# Human-Guided Safe and Efficient Trajectory Replanning for Unmanned Aerial Vehicles

Zezhong Zhang
*Mechanical and Aerospace Engineering*
*Nanyang Technological University*
Singapore
zezhong002@e.ntu.edu.sg

Hao Chen
*Mechanical and Aerospace Engineering*
*Nanyang Technological University*
Singapore
chen.h@ntu.edu.sg

Sun Woh Lye
*Mechanical and Aerospace Engineering*
*Nanyang Technological University*
Singapore
mswlye@ntu.edu.sg

Chen Lv
*Mechanical and Aerospace Engineering*
*Nanyang Technological University*
Singapore
lyuchen@ntu.edu.sg

***Abstract*—Safe and efficient local trajectory replanning is essential for the navigation of unmanned aerial vehicles (UAVs). Take the quadrotor as an example, most research works focus on the static or fully mapped environment. Flying in a dynamic environment for autonomous quadrotors is still a tricky problem. However, with the emergence of first-person-view Drone Racing in recent years, professional human pilots have shown highly-skilled techniques for navigating quadrotors to avoid collisions at high speed. Therefore, this work uses the intelligence of human users in perception and decision-making and proposes a human-guided trajectory replanning (HTP) system for the safe and efficient flight operation of quadrotors. A non-constraint optimization problem is formulated, and human guidance is designed as one term of the cost functions. The proposed approach is validated in the AirSim simulation environment. The result shows that HTP saves optimization time by 58% compared with the non-human guidance (Non-HG) baseline. In addition, the HTP can assist quadrotors to pass the specified target at a higher speed and comply better with human preferences than the Non-HG approach.**

***Index Terms*—human-machine collaboration, trajectory replanning, UAV**

## I. Introduction

Autonomous navigation in unfamiliar dynamic environments has been recognized as one of the ten most challenging tasks in robotics in recent years [1]. Many navigation algorithms are proposed to handle navigation problems in static and well-mapped environments. However, efficient and safe real-time trajectory replanning is still an open question for dynamic environments with complex terrains, cluttered obstacles, or changes to the original map.

Unmanned Aerial Vehicles (UAVs) can be deployed in various application scenarios such as power line fault inspection, cave exploration, and emergency delivery. However, the limited carrying capacity makes onboard sensors and computers of quadrotors as compact as possible, which significantly limits their computing power and perception capabilities. In addition, due to the fragility of rotor blades, quadrotors can hardly withstand even slight collisions during flight.

In recent years, with the rise of the FPV quadrotor competition, researchers have found that professional human pilots can stably navigate the quadrotor equipped with only simple sensors to perform obstacle avoidance and large-angle maneuvering flight in complex environments, with a speed up to 40 m/s. Up to now, no quadrotor controlled by a fully autonomous navigation algorithm can match the performance of human professional pilots. However, an expert pilot needs years of training, which puts forward a high threshold for users.

According to the latest study, the success of human experts lies in making decisions early, usually 1.5 seconds in advance [2]. Human experts can beat state-of-the-art autonomous navigation control algorithms in both decision-making and execution by virtue of advanced prediction and decision-making, supplemented by long-term control training. In addition, human vision is more robust to changes in light intensity or disturbances such as water mist and smoke than the current mainstream ranging sensors (depth cameras and lidars). Therefore, this work accomplishes a complementary trajectory replanning system for navigation in dynamic environments, by combining the superiorities of human users and autonomous systems. Different from previous teleoperated systems in the field of mobile robots [3]–[5], our collaboration system neither regards human instructions as the final decision, nor entirely depends on the autonomous system. The former relies heavily on human judgment, whereas the latter ignores humans strengths. In our understanding, a human-machine collaboration (HMC) system should be capable of utilizing their respective strengths and achieving the effect of "$1 + 1 > 2$".

