EVA-Planner: Environmental Adaptive Quadrotor Planning

Lun Quan, Zhiwei Zhang, Chao Xu and Fei Gao

Abstract— **Due to its superior agility and flexibility, quadrotor is popularly used in challenging environments. In these scenarios, trajectory planning plays a vital role in generating safe motions to avoid obstacles while ensuring flight smoothness. Although many works on quadrotor planning have been proposed, a research gap exists in incorporating self-adaptation into a planning framework to enable a drone to automatically fly slower in denser environments and increase its speed in a safer area. In this paper, we propose an environmental adaptive planner that effectively adjusts the flight aggressiveness based on the obstacle distribution and quadrotor state. Firstly, we design an environmental adaptive safety-aware method to assign the priority of obstacles according to the environmental risk level and instantaneous motion tendency. Then, we apply it into a multi-layered model predictive contouring control framework to generate adaptive, safe, and dynamical feasible local trajectories. Extensive simulations and real-world experiments verify our planning framework's efficiency and robustness and show superior performances in the benchmark comparison. Moreover, we will release our planning framework as open-source ros-packages[1].**

I. Introduction

Nowadays, quadrotors are increasingly used in dangerous scenarios such as mine exploration, quick response rescue, and target search. These applications may be difficult and dangerous for human beings and have high demands on the drone's autonomy and robustness.

Although there are extensive works on quadrotor motion planning, the self-adaptation of aggressiveness is seldom considered. Imagine this situation, for a quadrotor flies in a forest, as it moves more aggressively, there may occur more deviations in the state estimation, perception, and control. These errors result in a higher probability of a crash, especially while the quadrotor navigates between multiple obstacles in a narrow space. In contrast, if the quadrotor is operating in a wide-open space, a large safety margin naturally promises that the drone can fly faster. Moreover, in this scenario, the quadrotor should fly fast instead of limit its speed conservatively. Based on the above analysis, an ideal planner needs to leave sufficient planning margin by automatically adjusting its aggressiveness according to the degree of risk, which is directly decided by the obstacle distribution and quadrotor dynamics.

To this end, we propose an environmental adaptive quadrotor planning method, named EVA (**E**n**V**ironmental **A**daptive)-Planner, which effectively adjust the flight aggressiveness based on multi-layered adaptive planning. Our method takes the environmental risk level and instantaneous motion tendency into account. In this way, quadrotors can adaptively assign the priority of obstacles around it and plan a more safe and efficient flight trajectory. This method is implemented with a unified planning framework extending our recent results on quadrotor model predictive contouring control [1]. We summarize our contributions as:

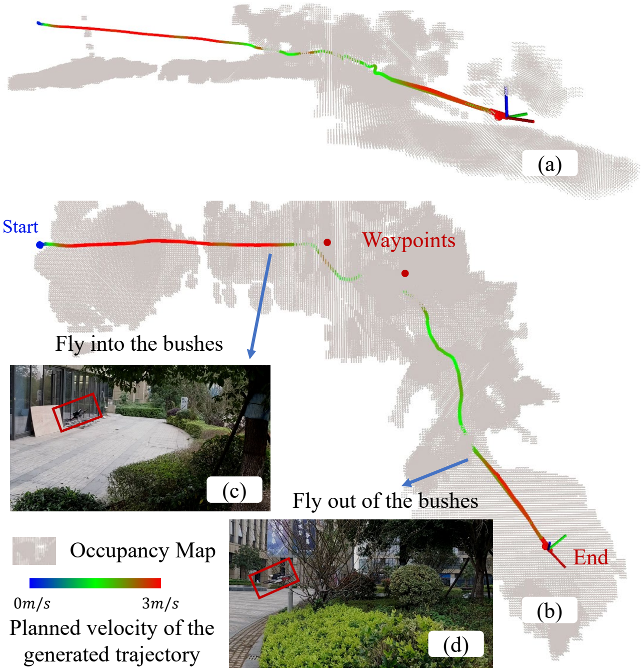

Fig. 1. (a) and (b) are the side and vertical views of the outdoor experiment. (c) and (d) are the snapshots of critical moments. The curve represents the generated trajectory, and the gradient color is the planned velocity. This experiment is described in detail in Sec. V-C. Video is available at `https://github.com/ZJU-FAST-Lab/EVA-planner`.

