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

# Learning Agile Flights through Narrow Gaps with Varying Angles using Onboard Sensing

Yuhan Xie, Minghao Lu, Rui Peng and Peng Lu, Member, IEEE

***Abstract*—This paper addresses the problem of traversing through unknown, tilted, and narrow gaps for quadrotors using Deep Reinforcement Learning (DRL). Previous learning-based methods relied on accurate knowledge of the environment, including the gap's pose and size. In contrast, we integrate onboard sensing and detect the gap from a single onboard camera. The training problem is challenging for two reasons: a precise and robust whole-body planning and control policy is required for variable-tilted and narrow gaps, and an effective Sim2Real method is needed to successfully conduct real-world experiments. To this end, we propose a learning framework for agile gap traversal flight, which successfully trains the vehicle to traverse through the center of the gap at an approximate attitude to the gap with aggressive tilted angles. The policy trained only in a simulation environment can be transferred into different domains with fine-tuning while maintaining the success rate. Our proposed framework, which integrates onboard sensing and a neural network controller, achieves a success rate of 87.36% in real-world experiments, with gap orientations up to 60°. To the best of our knowledge, this is the first paper that performs the learning-based variable-tilted narrow gap traversal flight in the real world, without prior knowledge of the environment.**

***Index Terms*—Learning agile flight, onboard sensing, motion control**

## I. Introduction

Quadrotors are highly agile and versatile flying machines, making them ideal for complex tasks in cluttered environments [1]. Meanwhile, reinforcement learning (RL) is recently developing rapidly in the robotics domain for its strong potential of exploiting the robots' agility [2], [3]. Therefore, research topics arise naturally to employ RL on quadrotors for aggressive tasks [4], [5], which have recently contributed to a significant increase in autonomy capabilities [6]. Among the agile flight tasks in complex environments, one of the fundamental challenges is flying through narrow gaps, in which the drone's position and attitude must be considered simultaneously, leading to a *SE(3)* planning and control problem.

Despite significant progress in learning-based gap traversing tasks, three main problems remain unsolved. Firstly, training a policy in simulation and successfully transferring it to real-world flights through aggressive angle narrow gaps has not been addressed [5], [7], [8]. The training algorithm is required to consider both an aggressive and robust *SE(3)* control policy and an effective Sim2Real transfer. Secondly, existing methods require prior knowledge of the gap pose and size in the world reference frame. Moreover, errors introduced by the gap detection would increase the risk of collision in real-world experiments. Lastly, some approaches rely on expert planners and controllers for imitation in training [9], which may end up with local optimal solutions similar to experts without sufficient exploration.

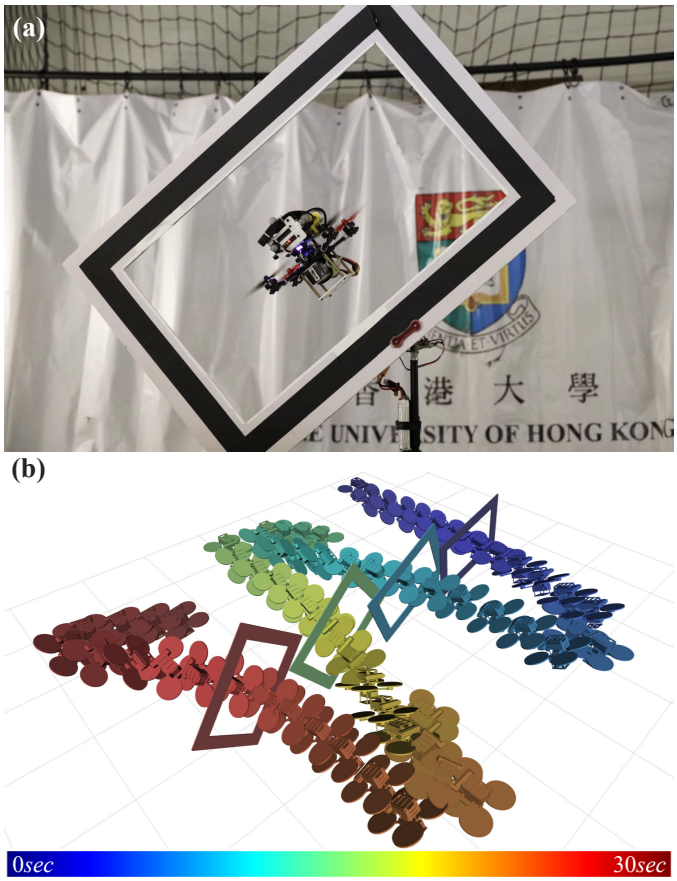


Fig. 1. Agile flights through tilted narrow gaps in real-world experiments. (a) Our quadrotor is traversing a tilted narrow gap. (b) Experiment results of four consecutive flights through narrow gaps. Each traversal is translated as a whole for visualization.

To overcome the aforementioned challenges, this paper proposes an end-to-end framework that includes a gap detection algorithm and a policy training algorithm, which enables quadrotors to autonomously detect and traverse gaps with variable-tilted attitude. The training algorithm takes generalization and domain adaption into account, thereby ensuring successful Sim2Real transfer for physical experiments. The main contributions of our work are summarized as below:

1) A novel learning framework is designed for variable-tilted narrow gap traversing tasks. The trained policy

Manuscript received: February 21, 2023; Revised: June 5, 2023; Accepted: July 9, 2023.

This paper was recommended for publication by Editor P. Pounds upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by General Research Fund under Grant 17204222, and in part by the Seed Funding for Strategic Interdisciplinary Research Scheme and Platform Technology Fund. *(Corresponding author: Peng Lu)*

The authors are with the Department of Mechanical Engineering, the University of Hong Kong, Hong Kong SAR, China (email: {yuhanxie, minghao0, pengrui-rio}@connect.hku.hk, lupeng@hku.hk).

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

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

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

achieves a precise *SE(3)* trajectory planning and control of a quadrotor.
2) With fine-tuning to transfer the policy from the training environment, repetitive tests in the software-in-the-loop (SITL) environments are conducted, maintaining a high success rate and demonstrating the effectiveness of the training algorithm.
3) Onboard sensing is introduced so that no prior knowledge of the gap is required, e.g., position, orientation, or size. To the best of our knowledge, this is the first work that integrates onboard sensing to a learning system for gap traversing tasks.
4) Repetitive real-world experiments demonstrate the robustness of the proposed framework. Our experiment results show that our quadrotor system can fly through variable-tilted narrow gaps with precise traversing posture for gap orientations up to $60^\circ$.

