}$$

with $\boldsymbol{R} \succ 0, \boldsymbol{Q} \succeq 0$ and $\boldsymbol{P} \succeq 0$ are the user-designed weighting matrix defining the optimization problem (12a). For the sake of simplicity, in (12a), we employ the same choice of weighting $\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R}$ for all $i \in \{1,2,3\}$.

Denote $\tilde{\boldsymbol{u}}_i^*(\boldsymbol{\xi}_k) \triangleq [\tilde{u}_{i,k}^*, ..., \tilde{u}_{i,k+N_p}^*]^\top$ the optimizer of (12). Then, the control applied to system (5) is given as:

$$\tilde{u}_{i,k} = \tilde{u}_{i,k}^{\text{mpc}}(\boldsymbol{\xi}_{i,k}) \triangleq \tilde{u}_{i,k}^*. \tag{14}$$

Moreover, with the system's linear representation (5), the solution for the QP (12) can be rewritten in its dual form via the Karush-Kuhn-Tucker (KKT) conditions. Then, depending on the state $\boldsymbol{\xi}_{i,k}$, a subset of constraints (12b) becomes active, on which is formulated a critical region. Over this activated domain, the constrained optimum for (12) can be explicitly parameterized by $\boldsymbol{\xi}_{i,k}$ as (Bemporad et al., 2002):

$$\underset{l\in\{1,...,\bar{l}_i\}}{\tilde{\boldsymbol{u}}_i^*(\boldsymbol{\xi}_{i,k})} = \boldsymbol{F}_{i,l}\boldsymbol{\xi}_{i,k} + \boldsymbol{\mu}_{i,l}, \text{ if } \boldsymbol{\xi}_{i,k} \in \mathcal{R}_{i,l}, \tag{15}$$

where $\mathcal{R}_{i,l} = \{\boldsymbol{\xi} : \boldsymbol{A}_{i,l}\boldsymbol{\xi} \leq \boldsymbol{b}_{i,l}\} \subset \mathbb{R}^2$ denotes the l-th polyhedral critical region, and the constant parameters $\boldsymbol{F}_{i,l}, \boldsymbol{\mu}_{i,l}$ are the associated optimal parameters in such a region. $\bar{l}_i$ denotes the total number of critical regions for each double integrator in (5).

Remark 2. *It is noteworthy that since $\mathcal{V}_c$ has the origin in its interior, there is always some subset $\mathcal{B}$ in the form of* (11) *with $\bar{\tilde{u}}_i$ being sufficiently small. Moreover, as one candidate, the maximum volume box $\mathcal{B}$ inscribed in $\mathcal{V}_c$, can be found by inflating a zonotope via an optimization problem presented in (Do & Prodan, 2023), since, $\mathcal{B}$ is, indeed, also a zonotope with the center at the origin (see Figure 1). Analytical solution for the largest $\mathcal{B}$ inscribed in $\mathcal{V}_c$ are formulated in the Appendix A.*

3.2. Simulation study

Previously, with the subset $\mathcal{B}$ as in (11), we sidestep the problem of dimensionality brought about by the 6D description of the model. To demonstrate such a computational trade-off and to better select the parameters for the proposed scheme, let us proceed by carrying out a simulation study for both approaches as the following scenarios:

- *Scenario 1* (Sce. 1): First, we employ the proposed MPC law generated from (12). The weighting $\boldsymbol{P}$ is computed from the algebraic Riccati equation. The terminal constraint set $\mathcal{X}_{f,i}$ is constructed with the polytopic maximal positive invariant (MPI) set with a local controller $\tilde{u}_{i,k}^{\text{loc}}(\boldsymbol{\xi}_{i,k}) \triangleq \boldsymbol{K}\boldsymbol{\xi}_{i,k}$ chosen from the LQR for $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{Q}, \boldsymbol{R})$, ensuring the stability and feasibility of system (5) under the control (14) (Mayne et al., 2000).
- *Scenario 2* (Sce. 2): Herein, we concatenate the dynamics (5) and formulate the following MPC:

$$\underset{\tilde{\boldsymbol{u}}_k,...,\tilde{\boldsymbol{u}}_{k+N_p-1}}{\arg\min} \sum_{j=0}^{N_p-1} \|\boldsymbol{\zeta}_{k+j}\|_{\underline{\boldsymbol{Q}}}^2 + \|\tilde{\boldsymbol{u}}_{k+j}\|_{\underline{\boldsymbol{R}}}^2 + \|\tilde{\boldsymbol{u}}_{k+N_p}\|_{\underline{\boldsymbol{P}}}^2, \tag{16}$$

$$\text{s.t} \begin{cases} \boldsymbol{\zeta}_{k+j+1} = \underline{\boldsymbol{A}}\boldsymbol{\zeta}_{k+j} + \underline{\boldsymbol{B}}\tilde{\boldsymbol{u}}_{k+j}, \\ \tilde{\boldsymbol{u}}_{k+j} \in \tilde{\mathcal{V}}_c \text{ as in } (10), \boldsymbol{\zeta}_{k+j} \in \underline{\mathcal{X}}, \boldsymbol{\zeta}_{k+N_p} \in \underline{\mathcal{X}}_f, \end{cases}$$

where $\boldsymbol{\zeta}_k \triangleq [\boldsymbol{\xi}_{1,k}^\top \boldsymbol{\xi}_{2,k}^\top \boldsymbol{\xi}_{3,k}^\top]^\top \in \mathbb{R}^6$ collects the position and velocity of the drone;

