

# TRACE: Trajectory Refinement with Control Error Enables Safe and Accurate Maneuvers

Zihan Yang, Jindou Jia, Yuhang Liu, Kexin Guo\*, Xiang Yu and Lei Guo, *Fellow, IEEE*

**Abstract**—Normal trajectory optimizations are carried out without the consideration of the closed-loop feasibility and performance. During aggressive flights, bias from the nominal trajectory could trigger feedback with larger control inputs, which might break through the actuator constraints, inducing stability issues. By leveraging the idea of TRAjectory refinement with Control Error (TRACE) for higher closed-loop performance, we develop a post-trajectory optimization approach for safe and accurate tracking control. A closed-loop model is established combining both control policy and saturations in order to model the dynamic progress of trajectory tracking. Subsequently, a time-varying optimal control problem (OCP) is constructed to enhance the tracking performance by weighted minimizing the tracking error and the variation from the nominal trajectory progress. Meanwhile, a flatness-based method is presented to provide aggressive time-optimal trajectories with high-order derivatives of the reference states. Examples show the effectiveness of our approach in refining aggressive trajectories, where safe and accurate tracking can be done using a baseline controller without fine-tuning and further control allocation design.

## I. INTRODUCTION

As maneuverable, flexible and small-sized aerial platforms, autonomous quadrotors appear in various applications with promising utility. The pursuit of mission completion with higher efficiency makes time-optimal and agile flights active topics with noticeable works [1], [2], [3]. To execute a task, a typical solution is to hierarchically decouple the quadrotor system into a higher-level planner and lower-level reference tracking controller. The decoupling brings us an explainable system architecture and allows us to construct straightforward designs. However, such treatment also brings a degradation to the tracking performance and even triggers safety issues [4].

This work was supported in part by the Major State Basic Research Development Program of China (No.2022YFB4701301), National Natural Science Foundation of China (No.62273023 and 61973012), Major Science and Technology Innovation Program of Hangzhou (No.2022AIZD0137), Defense Industrial Technology Development Program (No.JCKY2020601C016) and Beijing Nova Program (No.20230484266).

Zihan Yang and Yuhang Liu are with the School of Aeronautic Science and Engineering, Beihang University, 37 Xueyuan Road, Beijing, 100191, China.

Jindou Jia is with the School of Automation Science and Electrical Engineering, Beihang University, 37 Xueyuan Road, Beijing, 100191, China.

Kexin Guo is with the School of Aeronautic Science and Engineering, Beihang University, 37 Xueyuan Road, Beijing, 100191, China, and also with the Hangzhou Innovation Institute, Beihang University, Changhe Street, Hangzhou, 310051, China.

Xiang Yu and Lei Guo are with the School of Automation Science and Electrical Engineering, Beihang University, 37 Xueyuan Road, Beijing, 100191, China, and also with the Hangzhou Innovation Institute, Beihang University, Changhe Street, Hangzhou, 310051, China.

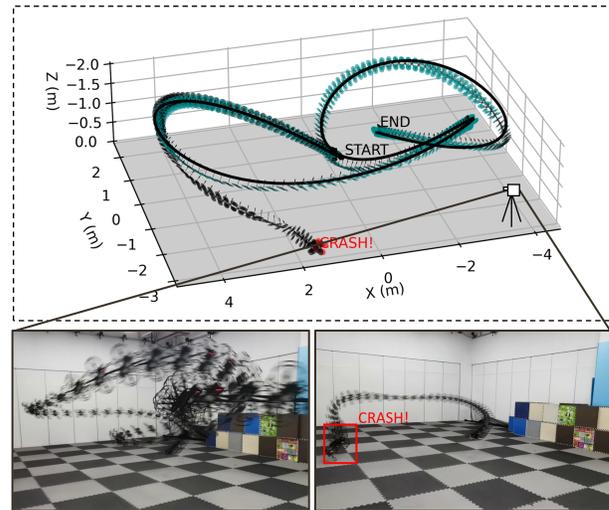


Fig. 1. The effect of the input saturation constraint in real-world time-optimal flight. The quadrotor fails (black) to track the time-optimal trajectory optimized with the open-loop input constraint and successfully tracks (cyan) the one refined with the input saturation constraint. The camera captions of real-world flight footage are presented. The complete video can be found at <https://youtu.be/2LhG1oYMrq8>.

Actuator constraints are typical input feasibility constraints for the safety and stability ensurance of maneuvers. Generally, they can be formulated as box constraints in trajectory optimizations. Despite the straightforward setup, such constraints are not exact guarantees of safety and stability during trajectory tracking. Tracking errors could be observed when references are violent. Such errors could bring extra input from the control policy. As the reference requires input values next to the boundary of actuator limits in aggressive trajectories [5], [4], it is possible to induce inputs overshooting the capability thresholds, leading to input saturations that trigger instability.

Accuracy is another major objective in many researches. Challenges are left to the controller level in common applications [6], [7]. A well-designed-and-tuned tracking controller is demanded for accurate trajectory tracking. However, both designing and fine-tuning works remain difficult in practice [8] and high-gain controllers are not practical in many applications [9]. Furthermore, tracking error can still be stimulated by saturations as the desired converging rate is loss due to the limits of inputs.

Since the reference trajectory is usually optimized without the consideration of the closed-loop properties, tracking error could not be directly controlled during motion planning. Our

approach (TRACE) refines the nominal trajectory optimized by an arbitrary trajectory optimizer, minimizing the tracking error while maintaining the original trajectory progress as much as possible.

Inside TRACE, a standard quadrotor open-loop model is constructed while integrating low-level control policy, forming a closed-loop one. We choose a differential flatness-based controller (DFBC) as the control policy. An input saturation constraint is established in the OCP for motor thrust saturation avoidance. Besides, TRACE comes with a flatness-based time-optimal trajectory generation method that provides aggressive trajectories for simulations and experiments. Our main contributions are summarized as follows.

- A differential flatness-based time-optimal trajectory optimization method of higher flexibility compared with polynomial-based approaches.
- A post-optimizer with time-varying OCP setup aimed at refining time parameters of the given nominal trajectories for safe and accurate tracking performance.
- Simulations and real-world experiments on tracking refined trajectories associated with parameter scanning and analysis.

The post-optimization strategy could be seen as an optimal coordination of pre-computed trajectories and a given control policy, potentially bridging the gap between planning and control for quadrotors as well as other modern robotics.