## II. Related Work

### A. Quadrotor Agile Flight

State-of-the-art agile quadrotor flight methods typically decouple trajectory planning and control. For a specific environment, the conventional approach usually follows the pipeline of planning a trajectory and then tracking the trajectory by the controller. The performance and success rate depend highly on both the quality of the planned trajectory and the robustness of the controller. For quadrotor trajectory generation, the modern frameworks exploit the differential flatness [10] of the vehicle using polynomial [11]–[13], or B-spline [14]–[16] representations. These trajectories are inherently smooth. Hence, they cannot represent the rapid state or input changes in a reasonable order, and only reach the input limits for an infinitesimal short duration [4]. The popular controllers for trajectory tracking include model predictive control (MPC) and differential flatness control [11]–[14]. However, most control approaches rely on physical assumptions and are dependent on modeling, making them struggle to handle disturbances during agile flight.

In contrast to the decoupled framework in optimization-based methods, learning-based methods address the problem by learning an end-to-end policy that predicts control commands directly from high-dimensional observations [4], [5], [8], [9], [17]. Recent works have shown that these methods can achieve superhuman performance in near-time-optimal flight for drone racing and high-speed flight in the wild [6].

### B. Agile Flight through Narrow Gaps

Aggressive flight through a narrow gap is one of the most challenging problems for quadrotors. A whole body planning and control considering position as well as attitude of the vehicle is required. Early work designed a sequence of control phases to execute an aggressive trajectory and reach the goal state [18]. Based on the differential flatness property [10], Loianno *et al.* [19] planned dynamically feasible trajectories which guide the drone to the window traversal state. The work also considers state estimation from a monocular camera and an IMU. Falangal *et al.* [20] further integrated state estimation and gap detection by onboard sensing and computing, and achieved the goal without prior knowledge of the pose of the gap.

Recently, some works have considered learning-based planning and control methods to address the gap traversal problem for quadrotors. Early work [9] follows the decoupled planning and control pipeline, and imitates a traditional planner [11] and controller [10], [21]. Additional reinforcement training is also required to fine-tune the policy network. The prior expert knowledge provides good initial conditions for the policy and accelerates the training process. However, the imitation learning may end up with local minimums similar to priors, which limits the exploration ability of RL. Moreover, the control command of desired attitude generated by the policy vibrates severely compared to the result of the traditional, indicating an unsatisfying control performance. To exploit the quadrotors' agility, recent work employs deep reinforcement learning for the gap traversal problem [5], [7], [8]. Our previous work [5] proposed a reinforcement learning framework augmented with curriculum learning and Sim2Real methods, which achieves successful real-world gap traversing flight using DRL. However, the tilted angle of the gap is fixed at $20^\circ$ in training and experiments. Chen *et al.* [8] considered narrow gaps with up to $60^\circ$ tilted angle in simulation, while the physical experiments were conducted with a very limited tilted angle. A successful Sim2Real transfer is not presented for aggressive angles. Overall, a learning-based control policy for traversal through aggressive angle gaps in the real world remains unsolved among these works. To tackle this problem, our training algorithm considers not only the aggressive and robust *SE(3)* control but also the effective Sim2Real transfer. Furthermore, the related work mentioned above relied on accurate prior knowledge of the gap, including the position, orientation, and size. Thus, these methods cannot address the problem when the gap state changes. In this work, we introduce an onboard sensing algorithm to detect the gap, which is necessary for real-world applications.

## III. Problem Statement

In this letter, we address the problem of controlling a quadrotor to fly through a narrow gap with varying tilted angles using an onboard camera.

### A. Problem Overview

Our approach consists of two subsystems: perception and control. The perception system estimates the position and orientation of the gap using a forward-facing depth camera, which is presented in Section V. The control system includes a neural network that maps from the observation of the drone and gap, directly to low-level control commands, guiding the quadrotor to complete the task. The trajectory should try to intersect the center of the gap while simultaneously attaining the exact orientation of the gap, as illustrated in Figure 2. Therefore, a precise *SE(3)* planning and control policy for quadrotor is required.

Variation of gap orientation is considered. In policy training, we keep the drone facing the gap and omit the yaw angle control. Pitch angles of the gap are ignored, as the gap on a

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

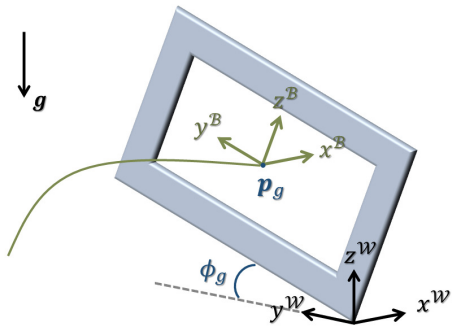


Fig. 2. Traversal Demonstration.

wall usually has a few pitches. Thus, we mainly cope with the variation of roll angle in this letter.

### B. Quadrotor Dynamics for Training

To simulate the quadrotor flight and the interaction between the vehicle and the gap for policy training, we formulate the quadrotor model. Consider a quadrotor with mass $m \in \mathbb{R}$ and diagonal moment of inertia matrix $\boldsymbol{J} \in \mathbb{R}^3$. The dynamic model of the system can be written as

$$\begin{aligned} \dot{\boldsymbol{p}} &= \boldsymbol{v}, \qquad m\dot{\boldsymbol{v}} = \boldsymbol{R}\boldsymbol{e}_3 f_T + \boldsymbol{R}\boldsymbol{f}_D + m\boldsymbol{g} \\ \dot{\boldsymbol{R}} &= \boldsymbol{R}\hat{\boldsymbol{\omega}}, \quad \boldsymbol{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{\tau}_T + \boldsymbol{\tau}_D \end{aligned} \tag{1}$$

