IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Efficient and Robust Time-Optimal Trajectory Planning and Control for Agile Quadrotor Flight

Ziyu Zhou, Gang Wang, Jian Sun, *Senior Member, IEEE*, Jikai Wang, and Jie Chen, *Fellow, IEEE*

***Abstract*—Agile quadrotor flight relies on rapidly planning and accurately tracking time-optimal trajectories, a technology critical to their application in the wild. However, the computational burden of computing time-optimal trajectories based on the full quadrotor dynamics (typically on the order of minutes or even hours) can hinder its ability to respond quickly to changing scenarios. Additionally, modeling errors and external disturbances can lead to deviations from the desired trajectory during tracking in real time. This letter proposes a novel approach to computing time-optimal trajectories, by fixing the nodes with waypoint constraints and adopting separate sampling intervals for trajectories between waypoints, which significantly accelerates trajectory planning. Furthermore, the planned paths are tracked via a time-adaptive model predictive control scheme whose allocated tracking time can be adaptively adjusted on-the-fly, therefore enhancing the tracking accuracy and robustness. We evaluate our approach through simulations and experimentally validate its performance in dynamic waypoint scenarios for time-optimal trajectory replanning and trajectory tracking.**

***Index Terms*—Optimization and optimal Control, integrated planning and control, control architectures and programming.**

Supplementary Material

Code: https://github.com/BIT-KAUIS/Fast-fly
Video: https://youtu.be/E6QVHWcvB6E

I. Introduction

THE use of quadrotors in both academic and commercial fields has garnered significant attention due to their excellent maneuverability and versatility [1], [2]. Quadrotors have numerous applications, including aerial photography and videography, search and rescue operations, agricultural monitoring, package delivery, and military reconnaissance, to just name a few. These applications require the quadrotor to visit multiple waypoints within a limited flight duration due to onboard battery capacity, which necessitates time-optimal flight to improve the efficiency and task completion [3].

Fig. 1. Quadrotor racing through circles at a maximum speed of 10.6 m/s. The circles have a radius of 0.8 m, with four stationary circles and one moving circle held by a person. The time-optimal trajectory is quickly replanned when the position of the moving circle changes. Our proposed methods enable the quadrotor to efficiently replan and accurately track the updated trajectory in a fast and precise manner.

Trajectory planning for time-optimal flight has always been a challenging problem [4]. Unlike the point-mass model, which has a closed-form solution with bang-bang acceleration and can quickly obtain the time-optimal trajectory by sampling at waypoints [5], the quadrotor model is underactuated. The thrust generated by the propellers can only act along the z-axis of the body, and both the thrust magnitude and direction need to be adjusted simultaneously to control the movement of the quadrotors [6]. Therefore, the coupling of acceleration and angular velocity makes the time-optimal trajectory of the quadrotors even more complex.

To address this challenge, a common approach uses nonlinear numerical optimization to plan the trajectory, by discretizing it in time and treating the full quadrotor dynamics as dynamical constraints [7]–[9]. However, for multi-waypoint time-optimal flight, each waypoint must be allocated as a constraint to a specific node on the trajectory. The optimal time of passing through each waypoint is unknown, making it difficult to determine which node a waypoint should be assigned to. To tackle this issue, the work [4] introduced the so-called progress measure variables to represent the completion of a waypoint (progress) and used complementary progress constraints (CPC) to allocate the waypoint constraints, thus ensuring generation of time-optimal trajectories that satisfy the full dynamics. However, the introduction of progress measure variables and CPC renders the entire optimization problem

Manuscript received: May 2, 2023; Revised August 22, 2023; Accepted September 24, 2023.

This paper was recommended for publication by Editor Hanna Kurniawati upon evaluation of the Associate Editor and Reviewers' comments. This work was partially supported by the National Natural Science Foundation of China under Grants 62173034, 61925303, 62088101, and by the Chongqing Natural Science Foundation under Grant 2021ZX4100027. (*Corresponding author: Gang Wang.*)

Ziyu Zhou, Gang Wang, Jian Sun, and Jikai Wang are with the National Key Lab of Autonomous Intelligent Unmanned Systems, Beijing Institute of Technology, Beijing 100081, China, and the Beijing Institute of Technology Chongqing Innovation Center, Chongqing 401120, China (e-mail: ziyuzhou@bit.edu.cn, gangwang@bit.edu.cn; sunjian@bit.edu.cn; jikaiwang@bit.edu.cn).

Jie Chen is with the National Key Lab of Autonomous Intelligent Unmanned Systems, Tongji University, Shanghai 201804, China (e-mail: chenjie@bit.edu.cn).

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

complicated and numerically more challenging, whose solving time typically ranges from minutes to even hours. Although the waypoint constraints can be relaxed by a certain tolerance, the accuracy and optimality of generated trajectories is sacrificed.

For trajectory tracking, the two leading frameworks are the nonlinear model predictive controller (NMPC) and the differential-flatness-based controller. Due to advances in computer hardware, the NMPC approach has demonstrated superior tracking accuracy and robustness [10]. However, when it comes to time-optimal trajectory planning, the quadrotor must operate at its motion limits, and dynamical model mismatch and unknown external disturbances can lead to trajectory tracking failure during extreme flight [11], making robust trajectory tracking a challenging task. To overcome this problem, [4] proposed to generate a time-optimal trajectory with a slightly lower input upper-bound than the quadrotor's real actuator limit to ensure tracking robustness. Obviously, this approach sacrifices time-optimality.

In this letter, we propose an efficient and robust framework for time-optimal waypoint flight. The framework divides the waypoint flight mission into two layers: the time-optimal path planning under waypoint constraints and the time-adaptive trajectory tracking control. The overall framework is based on the full quadrotor dynamics and nonlinear optimization, which minimizes the flight time while maximizing the quadrotor flight capability. Our contributions are summarized as follows.

- We propose a segmented time-optimal trajectory planning method with a much shorter solving time (seconds versus minutes), higher accuracy, and improved optimality in time than CPC.
- We propose a time-adaptive model predictive control (tMPC) method for tracking planned trajectories achieving a shorter tracking time and higher accuracy while ensuring the tracking robustness.

We implement the proposed time-optimal trajectory replanning method in real-world experiments with dynamic waypoints and validate its planning efficiency and tracking robustness. The source code of our system is publicly available at https://github.com/BIT-KAUIS/Fast-fly.