$\underline{\boldsymbol{A}} = \text{diag}(\boldsymbol{A}, \boldsymbol{A}, \boldsymbol{A})$; $\underline{\boldsymbol{B}} = \text{diag}(\boldsymbol{B}, \boldsymbol{B}, \boldsymbol{B})$; $\underline{\mathcal{X}} = \{\boldsymbol{\zeta}_k : \boldsymbol{\xi}_{i,k} \in \mathcal{X}_i, i \in \{1,2,3\}\}$ is the same state constraints chosen as in (12b). Similarly, the matrices $\underline{\boldsymbol{Q}} \triangleq \text{diag}(\boldsymbol{Q}, \boldsymbol{Q}, \boldsymbol{Q})$ and $\underline{\boldsymbol{R}} \triangleq \text{diag}(\boldsymbol{R}, \boldsymbol{R}, \boldsymbol{R})$ are chosen from choice of weighting for the problem (12). Lastly, the terminal ingredients $\underline{\boldsymbol{P}}, \underline{\mathcal{X}}_f$ are computed similarly as in Sce. 1. In other words, for the sake of comparison, the cost function, the state constraints and the terminal ingredients (the local controller, the quadratic terminal cost) are identically chosen with respect to Scenario 1. The explicit MPC law is then generated in the same fashion for the optimization problem (16). The main difference in this control synthesis is that the non-conservative approximation $\tilde{\mathcal{V}}_c$ as in (10) is employed, as opposed to $\mathcal{B}$ as in (11) for Sce. 1.

- *Scenario 3* (Sce. 3): In this case, the implicit MPC law (16) is implemented by solving the problem online with IPOPT solver in CasAdi.

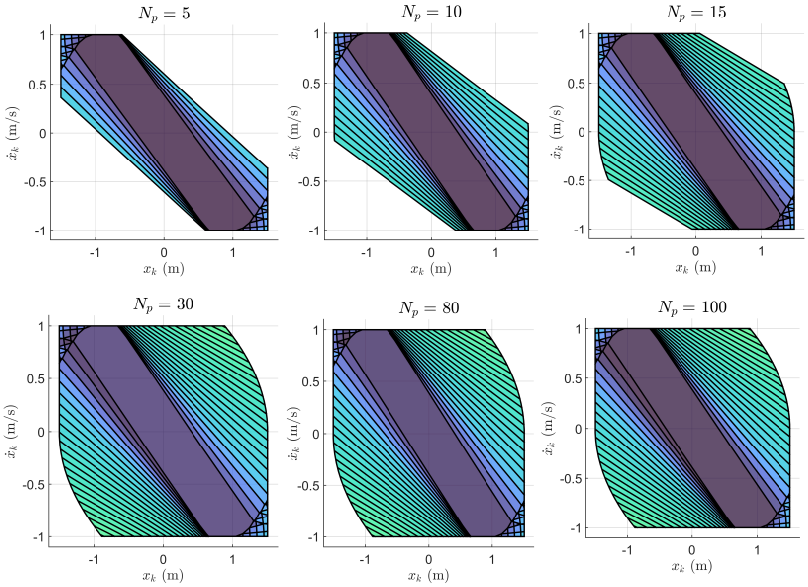

Figure 3. Explicit MPC solution with different choices of N_p.

The simulations are conducted for an interval of 60 seconds. The critical regions are calculated with the MPT3 toolbox on Matlab 2021b (Herceg, Kvasnica, Jones, & Morari, 2013) and stored as a `.mat` file data. The initial position for both scenarios was chosen as: $\boldsymbol{\xi}_{1,0} = [1.25, -0.8]^\top, \boldsymbol{\xi}_{2,0} = [0, 0.2]^\top, \boldsymbol{\xi}_{3,0} = [0.5, 0.2]^\top$ while the tuning adopts the values: $\boldsymbol{Q} = \text{diag}(50, 5), \boldsymbol{R} = 10$, resulting in the corresponding value of $\boldsymbol{P} = \begin{bmatrix} 524.37 & 223.75 \\ 223.75 & 225.97 \end{bmatrix}$, $\boldsymbol{K} = -[2.0003, 2.0978]$. The parameter setup is given in Table 2. Implementation code can be found in the following address `https://gitlab.com/huuthinh.do0421/explicitmpc-for-quadcopters`.

Discussion: Figure 3 presents how the critical regions expand and cover the state space with respect to different choices of the prediction horizon N_p. It is noticeable that, for a fixed choice of state constraints $\mathcal{X}_i$, after a certain value of N_p, the number of critical regions converges to a fixed value. This can also be seen in the result in Table 3 for Scenario 1. After $N_p > 30$, the number of regions in each of the three axes remains constant. Besides, even reaching the limit value, the computational requirement for data storage is relatively small. Meanwhile, with just 2 steps of prediction, the 6D

Table 2. Parameters for simulations and experimental tests

Parameters	Values
$T_{max}; \epsilon_{max}$ in (3)	$1.45g$ m/s^2; 0.1745 (rad)
Sampling time t_s as in (2)	100 ms
$\bar{\tilde{u}}_1, \bar{\tilde{u}}_2, \bar{\tilde{u}}_3$ as in (11)	0.8154,0.8154,3.27
$\bar{\mathrm{p}}_1, \bar{\mathrm{p}}_2, \bar{\mathrm{p}}_3$ for $\mathcal{X}_i$ in (12b)	1.5,1.5,1.5 (m)
$\bar{\dot{\mathrm{p}}}_1, \bar{\dot{\mathrm{p}}}_2, \bar{\dot{\mathrm{p}}}_3$ for $\mathcal{X}_i$ in (12b)	1,1,1.5 (m/s)

