Supplementary Material – 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 supplementary document is organized as follows. Section I introduces the problem to be addressed and presents an overview of our method. Section II presents a detailed quadrotor dynamics model, which is used for quadrotor simulation during policy training. For the detailed methodology and results, please refer to our letter at https://arxiv.org/abs/2302.11233. We refer the reader to the accompanying video for more experiment details at https://youtu.be/HUTWBclayT8.

I. Problem Overview

We assume a quadrotor equipped with a depth camera flying in a workspace with a tilted narrow window on a wall. The quadrotor initially hovers in front of the gap and aims to fly behind. Our approach consists of two subsystems: perception and control, illustrated in Figure 1. The perception system estimates the position and orientation of the gap in the world frame using a single-view image collected by a forward-facing depth camera.

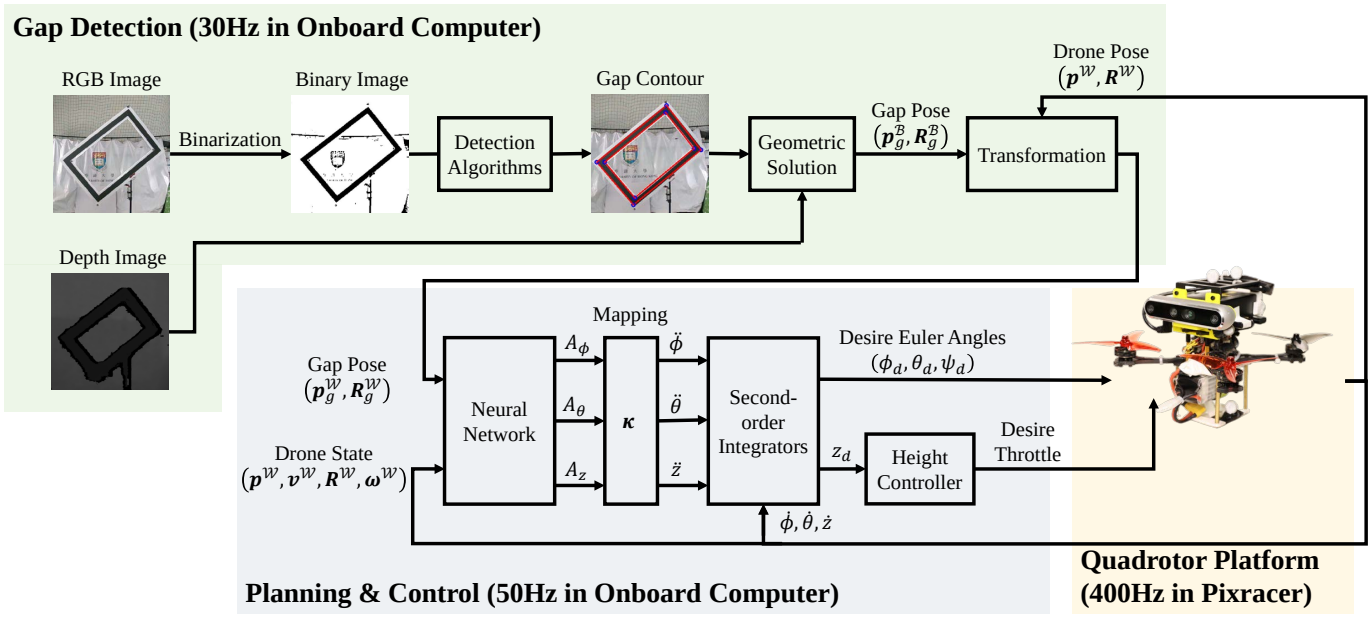


Fig. 1. The overall architecture in physical experiments.

Given the state of the drone and the gap, the control system uses a neural network to generate low-level control commands, which guide the quadrotor to complete the task. During the traversal, the drone has to maximize the distance to the gap edges to minimize the risk of collision. Thus, the planned traverse trajectory should try to intersect the center of the gap while simultaneously attaining the exact orientation of the gap, as illustrated in Figure 2. A precise *SE(3)* planning and control policy for quadrotor is required. We train our model in a self-supervised fashion in a physics simulation environment. In training, we assume the quadrotor and the window can be fully observed with reasonable estimation error.