II. Related Works

A. Time-optimal Trajectory Planning

Trajectory planning has advanced significantly in simulation and experimentation over the past decade. Early research focused primarily on planning collision-free and safe trajectories. Recent studies have focused more on planning smooth, dynamically feasible, and minimum-time trajectories to enable quadrotors to fly more flexibly and quickly [12]–[15]. Optimization-based trajectory planning methods have represented trajectories as time sequences of the quadrotor's state and control inputs, considering the objective of minimizing the flight time while complying with the quadrotor dynamics and input constraints [4]. A method by [16] achieved time-minimum flight by maximizing the velocity along the given path using a path parametrization method while considering the translational and rotational dynamics of quadrotors. However, this method only optimizes velocity along the given path and does not optimize the path further. The above trajectory planning methods did not consider the scenario where the quadrotors need to pass through given waypoints, which is common in drone racing. Time-optimal trajectory planning with waypoint constraints requires consideration of the time allocation problem, i.e., assigning waypoints to specific time steps, which is often non-trivial and hard to tackle. The recent work by [4] introduced the CPC method, considering the full-state dynamics constraints of the quadrotor and achieving truly time-optimal trajectory planning with waypoint constraints. However, the solution of CPC requires a large amount of computation, making it difficult to achieve real-time replanning. To address this issue, [5] performs real-time replanning of time-optimal trajectories based on a point-mass dynamics model and employs model predictive contouring control (MPCC) with full quadrotor dynamics [17] to track the trajectory. This method has both replanning capabilities and exceeds previous methods in time optimality.

B. Trajectory Tracking Control of Quadrotors

The differential-flatness based controller (DFBC) has shown significant improvement in tracking performance during high-speed flight [18], [19]. Quadrotors have been proven to be differentially flat [18], [20]; that is, once a time-parameterized 3D path with the heading angle is given, the attitude, angular velocity, and acceleration of quadrotors can be readily computed. These signals can be sent to the lower-level controller to form a feedforward term, which can address the issue of model mismatches and unknown external disturbances. The work [19] developed the differential control method by cascading an INDI controller after the DFBC. The INDI controller helps enhance robustness against external aerodynamic disturbances and achieves a maximum flight speed of nearly 13 m/s and an acceleration exceeding 2g on a real quadrotor.

Another control algorithm is MPC which predicts the future states for multiple time steps and computes the associated control inputs. Thanks to advances of hardware and nonlinear optimization solvers [21]–[25], NMPC based on full-state dynamics can meet real-time requirements. The work [26] used each rotor thrust as a control input, and [27] used the solved internal state as the reference input which is sent to the lower-level controller. NMPC can fully exploit the capabilities of drones and account for the actuator saturation constraints.

In a recent comparative study of NMPC and DFBC by [10], it was shown that although DFBC is more computationally efficient and easier to implement, NMPC performs better in high-speed flight. MPC can only track trajectories that satisfy dynamic feasibility. To this end, another predictive control method, namely MPCC [28], maximizes the trajectory tracking progress while considering the tracking accuracy. This method can not only track trajectories that are not dynamically feasible but also achieve approximately time-optimal flight [17].

III. Methodology

A. Quadrotor Dynamics

We represent vectors and matrices using bold lowercase and bold uppercase letters, respectively, throughout this letter. We

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

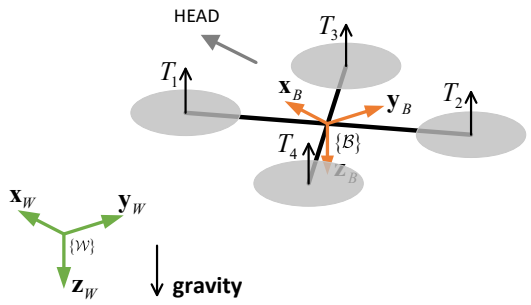

Fig. 2. A diagram showing the world frame and body frame of a quadrotor.

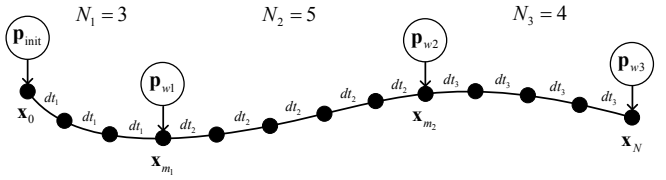

Fig. 3. Demonstration of the proposed time-optimal flight method with $N_w = 3$ waypoints. The trajectory is divided into three segments, each consisting of a pre-assigned number of discrete points, with $N_1 = 3$, $N_2 = 5$, and $N_3 = 3$. The sampling times for the three segments are denoted by dt_1, dt_2, and dt_3, respectively. The three waypoint constraints are allocated to nodes $m_1 = 3$ and $m_2 = 8$, as well as the last node N.

define the world frame, denoted by $\mathcal{W} : \{\mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W\}$, with $\mathbf{z}_W$ pointing downward and aligned with gravity. The body frame, denoted by $\mathcal{B} : \{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$, has $\mathbf{x}_B$ pointing forward and $\mathbf{z}_B$ pointing downward, opposite to the collective thrust. The body frame's origin is attached to the quadrotor's center of mass (CoM), as depicted in Fig. 2.

We consider the quadrotor dynamical model used in, e.g., [4], [17], whose dynamics are given as follows

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= g\mathbf{z}_W - c\mathbf{z}_B - \mathbf{R}\mathbf{K}\mathbf{R}^\top\mathbf{v} \\ \dot{\mathbf{q}} &= \tfrac{1}{2}\mathbf{q} \odot \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \\ \dot{\boldsymbol{\omega}} &= \mathbf{J}^{-1}(\boldsymbol{\tau} - \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}). \end{aligned} \tag{1}$$

Here, $\mathbf{p}$ and $\mathbf{v}$ denote the position and velocity of the quadrotor's CoM, respectively. The symbol $\mathbf{q} \in \mathbb{SO}(3)$ is the unit quaternion representing the rotation from $\mathcal{W}$ to $\mathcal{B}$, and $\mathbf{R}$ is the corresponding rotation matrix parameterized by $\mathbf{q}$. The symbol $\boldsymbol{\omega}$ is the angular velocity of $\mathcal{B}$ with respect to $\mathcal{W}$, c and $\boldsymbol{\tau}$ are the mass-normalized collective thrust and the resultant torque generated by rotors. The symbol $\mathbf{K} = \text{diag}(k_x, k_y, k_z)$ is the mass-normalized rotor-drag coefficients, and $\mathbf{J}$ is the quadrotor's inertia matrix.