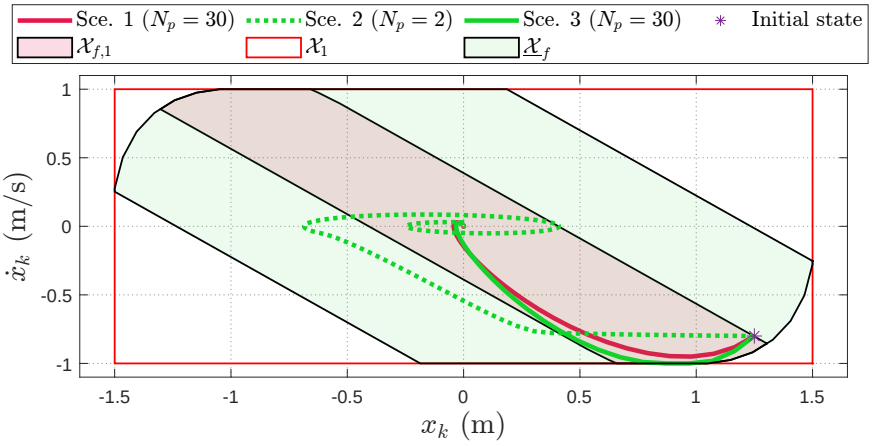

Figure 4. Comparison between the ExMPC for the proposed setup (Sce. 1), for the 6D model (Sce. 2) and the IMPC with the 6D model (Sce. 3).

explicit MPC solution in Scenario 2 accounts for a memory space that is 234 times larger than that of a 100-N_p in Scenario 1. Although regarded as an offline effort, the enumeration of all these regions required roughly 24 hours, in comparison with less than 3 minutes for the entire Scenario 1. Moreover, with an effortful online search (sequentially checking all the regions), the computational burden of the 6D explicit MPC is exposed via the runtime required (more than 13 times the sampling time t_s). This performance can be considered impractical for real-time implementation, highlighting the computational advantage of (12) over (16). Note that during the implementation, the sequential search among the critical regions was not turned off while the system enters the invariant terminal region. Therefore, the computational strength of the ExMPC scheme necessarily comes from the proposed decoupling settings.

Table 3. Simulation results and numerical specifications

	Number of regions	Size (MB)	N_p	Avg. CPU time	RMS errors (cm)
Sce.1	(99,99,11)	0.02	5	0.1563 ms	18.906
	(103,103,11)	0.11	30	0.3385 ms	18.906
	(103,103,11)	0.33	80	0.2864 ms	18.906
	(103,103,11)	0.41	100	0.1980 ms	18.906
Sce.2	49897	96.0	2	1318.5 ms	109.5
Sce.3	-	-	5	4.72 ms	11.46
	-	-	30	11.63 ms	11.319
	-	-	80	27.05 ms	11.319
	-	-	100	33.08 ms	11.319

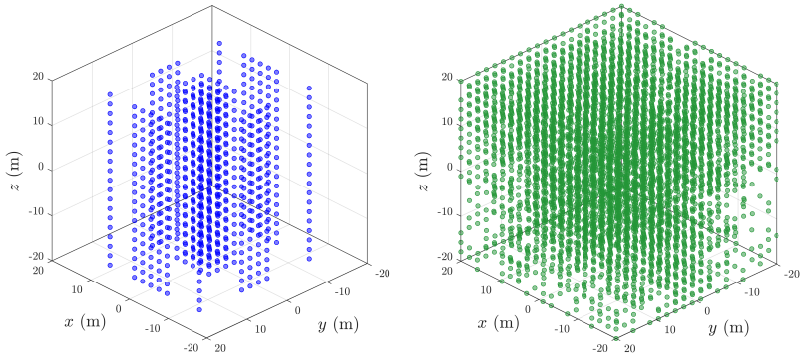

Figure 5. Feasible static initial conditions of ExMPC (left - Sce. 1) and IMPC (right - Sce. 2) without state constraints.

Furthermore, regarding the closed-loop performance for the same chosen parameters, it is certain that the Implicit MPC (IMPC) in Scenario 3 will outperform the ExMPC in Scenario 1 due to its less conservative input power (between $\mathcal{B}$ in (11) and $\tilde{\mathcal{V}}_c$ in (10), respectively). This advantage is not only shown by smaller root-mean-square (RMS) errors[1], but also indicated via the larger terminal invariant set (see Figure 4). With the interpretation as a safe region ensuring constraint satisfaction and asymptotic stability, the terminal region $\underline{\mathcal{X}}_f$ for Scenario 3 (as well as Scenario 2) encloses a larger domain in the state space in comparison with Scenario 1. This indicates that, for a fixed choice of initial condition, both Scenario 2 and Scenario 3 will require a smaller size of prediction horizon to guarantee the system's origin convergence and recursive feasibility. However, despite this advantage, the more efficient offline computation and implementation of Scenario 2 still remain open for further investigation from both combinatorial and geometric standpoints (Mihai, Stoican, & Ciubotaru, 2022; Xiu & Zhang, 2020). Lastly, this conservativeness of Scenario 1 exposes another limitation concerning its feasibility. More specifically, since the size of the terminal set for a fixed horizon directly affects the domain of attraction (Brunner, Lazar, & Allgöwer, 2015), it is certain that the feasible domain will be significantly limited when the set $\mathcal{B}$ is employed as apposed to $\tilde{\mathcal{V}}_c$. To demonstrate this drawback, the state constraints in Scenario 1 and 2 were disregarded and static feasible initial conditions for both cases were collected by checking the position space on a 20×20 grid sampled on $x, y, z \in [-20, 20]\,(m)$. The simulation result is depicted in Figure 5. As expected, for a fixed horizon of $N_p = 30$, Sce. 1 shows considerably less feasible initial conditions, while the polytopic approximation $\tilde{\mathcal{V}}_c$ in Sce. 2 enjoys more control power and, hence, a larger set of feasible states.