1) An environmental adaptive safety aware method is proposed with a reasonable judgment of the danger degree of surrounding obstacles according to the environmental information and the system's motion tendency.
2) A unified planning framework for generating aggressiveness-adaptive, obstacle-free, smooth, and dynamics feasible trajectory using multi-layered model predictive contouring control.
3) Extensive simulations and real-world experiments verify the robustness of our method. Benchmark shows the efficiency of our method in planning fast and safe trajectories.

This work is supported by Zhejiang University FAST Lab. All authors are with the State Key Laboratory of Industrial Control Technology, and the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, 310027, China. `lunquan@zju.edu.cn`, `fgaoaa@zju.edu.cn`

[1] https://github.com/ZJU-FAST-Lab/EVA-planner

The rest of the paper is organized as follows, related works are discussed in Sec.II. Our method for environmental adaptive safety aware is stated in Sec.III, and the multi-layered planning framework is described in detailed in Sec.IV. Simulation and real-world experiments results are discussed in Sec.V. This paper is concluded in Sec.VI.

II. Related works

A. Autonomous navigation algorithms

The problem of online autonomous navigation in the 3D complex environment has been investigated extensively. According to the planner structure, the planning algorithms can be divided into direct and hierarchical methods.

Direct methods plan trajectories on the abstract environment, which requires little computing power. By establishing a safe flight corridor (SFC) directly on depth map [2] or point clouds [3], trajectory planning can be carried out without mapping. Nanomap [5] generates rough k-d trees for collision check. However, direct methods tend to fail in a complex environment due to the memoryless of the map. Hierarchical methods such as [6]–[9] employ a planning framework with front-end path searching and back-end trajectory optimizing, which transforms the trajectory generation to a nonlinear optimization problem and formulate is as trading off smoothness, obstacle-free, and dynamical feasibility simultaneously. However, the executed trajectory may hit the obstacles due to the tracking errors caused by the high-speed motion, even if the desired obstacle-free trajectory is planned.

B. Adaptive fast and safe planning

FASTER [10] plans a safe trajectory and a fast trajectory at each time. Fast and Safe flight is realized through practical replanning design. However, the proposed method is inefficient and can only be used for planning in a small range. Some works mention the importance of considering the geometry of configuration space and system dynamics simultaneously, but few have shown convincing performance for a quadrotor in 3D dense environments. A planning framework is proposed in [11] with offline precomputed safety margins relative to different system dynamics and an online generated tree of trajectories that adaptively search a proper planning model. However, if the quadrotor needs to fly at high speed, substantial pre-computed models prevent it from being searched online. A new state-dependent directional metric is designed in [12] to adaptively adjust the influence of the environment on the system according to the vehicle's velocity. This metric changes the rule of collision checking, which is according to the current velocity of the system. In this way, obstacles parallel with velocity are considered more likely to cause collision than obstacles perpendicular to the velocity. However, this method treats the direction along and reverse with the velocity as same, and thus behaves rather conservative. For instance, even when the vehicle is escaping from an obstacle-rich area, it also lowers its speed to satisfy the collision checker.

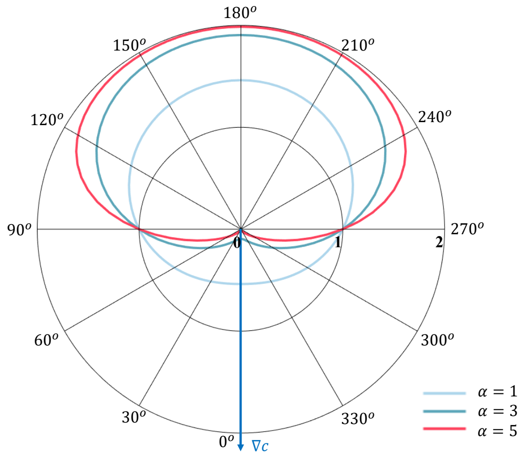

Fig. 2. The polar graph of η with different v when ∇c is constant to $(0,-1)$ in the 2D situation. The angle of the polar graph is between v and ∇c, and the length of the curve along the polar axis is the magnitude of η corresponds to velocity direction.

The above planning methods only consider the influence of the closest distance between the trajectory and the obstacles when dealing with safety, which is counterintuitive because the safety is related to the system dynamics. For instance, the degree of danger to the system is different when the quadrotor is flying towards and away from the obstacles, even if the distance is the same between the quadrotor and obstacles. In order to solve this problem, an environmental adaptive safety aware method is proposed in Sec.III.