For the configuration in Fig. 2, the thrust T_i at each rotor $i \in \{1, 2, 3, 4\}$ can be used to decompose c and $\boldsymbol{\tau}$ as follows

$$c = \frac{1}{m}(T_1 + T_2 + T_3 + T4) \tag{2}$$

$$\boldsymbol{\tau} = \begin{bmatrix} \frac{l}{\sqrt{2}}(T_1 + \mathrm{T}_4 - T_2 - T_3) \\ \frac{l}{\sqrt{2}}(T_1 + T_3 - T_2 - T_4) \\ c_\tau(T_3 + T_4 - T_1 - T_2) \end{bmatrix} \tag{3}$$

where m and l represent the quadrotor's mass and arm length respectively and c_τ is the rotor's torque constant. Additionally, the thrust values T_i must satisfy the following constraints

$$0 \leqslant T_{\min} \leqslant T_i \leqslant T_{\max}. \tag{4}$$

B. Time-optimal Trajectory Planner

The objective of a waypoint flight is, given a starting point, to guide the quadrotor through M given waypoints in a specific order. In the general optimization-based planning algorithm, the time required to reach each waypoint is predetermined, and the task of passing through the waypoint is expressed as position constraints of the quadrotor at the corresponding times [29], [30]. If $\{\mathbf{p}_{w_i} \in \mathbb{R}^3\}_{i=1}^{N_w}$ denote the positions of N_w waypoints, the optimization problem for the waypoint flight can be expressed as follows

$$\min_{\mathbf{x}_k, \mathbf{u}_k} \quad J = \sum_{k=0}^{N-1} (\|\mathbf{x}_{k+1}\|_{\mathbf{Q}} + \|\mathbf{u}_k\|_{\mathbf{Q_u}}) \tag{5a}$$

$$\text{s.t.} \quad \|\mathbf{p}_{m_i} - \mathbf{p}_{w_i}\|_2^2 \leqslant \delta_i^2, \quad i = 1, 2, \ldots, N_w \tag{5b}$$

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, dt) \tag{5c}$$

$$\mathbf{x}_{\text{lb}} \leqslant \mathbf{x}_k \leqslant \mathbf{x}_{\text{ub}} \tag{5d}$$

$$\mathbf{u}_{\text{lb}} \leqslant \mathbf{u}_k \leqslant \mathbf{u}_{\text{ub}} \tag{5e}$$

$$\mathbf{x}_0 = \mathbf{x}_{\text{init}}. \tag{5f}$$

where the state vector $\mathbf{x} = [\mathbf{p}^\top, \mathbf{v}^\top, \mathbf{q}^\top, \boldsymbol{\omega}^\top]^\top$, the control input $\mathbf{u} = [T_1, T_2, T_3, T_4]^\top$, and the weight matrix $\mathbf{Q} = \text{diag}\{\mathbf{Q_p}, \mathbf{Q_v}, \mathbf{Q_q}, \mathbf{Q_w}\}$ with $\mathbf{Q_p}$, $\mathbf{Q_v}$, $\mathbf{Q_q}$, $\mathbf{Q_w}$ and $\mathbf{Q_u}$ being usually diagonal and positive definite.

The objective function in (5a) minimizes the control cost and system energy, where $\|\mathbf{x}\|_{\mathbf{A}}$ is defined as $\mathbf{x}^\top\mathbf{A}\mathbf{x}$ for a positive definite matrix $\mathbf{A} = \mathbf{A}^\top$ of suitable dimensions. The waypoint constraints are represented in (5b), where $m_i \in \mathbb{N}$, $\mathbf{p}_{m_i}$ and $\delta_i \geq 0$ denote the time, the position of the quadrotor and the allowable position error, when passing through the i-th waypoint. The dynamics constraint is represented in (5c), which can be obtained from the continuous dynamics in (1) using the Runge-Kutta method with a sampling period of $dt > 0$. The state constraints, input constraints, and initial state constraint are represented in (5d)–(5f), respectively.

For time-optimal planning, time is treated as an optimization variable. To achieve the goal of minimizing the total flight time, the optimization objective function for time-optimal planning only considers minimizing the flight time T, which is not reflected in the objective of minimizing system energy and control cost in (5a).

To determine m_i, the work of [4] introduced complementary progress constraints to optimize the values of m_i, which considerably increases the computation time of the resulting nonlinear optimization problem. In our method, we fix the allocation of the waypoint constraints, i.e., by pre-assigning each m_i an appropriate value, and divides the trajectory into N_w segments using the N_w waypoints, each using a separate sampling interval denoted by $dt_i > 0$. The number of discrete points $N_i \in \mathbb{N}$ for each trajectory segment is pre-assigned based on the distance between adjacent waypoints; see a

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

pictorial illustration in Fig. 3. As a result, the total flight time $\mathcal{T}$ and the allocation m_i's of the waypoint constraints can be obtained as follows

$$\mathcal{T} = \sum_{i=1}^{N_w} N_i \, dt_i \tag{6a}$$

$$m_i = \sum_{j=1}^{i} N_j, \quad \forall i = 1, 2, \ldots, N_w. \tag{6b}$$

Furthermore, for each trajectory segment, we discretize the quadrotor dynamics using the corresponding sampling time dt_i in the following form

$$\mathbf{x}_{k_i+1} = \mathbf{f}(\mathbf{x}_{k_i}, \mathbf{u}_{k_i}, dt_i) \tag{7}$$

where $k_i \in \mathbb{N}$ and $m_{i-1} \leqslant k_i \leqslant m_i$, representing the nodes (i.e., discrete points) in the i-th trajectory segment.