Remark 3. *With the intention to adapt the stabilization problem* (12) *for trajectory tracking problem in the experiment, it is assumed that the reference trajectory satisfies:*

(1) $\boldsymbol{\xi}^{\text{ref}}_{i,k+1} = \boldsymbol{A}\boldsymbol{\xi}^{\text{ref}}_{i,k} + \boldsymbol{B}\tilde{u}^{\text{ref}}_{i,k}$ *and* $\tilde{\boldsymbol{u}}^{\text{ref}}_k \in \mathcal{V}^{\text{ref}}_c$, *where* $\boldsymbol{\xi}^{\text{ref}}_{i,k}, \tilde{\boldsymbol{u}}^{\text{ref}}_k$ *denote the reference for the state* $\boldsymbol{\xi}_{i,k}$ *and the input* $\tilde{\boldsymbol{u}}_k$, *respectively.* $\mathcal{V}^{\text{ref}}_c$ *is a time-invariant set enclosing the reference signal* $\tilde{\boldsymbol{u}}^{\text{ref}}_k$.

(2) The polyhedral set $\Delta\mathcal{V}_c \triangleq \tilde{\mathcal{V}}_c \ominus \mathcal{V}^{\text{ref}}_c$ *contains the origin as its interior point.*

[1] with the position $\boldsymbol{p}_k$ as in (1) and the desired reference $\boldsymbol{p}^{\text{ref}}_k = [p^{\text{ref}}_{1,k}, p^{\text{ref}}_{2,k}, p^{\text{ref}}_{3,k}]^\top$, the RMS error was computed as: RMS error $= \frac{1}{3}\sum_{i=1}^{3}\sqrt{N^{-1}_{sim}\sum_{k=0}^{N_{sim}}(p^{\text{ref}}_{i,k} - p_{i,k})^2}$ with N_{sim} denoting the final step of the test.

Then, with $\Delta\boldsymbol{\xi}_{i,k} \triangleq \boldsymbol{\xi}_{i,k} - \boldsymbol{\xi}_{i,k}^{\text{ref}}$, *the tracking error yields:*

$$\Delta\boldsymbol{\xi}_{i,k+1} = \boldsymbol{A}\Delta\boldsymbol{\xi}_{i,k} + \boldsymbol{B}\Delta\tilde{u}_{i,k}, \tag{17}$$

with the constraint $\Delta\tilde{\boldsymbol{u}}_k = \tilde{\boldsymbol{u}}_k - \tilde{\boldsymbol{u}}_k^{\text{ref}} \in \Delta\mathcal{V}_c$. *Consequently, by finding a box inside* $\Delta\mathcal{V}_c$, *the dynamics* (17) *can be stabilized by the control synthesis for* $\Delta\tilde{\boldsymbol{u}}_k$ *with the similar explicit MPC setup presented previously. Finally, the reference tracking control can be implemented as:*

$$\tilde{\boldsymbol{u}}_k = \Delta\tilde{\boldsymbol{u}}_k + \tilde{\boldsymbol{u}}_k^{\text{ref}}. \tag{18}$$

4. Experimental validation

Hereinafter, we implement the control scheme described in Figure 2 in the Crazyflie 2.1 nano-drone framework. While the quadcopter's position and attitude are estimated via eight motion capture Qualisys cameras, the Explicit MPC (15) is computed by carrying out a sequential search among the computed regions, respectively for all three axes (i.e, $i \in \{1, 2, 3\}$). The resulting control $\tilde{\boldsymbol{u}}_k$ then is transformed back to the real input $\boldsymbol{u}_k$ via (4) and sent to the drone through its Python API with the USB Crazyradio 2.0 dongle. In this work, we limit our problem to the outer-loop control (position control) for the vehicle and assume that the inner loop (attitude control) is stable. Within the experimental platform, after being computed in a station computer, the desired thrust (T_k), the roll (ϕ_k) and pitch angles (θ_k) are stacked with the desired yaw rate $\dot{\psi}^{\text{ref}} \triangleq 0$ and sent to the drone with an embedded inner-loop to follow (see Figure 6). For this attitude control layer, the built-in controller of Crazyflie 2.1 is exploited. It includes two cascade PID angle and angle rate controllers to compute the stabilizing torques in the roll, pitch and yaw orientation. These torques together with the thrust T_k will be transformed into PWM signals for the four rotors via a Control Mixer designed for their X-shape configuration. A detailed discussion for Crazyflie quadcopter's inner loop can be found in (C. Luis & Ny, 2016; Nguyen et al., 2020a).