The outline of this paper is as follows. Section II reviews the related works. Section III provides the preliminaries on notations, quadrotor dynamics and the DFBC scheme. Section IV goes into the details of the design of the flatness-based trajectory optimizer and the post-optimizer. Section V presents the simulations and numerical studies. Section VI shows the results of the real-world experiments. Section VII concludes this article with a discussion of the limitations of the proposed methods.

## II. RELATED WORK

### A. Time-Optimal Trajectory Optimization for Quadrotors

Minimum-time planning is demanded in numerous applications yet with huge challenges. Recent successful cases consist of various approaches are briefly discussed next. Point-mass model with bang-bang control could be derived using Pontryagin's Maximum Principle. Online replanning of this method is also carried out by combining sampling-based techniques [10]. Despite the advantages of computation, this policy introduces discontinuity into acceleration and does not inherently handle obstacle-avoidance constraints. Full-state model-based trajectory optimizations [11], [6] provide trajectories based on more accurate open-loop dynamics associated with feasibility constraints and therefore lead to higher tracking robustness compared with the simplified models. Yet point-mass models with sampling-based methods are more favored by obstacle-free path generation in complex environments [12].

Another category of trajectory optimization utilizes the differential flatness property of quadrotor systems. General

differential flatness-based planners apply piecewise polynomials with pre-assigned time collocation. For time-optimal implementations, one could use the bi-level optimization [13] technique to optimize the total trajectory time at the upper level, or convert to a convex problem [14] by utilizing the internal structure of the minimum-time problems. Additionally, these methods are favored by trajectory tracking since one can obtain high-order state references from the optimization, passing these terms to the tracking controller as feedforwards [15]. Despite the effectiveness of the flatness property, handling input limits require more complex constraints through the flatness mapping and polynomial-based representation lacks flexibility due to its pre-assigned collocation points. Our trajectory optimization method is flatness-based yet with a more flexible formulation of the time-optimal OCP.

Neither full-state nor flatness-based trajectory methods have access to the closed-loop dynamics. The security and accuracy remain unknown without the consideration of tracking performance and input saturations of the closed-loop system.

### B. Trajectory Tracking Controller for Quadrotors

As a state-of-the-art controller scheme, differential flatness-based controller (DFBC) has been gaining attention for its high performance in tracking aggressive trajectories [16], [17]. By leveraging the flat outputs, DFBC enjoys direct feedforward from trajectories. However, tracking time-optimal trajectories requires further modification to a standard DFBC. Improvements could be made by introducing control allocations and extra internal-model compensations at the cost of increasing complexity [18], [7].

Model predictive controllers (MPCs) are another group of controllers that are worth mentioning. MPC receives the information on the coming reference trajectories, potentially being able to achieve higher tracking accuracy. Optimal tracking with input constraints could provide better robustness since the saturations are avoided while minimizing the objectives. But the tracking accuracy of normal MPC controllers is undesirable in aggressive flights [4]. A model predictive contouring controller is presented in this work [6] for time-optimal trajectory tracking. This controller provides a trade-off between trajectory progress and tracking accuracy, gaining improvements in tracking performance and flight robustness. A time-adaptive MPC is designed in [19], in which the initial sampling time of the upcoming trajectory segment is optimized with control inputs, introducing flexibility into the original reference signal.

Above mentioned model predictive controllers leverage the idea of time-scaling into real-time control for better trajectory tracking. While appreciable working frequencies are retained thanks to modern embedded computing toolkits [20], [21], yet onboard CPUs are still required with heavy works on tuning. TRACE also includes time-scaling techniques but focuses on the planning stage with the consideration of low-level tracking performance. General controllers with continuous mapping are able to be adapted, making it generalized for various of platforms with microcontrollers.

### III. PRELIMINARIES

#### A. Notation

Several notations in this paper are explained as follows.  $\|\cdot\|_2$  represents the Euclidean norm of a vector.  $\text{diag}\{\mathbf{a}\}$  refers to a diagonal matrix with elements of vector  $\mathbf{a}$ .  $(\cdot)^\top$  is noted as the transpose of a matrix or a vector.  $(\cdot)^\vee$  represents the mapping from  $SO(3)$  to  $so(3)$ .  $\mathbf{H}_{m \times n}$  represents a  $m \times n$  dimensional matrix/vector whose elements are real number  $H$ .  $\mathbf{I}_n$  refers to a  $n$ -dimensional elementary matrix.  $\times$  refers to the multiplication of a number with another number, vector or matrix and also the cross product of two vectors of the same length. With a quaternion written as  $\mathbf{q} = [q_0, \mathbf{q}_v]^\top \in \mathbb{R}^{4 \times 1}$ ,  $\mathbf{q} \otimes \mathbf{x}$ ,  $\mathbf{x} \in \mathbb{R}^{3 \times 1}$  equals to:  $[-\mathbf{q}_v^\top \mathbf{x}, q_0 \mathbf{I}_3 \mathbf{x} + \mathbf{q}_v \times \mathbf{x}]^\top \in \mathbb{R}^{4 \times 1}$ .

#### B. Quadrotor Dynamics

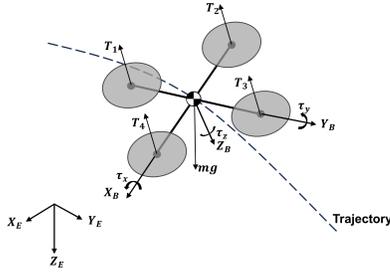


Fig. 2. Quadrotor schematic with definitions of the earth-fixed and the body-fixed frame.

Two coordinate systems are defined for the quadrotor: the earth-fixed frame  $\mathcal{E} = [\mathbf{X}_E, \mathbf{Y}_E, \mathbf{Z}_E]$  and the body-fixed frame  $\mathcal{B} = [\mathbf{X}_B, \mathbf{Y}_B, \mathbf{Z}_B]$ . The rotational matrix  $\mathbf{R}_{\mathcal{E}}^{\mathcal{B}} = [\mathbf{X}_B, \mathbf{Y}_B, \mathbf{Z}_B]$  refers to the rotational transformation that rotates  $\mathcal{E}$  to  $\mathcal{B}$ . A state-space model is constructed with a 13-dimensional state vector  $\mathbf{x} = [\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}]^\top$  and a 4-dimensional input vector  $\mathbf{u} = [T_1, T_2, T_3, T_4]^\top$  of motor thrusts.  $\mathbf{p}, \mathbf{v}$  represent the position and velocity  $\mathcal{E}$  while  $\boldsymbol{\omega}$  is the angular velocity defined in  $\mathcal{B}$ .  $\mathbf{q}$  refers to the quaternion for attitude expression.