Finally, the proposed time-optimal waypoint trajectory planning problem can be summarized as follows

$$\min_{\mathbf{x}_k, \mathbf{u}_k, dt_i} \quad \mathcal{T} \text{ in (6a)} \tag{8a}$$

$$\text{s.t.} \quad \|\mathbf{p}_{m_i} - \mathbf{p}_{w_i}\|_2^2 \leq \delta_i^2, \text{ with } m_i \text{ in (6b)} \tag{8b}$$

$$\text{Constraint in (7)} \tag{8c}$$

$$\mathbf{x}_{\text{lb}} \leqslant \mathbf{x}_k \leqslant \mathbf{x}_{\text{ub}} \tag{8d}$$

$$\mathbf{u}_{\text{lb}} \leqslant \mathbf{u}_k \leqslant \mathbf{u}_{\text{ub}} \tag{8e}$$

$$\mathbf{x}_0 = \mathbf{x}_{\text{init}}. \tag{8f}$$

Compared to solving the time-optimal flight optimization problem using CPC as presented in [4], our proposed Problem (8) can be efficiently tackled using interior-point methods, given that the allocation of each waypoint constraint remains fixed. However, due to the presence of nonlinear dynamics constraints (8c), Problem (8) inherently becomes nonconvex. When confronted with an absence of a suitable initial solution, directly solving the optimization problem (8) often results in infeasible solutions that breach the waypoint and dynamics constraints (8c). Moreover, even when an optimal solution can be finally reached, it requires a considerable number of iterations and computing time by e.g., the interior-point method. To ensure both the quality of the solution obtained through the interior-point method and a manageable solving time, we introduce a warm-up problem for (8). This warm-up problem integrates the waypoint and dynamics constraints into the objective function via penalty functions $L_w = \sum_{i=1}^{N_w} \|\mathbf{p}_{m_i} - \mathbf{p}_{w_i}\|_2^2$ and $L_d = \sum_{k=0}^{N-1} \|\mathbf{x}_{k+1} - \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, dt_0)\|_2^2$, respectively, where dt_0 is a constant.

There exist infinitely many solutions that fulfill both the relaxed waypoint and dynamics constraints. To ensure a unique optimal solution, we introduce a regularization term for the control inputs in the objective function. This term, which is defined as $L_c = \sum_{k=0}^{N-1} \|\mathbf{u}_k\|_{\mathbf{R}}$, considerably accelerates the convergence of the interior point method.

We can find an initial solution to Problem (8) by properly choosing dt_0 and solving the warm-up problem with an interior point method, which is formulated as follows

$$\min_{\mathbf{x}_k, \mathbf{u}_k} \quad L_w + L_d + L_c \tag{9a}$$

$$\text{s.t.} \quad \mathbf{x}_{\text{lb}} \leqslant \mathbf{x}_k \leqslant \mathbf{x}_{\text{ub}} \tag{9b}$$

$$\mathbf{u}_{\text{lb}} \leqslant \mathbf{u}_k \leqslant \mathbf{u}_{\text{ub}} \tag{9c}$$

$$\mathbf{x}_0 = \mathbf{x}_{\text{init}} \tag{9d}$$

whose solution is used as the initialization for Problem (8). Upon solving (8), we obtain the optimal sampling times $\{dt_i^*\}_{i=1}^{N_w}$ for all segments and the time-optimal trajectory points $\{\mathbf{x}_{k_i}^*\}_{k_i}$ between adjacent waypoints, given by

$$\mathbf{Traj}^* := \big\{(\mathbf{x}_{k_i}^*, dt_i^*) | i = 1, \ldots, N_w, \, k_i \in \mathbb{Z}, \, m_{i-1} \leqslant k_i \leqslant m_i\big\}. \tag{10}$$

C. Time-adaptive Trajectory Tracker

The standard NMPC computes the control commands by solving a finite-time optimal control problem with a receding horizon H, formulated as follows

$$\min_{\mathbf{u}} \quad \sum_{k=0}^{H-1} \left(\|\mathbf{x}_{k+1} - \mathbf{x}_{\text{ref},k}\|_{\mathbf{Q}} + \|\mathbf{u}_k - \mathbf{u}_{\text{ref},k}\|_{\mathbf{R}}\right) \tag{11a}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, dt) \tag{11b}$$

$$\mathbf{x}_{\text{lb}} \leqslant \mathbf{x}_k \leqslant \mathbf{x}_{\text{ub}} \tag{11c}$$

$$\mathbf{u}_{\text{lb}} \leqslant \mathbf{u}_k \leqslant \mathbf{u}_{\text{ub}} \tag{11d}$$

$$\mathbf{x}_0 = \mathbf{x}_{\text{init}} \tag{11e}$$

where $\mathbf{x}_{\text{ref},k}$ and $\mathbf{u}_{\text{ref},k}$ are the reference states and reference inputs generated from our high-level trajectory planner.

When the reference trajectory is time-optimal, it indicates that the quadrotor's performance has reached its limits. However, the nonlinearity, imprecision, and aerodynamic effect of the quadrotor's dynamical model can still affect the actual tracking error. Fast trajectory tracking using the NMPC algorithm (11) faces two main challenges.

- c1) The reference trajectory is typically specified as a sequence of discrete points, and its time interval may not match that of the trajectory tracking controller. Thus, the solution obtained from time-optimal control cannot be directly used as the reference state for NMPC.
- c2) Model mismatch, sensor noise, external disturbances, and delays may cause the actual tracking time progress to deviate from that obtained from high-level planning, resulting in an increased actual tracking error.

To address the first challenge, we represent the planned trajectory (10) as a time-parameterized one and use polynomials to interpolate the trajectory between discrete points. The quadrotor's differential flatness property, combined with the small-time intervals dt_i^* (less than 0.1 s in general) between discrete points $\mathbf{x}_k^*$, enables us to express the trajectory as

$$\mathbf{traj}(t) = \begin{cases} \mathbf{traj}_1(t - t_0), & t_0 \leqslant t < t_1 \\ \mathbf{traj}_2(t - t_1), & t_1 \leqslant t < t_2 \\ \quad \vdots & \\ \mathbf{traj}_N(t - t_{N-1}), & t_{N-1} \leqslant t < t_N \end{cases} \tag{12}$$

where t_k represents the time associated with the state $\mathbf{x}_k^*$ which can be obtained by accumulating dt_i^*, and $\mathbf{traj}_k$ is a function of time and satisfies the following conditions, where

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

$\mathbf{p}_k^*$ and $\mathbf{v}_k^*$ denote the position and velocity components of $\mathbf{x}_k^*$, respectively

$$\begin{aligned} \mathbf{traj}_k(0) &= \mathbf{p}_{k-1}^* \\ \mathbf{traj}_k(dt_k) &= \mathbf{p}_k^* \\ \left.\frac{\mathbf{traj}_k(t)}{dt}\right|_0 &= \mathbf{v}_{k-1}^* \\ \left.\frac{\mathbf{traj}_k(t)}{dt}\right|_{dt_k} &= \mathbf{v}_k^* \end{aligned} \tag{13}$$

which means the trajectory $\mathbf{traj}_k$ satisfies the condition of first-order continuity.