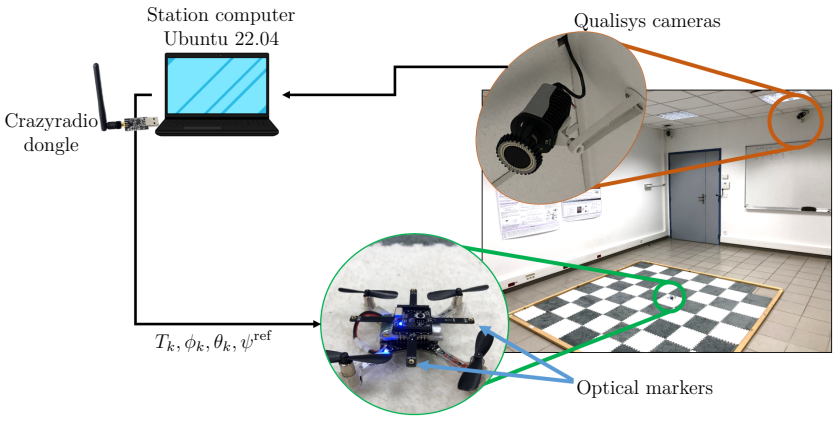

Figure 6. Experimental setup with Crazyflie quadcopter and Qualisys motion capture.

4.1. Validation setup

For the validation, we consider the two following tests.

- *Test 1:* With the objective of tracking the set-point at $x_f = 0.6, y_f = 0.6, z_f = 0.8$(m), the ExMPC computed in (15) is used. Moreover, a comparison between different choices of prediction horizon is given. A classical IMPC as in (16) is also tested in the same setup to highlight the simplicity of our formulation. IPOPT solver is employed with CasADi Python.
- *Test 2:* Herein, four quadcopters will be controlled sequentially with the presented method to track two groups of predefined trajectories: i) *Circular reference:* (Ref. 1) The four drones pursue a circular motion while maintaining a constant altitude of 1m; ii) *Square formation reference:* (Ref. 2) The four drones will follow a square formation trajectories while maintaining a one-meter altitude. The trajectory generation solution was adopted from minimum length B-spline parameterization framework (Prodan et al., 2019).

In the experiments, weighting matrices in Section 3.2 are readopted.

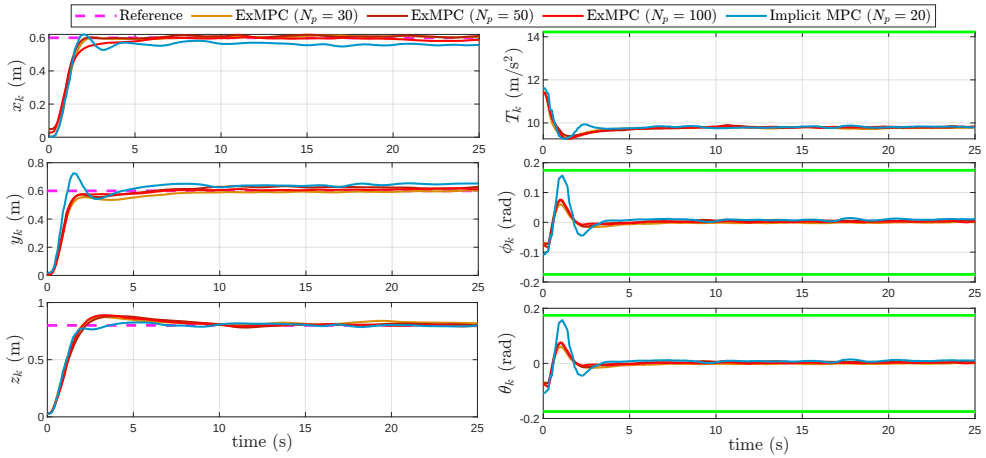

Figure 7. The quadcopter position and input $\boldsymbol{u}_k$ (with limits shown in green solid lines) in experiment with ExMPC and IMPC.

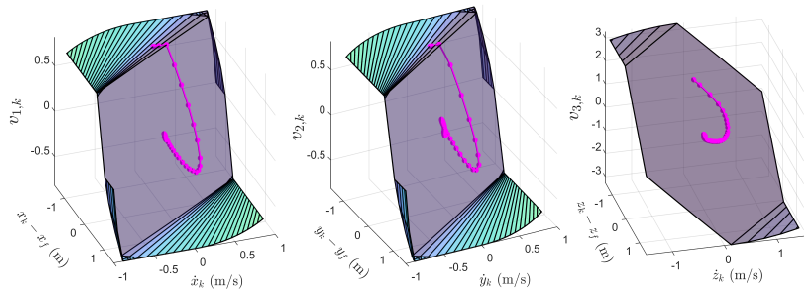

Figure 8. Experimental trajectory of $\tilde{\boldsymbol{u}}_k$ in Test 1, $N_p = 30$.

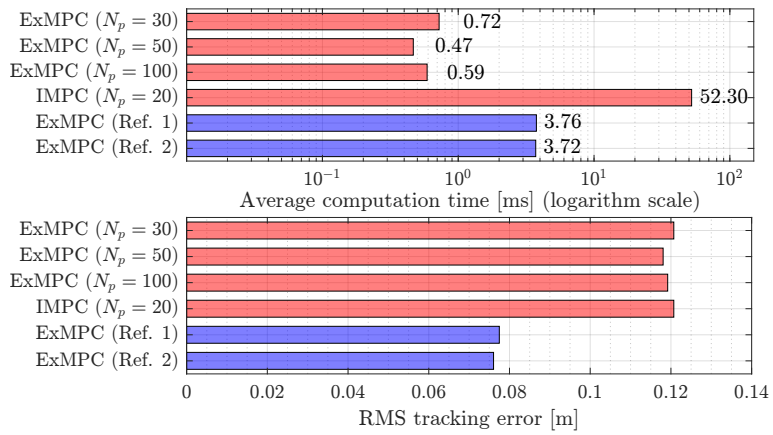

Figure 9. Computation time and RMS tracking errors with Explicit MPC and Implicit MPC in Test 1 (red) and Test 2 (blue).