Variation of gap orientation is also 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 wall usually has a few pitches. Thus, we mainly cope with the variation of roll angle in this paper.

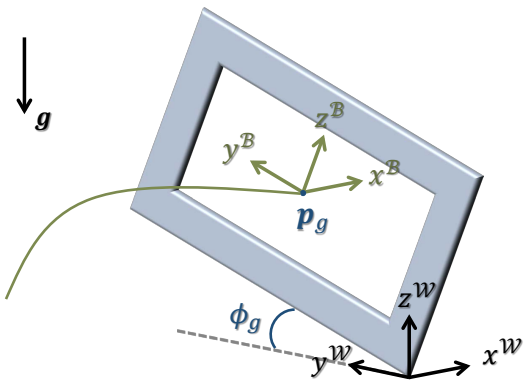


Fig. 2. Traversal Demonstration.

II. Quadrotor Dynamics for Training

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

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

where $\boldsymbol{p} = [p_x, p_y, p_z]^T$ and $\boldsymbol{v} = [v_x, v_y, v_z]^T$ are the position and velocity vector in the world frame. We use rotation matrix $\boldsymbol{R} \in \mathbb{SO}(3)$ to denote the rotation of the quadrotor and $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$ to represent the angular velocity in the body frame. The hat symbol $\hat{\boldsymbol{\omega}}$ denotes the skew-symmetric matrix form of the vector $\boldsymbol{\omega}$. $\boldsymbol{g}$ denotes the gravity vector in the world frame. $\boldsymbol{e}_3 = [0, 0, 1]^T$ is a constant vector in the body frame.

Additionally, f_T and $\boldsymbol{\tau}_T$ denote the vertical body thrust and three-dimensional body torques generated by four rotors. The propeller with rotor speed Ω_i is modeled as a first-order system $\dot{\Omega} = \frac{1}{k_{\text{delay}}}(\Omega_c - \Omega)$, where $\Omega_c \in [\Omega_{\min}, \Omega_{\max}]$ is the commanded rotor speed, and k_{delay} is the time delay constant. Individual motor thrusts f_i are then derived by thrust coefficient k_T as $f_i = k_T * \Omega_i^2$. Thus, body thrust f_T and body torques $\boldsymbol{\tau}_T$ can be calculated as

$$f_T = \sum_{i=1}^{4} f_i, \quad \boldsymbol{\tau}_T = \begin{bmatrix} (f_1 - f_2 - f_3 + f_4) \cdot l/\sqrt{2} \\ (f_1 + f_2 - f_3 - f_4) \cdot l/\sqrt{2} \\ (-f_1 + f_2 - f_3 + f_4) \cdot k_{TQ} \end{bmatrix} \tag{2}$$

where k_{TQ} denotes the ratio of moment coefficient to thrust coefficient, l is the arm length of the vehicle. Air drag is also modeled for aggressive motion. $\boldsymbol{f}_D$ represents air drag force and $\boldsymbol{\tau}_D$ is air drag torque, which are proportional to the square of linear body velocity $\boldsymbol{v}^{\mathcal{B}}$ and angular body velocity $\boldsymbol{\omega}$, respectively, with drag coefficients $\boldsymbol{k}_{f_D}, \boldsymbol{k}_{\tau_D}$.

Overall, the full-state and control input of quadrotor can be given as $\boldsymbol{x} = [\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{R}, \boldsymbol{\omega}]^T$, $\boldsymbol{u} = [f_T, \boldsymbol{\tau}_T]^T$. Additionally, we define the Euler angles of the quadrotor (ϕ, θ, ψ), which can be derived from the rotation matrix $\boldsymbol{R}$.

The parameters of the quadrotor dynamics used in training algorithm are summarized in Table I.

