2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

A Trajectory-based Flight Assistive System for Novice Pilots in Drone Racing Scenario

Yuhang Zhong, Guangyu Zhao, Qianhao Wang, Guangtong Xu, Chao Xu, and Fei Gao

***Abstract*— Drone racing has become a popular international competition and has attained wide attention in recent years. However, the requirements of high-level operation keep the novice pilots away from participating in it. This paper presents a trajectory-based flight assistive system that enables various operators to fly the drone in a racing scene at a high speed. The whole system is structured hierarchically, consisting of both offline and online components. In the offline part, a global time-optimal trajectory is generated as the expert reference, and a dense flight corridor is constructed to provide sufficiently large safe region. In the online part, a remote control-mapped primitive is designed to fast encapsulate pilots' inputs, and the time mapping based trajectory progress is customized to further capture intention. Then, a trajectory planner is proposed to generate intention-aligned, smooth, feasible, and safe trajectories periodically. Additionally, a yaw planning that provides the pilot with the best suitable view angle is employed to further alleviate the operation difficulty. Simulations and real world experiments are implemented to verify the performance of our system. The maximum flight speed can reach 6.0 m/s for a novice drone pilot in a real racing scene. Our code is released as an open-source package[1].**

I. Introduction

Drone racing has gained significant popularity in recent years. The thrill of engaging in high-speed races through complex obstacle courses, while experiencing the first-person perspective through first person view (FPV) goggles, captivates both participants and spectators alike. However, it must be acknowledged that flying at high speeds with minimal risk of crashing is a skill possessed only by professional drone racing pilots. To enable more novices to enjoy drone racing, this paper proposes an adaptive flight assistive system, dedicated to alleviate the challenges faced by beginners on their path to mastery.

An intuitive idea is to directly integrate assisted flight and autonomous racing into a unified system. Though both assisted flight and autonomous racing have witnessed remarkable advancements individually, their respective limitations come to the fore when collaboratively adapting assisted flight in high-speed drone racing, which are analyzed as follows.

Corresponding Author: Guangtong Xu; Fei Gao.

This work was supported by the National Natural Science Foundation of China under Grant 62322314 and Grant 62203256.

Yuhang Zhong, Guangyu Zhao, Qianhao Wang, Chao Xu, and Fei Gao are with the College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China, and Huzhou Institute, Zhejiang University, Huzhou 313000, China. (e-mail: {YuhangZhong, zhaoguangyu, qhwangaa, cxu, fgaoaa}@zju.edu.cn)

Guangtong Xu is with the Huzhou Institute, Zhejiang University, Huzhou 313000, China. (e-mail: guangtong_xu@163.com)

[1]https://github.com/ZJU-FAST-Lab/Assistive-Racing

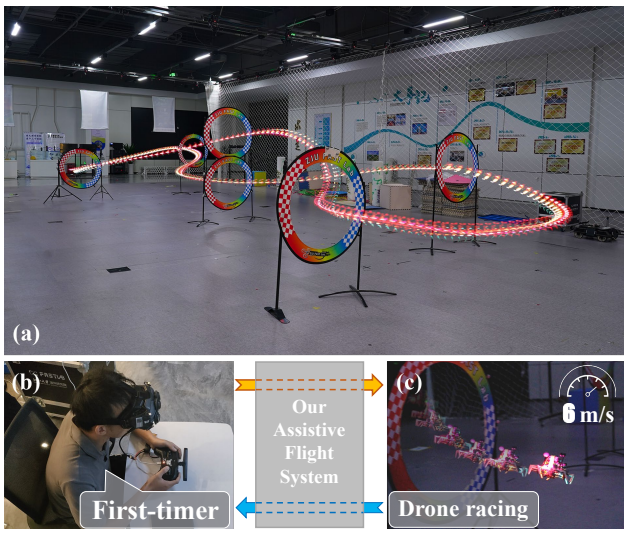

Fig. 1. Illustration of the real world experiments of our flight assistive system in a racing scene. (a) The demonstration of the real drone racing performance. (b) The first-time flying pilot wears goggles to access the FPV and operates the joystick to control the drone. (c) The snapshot of the racing drone at a high flight speed.

Assisted flight approaches [1]–[3] primarily focus on enabling the intent of user to guide flight while addressing obstacle avoidance in low-speed scenarios. It poses significant challenges to perform these approaches to high-speed and complex racing scenarios. Firstly, in the absence of expert trajectory references, these approaches allow users to solely rely on their own intentions for flight. However, this can present difficulties for novices in determining appropriate actions, particularly when faced with demanding scenarios like sharp turns often encountered in racing. Additionally, they struggle with balancing intent capture accuracy and computational overhead, both of which are crucial in racing scenarios. They either simplify the engagement of the pilots to ease the computational burden [1], or finely model the pilot's intent resulting in heavy computational overhead [2, 3], which limits them away from high-speed scenarios.