III. Environmental adaptive safety aware

As mentioned above, it is essential to adaptively adjust the system's safety aware according to the environmental risk level and instantaneous motion tendency. EVA-planner uses the Environmental adaptive safety aware (EASA) to calculate the risk weight η to regulate the system's flight behavior, which comprehensively considers the gradient ∇c of the Euclidean signed distance field (ESDF) with respect to the velocity v.

Inspired by [12], we realize that it is essential to assign the priority of obstacles depending on the system velocity. So we propose a method to adjust the priority of obstacles more intuitively than the above method, which uses the sigmoid function that maps the cosine of the angle between v and ∇c to the risk weight $\eta \in [0, 2]$. The function is defined as

$$\eta(\beta) = \frac{2}{1 + e^{\alpha\beta}}, \tag{1}$$

$$\beta = \frac{< v, \nabla c >}{\|v\|\|\nabla c\|}, \tag{2}$$

where $\alpha \in \mathbb{R}^+$ is the change rate coefficient and $<, >$ is the dot product operation.

As shown in Fig.2, η is maximum when v and ∇c are opposite, and η is minimum when v and ∇c are in the same direction. We simply regard the situation $\beta \in (0, 1]$ as safe and the situation $\beta \in [-1, 0]$ as dangerous. The impact of α in (1) is shown in the Fig.2. With the α increasing, the

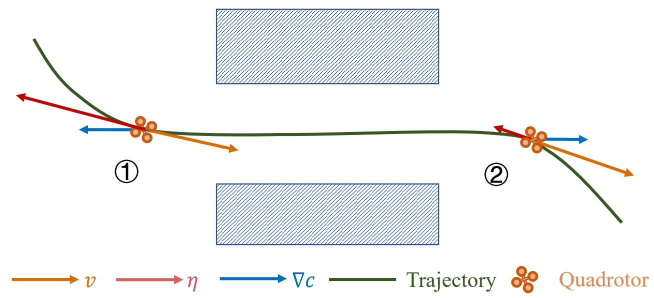

Fig. 3. Illustration of η changing in the planning process. The risk weight η is a scalar, but we put it in the opposite direction of v to show its slowing effect on the quadrotor.

difference of the risk weight η between dangerous and safe situations also increases.

EASA reasonably corresponds each pair of $\{v, \nabla c\}$ to a risk weight η. This method provides more intelligent environmental information for the planning process, reducing conservative planning, and providing clear signals in dangerous situations. As shown in Fig.3, η is large when a quadrotor flies into the corridor, which means it is in a dangerous situation and needs to fly cautiously at a slower velocity. Moreover, η is small when a quadrotor flies away from the corridor, which means it is in a safe state and can fly more aggressively. The application of EASA to the trajectory generation is described in detailed in Sec.IV-D.

IV. Multi-layered Model predictive contouring control framework

A. Multi-layered planning framework

Thanks to the differential flatness of the quadrotor dynamics stated in [13], we deal with the quadrotor as a particle model. In the planning process, the trajectories of the three axes x, y, z are decoupled, and the yaw angle ψ planning is carried out independently.

This paper designs a planning framework with three coarse-to-fine layers to generate receding-horizon local adaptive trajectories in real-time. The framework is unified among these layers and can be applied to quadrotors with different computing forces. This property enables planners to deal with approximations of the system's arbitrarily high order's fidelity, depending on the hardware.

In the first layer, we use the path finding algorithm, such as sampling-based method RRT* [14] and searching-based method A* [15] to generate a guiding path which connects the start and end points in the free space. In the middle layer, we use the low-order motion model to optimize the guiding path's geometry, named Low-level MPC. By doing so, the computing resources required to generate a reference trajectory can be reduced. In the last layer, we generate the local trajectory by optimizing EASA, tracking error, flight aggressiveness, and dynamical feasibility with high order system, named High-level MPCC.