4.2. Experimental validation and discussion

Figure 7 depicts the tested results for the Crazyflie quadcopter, where the tracking objective and input constraints are both respected. The constraint satisfaction can also be shown by the containment of $\tilde{\boldsymbol{u}}_k$ in $\mathcal{B}$ in Figure 8. Therein, it can be seen that all the ExMPC setups provide notably less input exploitation as a consequence of the conservative approximation of $\mathcal{B}$ in (11) with respect to $\tilde{\mathcal{V}}_c$ in (10). However, this difference does not necessarily imply slower convergence or a smaller overshoot. This is because the trade-off of control power for simplicity in these schemes contrarily provides the ability to have a solution with much longer prediction horizon, resembling the infinite horizon solution. Next, the computational advantage of the proposed scheme can also be observed via the execution time for each controller in Figure 9. Therein, the IMPC demands virtually 80 times more of the computation time in Test 1, although the tracking performance between the two controllers appears to be commensurate. Even when there are four drones being controlled in Test 2, the computation time of the ExMPC (around 4ms) is still lower than that of the IMPC tracking with a single drone in Test 1 (52.3ms). Beside the simplicity in digital implementation, qualitative comparisons in computational burden can also be addressed with the NMPC approaches proposed with the same theoretical guarantees in the literature (Gomaa et al., 2022; Nguyen et al., 2020c) (≈10-100ms) while the ExMPC requires under 1ms. This simplicity allows only the proposed setup to be implemented with four drones in a centralized manner without exceeding the chosen sampling time.

Moreover, although there are no collision avoidance constraints imposed, the experiments show the controller's reliability to closely follow a predefined trajectory with a low computational cost, input constraint satisfaction, and stability guaranteed. This satisfactory tracking opens possible implementations in a hierarchical structure where collision-free formation is obtained via online curve parameterization or consensus-based guidance (C. E. Luis, Vukosavljev, & Schoellig, 2020) and dynamical constraints are guaranteed at the lower level. Furthermore, from an implementation viewpoint, although restrictive, the fast feasible decision, prediction and collection of active constraints from the proposed setup also show a promising direction where they are exploited in a warm-start procedure to reduce online computational effort within the IMPC settings (Zeilinger, Jones, & Morari, 2011).

Finally, it is important to bring to light that, although successful validation under

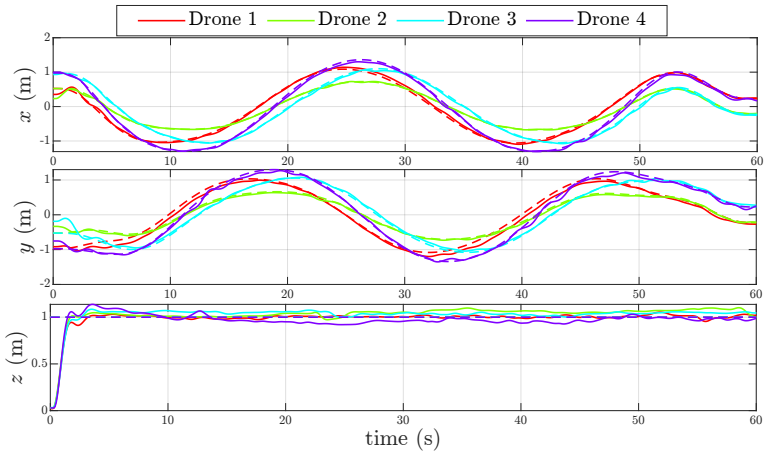

Figure 10. Quadcopters' trajectory tracking with Explicit MPC (Square formation reference, dashed and color-coded with the corresponding drone).

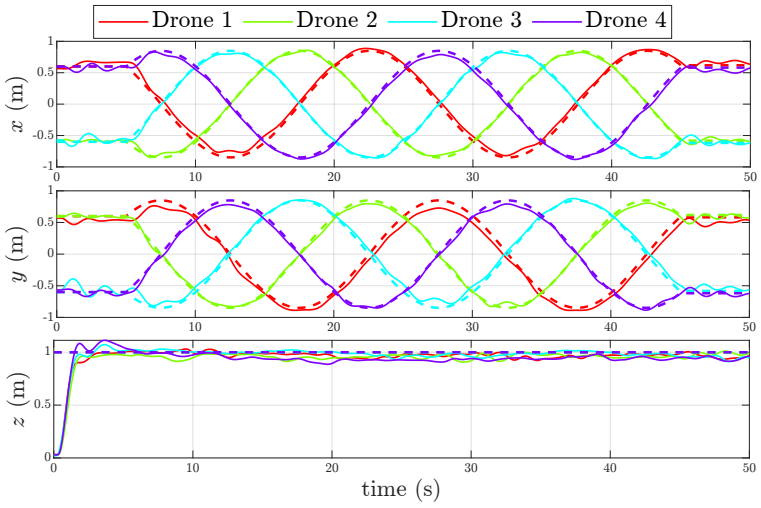

Figure 11. Multiple quadcopters' trajectory tracking with Explicit MPC (Circular reference, dashed and color-coded with the corresponding drone).

indoor environment was provided, the study with model (1) enjoyed several assumptions including little disturbances (e.g., drag), uncertainties or mismatch between the outer loop and the inner loop. These limitations will undeniably become more critical when larger models or outdoor applications are considered. Therefore, future completion of the method will concentrate on the robustification of the scheme via additional countermeasures such as disturbance rejection with estimators or robust MPC.