For autonomous racing, most approaches adopt to planning with simplified constraints to enhance efficiency, such as utilizing sparse and compact representations of safety regions [4] or directly requiring that a certain point on the trajectory must pass through the gate's center [5, 6]. These simplified constraints lead to a tighter solution space. However, when these racing approaches are employed in the assisted flight that utilizes the complex and variable user intents as guidance, the tight solution space can potentially result in trajectories that exhibit excessive intelligence or

become trapped, as shown in Fig. 3, leading to a perceived loss of control for the users.

In summary, despite the noteworthy strides in both assisted flight and autonomous racing, there persists a dearth of an apt solution assisted flight in drone racing. To bridge the gap, we establish a dedicated flight assistive system tailored explicitly for racing scenarios, with the capacity to empower novices to fully realize their racing ambitions. The system integrates offline and online modules. In the offline module, a global time-optimal trajectory and dense corridor are generated to provide expert reference and larger spatial freedom for the online module, respectively. In the online module, we propose an efficient replanning framework guided by user intent and capable of providing adaptive assistance. First, we generate a motion primitive based on the input of the user, indicating the user's intent. We then generate trajectories for drone execution using a trajectory optimization based approach, where the safety constraints are imposed on points of the trajectory through the dense corridor. Additionally, we propose a temporal mapping to adaptively find suitable guidance on the reference based on the user intention, which allows novices to learn and perform challenging maneuvers, as demonstrated in the Fig. 1. To allow the high operation engagement of pilots, the trajectory will be locally shaped by the intent-based primitive and globally guided by the time-optimal reference. Furthermore, the perception-aware yaw is planned for pilots to find gates favourably. Last but not least, benefiting from imposing all the above requirements in the form of convex constraints, our trajectory optimization can be solved efficiently to meet the high real-time demand. The contributions of this work are summarized below:

1) We propose a hierarchical flight system specifically designed for assisting drone racing that ensures the timeliness of executing user intents in high-speed scenarios, while maintaining accurate intent capture.
2) The introduction of time-optimal expert reference, time mapping-based trajectory progress, and dense corridor in the system achieve a harmonious blend of user engagement and guidance for completing racing aggressively.
3) Extensive simulations and real-world experiments are performed to confirm the adaptability and practicality of our system for various operators. The source code of our system is released .

II. Related Work

A. Autonomous Drone Racing

Numerous works and competitions are dedicated to drone racing. The DRL world drone racing championship [7] provides a platform for professional pilots around the world to compete. Besides, several international autonomous drone racing competitions, such as AlphaPilot Challenge [8] and Autonomous Drone Race in IEEE IROS [9], are held with the aim of winning races against human pilots. Pfeiffer et al. [10] explore the correlation between human pilot's operations and flight performance of drones to obtain a better understanding of the key factors influencing racing.

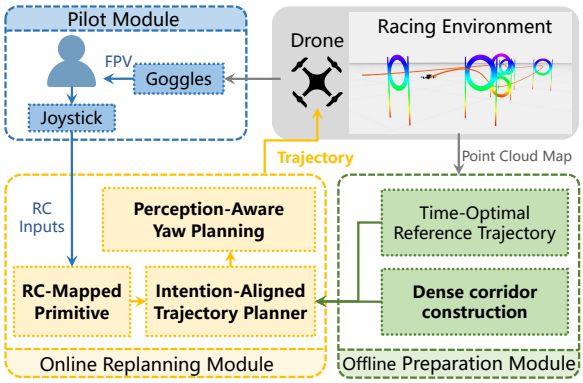

Fig. 2. A diagram of our flight assistive system architecture.

Foehn et al. [11] and Romero et al. [12] study the time-optimal trajectory planner for drones which drives a high-level autonomous flight comparable to elite pilots. Kuafmann et al. [13] propose Swift, a champion-level autonomous racing system only relying on onboard sensors, through deep reinforcement learning. Current investigations mainly focus on autonomous drone racing, and rare solutions of making racing accessible to novice pilots are available.

B. Assistive Aerial Teleoperation

Several previous studies [14, 15] design assistive system for aerial teleoperation, whereas the pilot's operation is represented as a low-level control input, resulting in a weak alignment of intentions. Yang et al. [2, 3] propose the Bias Incremental Action Sampling (BIAS) method to generate a set of primitives to encapsulate the intention of pilots and avoid obstacles. However, BIAS consumes relatively high computation time unsuited to high-speed drone racing, and suffers from potential loss of intention alignment due to the inherent discretized error of primitives. Works in [1, 16] map the intention to a target point and generate a safe target-guided trajectory, but the oversimplification on the intention representation leads to a significant loss of operation engagement for pilots.

III. System Overview

We aim to establish a flight assistive system that offers novice pilots, who lack proficiency in drone operation or have no drone fly experience, the opportunity to experience fast drone flight in racing scenarios. The architecture is depicted in Fig. 2, containing the pilot, offline preparation, and online replanning modules.

In the pilot module, the operator uses a joystick to send commands and receives the live FPV stream from goggles. In the offline preparation module, the point cloud map of the racing environment is established to provide the structure information for the subsequent global planning. The global planning algorithm by Han et al. [4] is utilized to generate a time-optimal trajectory which serves as the expert reference. Then, the dense flight corridor is constructed along the global trajectory to provide the safe region.