where $\boldsymbol{p} = [p_x, p_y, p_z]^T$ and $\boldsymbol{v}$ are the position and velocity vector in the world frame, $\boldsymbol{R} \in \mathbb{SO}(3)$ is the rotation of the quadrotor, $\boldsymbol{\omega}$ represents angular body velocity. $\hat{\boldsymbol{\omega}}$ is the skew-symmetric matrix of vector $\boldsymbol{\omega}$, $\boldsymbol{g}$ is the gravity vector, and $\boldsymbol{e}_3 = [0, 0, 1]^T$ is a constant vector. $f_T$ and $\boldsymbol{\tau}_T$ denote thrust in the body-z axis and body torque generated by four rotors. Air drag force $\boldsymbol{f}_D$ and torque $\boldsymbol{\tau}_D$ are also modeled for aggressive motion. Overall, the state and control input of quadrotor are $\boldsymbol{x} = [\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{R}, \boldsymbol{\omega}]^T$, $\boldsymbol{u} = [f_T, \boldsymbol{\tau}_T]^T$. We define the Euler angles of the quadrotor $(\phi, \theta, \psi)$, which can be derived from $\boldsymbol{R}$.

### C. Task Formulation

We model the task using an infinite-horizon Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, p, r)$, where the state space $\mathcal{S}$ and the action space $\mathcal{A}$ are continuous. At every control step $t$, given current state $\boldsymbol{s}_t \in \mathcal{S}$, an action $\boldsymbol{a}_t \in \mathcal{A}$ is sampled from a policy $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)$. Subsequently, the agent executes the action $\boldsymbol{a}_t$ and transits to the next state $\boldsymbol{s}_{t+1} \in \mathcal{S}$ with the unknown state transition probability $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, \infty)$, receiving a bounded reward $r : \mathcal{S} \times \mathcal{A} \to [r_{\min}, r_{\max}]$. Specifically, the state $\boldsymbol{s} \in \mathcal{S}$ includes the quadrotor state $\boldsymbol{x} \in \mathcal{X}$ and the gap pose $\boldsymbol{g} \in \mathcal{G}$. The goal of our algorithm is to learn a control model $\pi : \mathcal{X} \times \mathcal{G} \to \mathcal{A}$.

## IV. Learning to Control

This section presents the policy architecture, reward formulation, and training strategy employed in our approach for training a control policy for the tilted narrow gap traversal problem.

### A. Policy Architecture

The neural network architecture as well as the state and action spaces are illustrated in Figure 3. An additional $\tanh$ function is used at the last layer of the policy network to keep the actions within a fixed range.

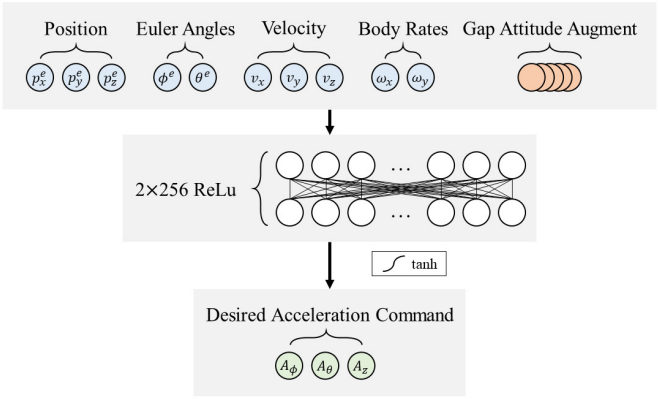


Fig. 3. Neural Network Architecture.

*1) States:* As stated in Section III-C, the state space of our neural network consists of two parts: the quadrotor state and the gap pose. We define the gap pose by center position $\boldsymbol{p}_g \in \mathbb{R}^3$ and the rotation matrix $\boldsymbol{R}_g \in \mathbb{SO}(3)$ in the world frame. The corresponding Euler angles are $(\phi_g, \theta_g, \psi_g)$.

*Drone States.* To facilitate traversal, the pose information of the quadrotor is given relative to the target. We denote $\boldsymbol{p}_T$ as a target position located behind the gap center that

$$\boldsymbol{p}_T = \boldsymbol{p}_g + \delta_T \cdot \boldsymbol{R}_g \boldsymbol{e}_1 \tag{2}$$

where $\delta_T$ is a target distance to the gap center and $\boldsymbol{R}_g \boldsymbol{e}_1$ represents the first column of the $\boldsymbol{R}_g$. The relative position vector $\boldsymbol{p}^e$ is designed as

$$p_i^e = \operatorname{sgn}(p_{T,i} - p_i)\sqrt{|p_{T,i} - p_i|}, \quad i \in \{x, y, z\} \tag{3}$$

The relative orientation is defined by subtracting the Euler angles of gap and quadrotor as

$$\phi^e = \phi_g - \phi, \quad \theta^e = \theta_g - \theta \tag{4}$$

Although the subtraction is physically meaningless, it is intuitive for policy training, guiding the quadrotor approaches gap's roll and pitch angle during traversal.

*Gap Attitude Augment.* Previous works only considered limited tilted angles in policy training or experiments. In contrast, we are interested in the variation of the gap orientation. Therefore, we implement a data augmentation technique on the state-based inputs to improve the data efficiency as well as the generalization ability of the policy [22]. Specifically, the random amplitude scaling method is introduced in this work for gap attitude, as shown in Figure 3.

*2) Actions:* Network actions are normalized second-order derivatives of desired Euler angles and altitude, while the low-level control commands for the vehicle are the desired orientation and altitude. Hence, after mapping the normalized network outputs to a fixed range, there is a second-order integrator before passing the signals to the low-level controller on the quadrotor.

There are two considerations for this design of network outputs. Firstly, the network outputs are physically meaningful and effective for agile quadrotor flight control. Based on the differential flatness of quadrotor dynamics, the control inputs $\boldsymbol{\tau}_T$ appear as functions of the second derivatives of orientation, and $\boldsymbol{f}_T$ appears as the function of the second derivatives of altitude. Thus, our policy can be considered as a motion

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

planner on thrust and torque, which has been demonstrated effective for agile quadrotor flight planning. Secondly, the network output processing framework can facilitate Sim2Real transfer, referring to our previous work [5] which demonstrated the framework could enhance generalization without utilizing real-world data.