A human-guided trajectory replanning (HTP) system is proposed in this work, which incorporates human intent into the time-parameterized optimization-based framework. HTP enhances sensors performance and accelerates the optimization

This work was supported in part by A*STAR project (No. W1925d0046), and the SUG-NAP project of Nanyang Technological University.

process so that the quadrotor can find a safe flight trajectory. The state-of-the-art autonomous trajectory replanning system [6] is chosen as the baseline and compared with the proposed HTP approach.

The main contributions of the current research are as follows:

- The human-guided trajectory replanning (HTP) system is proposed and tested in the quadrotor simulation platform. Furthermore, this framework can be applied to other mobile robot applications.
- The HTP problem is solved by formulating a time-parameterized B-spline optimization problem. The proposed system can converge to optimal results faster and fly with higher security in line with human intent.
- A reliable and straightforward interaction interface using the mouse cursor is developed to collect human intent.

## II. RELATIVE WORKS

In this section, we first review autonomous trajectory replanning approaches for quadrotors. After that, we analyze relative HMC studies for UAVs and point out the current research gap.

### A. Autonomous Trajectory Replanning for UAVs

Trajectory replanning for the quadrotor has become a hot topic in recent years [6]–[8]. Researchers are trying to increase flight speed under the premise of ensuring safety. The trajectory replanning approaches for quadrotors can be mainly divided into two streams: 1) optimization-based; 2) motion-primitives and path sampling. In the first stream, a shortest and collision-free path is firstly found by sampling or search methods, such as RRT or A*. Then cost functions with various concerns will be designed to formulate an optimization problem. For example, mini-snap [9] and polynomial optimization [10] formulate the problem as an unconstraint QP problem. Based on these two milestones studies, a time-parameterized real-time local trajectory replanning system is proposed [6]. B-spline curve is used to represent the replanned trajectory due to its local control property. The local trajectory is replanned by solving the unconstraint optimization problem iteratively in the flight process. Such heavy computation limits their flight speed to around 1m/s. Later, EGO-planner [8] shortens overall computing time by removing the building of Euclidian Distance Field (EDF). Nevertheless, the computation for optimization computing is still heavy.

The second stream is based on motion primitives and path sampling. Impressive rapid trajectory generation for badminton ball hitting [11] and catching [12] are achieved under this approach. But they rely on the state estimation (such as motion capture system) heavily and the success of finding a feasible trajectory depends on the selected discretization methods.

### B. Human-Machine Collaboration for UAV Operation

Several previous studies of collision avoidance have been developed for manually teleoperated UAVs [4] [5]. They predict possible collisions and push quadrotors to the opposite direction. The assisted system can hardly handle cluttered environments. Such assistance is reactive movement instead of real trajectory planning. Besides, human intent is not fully respected because the collision avoidance module only considers surrounding environment information.

Recently, Yang et al. [13] regard the human intent as guiding direction and model it as one cost function. They incorporate human intent with motion-primitives methods to generate a smooth trajectory. This approach cannot guarantee the optimal trajectory due to the inherent discretization limitation. Besides, the Remote Controller (RC) is not a user-friendly interface to express human intent directly. Wang et al. [14] adopt eye tracker and RC to capture users' expected direction and flight speed, respectively. One topological path will match extracted gaze point, and the path will be refined by following optimization module. However, due to the eyes blinking and erratic glance, the noise of gaze influences the system input dramatically. Proper expected speeds are also highly dependent on accurate control of RC lever users, which is not easy for fresh users without much training.

Though some works have studied human factors for solving the urgently-needed real-time local trajectory replanning problems, reliable interaction interfaces and efficient HMC systems are still open to further research. Our proposed approach falls into the optimization-based approaches. We adopt the users' mouse clicked point as a roughly guided direction and use it to accelerate the optimization of the B-spline time-parameterized trajectory in real-time. The human-machine interface is robust and straightforward to deploy, and the system does not require any precise control from users.