For the online replanning module, it receives the pilot's inputs, and generates the remote controller (RC)-mapped primitive. Then, a corridor-constrained and intention-aligned

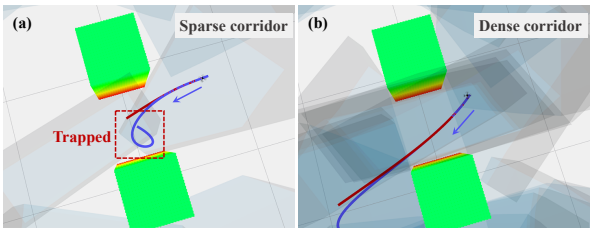

Fig. 3. A graphical interpretation of the merit of dense corridor. (a) Trapped trajectory (blue curve) due to the conflict between corridor and pilot's intention (red primitive). (b) Smooth, safe, intent-aligned trajectory.

trajectory is planned at a high frequency. In addition, we incorporate a perception-aware yaw planning to assist the pilot to swiftly align with the next passing gate and alleviate the high-level operation challenges.

IV. Generation of Corridor and Primitive

A. Dense Flight Corridor

Constructing an appropriate flight corridor is crucial for both the flight safety and effective trajectory generation. The intention-aligned trajectory may vary significantly due to the rapid and sudden change of pilots' inputs. The conventional sparse corridor construction method provides some of the trajectory pieces with the overlapping area of the polyhedrons as safe space. However, the relatively small overlapping space may induce significant conflict between corridor constraint and pilot's intention, resulting in unsatisfactory planning attempts such as trajectory bending (shown in Fig. 3). Instead, constructing a dense corridor can offer a significantly larger free space for each trajectory piece, thereby enhancing the robustness. The corridor construction method in [17] is utilized to generate the dense corridor $\mathcal{V}$ containing N polyhedrons along the global trajectory:

$$\mathcal{V} = \bigcup_{i=0}^{N} \mathcal{V}_i, \mathcal{V}_i = \{\boldsymbol{x} \in \mathbb{R}^3 | \boldsymbol{A}_i \boldsymbol{x} - \boldsymbol{b}_i \leq 0\}. \quad (1)$$

For each sampled point $\boldsymbol{p}_{glb}$ on the global trajectory, we assign a suitable polyhedron safe space with index $I(\boldsymbol{p}_{glb})$ by the following distance-based metric:

$$I(\boldsymbol{p}_{glb}) = \underset{i \in \{0,\dots,N\}}{\operatorname{argmax}} \ \mathcal{F}_i(\boldsymbol{p}_{glb}), \quad (2)$$

$$\mathcal{F}_i(\boldsymbol{p}_{glb}) = \min_{i \in \{0,\dots n\}} \left\| \boldsymbol{b}_i - \boldsymbol{A}_i \boldsymbol{p}_{glb} \right\|_\infty, \quad (3)$$

where $\|\cdot\|_\infty$ denotes maximum norm, and $\mathcal{F}_i(\boldsymbol{p}_{glb})$ is defined to compute the minimum Euclidean distance from $\boldsymbol{p}_{glb}$ to the surfaces with respect to the n nearest polyhedrons.

B. RC-Mapped Primitive

By using differential flatness of drone's dynamics, the control of drones can be transformed into controlling the flat outputs $\boldsymbol{\sigma} = [x, y, z, \psi]$, where x, y, z are the position, and ψ is the yaw angle. Thus, establishing a relationship between pilot's RC inputs and flat outputs significantly reduces the complexity of intention representation.

To achieve the mapping from RC inputs to flat outputs, the RC-mapped motion primitive is designed. This primitive is a polynomial parameterized trajectory $\mathcal{T}_{rc}$ generated responding to RC inputs. The primitive evolves in accordance with a direct velocity model and the RC control inputs are set as $\boldsymbol{a}_{cmd} = [v_x, v_y, v_z, \dot{\psi}]^\top$. The model is intuitive and user-friendly to novice pilots compared with the unicycle model used in [3], since it only requires pilots to understand a standard second order system and can explore the flat outputs sufficiently. To guarantee the smoothness and continuity of primitives, we use a 7th order parameterized trajectory with jerk continuity. The RC primitive is defined as:

$$\begin{aligned} \mathcal{T}_{rc}(t) &= \sum_{i=0}^{7} c_i t^i, \quad (4) \\ s.t.\ \mathcal{T}_{rc}^{(i)}(0) &= \boldsymbol{x}_0^{(i)},\ i = 0, 1, 2, 3, \\ \dot{\mathcal{T}}_{rc}(T) &= \boldsymbol{R}_w^b \boldsymbol{a}_{cmd}, \\ \mathcal{T}_{rc}^{(i)}(T) &= \boldsymbol{0},\ i = 2, 3, 4, \end{aligned}$$

where c_i is the coefficients of the polynomial trajectory, $\boldsymbol{x}_0$ is the initial UAV state, $\boldsymbol{R}_w^b$ is the transformation from the body frame to the world frame, and T is the duration of the primitive. More details about primitives can be found in Liu's paper [18]. Once T is predefined, this linear system is fully constrained and can be solved in closed form. To respond to the RC inputs swiftly, the primitive should reach the desired velocity as soon as possible. We use dichotomy to dynamically adjust the T from an initial time and check the maximum acceleration of the primitive by Sturms Theorem [19]. The RC-mapped primitive is generated at a high frequency, and the latest primitive is adopted to guide the generation of the intention-aligned trajectory.