### B. Reward Function Design

The reward function consists of four designs. The main objective of our task is to guide the drone to fly to the back side of the gap. The position distance between the quadrotor position $\boldsymbol{p}$ and the target point $\boldsymbol{p}_T$ defined by (2), is calculated as the position reward as follows

$$r_p(t) = -\left\|\boldsymbol{p}(t) - \boldsymbol{p}_T(t)\right\| \tag{5}$$

Meanwhile, to increase the margin between gap while traversal, the quadrotor should raise its roll angle to the same attitude as the gap and reduce its pitch angle to zero, as illustrated in Figure 2. Thus, we design an attitude reward of relative roll between quadrotor and gap when the vehicle approaches the gap.

$$r_a(t) = \begin{cases} -\min(\tan\left|\phi^e\right|, 50) & \text{approach gap} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Note that there is no constraint on pitch, leaving more space for policy exploration. A penalty on control input is also given for smooth control

$$r_u(t) = -\left\|\boldsymbol{u}(t)\right\| \tag{7}$$

Lastly, a terminal reward $r_T$ is given only when the vehicle successfully passes through the window without any collision detected. The total reward $r(t)$ at time $t$ is defined as

$$r(t) = \lambda_p r_p(t) + \lambda_a (r_a(t) + b_a) + \lambda_u r_u(t) + \begin{cases} r_T & \text{win} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $\lambda_p, \lambda_a, \lambda_u \in \mathbb{R}$ are hyperparameters that trade-off between each reward components, $b_a \in [0, +\infty)$ is a positive offset for relative attitude reward.

### C. Training Details

The policy is trained using Soft Actor-Critic (SAC) [23], an off-policy algorithm that features entropy regularization. In our training environment, a quadrotor with dynamics (1) and a static window on a wall are simulated. The vehicle is simulated at a frequency of 80Hz, while the control frequency, i.e., the frequency of collecting state and action data for training, is only 20Hz, which balances the training acceleration and data efficiency. The episodes terminate when the edge of state space $\mathcal{S}$ is reached, or the terminal reward $r_T$ is obtained.

*1) Curriculum Learning:* The terminal reward is hard to obtain directly due to the narrow gap. Only a precise control policy can complete the task and win the terminal reward. To overcome reward sparsity, a curriculum strategy is employed for policy training in multiple stages. Specifically, we refer to our previous work [5] and introduce a difficulty factor $d_f$ to adjust gap size with training episodes. As training episodes increase, the gap narrows so the feasible traversal trajectories converge. To augment the policy for aggressive cases ($|\phi_g| \geq 50°$), after the gap shrinks to the goal size, we further add a curriculum that makes the probability of large roll angles greater.

*2) Randomization:* Several randomization strategies are employed to make the policy robust against unknown dynamics effects and facilitate domain adaptation. For each episode, the vehicle and the window are initialized with randomization: the initial state of quadrotor is normally distributed. The dynamics parameters of vehicle are also randomized with normal distributions. For each step, observation noises are introduced in zero-mean normal distributions.

## V. Onboard Sensing

This section introduces the gap detection method, which aims to identify the black-and-white rectangular frames with uncertain sizes in physical experiments. The method employs an RGB-D camera to obtain both an RGB image and a depth image.

To extract and refine the edges from the binary image, we perform the closing operation, Canny edge detection, edge undistortion, and edge grouping consecutively. Subsequently, we apply the Douglas-Peucker algorithm to fit the edge groups into polygons, followed by generating their respective convex hulls using the Quickhull algorithm. The convex hull $\mathcal{H}_k$ is defined as a set of pixels $[u, v]^T$, given by

$$\mathcal{H}_k = \left\{[u_n, v_n]^T \mid n = 1, 2, \ldots, N\right\} \tag{9}$$

To identify rectangles among the convex hulls, we consider the following conditions:

$$N = 4, \ (\arccos(\boldsymbol{e} \cdot \boldsymbol{e}_a) - 1)^2 < \epsilon_1 \tag{10}$$

where $\boldsymbol{e}$ and $\boldsymbol{e}_a$ denote adjacent edge vectors of a hull, and $\epsilon_1$ is a small constant factor.

We proceed by estimating the 3D positions of the detected rectangles using the aligned depth image. The depth $d$ of each vertex within a rectangle is obtained by providing its pixel position $[u, v]^T$. Given the camera intrinsic matrix $\boldsymbol{M}_1 \in \mathbb{R}^{3\times4}$ and the world-to-camera transformation matrix $\boldsymbol{M}_2 \in \mathbb{R}^{4\times4}$, the 3D position of each vertex $\boldsymbol{p}_v = [x_v, y_v, z_v]^T$ in the world frame can be calculated by

$$\boldsymbol{M}_1 \boldsymbol{M}_2 [x_v, y_v, z_v, 1]^T = d[u, v, 1]^T \tag{11}$$

Hence, each detected rectangle can be represented as a set of four vertices in 3D space:

$$\xi = \{\boldsymbol{p}_{v,j} \mid j = 1, 2, 3, 4\} \tag{12}$$

The outlines of the gap can be determined by finding two rectangles $\xi_1$, $\xi_2$ that satisfy the following conditions,

$$\begin{aligned} &\arccos(\boldsymbol{n}_1 \cdot \boldsymbol{n}_2) < \epsilon, \\ &\text{area}(\xi_1) \subseteq \text{area}(\xi_2) \end{aligned} \tag{13}$$

where $\boldsymbol{n}_i$ denotes the normal unit vector of a rectangle plane, and $\epsilon$ is a small constant factor. The term $\text{area}(\xi)$ refers to the area confirmed by the points in $\xi$. These conditions, as stated in (13), describe the relationship between the inner $\xi_2$ and outer $\xi_1$ rectangular borders of the gap, which should be in the same plane, and the inner area is a proper subset of the outer area.