To address the second problem, we propose optimizing the initial sampling time t_0 of the first reference point and subsequently sampling the reference points on the trajectory every interval dt, denoted as

$$\mathbf{p}_{\mathrm{ref},k} = \mathbf{traj}(t_0 + (k-1)dt)\,,\ \forall k = 1, 2, \ldots, H. \tag{14}$$

As the trajectory planning process accounts for the quadrotor's dynamics model, the trajectory is dynamically feasible. Therefore, we can focus on tracking the reference position of the trajectory using a time-adaptive model predictive control (tMPC) problem formulation as follows:

$$\min_{\mathbf{x}_k, \mathbf{u}_k, t_0} \quad \sum_{k=1}^{H} \|\mathbf{p}_k - \mathbf{p}_{\mathbf{ref},k}\|_2^2 \tag{15a}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, dt) \tag{15b}$$

$$\mathbf{x}_{\mathrm{lb}} \leqslant \mathbf{x}_k \leqslant \mathbf{x}_{\mathrm{ub}} \tag{15c}$$

$$\mathbf{u}_{\mathrm{lb}} \leqslant \mathbf{u}_k \leqslant \mathbf{u}_{\mathrm{ub}} \tag{15d}$$

$$\mathbf{x}_0 = \mathbf{x}_{\mathrm{init}} \tag{15e}$$

where $\mathbf{x}_k$ and $\mathbf{u}_k$ represent the state and control inputs at time step k, respectively, and H is the prediction horizon.

IV. Experiments

In this section, we present a set of simulations and experiments to validate the effectiveness of our proposed methods. We implemented the nonlinear optimization and optimal control problems using the open-source CasADi toolkit [21] and developed our program in the ROS environment. We designed separate ROS nodes for the trajectory planner and tracker, and implemented two nodes for simulation and actual flight, allowing for seamless switching between the two without modifying the program. The inertial and geometric parameters of the quadrotor used in our simulations and experiments are listed in Table I.

TABLE I
Quadrotor Configurations

Parameters		Values
$\mathbf{K}$	[s $^{-1}$]	$\mathrm{diag}(0.398, 0.316, 0.230)$
$\mathbf{J}$	[gm^2]	$\mathrm{diag}(1, 2, 3)$
m	[kg]	1.2
l	[m]	0.3
$(T_{\min}, T_{\max})$	[N]	(0, 6.9)
c_τ	[1]	0.2

TABLE II
Comparison with the CPC method

Number of waypoints	Solving time[1][s]		Optimal time[2][s]	
	CPC	Ours	CPC	Ours
$n = 4$	417.12	$0.598 + 1.11$	7.973	7.935
$n = 5$	1960	$1.15 + 1.12$	8.939	8.884
$n = 6$	1720	$1.24 + 1.20$	9.86	9.829
$n = 7$	1980	$1.33 + 1.92$	11.45	11.41
$n = 8$	2350	$1.69 + 2.24$	11.93	11.65

[1] Solving time is the total computation time for our method, i.e., the sum of solving time for the warm-up problem (9) and for the time-optimal planning problem (8).
[2] Optimal time is the total time of the time-optimal trajectory.

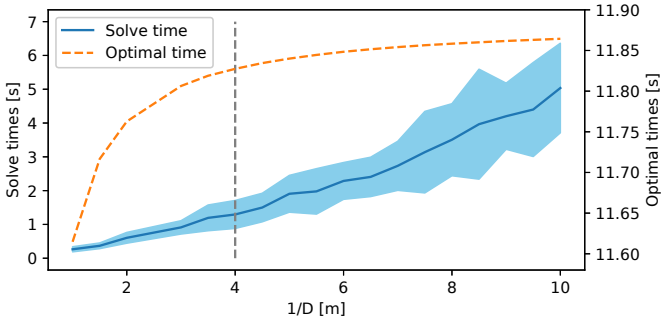

Fig. 4. The convergence trend of the optimal solution and the increasing trend of the solving time at different values of 1/D (average number of discrete points per meter). The location of each waypoint is fixed, and we measure the solving time 20 times with different initial solutions. The blue line represents the mean solving time, and the blue shadow represents the standard deviation.

A. Performance of Time-optimal Planner

Comparison with the state-of-the-art methods. Previous work by *Foehn et al.* [4] introduced CPC to achieve time-optimal trajectory planning. This method currently benchmarks the time-optimal flight performance, having even outperformed professional human pilots in drone races. However, the time-optimal trajectory generated by the CPC method requires a significant amount of computation time, typically taking several minutes to even hours to compute the full-state time-optimal trajectory.

Our study presents a time-optimal trajectory planning approach that addresses the same problem as the CPC method, but with considerably less computation time and better solution accuracy. To evaluate the performance of our method, we employed the drone configuration provided by Foehn et al. [4], which is summarized in Table I, and conducted a comparative analysis with the CPC method using an equal number of discretization points for 4 to 8 waypoints. Our experimental results in Table II reveal that our method is orders of magnitude faster than the CPC method, while maintaining superior solution quality. Moreover, our method offers the advantage of pre-assigning waypoint constraints and the flexibility to continuously adjust the optimal time for passing through each waypoint, leading to a higher precision on the waypoint constraints in contrast to the CPC method.

Convergence analysis of the optimal solution. In order to

improve the solution quality of our time-optimal planning method, we carefully determine the number of discretization points N_i between adjacent waypoints. Despite considering the accurate drone dynamics model in the optimization problem (8), the finite number of nodes used for discretizing the trajectory introduces a certain level of approximation. To ensure the uniformity of each node, we determine N_i based on the distance between adjacent waypoints. Specifically, we use the spatial density of nodes, denoted by D, to compute N_i as per the following equation

$$N_i = \left\lfloor \frac{\|\mathbf{wp}_i - \mathbf{wp}_{i-1}\|}{D} \right\rfloor \tag{16}$$

where $\lfloor \cdot \rfloor$ represents the floor function that outputs the greatest integer less than or equal to the given value.