V. Flight Assistive Planner

This section presents the intention-aligned trajectory planner including time mapping based trajectory progress, optimization problem formulation, and perception-aware yaw planning.

A. Time Mapping Based Trajectory Progress

In the racing scene, pilots are expected to fly the drone as fast as possible to win the competition. Therefore, our flight assistive system utilizes the global time-optimal trajectory as the reference to guide the generation of the intention-aligned trajectory. To implicitly encourage pilots to fly at a high speed, the trajectory should slightly extend beyond the pilot's intention. Inspired by the work [6, 20], we construct the trajectory $\boldsymbol{p}(t)$ in a fixed time duration T_f with a time mapping function $\tau(t)$:

$$\tau(t) = \sum_{i=0}^{5} e_i t^i. \quad (5)$$

where e_i is the corresponding coefficient of polynomial, and the order of $\tau(t)$ is predetermined as 5 to ensure the smoothness of the gradient throughout the optimization process. $\tau(t)$ maps the original time variable t to a new variable τ, and τ is used to find corresponding reference points $\boldsymbol{p}_{glb}(\tau(t))$ on the global time-optimal trajectory.

Based on the time mapping function $\tau(t)$, the trajectory progress cost is designed as

$$\mathcal{L}_{prog} = \int_0^{T_f} \rho_p \left\| \boldsymbol{p}(t) - \boldsymbol{p}_{glb}(\tau) \right\|_2^2 - \rho_\tau(\boldsymbol{a}_{cmd}) \dot{\tau}(t) dt, \quad (6)$$

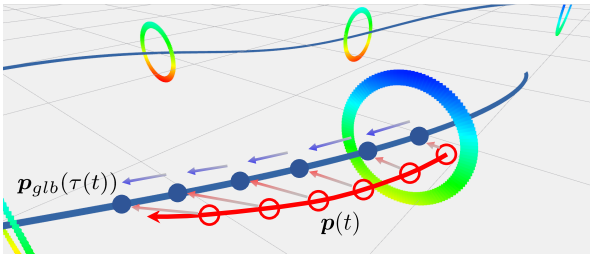

Fig. 4. Illustration of time mapping based trajectory progress. Red and blue curves indicate the planned local trajectory and pre-generated global time-optimal trajectory. Red and blue arrows denote the effects of the first and second terms in the trajectory progress cost $\mathcal{L}_{prog}$, respectively.

where ρ_p and $\dot{\tau}(t)$ denote weight coefficient and traveling progress, respectively. Red arrows in Fig. 4 show the effect of first part of the trajectory progress cost, which penalizes the position error between the time-optimal reference point $\boldsymbol{p}_{glb}(\tau(t))$ and the optimized point $\boldsymbol{p}(t)$. Note that the flight speed of the global reference point is defined as $\boldsymbol{v}_{glb}(\tau) = \dot{\boldsymbol{p}}_{glb}(\tau)\dot{\tau}(t)$, and adjusting $\dot{\tau}(t)$ to be greater than 1 enables a faster drone flight than reference $\dot{\boldsymbol{p}}_{glb}(\tau)$. Thus, introducing $\dot{\tau}(t)$ with a negative sign in (6) deliberately encourages a more aggressive flight behavior (shown as the blue arrows in Fig. 4). Additionally, to further capture the pilot's RC control input $\boldsymbol{a}_{cmd}$, a fine-tuning weight $\rho_\tau(\boldsymbol{a}_{cmd})$ is introduced in $\mathcal{L}_{prog}$ with a linear correlation:

$$\rho_\tau(\boldsymbol{a}_{cmd}) = a \left\|(\boldsymbol{a}_{cmd})_{(1:3)}\right\|_2 + b, \tag{7}$$

where a and b are scalars, and $(\cdot)_{(1:n)}$ denotes to select the first n elements from a vector. As the $\boldsymbol{a}_{cmd}$ increases which indicates the pilot's intention to fly more fast, the weight $\rho_\tau(\boldsymbol{a}_{cmd})$ grows. As a result, $\mathcal{L}_{prog}$ drives the optimization to find a more aggressive trajectory.