The 6-DOF pose of the gap can be calculated from the two point sets $\xi_1$ and $\xi_2$ using geometry calculations. The central position of the gap, $\boldsymbol{p}_g$, is determined as the average of the vertices, while the rotation matrix of the gap, $\boldsymbol{R}_g$, is

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

TABLE I
PARAMETERS OF TRAINING ALGORITHM

| | Parameter | Value |
|---|---|---|
| RL | position reward coefficient ($\lambda_p$) | 1.0 |
| | attitude reward coefficient ($\lambda_a$) | 10.0 |
| | attitude reward offset ($b_a$) | 0.2 |
| | control input reward coefficients ($\lambda_u$) | 0.05 |
| | terminal reward ($r_T$) | 500 |
| | window roll range | $[-60°, +60°]$ |
| | target distance ($\delta_T$ m) | 0.25 |
| | network outputs mapping scale ($\boldsymbol{\kappa}$) | $[80, 80, 24]$ |
| SAC [23] | optimizer | Adam |
| | learning rate | $3 \times 10^{-4}$ |
| | discount factor ($\gamma$) | 0.95 |
| | replay buffer size | $10^5$ |
| | batch size | 512 |
| | target smoothing coefficient ($\tau$) | 0.01 |
| | target update interval | 16 |
| Quadrotor | mass ($m$ [kg]) | 1.1 |
| | moment of inertia (diag($J$) [kg m$^2$]) | $[0.12, 0.12, 0.22]$ |
| | thrust coefficient ($k_T$) | $6 \times 10^{-6}$ |
| | moment to thrust coefficient ($k_{TQ}$) | 0.02 |
| | arm length ($l$ [m]) | 0.34 |

calculated using the sides of the rectangle and the normal vector of the rectangle plane. The rotation matrix $\boldsymbol{R}_g$ is then transformed to Euler angles $(\phi_g, \theta_g, \psi_g)$. Overall, we use $\boldsymbol{x}_g = [\boldsymbol{p}_g, \phi_g, \theta_g, \psi_g]^T$ to describe gap pose. To smooth the output gap pose, a third-order low-pass filter [24] is applied using the following equation,

$$\begin{aligned} \dot{\boldsymbol{x}}_1 &= \boldsymbol{x}_2 \\ \dot{\boldsymbol{x}}_2 &= \boldsymbol{x}_3 \\ \dot{\boldsymbol{x}}_3 &= \omega_1\omega_2^2(\boldsymbol{x}_{g,m} - \boldsymbol{x}_1) \\ &\quad - (2\zeta\omega_1\omega_2 + \omega_2^2)\boldsymbol{x}_2 - (\omega_1 + 2\zeta\omega_2)\boldsymbol{x}_3 \end{aligned} \tag{14}$$

where $\boldsymbol{x}_1 = \hat{\boldsymbol{x}}_g, \boldsymbol{x}_2 = \hat{\dot{\boldsymbol{x}}}_g, \boldsymbol{x}_3 = \hat{\ddot{\boldsymbol{x}}}_g$. The transfer function of this filter is $\omega_1\omega_2^2/(s+\omega_1)(s^2+2\zeta\omega_2 s+\omega_2^2)$. The gap detector operates at a frequency of 30 Hz in our system.

Compared to the method proposed in [20], our approach does not require any prior information on the gap size. Furthermore, our method is simpler to implement, and more computationally efficient, relying only on a binary and a depth image to calculate the pose. It is worth noting that the depth error of the D455 camera used in our system is less than 2% within a range of 4 meters. In contrast, the approach in [20] requires prior knowledge of the gap size, limiting its ability for unknown gap sizes.

## VI. RESULTS

In this section, we evaluate the proposed system. We first transfer the trained policy into a new simulation environment and validate the generalization ability to different domains without more training data. Ablation studies are presented to validate the policy algorithm designs for gap attitude variation. We perform repetitive real-world experiments, demonstrating the effectiveness and robustness of the proposed method. At last, we reproduce a traditional gap traversal method [20], and implement experiments to compare the control performances.

### A. Training Configuration

The parameters of the training algorithm defined in Section IV are summarized in Table I. The rewards and curriculum

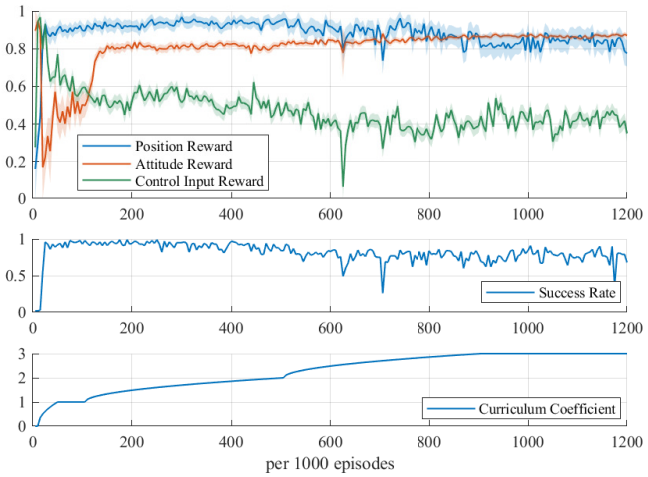


Fig. 4. Rewards Learning over Episodes. The rewards calculated by (5)-(7) are normalized.

difficulty factor over episodes are analyzed in Figure 4. As the curriculum difficulty increases, the policy continues to explore and learn, which can lead to local minima. This forms reward curves that exhibit generally ascending trends with intermittent spikes. Initially (the first 100k episodes), the gap is wide enough for the quadrotor to traverse with any attitude, resulting in a quick rise of the position reward $r_p$ in the first 15k episodes. As the training episodes proceed, the vehicle learns to follow the gap's attitude with guidance from the attitude reward and the constraint from the narrowing gap. Hence, the attitude rewards $r_a$ increase during 80k to 180k episodes. As the gap continues to narrow, the policy learns to maintain a high reward from completing the task, leading to the development of an accurate whole-body control policy. Providing the narrow gap goal from the outset of training, conversely, makes the terminal reward difficult to obtain, posing a challenge for the quadrotor to develop an accurate control policy.

### B. Sim2Real Validation

Before conducting real-world experiments, simulations in a different environment are implemented to validate the generalization ability of our policy. The software-in-the-loop (SITL) tests are all conducted utilizing Gazebo9 and PX4-Autopilot v1.11, running on a laptop featuring a 3.6GHz 8 core Intel Core i7-7700 CPU and an Intel HD Graphics 630. The algorithms are implemented in ROS with Ubuntu 18.04. The mass of the drone is set as 0.9kg with a motor constant of $1.5 \times 10^{-5}$. Note that the quadrotor dynamics parameters vary from the training environment. We only guarantee enough thrust-to-weight ratio, which is closely related to the ability of aggressive motion.