To yield a more accurate trajectory, we discretize the trajectory into more nodes. However, this results in increased computation time. Therefore, we investigate the convergence trend of our optimal solution and the computation time under different node densities. Fig. 4 depicts the convergence trend of the optimal solution and the computation time at various values of $1/D$ (average number of discrete points per meter) with a fixed location for each waypoint. We measure the solving time 20 times with different initial solutions and calculate the mean (blue line) and standard deviation (blue shadow).

As the plots illustrate, an incremental increase in the number of discrete points corresponds to a gradual enhancement in the precision of our optimal solution. However, this positive gain in accuracy is countered by a swift escalation in solving time, which in turn leads to an increasing instability in the time it takes to find a solution. Conversely, the importance of maintaining an adequate number of nodes cannot be understated, as it plays a pivotal role in ensuring optimal flight performance. A higher count of nodes contributes to the generation of a trajectory that closely adheres to the constraints imposed by the quadrotors' dynamics, particularly following the interpolation process. This alignment guarantees both the smoothness and optimality of the trajectory tracking.

Based on the outcomes of our experiments, it becomes evident that once $1/D$ surpasses 3, any further improvement in flight performance becomes minor. With the aim of achieving a well-calibrated balance between solution accuracy, flight performance, and computational efficiency, we make the judicious choice of $D = 0.3$.

B. Analysis of the Fast Trajectory Tracking Method

Upon obtaining the time-optimal trajectory from the high-level planner, we employ the trajectory tracking method in Section III-C to control the quadrotor and track the trajectory. To assess the tracking performance, we define the root-mean-square error (RMSE) of position tracking as follows

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \min_{t} \|\mathbf{p}_i - \mathbf{traj}(t)\|^2} \tag{17}$$

where $\mathbf{traj}(t)$ represents the time-parameterized trajectory in (12), $\mathbf{p}_i$ denotes the actual position of the quadrotor during the tracking process, and n is the number of sampling points.

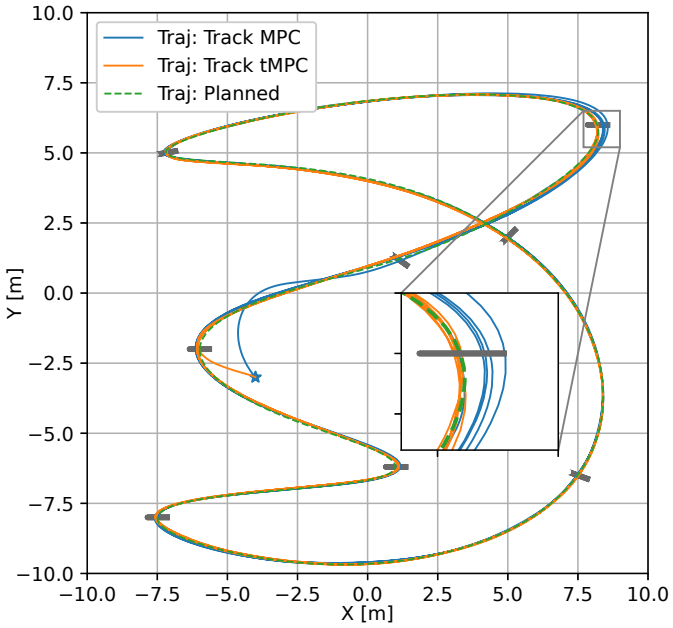

Fig. 5. Simulation comparison of the trajectory tracking performance between our tMPC and the plain-vanilla MPC. The gray line segments denote the pre-assigned waypoints, and the green dashed line represents the time-optimal trajectory. The quadrotor takes off from the blue star point and tracks the planned trajectory.

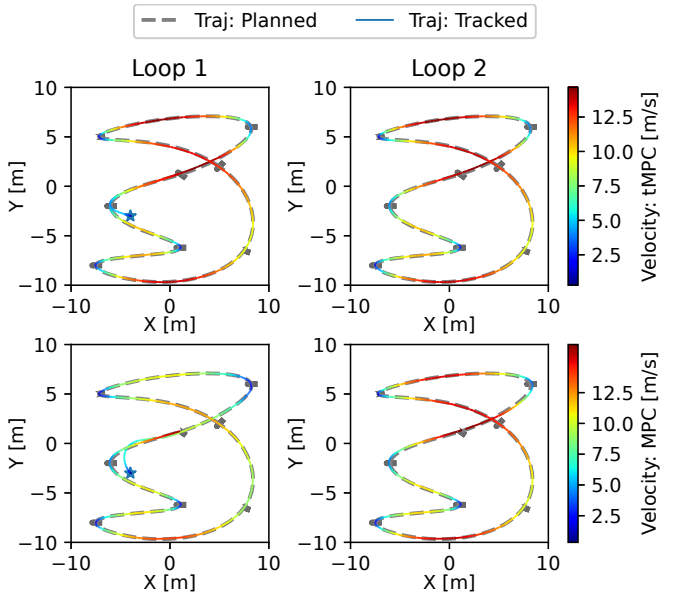

Fig. 6. Velocity tracking performance comparison between tMPC and MPC. The two plots at the top show the velocity tracking performance of tMPC for the first and second loops after takeoff, while the two at the bottom show the velocity tracking performance of MPC for the first and second loops.