B. Intention-Aligned Trajectory Optimization

MINCO [21] is utilized as the trajectory presentation inherently minimizing control effort, and the optimization variables are a set of waypoints $\boldsymbol{p}(t) = [p_0, p_1, ..., p_n]^\top$ and the time mapping parameters e_i in $\tau(t)$. We formulate the trajectory optimization problem considering smoothness, pilot's intention, time mapping based trajectory progress, feasibility, and safety, shown as

$$\min_{\boldsymbol{p}(t),\tau(t)} \mathcal{J} = \mathcal{L}_{smooth} + \mathcal{L}_{RC} + \mathcal{L}_{prog}, \tag{8}$$

$$s.t.\ \boldsymbol{p}^{(i)}(0) = \boldsymbol{x}_0^{(i)},\ i = 0, ..., s-1, \tag{9}$$

$$\tau^{(i)}(0) = \delta_0^{(i)},\ i = 0, ..., k-1, \tag{10}$$

$$\mathcal{G}_\star \le 0,\ \star = v, a, j, v_t, a_t, j_t, \tag{11}$$

$$\mathcal{G}_{safe} \le 0, \tag{12}$$

where $\boldsymbol{x}_0$ and δ_0 in (9) and (10) are the initial state of trajectory (up to $(s-1)$-th order derivative) and mapped time (up to $(k-1)$-th order derivative) respectively. In this particular case, s and k are set as 5 and 4, respectively. The costs and constraints are described below in detail.

1) Smoothness Cost: Since the smoothness of a trajectory is tightly related to the control effort, we directly employ the control effort of the trajectory as a cost function:

$$\mathcal{L}_{smooth} = \rho_s \int_0^{T_f} \left\|\boldsymbol{p}^{(s)}(t)\right\|_2^2 + \omega_\tau \cdot (\tau^{(k)}(t))^2 dt, \tag{13}$$

where ρ_s is the weight coefficient, and $\omega_\tau \in [0, 1]$ is a scalar for the inner weight allocation.

2) Intention Aligned Cost: The pilot's intention has already been represented as the RC-mapped primitive, as discussed in Section IV-B. The error between the optimized trajectory and RC-mapped primitive is formulated as a cost function to closely align with the pilot's intention and trade off with other requirements. The cost is designed as:

$$\mathcal{L}_{RC} = \int_0^{T_e} \alpha(t) \sum\nolimits_{i=0}^{2} \rho_{r_i} \left\|\boldsymbol{p}^{(i)}(t) - \boldsymbol{p}_{RC}^{(i)}(t)\right\|_2^2 dt, \tag{14}$$

where ρ_{r_i} is the corresponding weight, and T_e denotes the duration that the cost effects. $\boldsymbol{p}_{RC}(t)$ is the point on RC-mapped primitive with respect to the point $\boldsymbol{p}(t)$ in the optimized trajectory at the same time. $\alpha(t) = e^{-\beta \cdot t/T_e}$ is a function that gradually reduce the influence of pilot's intention as the time progresses, where β is a scalar to regulate the level of relaxation. $\alpha(t)$ allows more optimization freedom for the latter segments of trajectory, which can facilitate well balance among different costs and constraints. Even as the intention weakens along the trajectory, pilots can still perceive a rapid and accurate response of the drone. This is attributed to the fact that drones execute only the foremost segment of the trajectory, which exhibits a strong alignment with the pilot's intention, during each planning step.

3) Feasibility Constraint: To prevent excessive velocity caused by $\mathcal{L}_{prog}$ and mitigate motion blur and flutter in FPV, the bounds on velocity, acceleration, and jerk of the trajectory are constrained as:

$$\mathcal{G}_\mu = \left\|\boldsymbol{p}^{(n)}(t)\right\|_2^2 - \mu_{max}^2, \tag{15}$$

where $(n, \mu) \in \{(1, v), (2, a), (3, j)\}$. v_{max}, a_{max}, and j_{max} are the maximum velocity, acceleration, and jerk respectively.

Since $\mathcal{L}_{prog}$ also induces the excessive increase on the rate of the mapped time, constraints are imposed on the velocity, acceleration, and jerk of the mapped time to ensure its smooth variation within the specified bounds:

$$\mathcal{G}_\gamma = \left\|\tau^{(n)}(t)\right\|_2^2 - \gamma_{max}^2, \tag{16}$$

where $(n, \gamma) \in \{(1, v_t), (2, a_t), (3, j_t)\}$, and v_{tmax}, a_{tmax} and j_{tmax} denote the maximum velocity, acceleration and jerk of the mapped time respectively.

4) Safe Corridor Constraint: To guarantee the safety of drones, the dense corridor constraint $\mathcal{G}_{safe}$ in Section IV-A is introduced. The point on the trajectory is paired with the closest point $\boldsymbol{p}_{close}$ on the global trajectory and constrained as:

$$\mathcal{G}_{safe} = \mathbf{A}_i \boldsymbol{p}_t - \mathbf{b}_i,\ i = I(\boldsymbol{p}_{close}). \tag{17}$$