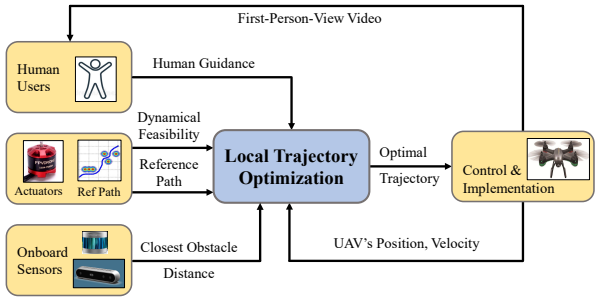



Fig. 1. The schematic diagram of system overview. The local trajectory optimization module is the key module developed in this work.

## III. SYSTEM OVERVIEW

The overall system is a general HMC framework for the navigation of UAVs, as illustrated in Fig. 1. The objective users of our system are those fresh users who receive a 30 minutes simple training.

In this HMC framework, the globally optimized reference path, computed through the previous study [10], is divided into many segments. UAVs will follow the reference path at first. The local trajectory optimization module will be triggered to replan the local trajectory of corresponding segments when the human users intervene, or obstacles are detected on the reference path.

The interaction interface is quite straightforward in our system. Human users watch the first-person-view video from onboard cameras. They are required to click the desired area on the screen by using a regular mouse cursor. Then the clicked point will be captured as the flight direction guidance.

The commonly used distance sensors such as depth and RGB-D cameras are used for collision-checking. We computed the scanned environment's Euclidean Distance Field (EDF) by using the algorithm proposed by Felzenszwalb and Huttenlocher [15]. Furthermore, we adopt trilinear interpolation to query the closest distance to obstacles of each grid and calculate the gradient of distance with respect to grid position [6].

By utilizing the rough perception and judgment at a long distance of human users and the precise measurement at close range of onboard sensors, the two complement shortcomings of each other and contribute to the local trajectory optimization. After that, the UAV is controlled by a fine-tuned PID controller to implement the optimal trajectory.

## IV. TRAJECTORY GEOMETRICAL REPRESENTATION

B-spline is selected as the representation curve for trajectory replanning due to its excellent local control and convex hull properties. To simplify the calculation and satisfy the fourth continuous trajectory, we use a quintic uniform B-spline curve in this work, which means that every pair of two subsequent nodes is equally spaced.

Given $n+1$ control points $q_i (i = 0, \ldots, n)$, degree $k$, and the knot vector $\{t_0, \ldots, t_{n+k+1}\}$ where $t_0 \leq \cdots \leq t_{n+k+1}$ are knots, the B-spline curve $f(t)$ can be computed using the following function:

$$f(t) = \sum_{i=0}^{n} Q_i N_i^k(t) \tag{1}$$

where $N_i^k(t)$are the basis functions with piecewise degree $k$ polynomials. The $t \in [t_k, t_{n+1}]$ represents the real-time sequence of curve starts from $t_k$, and ends at $t_{n+1}$; there are totally $(n+1-k)$ segments for the whole trajectory.

Since the trajectory replanning module is triggered when the human intervenes or unexpected obstacles block the reference path, we do not need to change the whole trajectory all the time. In this case of the degree $k = 5$, each point along the uniform B-spline is dependent on six control points. The local control property focuses on areas close to human-guided areas or obstacles and saves computation resources for unaffected areas along the reference path. Besides, each point of B-spline trajectory lies within the convex hull of $(k+1)$ neighboring control points, which is an apparent advantage for optimization solving. In addition, the basis function $N_i^k(t)$ is $C^{k-m}$ continuous at a knot of multiplicity m. This property removes the continuity constraints at the junction between any two segments, which is superior to the polynomial and Bezier curves.

For the segment $t \in [t_i, t_{i+1}]$, where $i \in [k, n]$, the control points set is $Q_i = [q_{i-5}, q_{i-4}, q_{i-3}, q_{i-2}, q_{i-1}, q_i]^T$. We define $u(t) = (t - t_i)/\Delta t$, which ranges from 0 to 1, corresponding to the start to the end of each segment, where $\Delta t = t_{i+1} - t_i$ is the constant interval. Then, the De Boor-Cox formula can be represented as matrix form:

$$f(u(t)) = U^T M^6 Q_i \tag{2}$$

where $U = \left[1, u(t), u(t)^2, u(t)^3, u(t)^4, u(t)^5\right]^T$, and $M^6 \in \mathbb{R}^{6\times 6}$ is a constant matrix [16].