B. System representation

The same system representation is used in Sec.IV-C and Sec.IV-D with different system dimension and order. The different parameter selections can be found in Tab.I. The time interval between the adjacent state points is fixed as δt. We assume that the system input u_i is constant in δt. Therefore the kinematic relationship between the two adjacent points can be expressed as

$$\mathbf{s}_{i+1} = A_d \mathbf{s}_i + B_d \mathbf{u}_i, \tag{3}$$

where A_d and B_d are the state-transfer matrixes governed by the d^{th}-order integral model. Based on the above assumptions, state sequence $\{\mathbf{s}_i\}$ can be represented as a mapping of the input sequence $\{\mathbf{u}_i\}$ by using the state transition matrix

$$\mathbf{S}_\mu = A\mathbf{U}_\mu + B\mathbf{s}_0, \tag{4}$$

$$A = \begin{pmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ B_d & 0 & \cdots & \cdots & 0 \\ A_d B_d & B_d & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_d^{H-1} B_d & A_d^{H-2} B_d & A_d^{H-3} B_d & \cdots & B_d, \end{pmatrix}, \tag{5}$$

$$B = (I, A_d, {A_d}^2 \cdots \ {A_d}^H)^T, \tag{6}$$

where state vector $\mathbf{S}_\mu = [\mathbf{s}_0, \mathbf{s}_1 \cdots \mathbf{s}_H]^T$ and input vector $\mathbf{U}_\mu = [\mathbf{u}_1, \mathbf{u}_2 \cdots \mathbf{u}_H]^T$.

From (4), all system states can be represented by the input vector $\mathbf{U}_\mu$ when the initial state $\mathbf{s}_0$ is known. Therefore the decision variables of the trajectory optimization problem are reduced to the input vector $\mathbf{U}_\mu$. Meanwhile, we construct the trajectory generation as an unconstrained optimization problem by transforming the constraints into penalty terms. The reference trajectory generation and the local trajectory optimization are explained in detail in Sec.IV-C and Sec.IV-D respectively.

C. Low-level MPC

The primary purpose of this layer is to optimize a geometrically continuous and obstacle-free reference trajectory quickly. Inspired by [16], the guiding path attracts the trajectory to escape the local minimum, and ESDF pushes the trajectory to find the local optimum. In this paper, we choose the velocity as the input to the reference trajectory, as shown in Tab.I. The object function is

$$\min \kappa_1 J_s + \kappa_2 J_c + \kappa_3 J_u, \tag{7}$$

where J_s is the similarity penalty of the distance between the trajectory and the guiding path, while J_c is the collision cost, and J_u is the smoothness term. And $\kappa_1, \kappa_2, \kappa_3$ are the weights of these penalty items

$$J_s = \sum_{i=1}^{M} \|p_i - g_i\|^2, \tag{8}$$

TABLE I

SYSTEM REPRESENTATION OF THE MULTI-LAYERED MPCC

	Low-level MPC	High-level MPCC
System dimension	$\mu \in \{x, y, z\}$	$\mu \in \{x, y, z, \theta\}$
System order	$d = 1$	$d = 3$
prediction horizons	$H = M$	$H = N$
State	$\mathbf{s}_i = p_i$ $i = 0, 1...M$	$\mathbf{s}_i = [p_i, v_i, a_i]^T$ $i = 0, 1...N$
input	$\mathbf{U}_i = v_i$ $i = 1, 2...M$	$\mathbf{U}_i = j_i$ $i = 1, 2...N$
transform matrix	$A_1 = 1$ $B_1 = \delta t$	$A_3 = \begin{pmatrix} 1 & \delta t & \frac{\delta t^2}{2} \\ 0 & 1 & \delta t \\ 0 & 0 & 1 \end{pmatrix}$ $B_3 = [\frac{\delta t^3}{6}, \frac{\delta t^2}{2}, \delta t]^T$

$$J_c = \sum_{i=1}^{M} F_c(c(p_i)), \tag{9}$$

$$J_u = \sum_{i=1}^{M-1} (u_{i+1} - u_i)^2, \tag{10}$$

where g_i is the i^{th} point of the guiding path and $c(p)$ is the distance of point p in the ESDF. Because the input u_i is constant in δt, the sum of squares of the difference between adjacent inputs is used to indicate the smoothness of trajectory in (10). $F_c(\cdot)$ is the penalty function of collision cost as

$$F_c(c(p)) = \begin{cases} (c(p) - c_{thr})^2, & \text{if } c(p) < c_{thr} \\ 0, & \text{if } c(p) \geq c_{thr} \end{cases}, \tag{11}$$

where the safe distance c_{thr} is the threshold. The collision cost starts to rise rapidly when the distance from the obstacles is less than this threshold.