C. Perception-Aware Yaw Planning

Since it is difficult for novice pilots to simultaneously control the movement and yaw of drones in racing scenes where the drone is required to pass through the narrow gates, the assistive system needs to provide perception-aware reference of yaw. Firstly, we extract the key points $\boldsymbol{p}_{kp}(t)$ along the global reference trajectory, which represent important positions requiring the FPV to orient, such as the center of gates. Then, the points on the intention-aligned trajectory are assigned to corresponding key points $\boldsymbol{p}_{kp}(t)$. The planning problem of perception-aware yaw $\psi(t)$ is described as:

$$\min_{\psi(t)} \mathcal{J} = \mathcal{I}_{smooth} + \mathcal{I}_{RC} + \mathcal{I}_{percep}, \tag{18}$$

$$s.t.\ \psi^{(i)}(0) = \psi_0^{(i)},\ i = 0, ..., n-1, \tag{19}$$

$$|\ \psi^{(1)}(t)\ | \leq v_{max}^{\psi}, |\ \psi^{(2)}(t)\ | \leq a_{max}^{\psi}, \tag{20}$$

where ψ_0 denotes the initial yaw up to 3rd order derivative. $\mathcal{I}_{smooth}$ and $\mathcal{I}_{RC}$ have the similar definitions to $\mathcal{J}_{smooth}$ and $\mathcal{J}_{RC}$, respectively. v_{max}^{ψ} and a_{max}^{ψ} are the bounds for the 1st and 2nd order derivative of yaw. The perception-aware cost $\mathcal{I}_{percep}$ guides the yaw to orient the key points $\boldsymbol{p}_{kp}(t)$ in the XY planar, expressed as:

$$\mathcal{I}_{percep} = \rho_{\psi}(1 - \frac{\boldsymbol{v}_{\psi} \cdot (\boldsymbol{p}_{kp}(t) - \boldsymbol{p}(t))_{(1:2)}}{\left\| (\boldsymbol{p}_{kp}(t) - \boldsymbol{p}(t))_{(1:2)} \right\|_2}), \tag{21}$$

$$\boldsymbol{v}_{\psi} = [cos(\psi(t)), sin(\psi(t))]^{\top}, \tag{22}$$

where ρ_{ψ} denote the corresponding weight coefficient.

The two optimization problems above are transformed to the unconstrained optimization problems by penalty method. The time integral penalties are reformulated as discretized form. The quasi-newton method L-BFGS is used to solve the problems. We recommend readers to [21] for more details.

VI. Experiments and Benchmarks

A. Implementation Details

The proposed flight assistive system is tested in simulations and real-world experiments. The simulations are implemented on a PC with the Intel Core i7-10700K CPU. In the real world experiment, the online replanning module is running onboard and processed by Jetson Xavier NX. The localization of drones is achieved by motion capture system.

We establish a racing scene as shown in Fig. 1 and invite five novice participants to compete. The racing route is designed to fly through multiple gates in a specified order. Participants are informed about the racing route in advance and practice in simulation for six times to familiarize the scene. Then they wear the DJI goggles to access the FPV and try to fly the drone at a high speed in two rounds.

B. Simulations

1) Benchmark: To validate the performance of our system in racing scene, the following key metrics are introduced:

- **AEIA**: Average error of the intention alignment.
- **ART & MRT**: Average and maximum replanning time.
- **CT:** Completion time of drone racing.

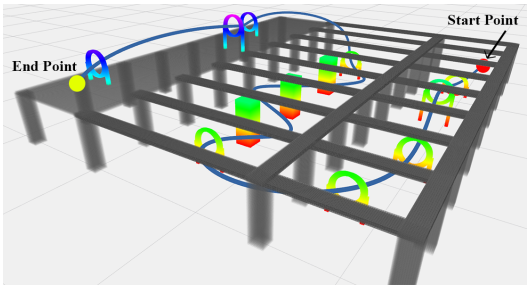

Fig. 5. Racing scene in simulations. The blue curve is global trajectory.

- **AFS & MFS:** Average and maximum flight speed in racing.

To evaluate the intention alignment, AEIA is computed by

$$AEIA = \frac{1}{N} \sum_{i=0}^{N} \int_0^{T_m} \sum_{j=0}^{2} \left\| \boldsymbol{p}_i^{(j)}(t) - \boldsymbol{p}_{rc_i}^{(j)}(t) \right\|_2 dt, \tag{23}$$

where N is the total number of evaluations in the test and T_m is flight time. The evaluation is running at 100Hz and T_m is set as 100s.

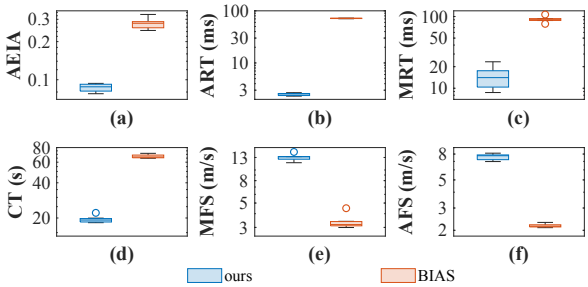

Fig. 6. Box plots about the comparison between BIAS and the proposed flight assistive system (ours). Due to the significant disparity in values along the y-axis, a logarithmic mapping is applied to the y axis to address this issue. The proposed system outperforms BIAS in terms of all aspects.