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{a} = -\frac{1}{m} \mathbf{Z}_B T + g \mathbf{Z}_E \\ \dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \otimes \boldsymbol{\omega} \\ \mathbf{J} \dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times (\mathbf{J} \boldsymbol{\omega}) + \boldsymbol{\tau} \\ [T, \boldsymbol{\tau}]^\top = \mathbf{C} [T_1, T_2, T_3, T_4]^\top \end{cases} \quad (1)$$

where  $\mathbf{C}$  is the mapping from the motor thrusts to the total thrust  $T$  and torque  $\boldsymbol{\tau} \in \mathbb{R}^{3 \times 1}$ .

#### C. Differential Flatness-based Controller

In a DFBC scheme, a dual-loop structure is constructed based on that the attitude loop is equipped with a bandwidth of over 10 times higher than that of the position loop. The output of the position loop forms the reference of the attitude loop. The position loop includes a PD feedback term with feedforward compensations:

$$\mathbf{a}_d = \mathbf{K}_v (\mathbf{K}_p (\mathbf{p}_r - \mathbf{p}) + \dot{\mathbf{p}}_r - \mathbf{v}) + \ddot{\mathbf{p}}_r - g \mathbf{Z}_E \quad (2)$$

where  $\mathbf{p}_r$  is the reference position and  $\mathbf{a}_d$  refers to the desired acceleration.

The desired acceleration indicates a desired attitude related to where the joint acceleration of total thrust and gravity force points. A mapping could be established from  $\mathbf{a}_d$  to the desired attitude  $\mathbf{R}_{\mathcal{E}d}^{\mathcal{B}} = [\mathbf{X}_{Bd}, \mathbf{Y}_{Bd}, \mathbf{Z}_{Bd}]$ . Note that the mapping may sometimes go singular and singularity-free mapping such as Hopf fibration can be adopted, for more detail please refer to [15], [16], [22], [23]. The desired body rate  $\boldsymbol{\omega}_d$  and angular acceleration  $\dot{\boldsymbol{\omega}}_d$  are given as:

$$\boldsymbol{\omega}_d = [-\mathbf{h}_\omega^\top \mathbf{Y}_{Bd} \quad \mathbf{h}_\omega^\top \mathbf{X}_{Bd} \quad [0, 0, \dot{\psi}_r] \mathbf{Z}_{Bd}]^\top \quad (3)$$

$$\dot{\boldsymbol{\omega}}_d = [-\mathbf{h}_\alpha^\top \mathbf{Y}_{Bd} \quad \mathbf{h}_\alpha^\top \mathbf{X}_{Bd} \quad [0, 0, \ddot{\psi}_r] \mathbf{Z}_{Bd}]^\top \quad (4)$$

where  $\psi_r$  refers to the reference yaw angle and

$$\begin{aligned} \mathbf{h}_\omega &\triangleq \boldsymbol{\omega} \times \mathbf{Z}_{Bd} = -\frac{m}{T} (\ddot{\mathbf{p}}_r - (\mathbf{Z}_{Bd}^\top \ddot{\mathbf{p}}_r) \mathbf{Z}_{Bd}) \\ \mathbf{h}_\alpha &\triangleq \dot{\boldsymbol{\omega}} \times \mathbf{Z}_{Bd} = -\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{Z}_{Bd}) \\ &+ \frac{m}{T} (\ddot{\mathbf{p}}_r^\top \mathbf{Z}_{Bd}) (\boldsymbol{\omega} \times \mathbf{Z}_{Bd}) + \frac{2m (\mathbf{Z}_{Bd}^\top \ddot{\mathbf{p}}_r)}{T} (\boldsymbol{\omega} \times \mathbf{Z}_{Bd}). \end{aligned} \quad (5)$$

The attitude loop provides a desired torque using techniques from the geometric controller [17]:

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{K}_a (\mathbf{R}_{\mathcal{E}}^{\mathcal{B}\top} \mathbf{R}_{\mathcal{E}d}^{\mathcal{B}} - \mathbf{R}_{\mathcal{E}}^{\mathcal{B}\top} \mathbf{R}_{\mathcal{E}}^{\mathcal{B}})^\vee \\ &+ \mathbf{K}_r (\boldsymbol{\omega}_d - \boldsymbol{\omega}) + \mathbf{J} \boldsymbol{\omega}_d \times \boldsymbol{\omega}_d + \mathbf{J} \dot{\boldsymbol{\omega}}_d. \end{aligned} \quad (6)$$

Current attitude  $\mathbf{R}_{\mathcal{E}}^{\mathcal{B}}$  can be derived from the Euler angles. The motor thrusts are finally given as:

$$\mathbf{u} = \mathbf{C}^{-1} [\|\mathbf{m} \mathbf{a}_d\|_2, \boldsymbol{\tau}]^\top. \quad (7)$$

### IV. METHODOLOGY

#### A. Flatness-based Time-Optimal Trajectory Generation

By choosing the positional flat outputs and their derivatives (up to the order of 4) as state vector  $\boldsymbol{\Psi}_p$ , and the snap (the 5<sup>th</sup> order derivatives) as input vector  $\boldsymbol{\Psi}_s$ , the positional flatness system appears to be linear:

$$\dot{\boldsymbol{\Psi}}_p = \mathbf{A}_p \boldsymbol{\Psi}_p + \mathbf{B}_p \boldsymbol{\Psi}_s \quad (8)$$

where  $\boldsymbol{\Psi}_p = [\mathbf{p}, \mathbf{v}, \mathbf{a}, \mathbf{j}]^\top$ ,  $\boldsymbol{\Psi}_s = [\mathbf{s}]^\top$ , and

$$\mathbf{A}_p = \begin{bmatrix} \mathbf{0}_{9 \times 3} & \mathbf{I}_9 \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 9} \end{bmatrix}, \mathbf{B}_p = \begin{bmatrix} \mathbf{0}_{9 \times 3} \\ \mathbf{I}_3 \end{bmatrix}.$$