The optimized reference trajectory $\rho_\mu(\theta)$ is parameterized by $\theta(\theta \in [0, M\delta t])$, which is equal to the time parameter.

D. High-level MPCC

After generating a reference trajectory, we take model predictive contouring control (MPCC) in the final layer. In this paper, we choose the jerk as the input and add the state $\{p_{\theta,i}, v_{\theta,i}, a_{\theta,i}\}$ of reference points which travels on the reference trajectory as another system dimension, as shown in Tab.I.

Inspired by [1, 17], we optimize the tracking accuracy and traveling progress simultaneously. Meanwhile we consider the environmental adaptive flight by using EASA which is detailed in Sec.III. Furthermore, there is also a penalty for violating the dynamical feasibility. The objective function is

$$\min \lambda_1 f_s + \lambda_2 f_p + \lambda_3 f_e + \lambda_4 f_c + \lambda_5 f_d, \tag{12}$$

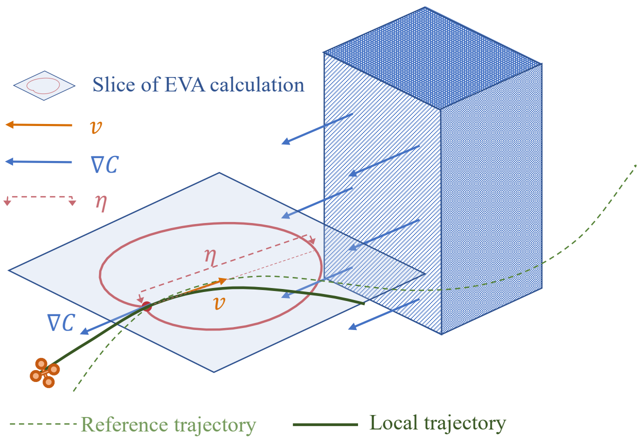

Fig. 4. Illustration of EASA calculation during trajectory optimization.

where f_s is the tracking error between the local trajectory and the reference trajectroy, f_p indicates the progress of reference points and f_e represents EASA penalty item. f_c is the collision cost same as (9) and f_d is the penalty for violating kinodynamics feasibility. $\lambda_i, i = 1 \cdots 5$ are the weights of these items

$$f_s = \sum_{i=1}^{N} \|p_i - p_{\theta,i}\|^2, \tag{13}$$

$$f_p = -\delta t \sum_{i=1}^{N} v_{\theta,i}, \tag{14}$$

$$f_e = \sum_{i=1}^{N} \eta(v_i, \nabla c(p_i)) F_c(c(p_i))(\|v_i\| - v_{thr})^2, \tag{15}$$

above three cost function items are the main determinants of the trajectory behavior. The optimization problem trades off f_s and f_p to make the trajectory as fast as possible while ensuring the tracking error small enough. However, when the quadrotor flies towards obstacles, as shown in Fig.4, η increases to make f_e become the primary determinant of the trajectory behavior, which causes the trajectory to slow down until the quadrotor is out of the dangerous area. The derivative of (13) and (14) can be found in [17]. The derivative of (15) is

$$\begin{aligned} \frac{\partial f_e}{\partial U_\mu} &= \sum_{i=1}^{N} \frac{2\alpha e^{\alpha\beta}}{(1+e^{\alpha\beta})^2} \frac{\partial \beta}{\partial U_\mu} F_c(c(p_i))(\|v_i\| - v_{thr})^2 \\ &\quad + \eta(v_i, \nabla c(p_i))(\nabla_\mu F_c(c(p_i)) \frac{\partial p_i}{\partial U_\mu} (\|v_i\| - v_{thr})^2) \\ &\quad + \eta(v_i, \nabla c(p_i))(2F_c(c(p_i)) \frac{v_\mu}{\|v_i\|} (\|v_i\| - v_{thr}) \frac{\partial v_i}{\partial U_\mu}), \\ \frac{\partial \beta}{\partial U_\mu} &= \frac{\nabla_\mu c(p_i)\|v\|^2 - v_\mu < v, \nabla c >}{\|v\|^3 \|\nabla c(p_i)\|} \frac{\partial v_i}{\partial U_\mu} \\ &\quad + \frac{v_\mu \|\nabla c(p_i)\|^2 - \nabla_\mu < v, \nabla c >}{\|v\| \|\nabla c(p_i)\|^3} {\nabla_\mu}^2 c(p_i) \frac{\partial p_i}{\partial U_\mu}, \end{aligned} \tag{16}$$