TABLE I
Parameters of Training Algorithm

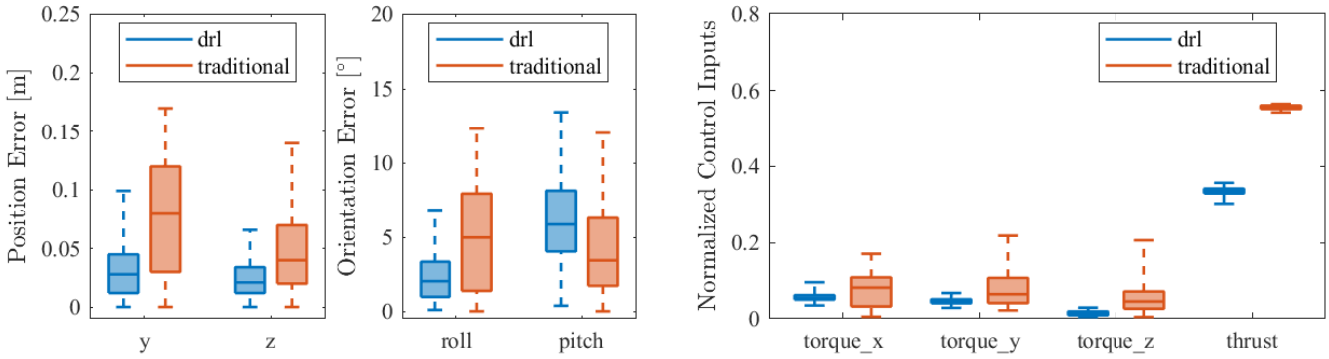
	Parameter	Value
Quadrotor	m [kg]	1.1
	$\mathrm{diag}(J)$ [kg m^2]	$[0.12, 0.12, 0.22]$
	$[\Omega_{\min}, \Omega_{\max}]$	$[50, 2000]$
	k_T	6×10^{-6}
	k_{TQ}	0.02
	l [m]	0.34
	$\boldsymbol{k}_{f_D}$ [N s^2 m^{-2}]	$[2.9, 2.9, 5.7] \times 10^{-2}$
	$\boldsymbol{k}_{\tau_D}$ [N s^2]	$[3.2, 3.2, 1.7] \times 10^{-3}$

III. Comparative Study with Traditional Method

We compare the proposed traversal policy with a traditional method proposed by Falanga *et al.* [1], which was imitated by [2]. The method 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 [1] and carefully tune the controller parameters for optimal performance. Comprehensive tests in both SITL and real-world experiments are conducted with the same platform as

described in our manuscript. As the maximum tilt angle of the gap is 45° in the experiments of [1], we limited the gap roll angle to no more than 45° in our tests.

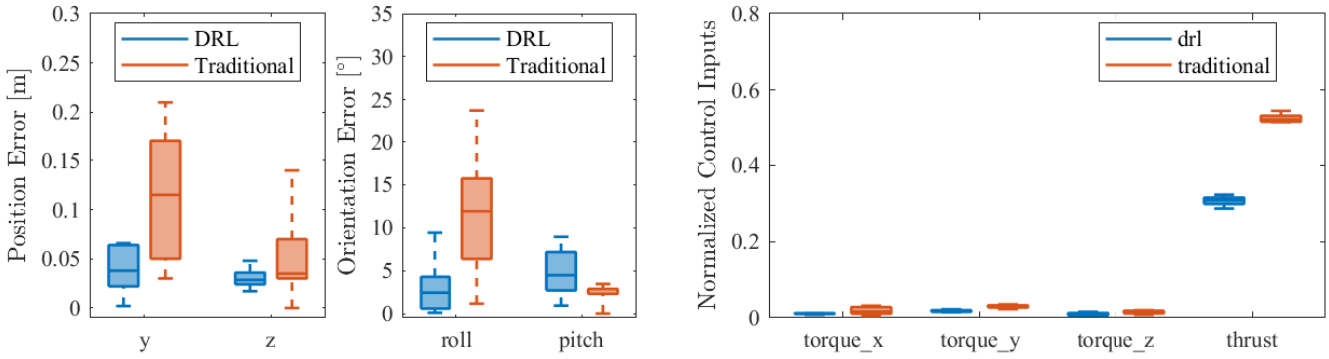
In the simulation, we evaluated our method across seven different scenarios, varying the gap roll angle at 0°, $\pm 15^\circ$, $\pm 30^\circ$, and $\pm 45^\circ$. Each scenario was repeated ten times to ensure statistical robustness. We compared the traversal state error and actuator control efforts of each method in simulation, and the results are presented in Figure 3. For the real-world experiments, we performed tests in five different scenarios with gap roll angles of 0°, $\pm 20^\circ$, and $\pm 45^\circ$. Each scenario was repeated twice. As our main focus was on the control performance comparison, we employed a motion capture system to accurately detect the gap pose in these experiments. The comparison results of real-world experiments are shown in Figure 4. The real-world experiment results have also been presented in Section VI-E of our letter.