To optimize the heading of the quadrotor, the flatness system of the yaw can be constructed as:

$$[\dot{\psi}, \ddot{\psi}, \dddot{\psi}]^\top = \mathbf{A}_\psi [\psi, \dot{\psi}, \ddot{\psi}]^\top + \mathbf{B}_\psi \dddot{\psi}, \quad (9)$$

where

$$\mathbf{A}_\psi = \begin{bmatrix} \mathbf{0}_{2 \times 1} & \mathbf{I}_2 \\ 0 & \mathbf{0}_{1 \times 2} \end{bmatrix}, \mathbf{B}_\psi = \begin{bmatrix} \mathbf{0}_{2 \times 1} \\ 1 \end{bmatrix}.$$

We note the flatness system with the augmentation of the two systems as

$$\dot{\boldsymbol{\Psi}}_x = \mathbf{F} (\boldsymbol{\Psi}_x, \boldsymbol{\Psi}_u) = \begin{bmatrix} \mathbf{A}_p & \\ & \mathbf{A}_\psi \end{bmatrix} \boldsymbol{\Psi}_x + \begin{bmatrix} \mathbf{B}_p & \\ & \mathbf{B}_\psi \end{bmatrix} \boldsymbol{\Psi}_u \quad (10)$$

in which  $\Psi_x = [\Psi_p \ \psi \ \dot{\psi} \ \ddot{\psi}]^\top$ ,  $\Psi_u = [\Psi_s \ \ddot{\psi}]^\top$ . With the two linear systems established, one can restore the 13 (with quaternion) or 12 (with Euler angles) dimensional state of the quadrotor system with flatness mapping.

Discretization of the system requires a timestep  $h$ . Here,  $h$  at each discrete point is defined as an extra decision variable, where minimizing  $h$  at each discrete point brings the trajectory to a time-optimal fashion. For OCP setup, the explicit 4<sup>th</sup> order Runge-Kutta method is applied for discretization and a box constraint is applied to  $h_k$  to prevent infeasible integration, we note the discrete flatness system as:

$$\Psi_{x_{k+1}} = F_{RK4}(\Psi_{x_k}, \Psi_{u_k}, h_k). \quad (11)$$

A sequence of position waypoints  $W = \{w_1, \dots, w_m\}$  is given and the waypoint constraints are activated only on specified discrete nodes  $M$ . These indices could be determined according to the distances between waypoints. Waypoint constraints are relaxed and become box constraints with tolerance  $\delta$ . The time-optimal OCP can be formulated as a nonlinear programming (NLP) using direct multiple shooting:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \sum_{k=1}^N h_k^2 \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{X}) = \mathbf{0} \\ & \mathbf{g}(\mathbf{X}) \leq \mathbf{0} \\ & \mathbf{u}_{lb} \leq \boldsymbol{\pi}_0(\Psi_x) \leq \mathbf{u}_{ub} \\ & 0 \leq h_k \leq \bar{h}, k = 1, \dots, N \\ & -\delta \leq [\mathbf{1}_{3 \times 3} \ \mathbf{0}_{3 \times 12}] \Psi_{x_k} - \mathbf{w}_k \leq \delta, k \in M \end{aligned} \quad (12)$$

where:

$$\mathbf{X} = [\Psi_{x_1}, \Psi_{u_1}, h_1, \dots, \Psi_{x_N}, \Psi_{u_N}, h_N]^\top \quad (13)$$

$$\mathbf{c}(\mathbf{X}) = \begin{bmatrix} \Psi_{x_1} - F_{RK4}(\Psi_{x_0}, \Psi_{u_0}, h_1) \\ \vdots \\ \Psi_{x_N} - F_{RK4}(\Psi_{x_{N-1}}, \Psi_{u_{N-1}}, h_{N-1}) \end{bmatrix} = \mathbf{0} \quad (14)$$

where  $\mathbf{g}(\mathbf{X})$  is defined for obstacle-avoidance.  $\boldsymbol{\pi}_0$  refers to the mapping from the flat output to the motor thrusts.

NLP (12) can be solved using the interior point method, which is done using Ipopt and CasADi toolkit [24] in this letter. In waypoint trajectory optimizations,  $h_k$  can be set to be the same in the segment between waypoints, where the OCP performs more like a polynomial-based method but without waypoint time collocations. This property makes it more flexible than polynomial-based approaches while still providing high-order derivatives of flat outputs, and considering motor thrust constraints as methods with full-state dynamics do.

This method might produce aggressive trajectories that DFBC is not able to track them accurately, even failing to stay stable if control inputs are saturated. Similar issues could be found using other methods since the closed-loop performance is not considered. Next, we show how such issues can be solved using closed-loop modeling and trajectory refinement.

## B. Closed-loop Quadrotor Model

A closed-loop dynamic  $\phi$  is obtained by combining the DFBC policy  $\pi$  with the quadrotor dynamics  $\mathbf{f}$ :

$$\dot{x} = \mathbf{f}(x, \pi(x, \Psi_x)) = \phi(x, \Psi_x). \quad (15)$$

where  $\mathbf{u} = \pi(x, \Psi_x)$  refers to the inputs of the open-loop system, i.e., the motor thrusts generated by the DFBC policy  $\pi$ . Input saturations are avoided using the constraint:

$$\mathbf{u}_{lb} \leq \pi(x, \Psi_x) \leq \mathbf{u}_{ub}. \quad (16)$$

Enabling the constraint allows us to get access to the real dynamic feasibility of trajectory tracking.

The singularity at 90 degrees of pitch could be found in some controllers [15], [16]. To avoid such singularity, state constraints on attitude are constructed as ( $\epsilon$  is sufficiently small for the approximation of an open set):

$$-\pi/2 + \epsilon \leq [0 \ 1 \ 0] \Theta \leq \pi/2 - \epsilon. \quad (17)$$

$\Theta$  refers to the Euler angles. Such a setting guarantees the safe usage of DFBC policy while normal constraints without closed-loop consideration would not.