We compared our proposed trajectory tracking algorithm with the standard MPC. Our method demonstrated superior performance in terms of position tracking error, flight speed, actual flight time, and tracking robustness. When disturbances or model mismatch were present, our proposed tMPC was capable of adaptively tracking the time-optimal trajectory, whereas standard MPC failed under time-optimal flight conditions. To address this issue, we slightly slowed down the

TABLE III
TRACKING ERROR UNDER MODEL MISMATCH

	RMES [m]		Max Error [m]		Track Time [s]	
	tMPC	MPC	tMPC	MPC	tMPC	MPC
Baseline	**0.036**	0.04	**0.133**	0.147	**9.75**	9.81
m 0.03 kg	**0.031**	0.037	**0.098**	0.151	**9.64**	9.81
$m + 0.03$ kg	**0.052**	0.081	**0.208**	0.329	9.87	**9.83**
0.9**K**	0.044	**0.043**	0.223	**0.134**	**9.73**	9.83
1.1**K**	**0.029**	0.039	**0.092**	0.148	**9.74**	9.82

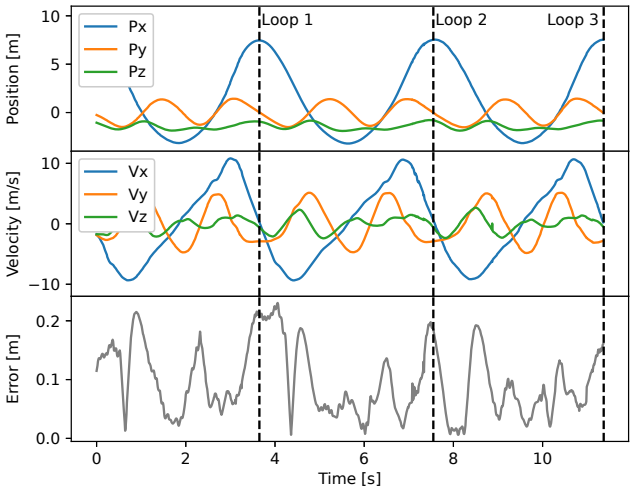

Fig. 7. Performance of time-optimal flight over static waypoints in real world. Panels from top to bottom are the position, velocity, and tracking error for three loops of actual flight. The maximum flight speed reached 10.6 m/s, and the maximum position error during the entire trajectory tracking was 0.22 m.

tracking progress of MPC to cope with external disturbances and actuator delays. Fig. 5 shows the trajectory tracking performance of tMPC and MPC under constraints of eight waypoints. Table III presents the average and maximum tracking errors and actual tracking time when the dynamic model is accurate and when there are mismatches in drone mass or rotor drag coefficient. The results indicate that due to the inability of MPC to adapt its tracking time, the tracking accuracy of MPC varied significantly when there were model mismatches, while tMPC automatically adjusted the tracking time and maintained tracking accuracy. Further, we compared the velocity tracking performance of tMPC and MPC, revealing that tMPC exhibits better tracking performance. As shown in Fig. 6, tMPC can quickly track the time-optimal trajectory with desired speed for both the first and second loops after takeoff. In contrast, MPC requires a longer transition process in the first loop.

C. Real-world Waypoint Racing

To further validate the performance of our algorithm, we conducted experiments on a physical quadrotor platform. The physical platform utilized the PX4-Vision frame and power kit and was integrated with an Intel NUC11 as the onboard computer. For the lower-level control, we employed the Pixhawk4 flight controller, which can accept angular velocity

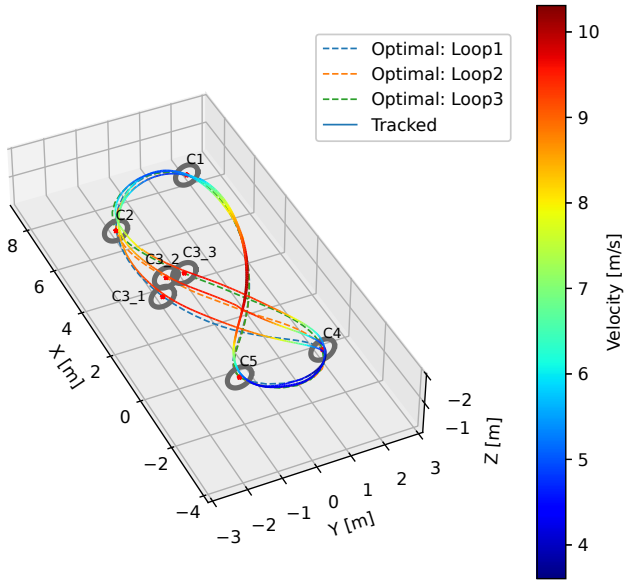

Fig. 8. Online replanning and tracking performance under dynamic waypoints in real world. In the figure, C3_1, C3_2, and C3_3 show the positions of the third circle at different times. The dashed line represents the corresponding time-optimal trajectory. The solid lines with colors indicate the real-time position of the quadrotor, with the color signifying the flight speed.

and thrust as control inputs and runs the angular velocity controller at $1,000$ Hz. In addition, the PX4 firmware running on the Pixhawk4 provided the extended Kalman filter for state estimation, which fused Inertial Measurement Unit (IMU) data with external auxiliary positioning provided by an OptiTrack motion capture system at 100 Hz.

To demonstrate the accuracy of trajectory tracking, we conducted an experiment where the drone flew through a circle with a diameter of 0.8 m. The experiment involved five circles, each representing a waypoint that the drone has to navigate through. We first conduct time-optimal flight experiments with static circles. In the experiment, we measure the position of each circle in advance and then use our proposed algorithm to plan the time-optimal trajectory and track it. We collect position and velocity data for the flight of three loops and calculate the position tracking error with respect to the time-optimal trajectory, as shown in Fig. 7. The results indicate that in the small indoor environment, we achieve a maximum flight speed of 10.6 m/s and a tracking error of less than 0.22 m.

To demonstrate the replanning capability of our algorithm, we conducted time-optimal flight experiments under dynamic waypoints. In the experiment, we randomly change the position of one of the waypoints and regenerate optimal trajectories in real-time, as shown in Fig. 8. We achieved a maximum flight speed of 10.2 m/s, with the throttle maintaining at around 21 m/s^2, which is very close to the theoretical maximum throttle (generated by the maximum thrust of four propellers, i.e., $4T_{\max}/m = 4 \times 6.9/1.2 = 23$ m/s^2) of the quadrotor in the experiment, as shown in Fig. 9. The experimental data corroborate that our algorithm can quickly replan new time-optimal trajectories upon changing the waypoint positions (in our experiment, the replanning time was 0.12 s) and accurately control the drone to fly along the new trajectory.