It is straightforward to compute the derivative with respect to time or control points with (2). The integral of squared time derivatives can be solved in the closed form, take the integral over squared jerk as an example, which will be used for designing the smoothness penalty in section V.

$$L_J = \int_{t_{\min}}^{t_{\max}} \omega_J \left(f'''(t)\right)^2 dt = \lambda_J {Q_i}^T {M^6}^T H_J M^6 Q_i \tag{3}$$

where $\omega_J$ is the weight parameter of jerk, and

$$H_J = \frac{1}{\Delta t^5} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 36 & 72 & 120 \\ 0 & 0 & 0 & 72 & 192 & 360 \\ 0 & 0 & 0 & 120 & 360 & 720 \end{bmatrix} \tag{4}$$

In the case of uniform B-spline, the hessian matrix $H_J$ is constant and can be calculated in advance. The hessian matrixes of second or fourth derivatives can also be computed using the same way.

## V. TRAJECTORY REPLANNING PROBLEM FORMULATION

### A. Overall Cost Function

The local trajectory replanning can be modeled as a non-linear optimization problem without constraints. The cost function includes five terms and is computed as follows:

$$L_{Total} = L_{hg} + L_{col} + L_{ref} + L_{smooth} + L_{fea} \tag{5}$$

where $L_{hg}$ is the penalty of deviation between the endpoint of each replanned segment and the human clicked point; $L_{col}$ is the penalty of possible collision; $L_{ref}$ is the penalty of deviation between the replanned trajectory and reference path. The desired point is confirmed by sharing the same time parameter with the endpoint of replanned trajectory. $L_{smooth}$ is the penalty of trajectory smoothness; $L_{fea}$ is the penalty for violation of dynamical feasibility of quadrotors. Users can assign different weights to each terms according to situations and task priority. In this paper, we set higher weights to the human preference and collision avoidance.

### B. Human Guidance Penalty

The purpose of human guidance penalty is to guide the replanned trajectory to approach the human clicked point. We define this cost term as:

$$L_{hg} = \omega_{hg} \left(f\left(t_{hg}\right) - p_{hg}\right)^2 \tag{6}$$

where $f(t_{hg})$ is the endpoint position of each replanned trajectory, $p_{hg}$ is the human clicked point position, $\omega_{hg}$ is the weight parameter.

In the abovementioned human-machine interface, the human clicked point is 2-D information, including horizontal and vertical position on the front view plane of the quadrotor. However, the depth information is calculated by multiplying the real-time velocity of quadrotor and a prediction time of human. According to work [2], expert pilots usually predict 1.5s in advance for avoiding collisions. By considering our test subjects are fresh users, we adjust the value as 1s.

### C. Other Penalty Terms

*1) Collision penalty:* The aim of collision penalty is to push trajectory away from obstacles. We defined the cost term as:

$$\begin{aligned} L_{col} &= \omega_{col} \int_{t_{\min}}^{t_{\max}} g(f(t)) ds \\ &= \omega_{col} \int_{t_{\min}}^{t_{\max}} g(f(t)) \left\| f'(t) \right\| dt \end{aligned} \tag{7}$$

where $f(t)$ is the position of the point at time $t$, $ds$ is the differential element of trajectory, $\|f'(t)\|$ is size of the real-time velocity, $g(x)$ is a continuous punishment function of the distance, which is defined by using below function. The input x is position of quadrotors, $d(x)$ is the closest distance between quadrotors and obstacles, $\epsilon$ is the safe threshold.

$$g(x) = \begin{cases} \frac{1}{2\epsilon}(d(x)-\epsilon)^2 & \text{if } d(x) \le \epsilon \\ 0 & \text{if } d(x) > \epsilon \end{cases} \tag{8}$$

Equation (7) defines the penalty as the multiplication of distance punishment and the differential element of trajectory instead of differential element of time. Because if using the latter one, possible results of collision penalty term may minimize this term by shortening the time interval, which will increase the speed of quadrotors and make the flight more dangerous. However, the use of differential elements of trajectory can push trajectory away from obstacles, which obeys our aims better.