## C. OCP Formulation of TRACE

The key idea of trajectory refinement is to re-scale the trajectory progress by optimizing a scale multiplier  $\alpha$ . By introducing such multipliers, the original trajectory timestep is refined as  $h \leftarrow h/\alpha$ , where some trajectory segments can be relaxed for higher tracking accuracy. Correspondingly, the variation of flat output  $\Psi_x \in \mathbb{R}_{15 \times 1}$  could be given by the chain rule:

$$\Psi_x(\alpha) = [\mathbf{1}_{3 \times 1} \ \alpha \mathbf{1}_{3 \times 1} \ \alpha^2 \mathbf{1}_{3 \times 1} \ \alpha^3 \mathbf{1}_{3 \times 1} \ 1 \ \alpha^2] \Psi_x. \quad (18)$$

The chain rule indicates that the continuous trajectory time  $t$  is modified as  $t \leftarrow \alpha t$ , and  $\alpha$  gets powered with derivatives orders.  $\alpha \leq 1$  refers to trajectory relaxation where tracking accuracy could be recovered, as illustrated in Fig. 3.

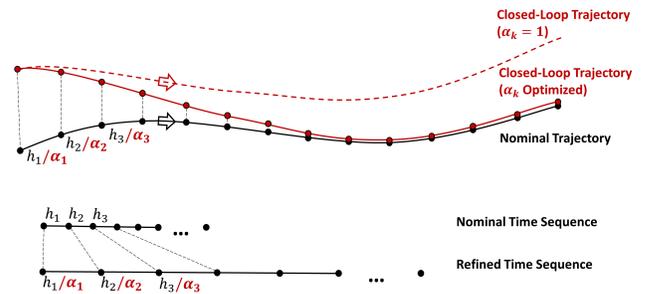


Fig. 3. The rationale of the post-optimization process. A scale multiplier is divided by the original timestep to reduce the rate of trajectory progress.

The process-scaling scheme is adapted to the closed-loop dynamic via discretization using the explicit 4<sup>th</sup> order Runge-Kutta method, where  $h_k$  and  $\Psi_{x_k}$  are pre-computed trajectory parameters:

$$\mathbf{x}_{k+1} = \phi_{RK4}(\mathbf{x}_k, \Psi_x(\alpha_k), h_k/\alpha_k). \quad (19)$$

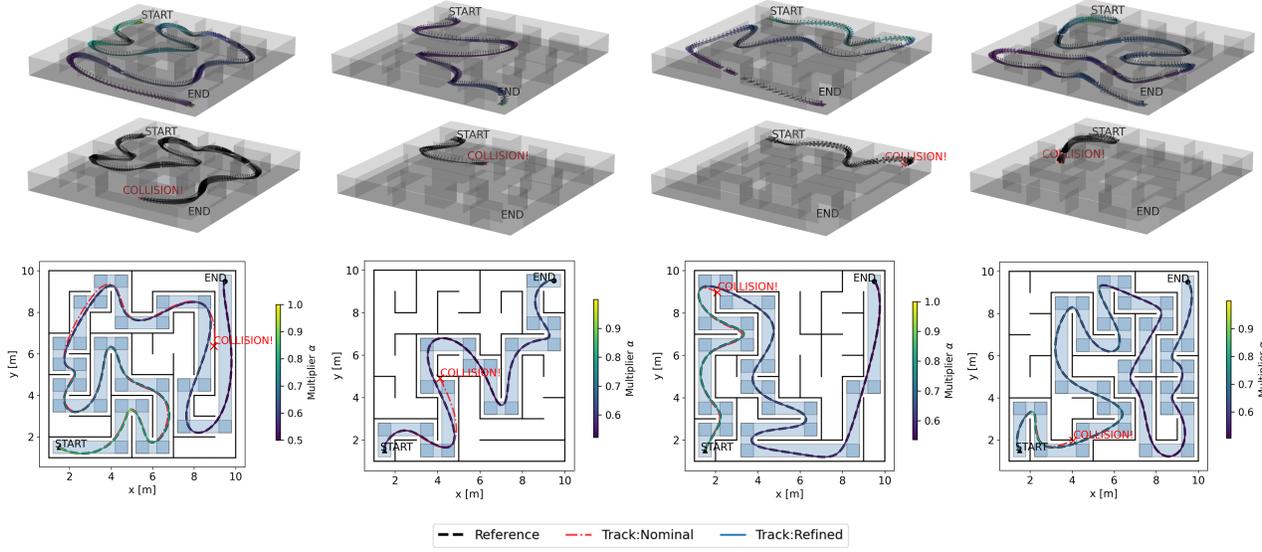


Fig. 4. Simulation scenario: Time-optimal flight in narrow corridors. Four random mazes sized  $10 \times 10$  with safe flight corridors as the union of blue rectangles areas. Colorations illustrate the variation in the multiplier of the refined trajectories. Tracking trajectories without refinement is visualized as red dash-dot lines with collisions (highlighted as a red cross) taking in place. Each 3D plot shows the attitude and the position of the quadrotor motion in tracking both original (black) and refined trajectories (colored).

The reference trajectory varies during the tracking progress, therefore, such a dynamic system is time-varying with nonlinearity. The closed-loop system takes the multiplier as input, with the states as same as the open-loop one. In the objective function, we only compare the tracking error of position and velocity in the earth-fixed frame, where the quadrotor states and the flat outputs share the same definition. To match the sizes of states and flat outputs, we remove the extra terms by defining:  $P_1 = [I_6 \ 0_{6 \times 7}]$  and  $P_2 = [I_6 \ 0_{6 \times 9}]$ , Therefore, the cost for tracking error is established:  $L_k(x_k, \Psi_x(\alpha_k)) = (P_1 x_k - P_2 \Psi_x(\alpha_k))^T Q (P_1 x_k - P_2 \Psi_x(\alpha_k))$ , where  $Q = \text{diag}\{q_l\}$ . The time-varying OCP for post-optimization could then be constructed as an NLP using direct multiple shooting formulation.