5. Conclusion

With the advantage of the linear representation in the flat output space of the quadcopter's translational dynamics, this work proposes the constrained position tracking problem of the vehicle within the framework of the explicit MPC design. The successfully validated results highlight the computational advantage of the explicit formulation, while the conservativeness of the approximated input constraints can be compensated via appropriate tuning and a sufficiently large prediction horizon. Future direction concerns the implementation of more efficient point location algorithms to make use of the less conservative constraint set $\tilde{\mathcal{V}}_c$ in (10), while respecting the high sampling frequency of the vehicle. Furthermore, robust design subject to external and aerodynamics disturbances will be considered for larger scale models.

Appendix A. Largest box inscribed in $\mathcal{V}_c$

With the zonotopic setup in (Do & Prodan, 2023), the maximum volume set $\mathcal{B}$ as in (11) inscribed in $\mathcal{V}_c$ in (9) is found by solving:

$$(\bar{\tilde{u}}_1, \bar{\tilde{u}}_2, \bar{\tilde{u}}_3)^* = \arg\max \bar{\tilde{u}}_1 \bar{\tilde{u}}_2 \bar{\tilde{u}}_3, \tag{A1a}$$

$$\text{s.t: } \mathcal{I}(\bar{\tilde{u}}_1, \bar{\tilde{u}}_2, \bar{\tilde{u}}_3) \in \mathcal{V}_c \text{ as in (9)}. \tag{A1b}$$

where $\mathcal{I}(\bar{\tilde{u}}_1, \bar{\tilde{u}}_2, \bar{\tilde{u}}_3) \triangleq \{\sum_{i=1}^{3} 0.5\gamma_i \bar{\tilde{u}}_i \boldsymbol{e}_i : |\gamma_i| = 1\}$ denote the finite set which contains all the vertices of $\mathcal{B}$, with $\boldsymbol{e}_i$ denoting the i-th column of the 3×3 identity matrix. The finite condition (A1b) implies and is implied by the containment $\mathcal{B} \subset \mathcal{V}_c$ thanks to the convexity of $\mathcal{V}_c$, while the cost function (A1a) maximizes the volume of $\mathcal{B}$. With the continuity of the cost function and the boundedness of the constraints in (A1), the existence of an optimal solution is certain. By analyzing the KKT necessary conditions, the description of the set can be briefly expressed as follows:

- if $2c^\star(g + c^\star) - \tan(\epsilon_{max})^2(c^\star - g)^2 < 0$, then:

$$\bar{\tilde{u}}_3 = c^\star; \bar{\tilde{u}}_1 = \bar{\tilde{u}}_2 = \sqrt{c^\star(c^\star + g)}. \tag{A2}$$

- Otherwise,

$$\bar{\tilde{u}}_3 = g/3; \bar{\tilde{u}}_1 = \bar{\tilde{u}}_2 = \tan(\epsilon_{max})\sqrt{\bar{\tilde{u}}_3(g - \bar{\tilde{u}}_3)}, \tag{A3}$$

where $c^\star \triangleq \left(-2g + \sqrt{g^2 + 3T_{max}^2}\right)/3$.

References

Bemporad, A., Morari, M., Dua, V., & Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*.

Bertrand, S., Guénard, N., Hamel, T., Piet-Lahanier, H., & Eck, L. (2011). A hierarchical controller for miniature vtol uavs: Design and stability analysis using singular perturbation theory. *Control Engineering Practice*, *19*(10), 1099–1108.

Brunner, F. D., Lazar, M., & Allgöwer, F. (2015). Stabilizing model predictive control: On the enlargement of the terminal set. *International Journal of Robust and Nonlinear Control*, *25*(15), 2646–2670.

Cao, N., & Lynch, A. (2021). Predictor-based control design for uavs: robust stability analysis and experimental results. *International Journal of Control*, *94*(6), 1529–1543.

Cao, N., & Lynch, A. F. (2015). Inner–outer loop control for quadrotor uavs with input and state constraints. *IEEE Transactions on Control Systems Technology*, *24*(5), 1797–1804.

Chen, X., & Wang, L. (2013). Cascaded model predictive control of a quadrotor uav. In *2013 australian control conference* (pp. 354–359).

Chen, X., & Wang, L. (2016). Quadrotor cascade pid controller automatic tuning. In *2016 australian control conference (aucc)* (p. 311-316).

Cohen, M. R., Abdulrahim, K., & Forbes, J. R. (2020). Finite-horizon LQR control of quadrotors on $SE_2(3)$. *IEEE Robotics and Automation Letters*, *5*(4), 5748-5755.

Do, H. T., & Prodan, I. (2023). Indoor experimental validation of MPC-based trajectory tracking for a quadcopter via a flat mapping approach. In *European control conference.*

Do, H.-T., Prodan, I., & Stoican, F. (2021). Analysis of alternative flat representations of a uav for trajectory generation and tracking. In *2021 25th international conference on system theory, control and computing (icstcc)* (pp. 58–63).

Fliess, M., Lévine, J., Martin, P., & Rouchon, P. (1993). On differentially flat nonlinear systems. In *Nonlinear control systems design 1992* (pp. 159–163). Elsevier.