(a) Comparison of Traversal State Error

(b) Comparison of Actuator Control Effort

Fig. 3. Comparison Study in Simulation.



(a) Comparison of Traversal State Error

(b) Comparison of Actuator Control Effort

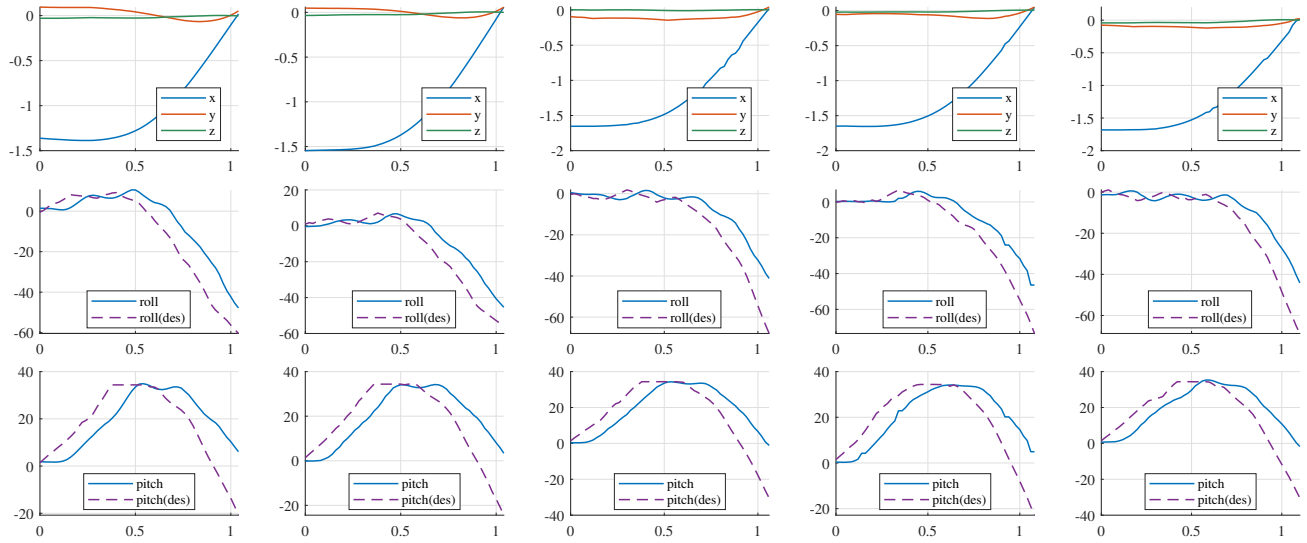
Fig. 4. Comparison Study in Experiment.

The simulation and experimental results consistently demonstrate the superior performance of our method. It is important to note that despite careful design and tuning of the traditional 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 $3\mathrm{m/s}$, angular velocity up to $4\mathrm{rad/s}$). 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 of actuator control efforts. This demonstrates the effectiveness of our control input penalty and the exploration capabilities of our proposed method.

IV. Robustness Study of Control Policy

A. Wind Disturbance

To test the policy performance under wind disturbance, we conduct several tests in SITL by applying a constant external force to simulate the effect of wind. Firstly, we estimate the external force F bring by wind speed v_w. The lateral area of the quadrotor in SITL is $A = 0.0517\mathrm{m/s^2}$. The air density is $\rho_a = 1.29\mathrm{kg/m^3}$. Then the lateral force can be as $F = \frac{1}{2}\rho_a A v_w^2$. In the simulation, We start at zero and gradually increase the wind speed at intervals of one. The drone can finally endure a maximum lateral wind speed of $v_w = 3\mathrm{m/s}$, which corresponds to the external force $F = 0.300\mathrm{N}$. For lateral wind speed $v_w \geq 4\mathrm{m/s}$, the drone is not able to hover stably. We conduct five tests with a wind speed of $y = 0\mathrm{m/s}$, $y = \pm 2\mathrm{m/s}$, $y = \pm 3\mathrm{m/s}$, as shown in figure 5.