$$\begin{aligned} \min_{\mathbf{X}} \quad & \sum_{k=1}^N L_k(x_k, \Psi_x(\alpha_k)) + r(\alpha_k - 1)^2 \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{X}) = \mathbf{0} \\ & lb_\alpha \leq \alpha_k \leq ub_\alpha, \\ & \mathbf{u}_{lb} \leq \boldsymbol{\pi}(x_k, \Psi_{x_k}) \leq \mathbf{u}_{ub}. \end{aligned} \quad (20)$$

where

$$\mathbf{X} = [x_1, \alpha_1, \dots, x_N, \alpha_N]^T \quad (21)$$

$$\Psi_x(\alpha_k) = [1_{3 \times 1} \ \alpha 1_{3 \times 1} \ \alpha^2 1_{3 \times 1} \ \alpha^3 1_{3 \times 1} \ 1 \ \alpha \ \alpha^2] \Psi_{x_k} \quad (22)$$

and

$$\mathbf{c}(\mathbf{X}) = \begin{bmatrix} x_1 - \phi_{RK4}(x_0, \alpha_1, 0) \\ \vdots \\ x_N - \phi_{RK4}(x_{N-1}, \alpha_{N-1}, N-1) \end{bmatrix} = \mathbf{0}. \quad (23)$$

If attitude singularity occurs, one can apply

$$-\pi/2 + \epsilon \leq [0 \ 1 \ 0] \boldsymbol{\Theta} \leq \pi/2 - \epsilon. \quad (24)$$

The OCP optimizes a dual-objective problem, as targets of high accuracy and progress maintenance (accuracy and aggressiveness for time-optimal cases) are weighted by  $q_l$  and  $r$ . The OCP applies the original closed-loop state trajectory together with the multiplier  $\alpha_k = 1$  as an initial guess of the solution. We again use the Ipopt and CasADi toolkit [24] to obtain the solution.

For parameter tuning, one may utilize warm-starting techniques, i.e. construct a sequence of problems with different  $q_l$  setup and apply the previous result as the initial guess of the next problem. Starting with a relatively low penalty on tracking accuracy, one could increase it until the desired trade-off between aggressiveness and accuracy is achieved.

Note that the method does not even require a stable closed-loop system since the integration has a limited horizon, but numerical tests show that a closed-loop system with poor tracking performance can sometimes destroy the convergence of TRACE. A general controller with light-tuning work would be fine for TRACE to work. Since the multiplier is one-dimensional and serves as the input of the closed-loop dynamics, TRACE has fewer decision variables compared with an open-loop full-state trajectory optimization problem formulated using direct multiple shooting, and they share a similar computational complexity.

Trajectories provided by conventional trajectory optimization methods could be refined by TRACE since it does not rely on continuous-time parameterization of the nominal trajectory. Yet a conversion from the quadrotor states to the flat outputs is required.

## V. SIMULATIONS

In this section, two scenarios: time-optimal flights in narrow corridors and a time-optimal waypoint flight, are presented to illustrate the effectiveness of TRACE in enhancing the tracking performance and avoiding input saturations.

TABLE I  
PARAMETER SETUP IN (12) AND (20).

Parameters	Value	Parameters	Value
$q$	100.0	$\mathbf{u}_{ub}$	$\mathbf{4}_{4 \times 1}$
$r$	1.0	$\mathbf{u}_{lb}$	$\mathbf{0}_{4 \times 1}$
$N$	500	$lb_{\alpha}$	0.5
$\delta$	$\mathbf{0.05}_{3 \times 1}$	$ub_{\alpha}$	1.0
$\bar{h}$	0.05	$\epsilon$	$1e-2$

### A. Time-Optimal Flight in Narrow Corridors

This scenario creates random mazes with narrow corridors that require accurate trajectory tracking. We construct safe flight corridors inside the maze as obstacle-avoidance constraints. As shown in Fig. 4, four random mazes of size  $10 \times 10$  are generated, each of them is associated with safe flight corridors with margin (highlighted in blue) as box constraints of states to compute collision-free time-optimal trajectories by solving (12). Such safe flight corridors are built based on the shortest path of the maze. Subsequently, the trajectories are refined using TRACE for safe and accurate tracking. The parameters of (12) and (20) OCP setups are given in table I. For convenience, we use  $q$  to scale the cost parameter  $\mathbf{Q}$  in (20) by making  $\mathbf{Q} = q \times \text{diag}\{\mathbf{1}_{6 \times 1}\}$ . The nominal quadrotor dynamics and controller parameters are consistent with our experiment platform and will be introduced in the next section.

Despite the success of time-optimal trajectory optimization in mazes with narrow corridors, the current DFBC controller fails to provide safe and accurate tracking performance as quadrotors collide with the walls. After the refinement of the original trajectories,  $\alpha < 1$  could be found at aggressive corners inside the maze, where the closed-loop model outputs an increase in its tracking error and reduced by solving the OCP (20). All four cases show the relaxation in trajectory progress in exchange for extraordinary tracking accuracy with control inputs bounded.

### B. Time-Optimal Waypoint Flight

A time-optimal waypoint flight is carried out for parameter scanning and analysis. From Fig. 5, direct trajectory tracking leads to the divergence of tracking control as the inputs are saturated and fail to provide effective feedback. After the post-optimization, the multipliers are optimized to average around 0.7 along most of the trajectory. The relaxation brings back the capability for tracking error elimination.

Next, the influence of parameter setup is studied since the OCP in (20) is in a dual-objective fashion. QR ratio  $q/r$  represents the weight on tracking error elimination and trajectory progress. A higher weight on  $q$  outputs a more accurate but slower trajectory. The Euclidean norm of positional tracking error  $\|e_p\|_2$  is applied to characterize the

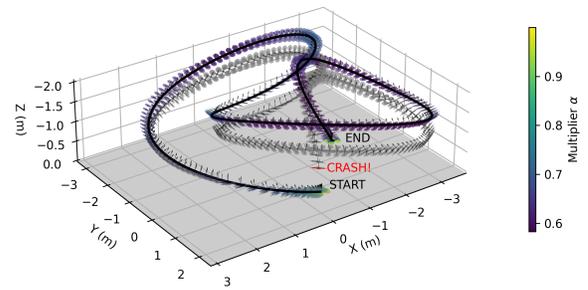


Fig. 5. Simulation scenario: Time-optimal waypoint flight. The gray quadrotor motions refer to direct trajectory tracking while colored ones represent the trajectory tracking after refinement. Colorations indicate the magnitude of the multiplier along the nominal trajectory.

tracking performance. As demonstrated in Fig. 6, the tracking error starts to level off as  $q/r$  reaches beyond a certain value while the trajectory time still rises before it reaches a plateau. These could be varied throughout different tasks and the warm-starting techniques introduced in section IV-C could be utilized for tuning  $q/r$ .