The control frequencies in the Gazebo simulation and the following real-world experiment are given as 50Hz, which is different from our training environment, as stated in Section IV-C. Considering the Sim2Real gap in control frequency, quadrotor dynamics, low-level controllers etc., we tune the linear mapping scale of the network outputs as $\boldsymbol{\kappa} = [160.0, 160.0, 24.0]$ in SITL and real-world experiments. After that, the policy trained only in our training environment can work well in unknown environments. We count the success

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

TABLE II
EVALUATION OF THE POLICY AND ABLATION STUDY IN SITL SIMULATION COMPARED WITH TRAINING RESULTS.

| | Methods | -60° | -50° | -40° | -30° | -20° | -10° | 0° | +10° | +20° | +30° | +40° | +50° | +60° |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Training Results [1] | **98.3%** | **99.1%** | **99.6%** | **99.4%** | **99.7%** | **99.9%** | **99.9%** | **99.9%** | **99.7%** | **98.8%** | **98.3%** | **96.2%** | **90.5%** |
| Config. 1 | Ours [2] | **83%** | **92%** | **94%** | **96%** | **98%** | **98%** | **98%** | **99%** | **97%** | **97%** | **91%** | **92%** | **76%** |
| | w/o attitude reward | 47% | 62% | 73% | 82% | 95% | 96% | 96% | 98% | 94% | 82% | 80% | 69% | 43% |
| | w/o attitude augment | 48% | 49% | 76% | 80% | 92% | 88% | 91% | 90% | 84% | 83% | 70% | 69% | 36% |
| Config. 2 | Ours | **95%** | **98%** | **99%** | **100%** | **100%** | **100%** | **100%** | **100%** | **99%** | **100%** | **99%** | **98%** | **88%** |
| | w/o attitude reward | 79% | 85% | 95% | 99% | 100% | 100% | 99% | 99% | 100% | 97% | 96% | 83% | 67% |
| | w/o attitude augment | 73% | 82% | 94% | 98% | 100% | 100% | 100% | 100% | 99% | 99% | 95% | 88% | 75% |

[1] 1000 tests for each case in training environment.
[2] 100 tests for each case in SITL environment.
[3] Configuration 1&2 are for SITL tests. Configuration 1 follows the training environment with drone size of $0.47\text{m} \times 0.17\text{m}$ and gap size of $0.70\text{m} \times 0.30\text{m}$. Configuration 2 follows the real-world experiments with drone size of $0.35\text{m} \times 0.20\text{m}$ and gap size of $0.70\text{m} \times 0.40\text{m}$.

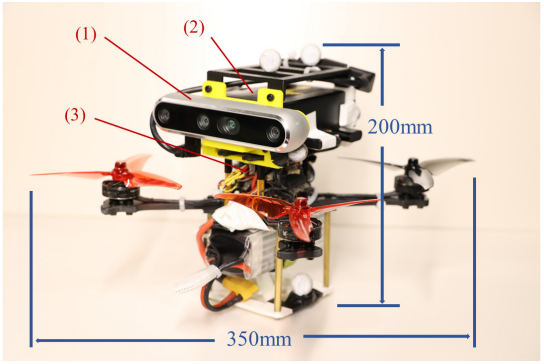


Fig. 5. Our quadrotor platform for real-world validation. (1) Intel D455i depth camera. (2) DJI Manifold 2-C onboard computer. (3) Pixracer.

rate through thousands of SITL tests with respect to different roll angles of the gap, as shown in Table II. The tests are implemented in two configurations related to the training environment and real-world experiments. The high success rates maintained from training environment demonstrate the robustness of the proposed algorithm. Please refer to the supplementary material for further evaluation of our policy.

### C. Ablation Study

We perform ablation studies to validate the designs of the proposed approach. Specifically, we focus on the effect of attitude augmentation (on network inputs) and attitude reward, which are designed to handle the variation of gap attitude. We replace the attitude augmentation with a plain attitude input or ablate the attitude reward in training and deploy the resulting policy into SITL tests. The success rates are counted and summarized in Table II.

For the ablation study on attitude reward, the results show that success rate is not affected significantly when the roll angle is small, i.e., $|\phi_g| \leq 20°$. However, when the task gets more complicated, e.g., increasing the attitude of gap or decreasing the size ratio of gap to drone, training without attitude reward is insufficient to achieve the traversal. Meanwhile, the attitude augmentation on network inputs increases the success rate and robustness of the policy for most situations. The results indicate the necessity of attitude augmentation for our policy training.

### D. Real-World Experiments

*1) Experiment Setup:* We validate our proposed system in the real world. Figure 5 shows our quadrotor platform used in the experiments. The target gap is detected by an Intel D455i depth camera. The gap observation algorithm and control policy runs on a DJI Manifold 2-C computer, sending low-level attitude and altitude control commands to a Pixracer. All real-world experiments are conducted indoors with a motion capture system, which facilitates state observation of the vehicle.

TABLE III
GAP DETECTION ERROR STATISTICS

| | Position error $[m]$ | | | Orientation error $[°]$ | | |
|---|---|---|---|---|---|---|
| | $\Delta x$ | $\Delta y$ | $\Delta z$ | $\Delta\phi$ | $\Delta\theta$ | $\Delta\psi$ |
| $\mu$ | 0.058 | 0.045 | 0.022 | 2.907 | 4.494 | 2.240 |
| $\sigma$ | 0.006 | 0.007 | 0.006 | 1.759 | 1.914 | 1.578 |
| 10% CI | 0.050 | 0.037 | 0.015 | 0.670 | 1.682 | 0.180 |
| 95% CI | 0.069 | 0.055 | 0.032 | 6.087 | 6.869 | 4.427 |

TABLE IV
POSE ERROR STATISTICS AT TRAVERSAL POINT.

| | Position error $[m]$ | | Orientation error $[°]$ | |
|---|---|---|---|---|
| | $\Delta y$ | $\Delta z$ | $\Delta\phi$ | $\Delta\theta$ |
| $\mu$ | 0.065 | 0.033 | 5.297 | 6.749 |
| $\sigma$ | 0.047 | 0.036 | 4.494 | 6.745 |
| 10% CI | 0.004 | 0.005 | 0.6207 | 0.9167 |
| 95% CI | 0.145 | 0.093 | 12.865 | 19.538 |

The overall weight of our quadrotor is $1.1\text{kg}$, with a thrust-to-weight ratio of 3.5. The arm length of the quadrotor is $22\text{cm}$, and the overall dimension is $35\text{cm} \times 20\text{cm}$ (the largest length measured between propeller tips), while the size of the gap used in experiments is $70\text{cm} \times 40\text{cm}$. When the vehicle is at the center of the gap, the long and short sides tolerances are only $17.5\text{cm}$ and $10\text{cm}$, respectively. In our experiments, the drone aims to fly through a variable-angle narrow gap back and forth.