*2) Reference path penalty:* Reference path penalty is designed to pull the replanned trajectory to reference one since the reference path is pre-optimized. The replanned trajectory will overlap with the reference if there are no human interventions and obstacles. By choosing the endpoint of each segment as expected point, we define the cost term as follows:

$$L_{\text{ref}} = \omega_P \left( f\left(t_{\text{exp}}\right) - p_{\text{exp}} \right)^2 + \omega_V \left( f'\left(t_{\text{exp}}\right) - p'_{\text{exp}} \right)^2 \tag{9}$$

where $t_{exp}$ is the end time of each segment, $f(t)$ is the trajectory to be optimized, $p_{exp}$ and $p'_{\text{exp}}$ are position and velocity of the expected point along the reference path, respectively. $\omega_P$ and $\omega_V$ are weight parameters.

*3) Smoothness penalty:* The acceleration and jerk of trajectory are considered in the smoothness term. They can be computed in a close form, as illustrated in section IV. The smoothness penalty is defined as follows:

$$L_{\text{smooth}} = \omega_A \int_{t_{\min}}^{t_{\max}} f''(t)^2 dt + \omega_J \int_{t_{\min}}^{t_{\max}} f'''(t)^2 dt \tag{10}$$

where $f''(t)$ and $f'''(t)$ are second and third derivates of trajectory, respectively. $\omega_A$ and $\omega_J$ are weight parameters.

*4) Dynamical Feasibility penalty:* The Dynamical Feasibility penalty is designed in case the replanned trajectory exceeds the physical limitations of quadrotors. We design this soft constraint by using the cubic function.

$$\begin{aligned} L_{fea} = \omega_{fea} * \\ \sum_{i=1}^{2} \int_{t_{\min}}^{t_{\max}} \max \left( 0, \left( \left( f^{(i)}(t) \right)^2 - \left( p_{\max}^{(i)} \right)^2 \right)^3 \right) dt \end{aligned} \tag{11}$$

where $f^{(i)}(t)$ is the $i_{th}$ derivates of trajectory and ${p_{\max}}^{(i)}$ are maximum of speed and acceleration, respectively. We substitute 90% of the maximum speed and acceleration in the real implementation because some power resources must be reserved for feasible trajectories.

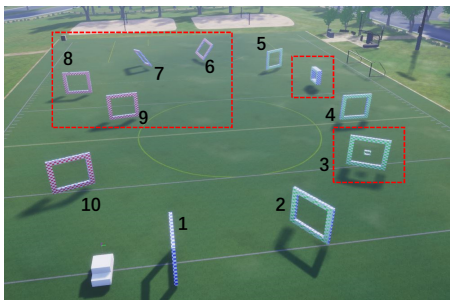



Fig. 2. The changed environment for testing trajectory replanning algorithms. Gates are placed in order in the original map, where the reference path is built from. Thereafter, the new map is obtained through translating Gate 8&9; rotating Gates 6&7, and adding new obstacles at Gate3 and the area between Gates 4&5.

### D. Gradient of Cost Functions

The gradients of cost functions are straightforward for most terms except collision penalty since the distance function is not continuous. First, we discretize each segment by dividing it into n sub-segments, $\delta t = \Delta t / n$ is the time duration for one sub-segment. Then the collision penalty is:

$$L_{\text{col}} = \omega_{\text{col}} \sum_{i=1}^{n} g\left(f\left(t_i\right)\right) \left\| f'\left(t_i\right) \right\| \delta t \tag{12}$$