where $\nabla_\mu{}^2 c(p_i)$ is the second derivative in the third-order linear interpolation ESDF and v_{thr} is the slowest safe speed, which is generally set as $0.1m/s$ to prevent the optimized trajectory velocity from being zero. $\frac{\partial p_i}{\partial U_\mu}$ and $\frac{\partial v_i}{\partial U_\mu}$ are the corresponding row vectors of A in (5). The obstacle avoidance and kinodynamics constraints are

$$f_c = \sum_{i=1}^{N} F_c(c(p_i)), \tag{17}$$

$$f_d = \sum_{i=1}^{N} (F_d(v_i) + F_d(a_i) + F_d(j_i) + F_d(v_{\theta,i})), \tag{18}$$

where $F_c(\cdot)$ is the penalty function of collision cost same as (11). $F_d(\cdot)$ is the penalty function of kinodynamics feasibility

$$F_d(D) = \sum_{\mu} f_d(d_\mu), \tag{19}$$

$$f_d(d_\mu) = \begin{cases} -(d_\mu - d_{min})^3, & d_\mu < d_{min}, \\ 0, & d_{min} \le d_\mu \le d_{max}, \\ (d_\mu - d_{max})^3, & d_\mu > d_{max}, \end{cases} \tag{20}$$

where d_{min} and d_{max} are the lower and upper bound of each variable.

V. Results

A. Implementation Details

Extensive simulation and real-world flight experiments are carried out to test the proposed method's effectiveness and robustness. We use the flight platform proposed in [18] with Intel Realsense D435[2] for perception and mapping. Modules including state estimation, environment perception, trajectory planning, and flight control are all running onboard computer Manifold2[3] with Intel i7-8550U in real-time. All simulations run on a laptop with Intel i7-6500U.

In both simulation and real-world experiments, we generate the reference trajectory by solving (7) at a fixed interval of 2 seconds and check whether the reference trajectory has a collision with the new explored environment at a frequency of 100Hz. If the collision is detected, the reference trajectory replanning is carried out immediately. For local trajectory generation, we solve the unconstrained optimization problem (12) with 40 prediction horizons and time step of 0.05 second, $\delta t = 0.05s$, $N = 40$ and $T = 2s$. The trajectory optimization problem is solved by a nonlinear optimization open-source library NLopt[4]. The optimization times of Low-level MPC and High-level MPCC are less than 10ms each time, respectively.

TABLE II
COMPARISON OF RUNNING TIMES IN DIFFERENT MAPS

Density($obs./m^2$)	Random forest map				Gate map
	0.04	0.16	0.28	0.40	
SDDM time(s)	20.74	29.50	40.05	42.90	42.71
EVA time(s)	19.28	25.02	31.04	33.01	32.99

B. Benchmark

We design a benchmark comparison to validate the effectiveness of EVA-planner. We compare EVA with SDDM [12], which also incorporates the velocity direction into the judgement of obstacles. However, this method is still conservative when the system flies away from the obstacles, as shown in Fig.5. The performance of SDDM is adjusted to the best according to the parameters in [12]. The velocity limit is set as $3m/s$, and the same reference trajectories are generated. $40m \times 5m$ random forest map and narrow gate map are used in tests. Comparisons of the running times are shown in Tab.II. When the number of obstacles in the random forest map increases, the running time gap between EVA and SDDM gradually increases.

The benchmark shows that EVA-planner has better adaptability to the environment and can generate a more aggressive trajectory while ensuring safety.

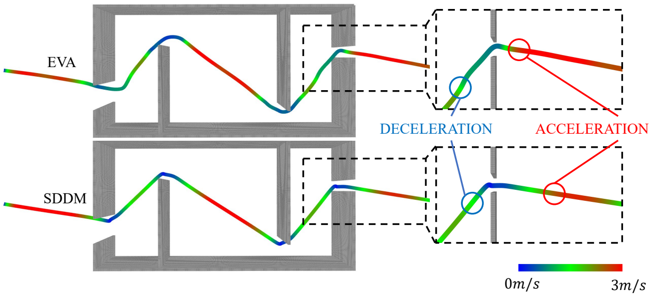

Fig. 5. Velocity curve comparison of EVA and SDDM in the narrow gate map. Our method shows a more reasonable change in velocity when passing through a narrow gate in contrast to the zoom-in trajectories.