*2) Experiment Results:* We design groups of experiments to demonstrate the robustness of the proposed gap detection algorithm as well as the control policy.

The accuracy of our onboard sensing method is first evaluated. We detect the narrow gap placed in different poses and compare the results with ground truth data from a motion capture system. The statistics of the measurement error are shown in Table III.

We then implement repetitive experiments to evaluate the whole system, where the quadrotor is required to traverse multiple gaps with different roll angles. Overall, we ran 87 traversals with the roll angle ranging from $-60°$ to $+60°$,

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

TABLE V
SUCCESS RATE IN REAL-WORLD EXPERIMENTS.

| Gap Roll Range[°] | [-60, -50) | [-50, -40) | [-40, -30) | [-30, -20) | [-20, -10) | [-10, 10] | (10, 20] | (20, 30] | (30, 40] | (40, 50] | (50, 60] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Experiment Results [1] | 100.0% | 87.5% | 85.7% | 88.9% | 100.0% | 100.0% | 100.0% | 100.0% | 72.7% | 71.4% | 80.0% |

[1] At least 5 traversal flights for each case.

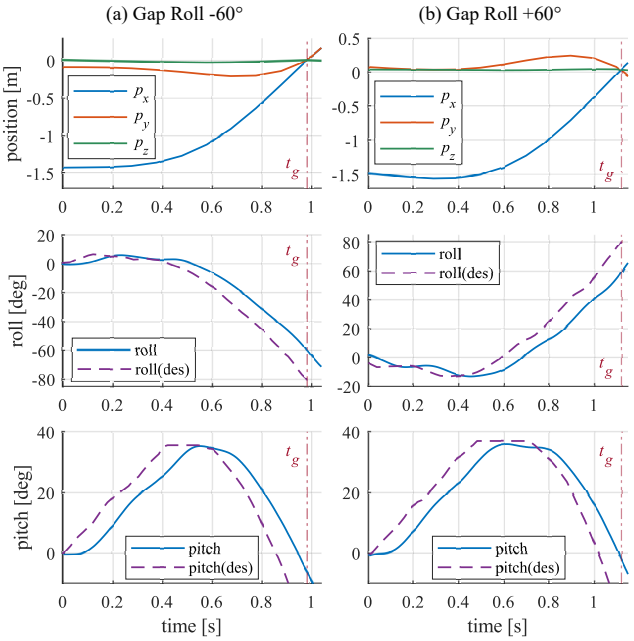


Fig. 6. Drone states over time during narrow gap traversal. Each column depicts the results of an experiment performed under distinct gap attitudes: (a) $-60°$ roll and (b) $+60°$ roll. The quadrotor reaches the center of the gap at $t = t_g$.

achieving a remarkable success rate of 87.36%. Success rates with respect to different angle ranges are calculated in Table V. Figure 6 shows the traversal motion with estimated position and orientation over time in two representative experiments. It can be observed that the drone orientations are planned precisely by the policy, resulting in an almost perfect posture of the vehicle when it reaches the gap plane. Specifically, at time $t_g$, the roll is close to the gap, and the pitch reduces to zero, while the position is close to the gap center. Table IV reports the statistics of the pose errors at time $t = t_g$, measured as a distance between drone posture and the gap. The errors include control errors introduced by control policy and detection errors introduced by gap detection algorithm. The statistics include both successful and unsuccessful experiments. Compared to the traversal error statistics result using traditional optimization-based method in [20], our framework achieves comparable results, demonstrating the robustness and the potential of exploiting the quadrotor's agility of learning-based methods.

### *E. Comparative Study with Traditional Method*

We compare the proposed traversal policy with a traditional method in [20], which designed a two-stage traversal trajectory based on the differential flatness property of quadrotors. We implement the trajectory planning method and control algorithm used in [20] on the same platform specified in Section VI-D1. As our main focus was on the control performance comparison, we employed a motion capture system to accurately detect the gap pose. As the maximum tilt angle of the

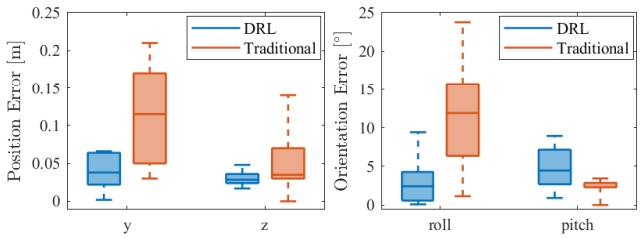


Fig. 7. Comparison of Traversal State Error.

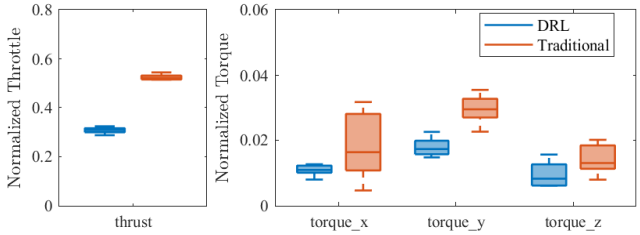


Fig. 8. Comparison of Actuator Control Effort.

gap is $45°$ in the experiments of [20], we performed tests in five different scenarios with gap roll angles of $0°$, $\pm 20°$, $\pm 45°$. Each scenario was repeated twice. We compared the traversal state error and actuator control efforts of each method, and the results are presented in Figures 7 and 8 respectively.