Furthermore, the effect of the controller gain is analyzed. Based on the assumption of time-scale separation, our controller gain of the translational loop in the simulations stays below or slightly reaching over  $1/10$  of that of the attitude loop, where the latter is determined by real-world practice. Higher controller gain in simulations might work but would enlarge the uncertainty originated from localization, filtering and disturbances in real-world flights. Without the constraints on input saturations, the DFBC fails to stay stable during trajectory tracking. Therefore, in the analysis, the saturations are removed during simulations. After that, the  $\|e_p\|_2$  of each group with different multiples (ranging from 0.2-1.4) of the  $\mathbf{K}_p$ , are recorded. Reaching over 1.4 would lead to an unreasonable value for holding the time-scale separation assumption, leading to stability issues. From the result in Fig. 6, although increasing the gain in a rational range can bring down the tracking error, refining trajectories still significantly improves the accuracy.

## VI. REAL-WORLD EXPERIMENTS

Real-world experiments are constructed for further validation of the proposed methods. Our experiment platform comes with  $m = 0.83kg$  and  $\mathbf{J} = \text{diag}\{3e-3 \ 3e-3 \ 3 \ 4e-3\}kg \cdot m^2$ . The controller with its parameters shown in table II runs onboard with an STM32-F7 microcontroller. An STM32-F4 microcontroller is set up for data reception and transmission with an ultra-wideband (UWB) telemetry module. Localization of the quadrotor UAV is done using a motion-capturing system running on a ground station.

First, we show the effect of the input saturation constraint in agile flight. Two trajectories of the same  $q/r = 1$  are post-optimized with and without the input saturation constraint. As shown in Fig. 1, tracking reference trajectory with the original open-loop input constraint is unsafe under the existence of tracking errors. However, with the input saturation

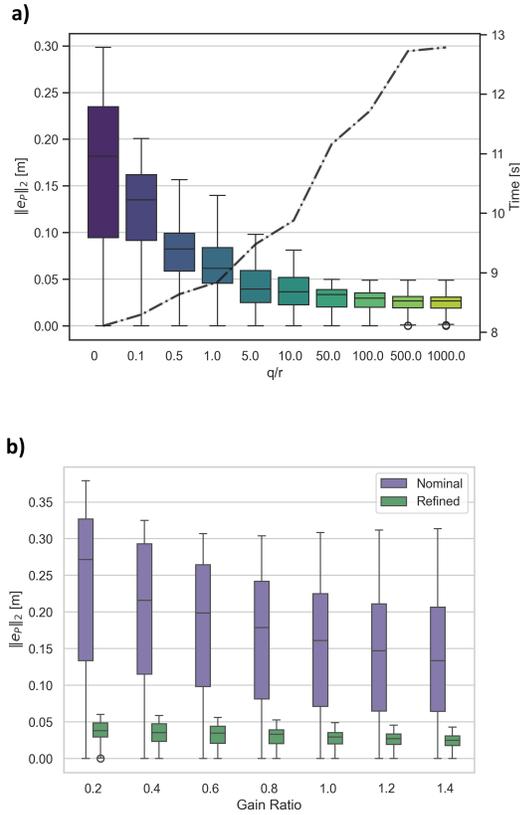


Fig. 6. Parameter scanning and analysis. a) Study on the dual-objective parameter setup of TRACE, i.e.,  $q/r$  ratio, b) Study on the influence of controller gain to the optimization result.

constraint activated, the nominal trajectory is further relaxed for safe tracking as motor thrusts maintain below their limits.

TABLE II  
CONTROLLER PARAMETER SETUP.

Parameters	Value
$K_p$	$diag\{[1.0 \ 1.0 \ 0.7]\}$
$K_v$	$diag\{[4.0 \ 4.0 \ 2.8]\}$
$K_a$	$diag\{[10.0 \ 10.0 \ 4.0]\}$
$K_r$	$diag\{[0.25 \ 0.25 \ 0.1]\}$
$u_{lb}$	$[0.0 \ 0.0 \ 0.0 \ 0.0]$
$u_{ub}$	$[4.0 \ 4.0 \ 4.0 \ 4.0]$

Next, four cases with varied  $q/r$  are carried out in the experiment, the actual trajectories with references are plotted in Fig. 7, each closed-loop trajectory is colored the magnitude of the multipliers. Table III presents the results of the real-world experiments. The tracking accuracy is related to the mean square error (MSE) of the reference position and the actual position. Specifically, the last case is prepared with  $q = 0$  and the input saturation constraint is activated, but it results in unstable trajectory tracking. An explanation could be made that there might exist model mismatches and disturbances in real-world flights.

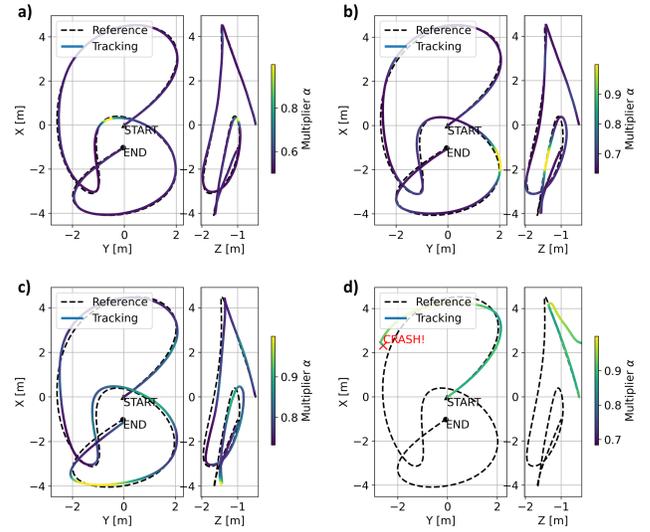


Fig. 7. Four cases of trajectory tracking experiments with different  $q/r$ . a)  $q/r = 100$ , b)  $q/r = 10$ , c)  $q/r = 1$ , d)  $q/r = 0$ .

TABLE III  
EXPERIMENTAL CASES AND RESULTS.