C. real-world test

We design several real-world experiments to validate the feasibility and robustness of our method. All calculations are run on the onboard computer in real-time.

In order to verify the characteristics of EVA-planner, we set up the quadrotor to go through a loop in the straight line, as shown in Fig.6. Velocity limit is set as $3m/s$. Composite snapshots and executed trajectory show that a quadrotor slows down when it approaches the loop and accelerates as soon as it passes through it. The quadrotor flies in maximum velocity on the other times. This experiment verifies that EVA-planner can increase the aggressiveness of the flight trajectory while ensuring safety.

To verify the necessity of our method, we compare the flight results with & without EASA in a complex indoor

[2] https://www.intelrealsense.com/depth-camera-d435/
[3] https://www.dji.com/cn/manifold-2
[4] https://nlopt.readthedocs.io/

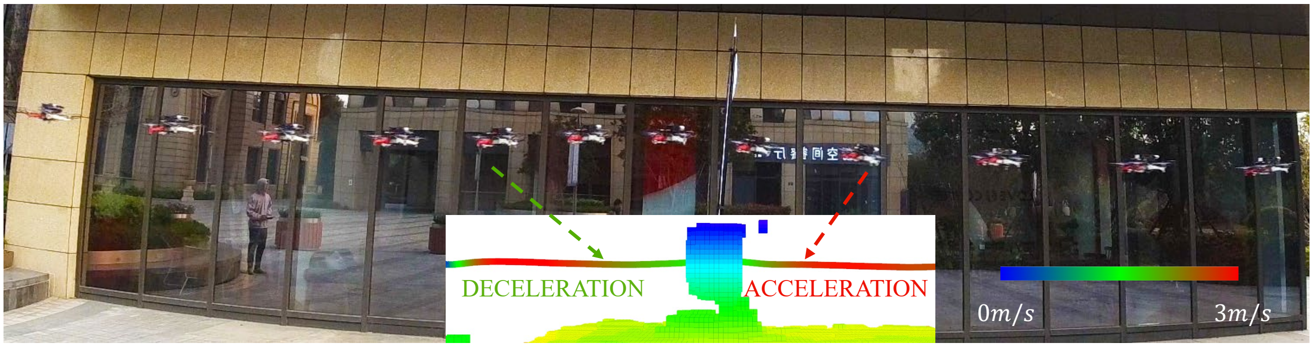

Fig. 6. Composite snapshots and executed trajectory of flying through a loop

environment, as shown in Fig.7. We design a challenging scenario for UAV planning in response to the limited FOV of visual cameras. The quadrotor flies through a narrow gap and immediately turns to the right where an obstacle is placed in its path. In Fig.7b and Fig.7d, trajectory without EASA performs unsafe aggressiveness when it flies through the narrow gap. This behavior causes the quadrotor to have no time to avoid the sudden obstacle. By contrast, from Fig.7b and Fig.7d, the quadrotor with EASA performs conservative velocity when passing the dangerous area, which makes the system have enough time to deal with the sudden danger. We repeat the experiments with EASA seven times with a success rate of 100%. However, two out of three experiments without EASA fail and hit the obstacles.

In the outdoor experiment, the quadrotor flies from an open area into the bushes, then out of the bushes into another open area, as shown in Fig.1. Although the bushes are rugged and leafy, the quadrotor can still carry out adaptive deceleration while flying into the dangerous area and adaptive acceleration while flying out of the dangerous area. As validated in the attatched video, we repeat ten flights, and all are successful. This experiment verifies the robustness of our algorithm in the complex unknown environment.

VI. Conclusions and Future Work

In this paper, we propose a novel environmental adaptive safety aware method according to the gradient of ESDF and the velocity. Compared with the benchmark method, this method is more reasonable in judging the degree of danger. Therefore it can plan a more aggressive trajectory without affecting safety.

To apply this method into the trajectory generation, we design a unified multi-layered planning framework to generate smooth, adaptively aggressive, safe, and dynamical feasible local trajectories. Extensive simulations and real-world experiments validate the robustness and effectiveness of our planning framework. In the future, we plan to generate environmental adaptive trajectories in complex dynamic environments and intend to challenge our quadrotor system in more extreme situations.