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

Gomaa, M. A., De Silva, O., Mann, G. K., & Gosine, R. G. (2022). Computationally efficient stability-based nonlinear model predictive control design for quadrotor aerial vehicles. *IEEE Trans. on Control Systems Technology*.

Herceg, M., Kvasnica, M., Jones, C., & Morari, M. (2013). Multi-Parametric Toolbox 3.0. In *2013 european control conference* (pp. 502–510).

Jiajin, L., Rui, L., Yingjing, S., & Jianxiao, Z. (2017). Design of attitude controller using explicit model predictive control for an unmanned quadrotor helicopter. In *2017 chinese automation congress* (pp. 2853–2857).

Lee, J. H. (2011). Model predictive control: Review of the three decades of development. *International Journal of Control, Automation and Systems*, *9*, 415–424.

Lei, W., Li, C., & Chen, M. Z. (2018). Robust adaptive tracking control for quadrotors by combining pi and self-tuning regulator. *IEEE Trans. Control Syst. Technol.*, *27*(6), 2663–2671.

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

Liu, C., Lu, H., & Chen, W.-H. (2015). An explicit mpc for quadrotor trajectory tracking. In *2015 34th chinese control conference.*

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

Luis, C. E., Vukosavljev, M., & Schoellig, A. P. (2020). Online trajectory generation with distributed model predictive control for multi-robot motion planning. *IEEE Robot. Autom. Lett.*, *5*(2).

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. (2000). Constrained model

predictive control: Stability and optimality. *Automatica*, *36*(6), 789–814.
Mihai, S. S., Stoican, F., & Ciubotaru, B. D. (2022). On the link between explicit MPC and the face lattice of the lifted feasible domain. *IFAC-PapersOnLine*.
Mueller, M. W., & D'Andrea, R. (2013). A model predictive controller for quadrocopter state interception. In *European control conference* (pp. 1383–1389).
Nguyen, N. T., Prodan, I., & Lefèvre, L. (2018). Effective angular constrained trajectory generation for thrust-propelled vehicles. In *2018 european control conference (ecc)* (pp. 1833–1838).
Nguyen, N. T., Prodan, I., & Lefevre, L. (2019). A stabilizing NMPC design for thrust-propelled vehicles dynamics via feedback linearization. In *2019 american control conference* (pp. 2909–2914).
Nguyen, N. T., Prodan, I., & Lefèvre, L. (2020a). Flat trajectory design and tracking with saturation guarantees: a nano-drone application. *International Journal of Control*, *93*(6), 1266–1279.
Nguyen, N. T., Prodan, I., & Lefèvre, L. (2020b). Multicopter attitude control through nmpc design with guaranteed stability. *IFAC-PapersOnLine*, *53*(2).
Nguyen, N. T., Prodan, I., & Lefèvre, L. (2020c). Stability guarantees for translational thrust-propelled vehicles dynamics through nmpc designs. *IEEE Transactions on Control Systems Technology*, *29*(1), 207–219.
Pereira, J. C., Leite, V. J., & Raffo, G. V. (2021). Nonlinear model predictive control on SE(3) for quadrotor aggressive maneuvers. *Journal of Intelligent & Robotic Systems*, *101*, 1–15.
Prodan, I., Olaru, S., Bencatel, R., de Sousa, J. B., Stoica, C., & Niculescu, S.-I. (2013). Receding horizon flight control for trajectory tracking of autonomous aerial vehicles. *Cont. Engineering Practice*, *21*(10).
Prodan, I., Stoican, F., & Louembet, C. (2019). Necessary and sufficient LMI conditions for constraints satisfaction within a B-spline framework. In *Ieee 58th conf. on decision and control.*
Sun, S., Romero, A., Foehn, P., Kaufmann, E., & Scaramuzza, D. (2022). A comparative study of nonlinear mpc and differential-flatness-based control for quadrotor agile flight. *IEEE Transactions on Robotics*.
Teel, A. R. (1992). Global stabilization and restricted tracking for multiple integrators with bounded controls. *Systems & control letters*, *18*(3), 165–171.
Torrente, G., Kaufmann, E., Föhn, P., & Scaramuzza, D. (2021). Data-driven mpc for quadrotors. *IEEE Robotics and Automation Letters*, *6*(2), 3769–3776.
Tran, T.-T., Sam, S., He, W., Luu-Trung-Duong, P., & Truong, N.-V. (2018). Trajectory tracking control of a quadrotor aerial vehicle in the presence of input constraints. *Int. Journal of Control, Auto. and Systems*, *16*, 2966–2976.
Wang, D., Pan, Q., Shi, Y., Hu, J., & Zhao, C. (2021). Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment. *IEEE Transactions on Cybernetics*.
Xie, H., & Veitch, D. (2020). Nested saturation control of multiple vector integrators and its application to motion control of uavs. *International Journal of Robust and Nonlinear Control*, *30*(1), 246–265.
Xiu, X., & Zhang, J. (2020). Grid kd tree approach for point location in polyhedral data sets–application to explicit mpc. *International Journal of Control*, *93*(4), 872–880.
Zanelli, A., Horn, G., Frison, G., & Diehl, M. (2018). Nonlinear model predictive control of a human-sized quadrotor. In *European control conference.*
Zeilinger, M. N., Jones, C. N., & Morari, M. (2011). Real-time suboptimal model predictive control using a combination of explicit mpc and online optimization. *IEEE Transactions on Automatic Control*.