Case	Mode	Max.Vel	Max.Acc	Time	MSE
Time-Optimal	Nominal	7.9759	16.7547	7.6496	/
	Refined	N/A			
$q/r = 0$	Nominal	7.3640	14.8231	8.3507	/
	Refined	Crashed			
$q/r = 1$	Nominal	6.0781	10.4445	9.4216	/
	Refined	6.1937	9.6133		<b>0.1126</b>
$q/r = 10$	Nominal	5.2372	7.6709	11.0265	/
	Refined	5.2621	7.4401		<b>0.0115</b>
$q/r = 100$	Nominal	4.1848	5.4149	13.7127	/
	Refined	4.2793	5.7026		<b>0.0060</b>

## VII. CONCLUSION

In this paper, the methodologies of generating time-optimal trajectories and trajectory refinement are carried out. Flatness-based time-optimal trajectory optimizations with narrow corridors and waypoints are conducted in simulations and experiments. TRACE refines the pre-computed trajectories, making the trajectory-tracking process safer and more accurate. Some limitations still exist in our approach. Our method ignores the influence of model mismatches as well as other kinds of uncertainties. This has led to a failed case in real-world experiments. Online application of such method can be further developed, which allows real-time adjustment of multipliers to improve robustness. Future work might focus on extending the proposed idea into robust trajectory refinement to handle the uncertainties, as well as direct trajectory optimization in the presence of multiple bounded disturbances.

## REFERENCES

- [1] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Learning high-speed flight in the wild," vol. 6, no. 59, p. eabg5810. [Online]. Available: <https://www.science.org/doi/10.1126/scirobotics.abg5810>

- [2] Y. Song, K. Shi, R. Penicka, and D. Scaramuzza, "Learning Perception-Aware Agile Flight in Cluttered Environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1989–1995. [Online]. Available: <https://ieeexplore.ieee.org/document/10160563>
- [3] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," vol. 620, no. 7976, pp. 982–987. [Online]. Available: <https://www.nature.com/articles/s41586-023-06419-4>
- [4] Y. Song, A. Romero, M. Müller, V. Koltun, and D. Scaramuzza, "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning," vol. 8, no. 82, p. eadg1462. [Online]. Available: <https://www.science.org/doi/10.1126/scirobotics.adg1462>
- [5] G. Ryou, E. Tal, and S. Karaman, "Multi-fidelity black-box optimization for time-optimal quadrotor maneuvers," vol. 40, no. 12–14, pp. 1352–1369. [Online]. Available: <https://doi.org/10.1177/02783649211033317>
- [6] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, "Model Predictive Contouring Control for Time-Optimal Quadrotor Flight." [Online]. Available: <http://arxiv.org/abs/2108.13205>
- [7] S. Sun, A. Romero, P. Foehn, E. Kaufmann, and D. Scaramuzza, "A Comparative Study of Nonlinear MPC and Differential-Flatness-Based Control for Quadrotor Agile Flight." [Online]. Available: <http://arxiv.org/abs/2109.01365>
- [8] A. Loquercio, A. Saviolo, and D. Scaramuzza, "AutoTune: Controller Tuning for High-Speed Flight," vol. 7, no. 2, pp. 4432–4439. [Online]. Available: <https://ieeexplore.ieee.org/document/9696365>
- [9] J. Jia, K. Guo, X. Yu, W. Zhao, and L. Guo, "Accurate high-maneuvering trajectory tracking for quadrotors: A drag utilization method," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6966–6973, 2022.
- [10] A. Romero, R. Penicka, and D. Scaramuzza, "Time-Optimal Online Replanning for Agile Quadrotor Flight," vol. 7, no. 3, pp. 7730–7737. [Online]. Available: <http://arxiv.org/abs/2203.09839>
- [11] P. Foehn, A. Romero, and D. Scaramuzza, "Time-optimal planning for quadrotor waypoint flight," vol. 6, no. 56, p. eabh1221. [Online]. Available: <https://www.science.org/doi/10.1126/scirobotics.abh1221>
- [12] R. Penicka and D. Scaramuzza, "Minimum-Time Quadrotor Waypoint Flight in Cluttered Environments," vol. 7, no. 2, pp. 5719–5726. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9721033>
- [13] W. Sun, G. Tang, and K. Hauser, "Fast UAV Trajectory Optimization using Bilevel Optimization with Analytical Gradients." [Online]. Available: <http://arxiv.org/abs/1811.10753>
- [14] F. Gao, W. Wu, J. Pan, B. Zhou, and S. Shen, "Optimal Time Allocation for Quadrotor Trajectory Generation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4715–4722. [Online]. Available: <https://ieeexplore.ieee.org/document/8593579>
- [15] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2520–2525. [Online]. Available: <https://ieeexplore.ieee.org/document/5980409>
- [16] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential Flatness of Quadrotor Dynamics Subject to Rotor Drag for Accurate Tracking of High-Speed Trajectories," vol. 3, no. 2, pp. 620–626. [Online]. Available: <https://ieeexplore.ieee.org/document/8118153>
- [17] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 5420–5425. [Online]. Available: <https://ieeexplore.ieee.org/document/5717652>
- [18] E. Tal and S. Karaman, "Accurate Tracking of Aggressive Quadrotor Trajectories Using Incremental Nonlinear Dynamic Inversion and Differential Flatness," vol. 29, no. 3, pp. 1203–1218. [Online]. Available: <https://ieeexplore.ieee.org/document/9121690>
- [19] Z. Zhou, G. Wang, J. Sun, J. Wang, and J. Chen, "Efficient and Robust Time-Optimal Trajectory Planning and Control for Agile Quadrotor Flight." [Online]. Available: <http://arxiv.org/abs/2305.02772>
- [20] R. Quirynen, M. Vukob, M. Zanon, and M. Diehl, "Autogenerating Microsecond Solvers for Nonlinear MPC: a Tutorial Using ACADO Integrators," *Optimal Control Applications and Methods*, 2014.
- [21] R. Verschuere, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados: a modular open-source framework for fast embedded optimal control," 2020.
- [22] M. Watterson and V. Kumar, "Control of Quadrotors Using the Hopf Fibration on SO(3)," in *Robotics Research*, ser. Springer Proceedings in Advanced Robotics, N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti, Eds. Springer International Publishing, pp. 199–215.
- [23] B. Morrell, M. Rigter, G. Merewether, R. Reid, R. Thakker, T. Tzanetos, V. Rajur, and G. Chamitoff, "Differential Flatness Transformations for Aggressive Quadrotor Flight," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5204–5210. [Online]. Available: <https://ieeexplore.ieee.org/document/8460838>
- [24] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.