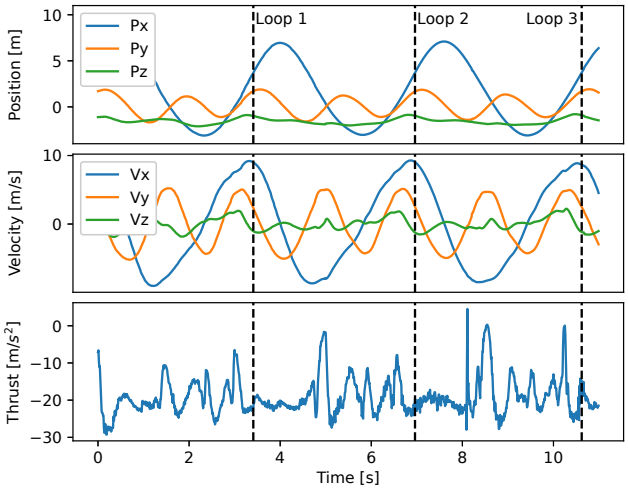

Fig. 9. Time-optimal flight data under dynamic waypoints in real world. Plots from top to bottom are the quadrotor's real-time position, velocity, and throttle. The maximum speed is 10.2 m/s and the average throttle is 21 m/s^2.

V. Conclusion and Future Work

In this letter, we have presented a novel (online) time-optimal (re)planning method for quadrotor navigation through dynamic waypoints. Our proposed tMPC algorithm effectively addressed the issue of poor robustness in time-optimal trajectory tracking. Our experimental results demonstrated that our approach can quickly plan new time-optimal trajectories and enable the quadrotor to rapidly switch to the new trajectories once the waypoint positions change. Compared to previous work, our approach achieved truly time-optimal trajectory replanning. However, we acknowledge that the proposed method does not constrain the quadrotor's yaw angle, which is often required in many applications. In future work, we will consider onboard sensor-based state estimation and gate position estimation to reduce dependence on motion capture systems. Moreover, collision-free time-optimal trajectory planning constitutes an interesting topic for future research.

References

[1] G. Loianno and D. Scaramuzza, "Special issue on future challenges and opportunities in vision-based drone navigation," *J. Field Robot.*, vol. 37, no. 4, pp. 495–496, 2020.

[2] J. Chen, J. Sun, and G. Wang, "From unmanned systems to autonomous intelligent systems," *Eng.*, vol. 12, pp. 16–19, 2022.

[3] D. Hanover, A. Loquercio, L. Bauersfeld, A. Romero, R. Penicka, Y. Song, G. Cioffi, E. Kaufmann, and D. Scaramuzza, "Autonomous drone racing: A survey," *arXiv:2301.01755*, 2023.

[4] P. Foehn, A. Romero, and D. Scaramuzza, "Time-optimal planning for quadrotor waypoint flight," *Sci. Robot.*, vol. 6, no. 56, p. eabh1221, 2021.

[5] A. Romero, R. Penicka, and D. Scaramuzza, "Time-optimal online replanning for agile quadrotor flight," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7730–7737, 2022.

[6] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, 2012.

[7] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *J. Guid. Control Dyn.*, vol. 10, no. 4, pp. 338–342, 1987.

[8] M. Geisert and N. Mansard, "Trajectory generation for quadrotor based systems using numerical optimal control," in *IEEE Int. Conf. Robot. Autom.* IEEE, 2016, pp. 2958–2964.

[9] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "EGO-planner: An ESDF gradient-based local planner for quadrotors," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 478–485, 2021.

[10] 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," *IEEE Trans. Robot.*, vol. 38, no. 6, pp. 3357–3373, 2022.

[11] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[12] M. Hehn, R. Ritz, and R. D'Andrea, "Performance benchmarking of quadrotor systems using time-optimal control," *Autonom. Rob.*, vol. 33, pp. 69–88, 2012.

[13] W. Van Loock, G. Pipeleers, and J. Swevers, "Time-optimal quadrotor flight," in *Euro. Control Conf.* IEEE, 2013, pp. 1788–1792.

[14] S. Spedicato and G. Notarstefano, "Minimum-time trajectory generation for quadrotors in constrained environments," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1335–1344, 2017.

[15] W. Zhang, J. Jia, S. Zhou, K. Guo, X. Yu, and Y. Zhang, "A safety planning and control architecture applied to a quadrotor autopilot," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 680–687, 2022.

[16] I. Spasojevic, V. Murali, and S. Karaman, "Perception-aware time optimal path parameterization for quadrotors," in *IEEE Int. Conf. Robot. Autom.* IEEE, 2020, pp. 3213–3219.

[17] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, "Model predictive contouring control for time-optimal quadrotor flight," *IEEE Trans. Robot.*, vol. 38, no. 6, pp. 3340–3356, 2022.

[18] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 620–626, 2017.

[19] E. Tal and S. Karaman, "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 3, pp. 1203–1218, 2020.

[20] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE Int. Conf. Robot. Autom.* IEEE, 2011, pp. 2520–2525.

[21] 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.

[22] X. Wang, J. Sun, F. Deng, G. Wang, and J. Chen, "Event-triggered consensus control of heterogeneous multi-agent systems: Model-and data-based approaches," *Sci. China Inf. Sci.*, vol. 66, no. 9, pp. 1–17, 2023.

[23] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit—An open-source framework for automatic control and dynamic optimization," *Optim. Control Appl. Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[24] W. Liu, J. Sun, G. Wang, F. Bullo, and J. Chen, "Data-driven resilient predictive control under Denial-of-Service," *IEEE Trans. Autom. Control*, vol. 68, no. 8, pp. 4722–4737, Aug. 2023.

[25] ——, "Data-driven self-triggered control via trajectory prediction," *IEEE Trans. Autom. Control*, pp. 1–8, 2023, DOI:10.1109/TAC.2023.3244116.

[26] D. Bicego, J. Mazzetto, R. Carli, M. Farina, and A. Franchi, "Nonlinear model predictive control with enhanced actuator model for multi-rotor aerial vehicles with generic designs," *J. Intell. Robot. Syst.*, vol. 100, pp. 1213–1247, 2020.

[27] G. Torrente, E. Kaufmann, P. Föhn, and D. Scaramuzza, "Data-driven MPC for quadrotors," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3769–3776, 2021.

[28] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1: 43 scale RC cars," *Optim. Control Appl. Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[29] T. R. Jorris and R. G. Cobb, "Three-dimensional trajectory optimization satisfying waypoint and no-fly zone constraints," *J. Guid. Control Dyn.*, vol. 32, no. 2, pp. 551–572, 2009.

[30] K. Bousson and P. F. Machado, "4D trajectory generation and tracking for waypoint-based aerial navigation," *Trans. Syst. Control*, no. 3, pp. 105–119, 2013.