The proposed assistive system is compared with Yang's work [3] (denoted as BIAS). An industry-like environment is established with a size of $40\text{m} \times 40\text{m} \times 10\text{m}$ as shown in Fig. 5. To ensure the fairness of comparison, the identical global trajectory is employed to BIAS and ours. Each method is tested for 10 times, and the comparative results are visualized as the box plots in Fig. 6. The AEIA in Fig. 6(a) by ours is less than 0.1 and much smaller than that of BIAS. The comparison on ART and MRT in Figs. 6(b)&(c) demonstrates that the computation efficiency of our online replanning module is almost two orders of magnitude faster than BIAS. The efficiency of BIAS is limited by the sampling size highly related to the tree library size and the discretization number. When modifying BIAS for racing, the discretization number is increased, resulting in heavy computation. In contrast, ours converges to a feasible solution with the convex formulation on constraints and the sufficient safe space provided by the dense corridor. Figs. 6(e)&(f) exhibit that our method enables the drone to achieve a maximum flight speed of 14 m/s and an average flight speed of 8 m/s, which leads to a shorter completion time against BIAS shown in Fig. 6(d).

2) Analysis of Assistive Perception-Aware Yaw Guidance: Based on the data collected from the practices of pilots in simulations, we evaluate the impact of the assistive yaw

guidance. The CT of racing is shown in Fig. 7(b), where the median value of CT with assisted yaw is lower than that without it. Besides, the flight trajectories of a participant with assisted yaw or not are compared in Fig. 7(a). Without assisted yaw, the pilot fails to find the coming gate due to the high flight speed in Aera1. While the pilot loses the direction in the Aera2 because of the sharp turning, which induces a sudden stop. Conversely, this pilot can achieve a smooth gate traversal with the suitable yaw guidance.

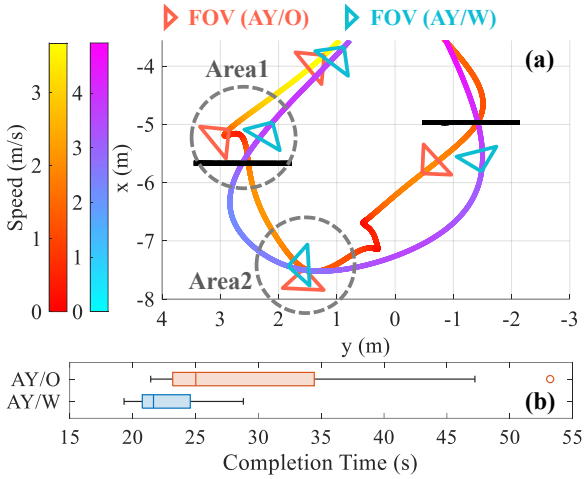

Fig. 7. Comparisons on the yaw assistance. (a) The flight trajectories of a novice pilot, and the blue trajectory indicates the flight with assisted yaw (denoted as AY/W) and red one represents the flight without it (AY/O). (b) The box plot of completion time with assisted yaw and without it.

C. Real World Experiments

1) Pilot Adaptation: To validate that the proposed system can accommodate different desired flight speeds of pilots in real world, the pilot is requested to fly the drone slowly in the loop one and accelerate to a high speed in the loop two. The results are shown in the Fig. 8. In the first loop, due to the requirement of low flight speed, all normalized values of RC channels are under 0.4, which conducts a trajectory with the speed under 2m/s. During the second loop, the pilot operates the RC inputs in a more aggressive manner, while our system assists the pilot in achieving significantly higher flight speeds. This experiment validates that our system is user-friendly for pilots, as it offers a wide range of flight speed options for pilots considering different operation levels.

TABLE I

RANKS WITH EVALUATING DATA

Rank	1	2	3	4	5
CT (s)	22.591	23.5031	27.4246	33.2338	34.4221
MFS (m/s)	6.0011	4.4502	4.4777	3.795	3.0377
AFS (m/s)	2.9838	2.7913	2.382	1.9116	1.8533

2) Drone Racing Competition Under Our Assistive System: After practicing in simulations, five participants are invited to compete with each other in a real world drone racing scene. For two trails of each pilot, the shortest completion time is recorded and used to determine the final rank. All flight trajectories and captured shots are shown in Fig. 9 and the rank information is demonstrated in Table. I. All pilots successfully complete the racing in the real world

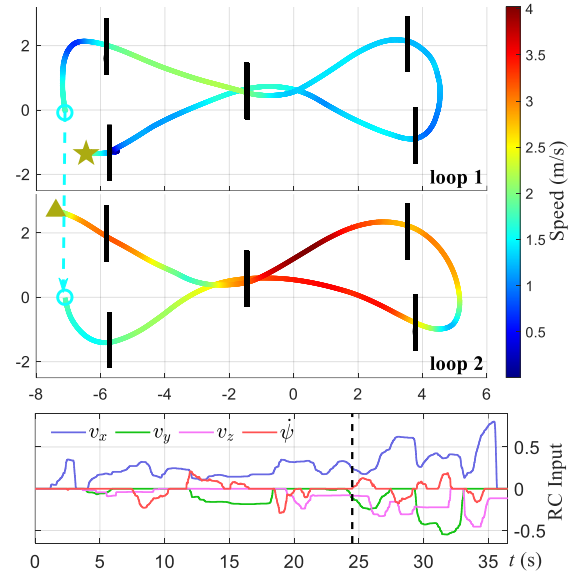

Fig. 8. Demonstration of adaptation test. (Top) The trajectory in the full-course flight is separated according to different loops, and the color is varied with flight speed profile. The star and triangle denote the start point and end point. (Bottom) The RC inputs are illustrated along the time serial, and the black dotted line represents the completion of loop one.

scene, achieving a maximum speed greater than 3.0 m/s. The pilot who wins the competition demonstrates an impressive performance, attaining a maximum flight speed of 6.0 m/s and an average flight speed close to 3.0 m/s. More experiment validations can be found in the attached video.