The gradient is computed as follows:

$$\begin{aligned} &\frac{\partial L_{col}}{\partial Q_{i(\mu)}} \\ &= \omega_{col} \sum_{i=1}^{n} \left\{ \frac{\partial g\left(f\left(t_i\right)\right)}{\partial Q_{i(\mu)}} \left\| f'\left(t_i\right) \right\| + g\left(f\left(t_i\right)\right) \frac{\partial \left\| f'\left(t_i\right) \right\|}{\partial Q_{i(\mu)}} \right\} \delta t, \\ &= \omega_{col} \sum_{i=1}^{n} \left\{ \frac{1}{\epsilon}(d(x)-\epsilon) \frac{\partial d\left(f\left(t_i\right)\right)}{\partial f\left(t_i\right)} \frac{\partial f\left(t_i\right)}{\partial Q_{i(\mu)}} \left\| f'\left(t_i\right) \right\| + \right. \\ &\left. g\left(f\left(t_i\right)\right) \frac{f'\left(t_i\right)_{(\mu)}}{\left\| f'\left(t_i\right) \right\|} \frac{\partial f'\left(t_i\right)}{\partial Q_{i(\mu)}} \right\} \delta t, \quad \mu \in \{x, y, z\} \end{aligned} \tag{13}$$

Then, we can computed the desired gradient of collision penalty. The Classical algorithm such as MMA [17] is applied to solve this unconstraint optimization problem.

## VI. Experiments and Results

### A. Experiment Setup and Design

The HTP algorithm is tested on the high-fidelity simulation platform AirSim [18]. The dynamic soccer field environment shown in Fig. 2 is from the AirSim Drone Racing Lab [19]. The non-linear optimization solver NLopt [20] is used to solve the above optimization problem. For the hardware, the intel core i7-9700 CPU is used for computing.

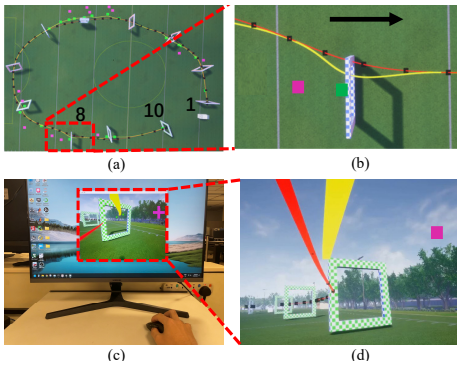



Fig. 3. The diagram of overall replanned trajectory and users operation interface. (a) the overall top view of the test environment. (b) One typical situation of trajectory replanning: the top view of Gate8. Red and yellow curves are reference and replanned trajectories, respectively. The pink, green and black points are human-guided point, re-optimized control point and reference control points respectively (below figures use the same legend). The black arrow indicates flight direction. (c) user operation interface: the pink crossing is controlled by human through the mouse cursor; (d) the first-person-view (FPV) video provided for users.

In this task, quadrotors are required to pass through Gate1 to Gate10 with no collisions and minimum time-consuming. The user interface is illustrated in Fig. 3(c) and Fig. 3(b). To test the trajectory replanning capability, we changed the original soccer field map. Obstacles are added along with the reference, and gates' positions and orientations are changed so that following the original reference path will cause collisions or miss some gates. The state-of-the-art B-spline autonomous replanning algorithm [6] is selected as the Non-HG baseline to compare with our proposed HTP approach. In all tests, flight speed is adjusted by the setting of overall time of the whole trajectory; and the time assigned to each sector between two gates is based on their distance. We set default values as: 1) the sensing distance of quadrotors is 2.5m; 2) safe distance threshold $\epsilon$ for collision checking is 0.25m; 3) soccer field map is a 55m×40m×5m space with a 0.05m grid resolution.