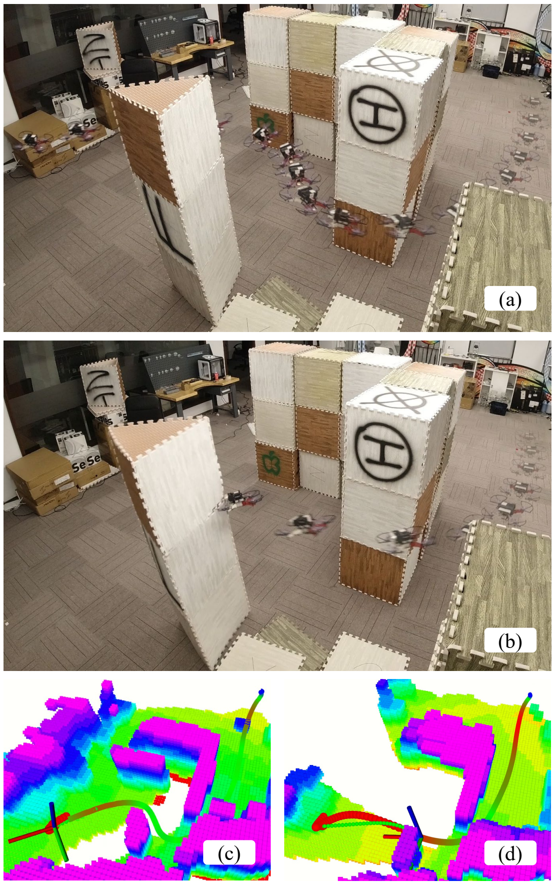

Fig. 7. Composite snapshots and executed trajetcory of flying in a complex indoor environment with & without EASA. The flight results in (a) and (c) are with EASA while (b) and (d) are without EASA.

References

[1] J. Ji, X. Zhou, C. Xu, and F. Gao, "Cmpcc: Corridor-based model predictive contouring control for aggressive drone flight," *arXiv preprint arXiv:2007.03271*, 2020.

[2] B. T. Lopez and J. P. How, "Aggressive 3-d collision avoidance for high-speed navigation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5759–5765.

[3] F. Gao, W. Wu, W. Gao, and S. Shen, "Flying on point clouds: Online trajectory generation and autonomous navigation for quadrotors in cluttered environments," *Journal of Field Robotics*,

vol. 36, no. 4, pp. 710–733, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21842

[4] N. Bucki, J. Lee, and M. Mueller, "Rectangular pyramid partitioning using integrated depth sensors (rappids): A fast planner for multicopter navigation," 03 2020.

[5] P. R. Florence, J. Carter, J. Ware, and R. Tedrake, "Nanomap: Fast, uncertainty-aware proximity queries with lazy search over local 3d data," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7631–7638.

[6] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, May 2009.

[7] H. Oleynikova, M. Burri, Z. Taylor, J. Nieto, R. Siegwart, and E. Galceran, "Continuous-time trajectory optimization for online uav replanning," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Daejeon, Korea, Oct. 2016.

[8] F. Gao, Y. Lin, and S. Shen, "Gradient-based online safe trajectory generation for quadrotor flight in complex environments," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Sept 2017.

[9] B. Zhou, F. Gao, J. Pan, and S. Shen, "Robust real-time uav replanning using guided gradient-based optimization and topological paths," *arXiv preprint arXiv:1912.12644*, 2019.

[10] J. Tordesillas, B. T. Lopez, and J. P. How, "FASTER: Fast and safe trajectory planner for flights in unknown environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.

[11] D. Fridovich-Keil, S. Herbert, J. Fisac, S. Deglurkar, and C. Tomlin, "Planning, fast and slow: A framework for adaptive real-time safe trajectory planning," 05 2018, pp. 387–394.

[12] Z. Li, . Arslan, and N. Atanasov, "Fast and safe path-following control using a state-dependent directional metric," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020.

[13] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Shanghai, China, May 2011, pp. 2520–2525.

[14] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, pp. 846–894, 2011.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.

[16] B. Zhou, J. Pan, F. Gao, and S. Shen, "Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight," 2020.

[17] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale rc cars," *Optimal Control Applications and Methods*, vol. 36, pp. 628–647, 09 2015.

[18] F. Gao, L. Wang, B. Zhou, L. Han, J. Pan, and S. Shen, "Teach-repeat-replan: A complete and robust system for aggressive flight in complex environments," *arXiv preprint arXiv:*, 2019.