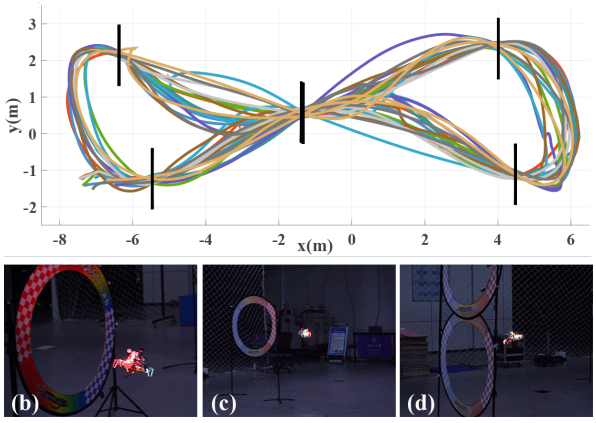

Fig. 9. Demonstration of real world flight. (a) All the flight trajectories by pilots in different colors. (b)-(d) Captured snapshots of the drone flight in racing.

VII. Conclusion and Future Work

In this paper, we propose a trajectory-based flight assistive system to bridge the gap between drone racing and novice pilots. The key property of our system is that it seamlessly assists pilots to fly the drone with safety insurance while encouraging them to fly at a higher speed in racing sense by solving a joint trajectory optimization problem. Both simulations and real world experiments validate the performance of our system, and the flight speed of the drone operated by a novice pilot can reach 6.0 m/s in a real world racing scene. In future work, we will implement our system in outdoor environments to realize a drone racing in the wild.

REFERENCES

[1] Q. Wang, B. He, Z. Xun, C. Xu, and F. Gao, "Gpa-teleoperation: Gaze enhanced perception-aware safe assistive aerial teleoperation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5631–5638, 2022.

[2] X. Yang and N. Michael, "Assisted mobile robot teleoperation with intent-aligned trajectories via biased incremental action sampling," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 998–11 003.

[3] X. Yang, J. Cheng, and N. Michael, "An intention guided hierarchical framework for trajectory-based teleoperation of mobile robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 482–488.

[4] Z. Han, Z. Wang, N. Pan, Y. Lin, C. Xu, and F. Gao, "Fast-racing: An open-source strong baseline for $\mathrm{SE}(3)$ planning in autonomous drone racing," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8631–8638, 2021.

[5] Q. Wang, D. Wang, C. Xu, A. Gao, and F. Gao, "Polynomial-based online planning for autonomous drone racing in dynamic environments," *arXiv preprint arXiv:2306.14461*, 2023.

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

[7] W. Standaert, "Digital growth strategies at drone racing league," *Journal of Information Technology Teaching Cases*, vol. 11, no. 1, pp. 2–7, 2021.

[8] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "Flightgoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6941–6948.

[9] J. A. Cocoma-Ortega and J. Martínez-Carranza, "Towards high-speed localisation for autonomous drone racing," in *Advances in Soft Computing: 18th Mexican International Conference on Artificial Intelligence, MICAI 2019, Xalapa, Mexico, October 27–November 2, 2019, Proceedings 18*. Springer, 2019, pp. 740–751.

[10] C. Pfeiffer and D. Scaramuzza, "Human-piloted drone racing: Visual processing and control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3467–3474, 2021.

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

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

[13] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.

[14] M. Nieuwenhuisen, D. Droeschel, J. Schneider, D. Holz, T. Läbe, and S. Behnke, "Multimodal obstacle detection and collision avoidance for micro aerial vehicles," in *2013 European Conference on Mobile Robots*. IEEE, 2013, pp. 7–12.

[15] M. Odelga, P. Stegagno, and H. H. Bülthoff, "Obstacle detection, tracking and avoidance for a teleoperated uav," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 2984–2990.

[16] Z. Zhang, H. Chen, S. W. Lye, and C. Lv, "Human-guided safe and efficient trajectory replanning for unmanned aerial vehicles," in *2022 International Conference on Control, Automation and Diagnosis (ICCAD)*. IEEE, 2022, pp. 1–6.

[17] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.

[18] S. Liu, N. Atanasov, K. Mohta, and V. Kumar, "Search-based motion planning for quadrotors using linear quadratic minimum time control," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 2872–2879.

[19] J.-C. Pont, *Collected Works of Charles François Sturm*. Springer, 2009.

[20] J. Ji, X. Zhou, C. Xu, and F. Gao, "Cmpcc: Corridor-based model predictive contouring control for aggressive drone flight," in *Experimental Robotics: The 17th International Symposium*. Springer, 2021, pp. 37–46.

[21] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3259–3278, 2022.