### B. Comparison of Collision Avoidance

We compare the performance of HTP and Non-HG approaches from three aspects: obstacle avoidance, human intent compliance, and optimization time. After ten repeated tests at each flight speed (3.75, 4.30, 5.00, 6.00 $\mathrm{m/s}$), when the total flight time is set at $T$ = 25s, that is, when the average speed is $\bar{v} = 6.00$ $\mathrm{m/s}$, the success rate of avoiding obstacles remains above 50%. One case is shown in Fig. 6(a); while the Non-HG approach can only avoid obstacles successfully when $T$ = 35s, that is $\bar{v} = 4.30$ $\mathrm{m/s}$, as shown in Fig. 6(c); if faster, a collision will occur, as shown in Fig. 6(b). In our test environment, HTP can effectively increase flight speed by up to 40% compared to the Non-HG approach.

### C. Comparison of Human Intent Compliance

In the case of $\bar{v} = 4.30$ $\mathrm{m/s}$ ($T$ = 35s) , even though the Non-HG approach can navigate quadrotors without collisions, the replanned trajectory does not pass through the gate, as shown in Fig. 6(c). However, the trajectory planned by HTP not only avoids obstacles, but also passes through the gate in compliance with the intent of human users, as shown in Fig. 6(a). Especially, HTP is more flexible than the Non-HG approach when human intent changes from time to time.

### D. Comparison of Optimization Time

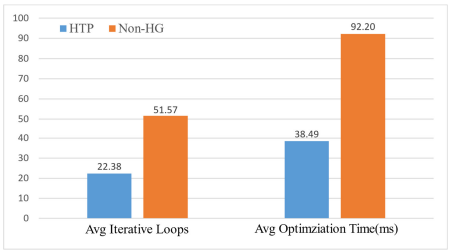



Fig. 4. Comparison of the average computation time of each segment between HTP and Non-HG approaches.

In the form of a B-spline trajectory replanning, we divided the entire path into 60 segments, and each length is about 2.5m. The replanning algorithm will optimize the segment curve when triggered. As shown in Fig. 4, the average optimization time (38.49 ms) of HTP in each segment is 58% faster than the performance of the Non-HG method (92.20 ms). The number of optimization iterations per segment of 22.38 laps of HTP also significantly outperforms the Non-HG performance of 51.57 laps.

### E. Analysis

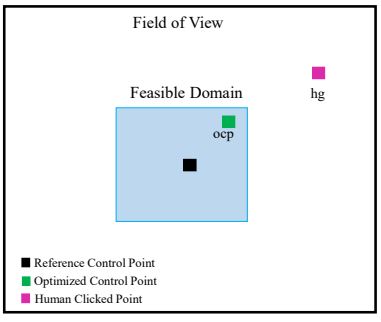
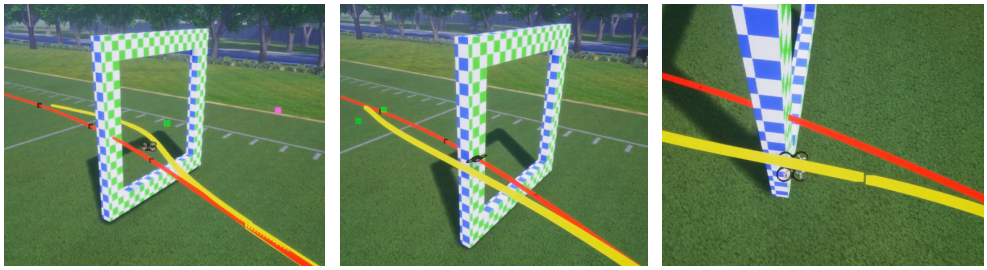



Fig. 5. The diagram of human-guided optimization process.



(a) (b) (c)

Fig. 6. The situation at Gate 8, where reference trajectory collides with gate. (a) HTP replans a safe trajectory (yellow curve) that passes through the gate under the influence of human guidance (pink point), in the case of total flight time is set at $T$=25s; (b) Non-HG replans an unsafe trajectory that leads to a collision, $T$=25s; (c) when prolonging flight time ($T$=35s), Non-HG replans a collision-free trajectory but miss the gate.

This section attempts to explain the question: "why does human guidance shorten the optimization time?".