The computation time is compared in Python. For each control step, the traditional method takes $1.025\text{ms}$ to re-plan the trajectory and compute control commands, while the proposed method only requires $0.615\text{ms}$ to generate and process network outputs. Further discussions of the comparison results are presented in Section VII-A.

## VII. DISCUSSION

In this section, we discuss our system and provide more insights into the proposed methods.

### *A. Gap Traversal Control Policy*

Traditional quadrotor agile flight methods typically decouple trajectory planning and control. The performance and success rate depend highly on both the quality of the planned trajectory and the controller. Towards narrow gap traversal flight, the conventional approaches focus on planning dynamically feasible trajectories by exploiting differential flatness of the quadrotor [19], [20]. It is important to note that despite careful design and tuning of the algorithms, the minimal low-level control delays in the real-world implementations will lead to certain control errors during aggressive motions (i.e., linear velocity up to $3\text{m/s}$, angular velocity up to $4\text{rad/s}$), as shown in Figure 7. When the motion is relatively moderate (e.g., the pitch angle), the traditional method can perform better. In contrast, our learning-based method provides an end-to-end policy that learns and adapts to control response features during training. This eliminates the need for extensive controller design and tuning while still achieving better performance in aggressive maneuvers. Moreover, our method requires fewer control efforts to accomplish the task, as indicated by the

comparison result in Figure 8. This demonstrates the effectiveness of the control input penalty (7) and the exploration capabilities of our proposed method. Although the traditional method has derived closed-form solutions for re-planning, the proposed method with a lightweight end-to-end policy exhibits superior performance in computation time. One limitation of the proposed learning-based method is that it only has soft constraints by giving rewards.

### B. Scalability and Generalizability

We further explore the scalability and generalizability of the system through complementary experiments. Regarding gap detection, we successfully test our algorithm under varying illuminations and changing the gap size in the real-world experiments, respectively. However, the proposed gap detection method is limited to the marked rectangular frame. A structureless gap detection method could be considered in future work refer to [25]. For the control policy, we successfully test our algorithm with different quadrotor dynamics in SITL, while only requiring enough thrust-to-weight ratio (more than 2.5 for up to $60^\circ$ maneuvers). Furthermore, to test the ability of the whole proposed system, we considered a scenario for the quadrotor to fly back and forth through a gap with an increasing inclined angle. The experiment results are provided in Figure 1(b). We refer the reader to the accompanying video for more experiment details at https://youtu.be/06F6YDsypPQ.

## VIII. Conclusion

This letter presented a learning-based system for a quadrotor to fly through an unknown tilted narrow gap. Compared to our previous work, the training algorithm incorporated an input augmentation and a carefully designed reward function to handle variation in gap attitude. Additionally, an onboard sensing method is introduced for autonomous gap detection, eliminating the need for prior environmental knowledge. The end-to-end system is validated through real-world experiments, achieving a success rate of 87.36% in 87 traversals. To the best of our knowledge, this is the first work that performs the learning-based traversal of variable-tilted narrow gaps in the real world without prior knowledge of the environment.

One limitation of this study is that using Euler angles to represent orientations may introduce singularity issues for some extreme states. Future work will explore full-state *SE(3)* flight using rotation matrix or quaternion representations.

## Acknowledgement

The authors gratefully thank Yunfan Ren and Yixi Cai for their help in picture-making and helpful discussions.

## References

[1] M. Lu, H. Chen, and P. Lu, "Perception and avoidance of multiple small fast moving objects for quadrotors with only low-cost rgbd camera," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 657–11 664, 2022.

[2] M. O'Connell, G. Shi, X. Shi, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural-fly enables rapid learning for agile flight in strong winds," *Science Robotics*, vol. 7, no. 66, p. eabm6597, 2022.

[3] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[4] R. Penicka, Y. Song, E. Kaufmann, and D. Scaramuzza, "Learning minimum-time flight in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7209–7216, 2022.

[5] C. Xiao, P. Lu, and Q. He, "Flying through a narrow gap using end-to-end deep reinforcement learning augmented with curriculum learning and sim2real," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[6] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Learning high-speed flight in the wild," *Science Robotics*, vol. 6, no. 59, p. eabg5810, 2021.

[7] Q. Sun, J. Fang, W. X. Zheng, and Y. Tang, "Aggressive quadrotor flight using curiosity-driven reinforcement learning," *IEEE Transactions on Industrial Electronics*, 2022.

[8] S. Chen, Y. Li, Y. Lou, K. Lin, and X. Wu, "Learning real-time dynamic responsive gap-traversing policy for quadrotors with safety-aware exploration," *IEEE Transactions on Intelligent Vehicles*, 2022.

[9] J. Lin, L. Wang, F. Gao, S. Shen, and F. Zhang, "Flying through a narrow gap using neural network: an end-to-end planning and control approach," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3526–3533.

[10] 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.

[11] 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.

[12] Y. Ren, F. Zhu, W. Liu, Z. Wang, Y. Lin, F. Gao, and F. Zhang, "Bubble planner: Planning high-speed smooth quadrotor trajectories using receding corridors," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 6332–6339.

[13] 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.

[14] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3725–3732, 2018.

[15] B. Zhou, J. Pan, F. Gao, and S. Shen, "Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1992–2009, 2021.

[16] 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.

[17] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2096–2103, 2017.

[18] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[19] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, "Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 404–411, 2016.

[20] D. Falanga, E. Mueggler, M. Faessler, and D. Scaramuzza, "Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 5774–5781.

[21] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se (3)," in *49th IEEE conference on decision and control (CDC)*. IEEE, 2010, pp. 5420–5425.

[22] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas, "Reinforcement learning with augmented data," *Advances in neural information processing systems*, vol. 33, pp. 19 884–19 895, 2020.

[23] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[24] P. Lu, T. Sandy, and J. Buchli, "Nonlinear disturbance attenuation control of hydraulic robotics," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 1451–1458.

[25] N. J. Sanket, C. D. Singh, K. Ganguly, C. Fermüller, and Y. Aloimonos, "Gapflyt: Active vision based minimalist structure-less gap detection for quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2799–2806, 2018.

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