Fig. 5 illustrates an optimization process with human guidance. Since the weight of human guidance is set much higher than that of the reference path, when the human intervenes the replanning process, the clicked point gives a strong guidance for the optimization direction. Thus, the control point to be optimized will quickly move along the direction to the human clicked point, and the iterative search stops when it reaches the edge of feasible domain. Then, the optimal solution can be found in a short time.The experiment results validate the above analysis, which shows that HTP approach consumes 26.30 ms for one replanning, however, the Non-HG approach takes 91.94 ms.

It should be noted that the above case is an simplified example without obstacles. In more general situations, the optimization performance of HTP is influenced by all penalty terms in section V. Besides, the choice of the feasible domain size is a compromise between optimization space and computation time, which should be set based on specific tasks.

## VII. Conclusion

This paper proposes a human-guided trajectory replanning approach for UAVs in dynamic environments. The human guidance is integrated into the non-linear unconstrained optimization problem and modelled as one term of cost functions. Relative experiments show that the proposed HTP approach outperforms the baseline autonomous trajectory replanning approach in collision avoidance, human intent compliance, and optimization time. In the future, more experiment trails and complex environments with moving obstacles will be used to test the proposed approach.

## References


[1] G.-Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield *et al.*, "The grand challenges of science robotics," *Science robotics*, vol. 3, no. 14, p. eaar7650, 2018.

[2] C. Pfeiffer and D. Scaramuzza, "Human-piloted drone racing: Visual processing and control," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 03 2021.

[3] J. Israelsen, M. Beall, D. Bareiss, D. Stuart, E. Keeney, and J. van den Berg, "Automatic collision avoidance for manually tele-operated unmanned aerial vehicles," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6638–6643.

[4] D. Bareiss, J. van den Berg, and K. K. Leang, "Stochastic automatic collision avoidance for tele-operated unmanned aerial vehicles," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4818–4825.

[5] S. J. Anderson, S. B. Karumanchi, K. Iagnemma, and J. M. Walker, "The intelligent copilot: A constraint-based approach to shared-adaptive control of ground vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 2, pp. 45–54, 2013.

[6] V. Usenko, L. Von Stumberg, A. Pangercic, and D. Cremers, "Real-time trajectory replanning for mavs using uniform b-splines and a 3d circular buffer," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 215–222.

[7] P. Foehn, D. Brescianini, K. E *et al.*, "Alphapilot: autonomous drone racing." *Autonomous Robot*, vol. 46, p. 307–320, 2022.

[8] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2020.

[9] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 2520–2525.

[10] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics research*. Springer, 2016, pp. 649–666.

[11] M. Hehn and R. D'Andrea, "Quadrocopter trajectory generation and control," *IFAC proceedings Volumes*, vol. 44, no. 1, pp. 1485–1491, 2011.

[12] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE transactions on robotics*, vol. 31, no. 6, pp. 1294–1310, 2015.

[13] 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.

[14] Q. Wang, B. He, Z. Xun, C. Xu, and F. Gao, "Gpa-teleoperation: Gaze enhanced perception-aware safe assistive aerial teleoperation," *arXiv preprint arXiv:2109.04907*, 2021.

[15] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of computing*, vol. 8, no. 1, pp. 415–428, 2012.

[16] K. Qin, "General matrix representations for b-splines," in *Proceedings Pacific Graphics' 98. Sixth Pacific Conference on Computer Graphics and Applications (Cat. No. 98EX208)*. IEEE, 1998, pp. 37–43.

[17] K. Svanberg, "A class of globally convergent optimization methods based on conservative convex separable approximations," *SIAM journal on optimization*, vol. 12, no. 2, pp. 555–573, 2002.

[18] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*. Springer, 2018, pp. 621–635.

[19] M. Ratnesh, G. Nicholas, V. Sai *et al.*, "Airsim drone racing lab," 2020. [Online]. Available: https://arxiv.org/abs/2003.05654

[20] S. G. Johnson, "The nlopt nonlinear optimization package," 2022. [Online]. Available: https://nlopt.readthedocs.io/en/latest/