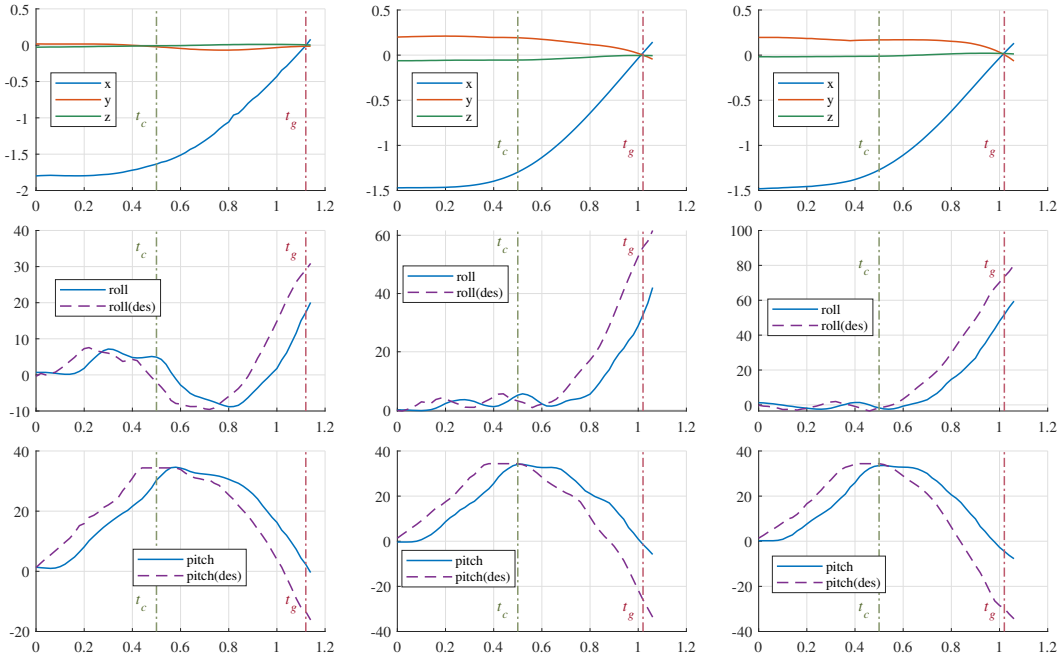
(a) wind speed $y = +3\text{m/s}$ (b) wind speed $y = +2\text{m/s}$ (c) wind speed $y = 0\text{m/s}$ (d) wind speed $y = -2\text{m/s}$ (e) wind speed $y = -3\text{m/s}$

Fig. 5. External Disturbance Test

As shown in Figure 1, there is a height controller (runs on onboard computer) and a low-level attitude controller (runs on Pixracer) before processing policy commands to the drone dynamics. Therefore, the external force resistance ability mainly depends on these controllers. The simulation results are basically consistent with our analysis.

B. Gap Orientation Change Mid-flight

We conduct tests in SITL with the gap orientation changed at 0.5s after the traversal policy runs. Specifically, the gap roll angle is initially set as $-40°$ and is changed to $+20°$ at 0.5s in the first test. We gradually increase the angle of change. In the second and third tests, the roll is reset from $-40°$ to $+40°$, and from $-60°$ to $+60°$, respectively. The results are shown in figure 6. t_c denotes the gap change time, and t_g denotes the time when the drone reaches the gap plane. As indicated in the figure, the attitude of the quadrotor is adjusted in time, and the drone successfully achieves the goal orientation exactly when it reaches the gap plane.



(a) Roll from $-40°$ to $+20°$ (b) Roll from $-40°$ to $+40°$ (c) Roll from $-60°$ to $+60°$

Fig. 6. Gap Orientation Change Test

In each control step, the neural network takes the gap orientation as input and generates control commands. Therefore, the control policy could adjust the control commands immediately after it receives a new orientation input, unless the velocity of the drone is too quick to brake. In our flight performance, the movement in the first $0.5\mathrm{s}$ is relatively moderate. Thus, we choose $0.5\mathrm{s}$ as the orientation change time, which is approximately the half time for the overall flight.

REFERENCES

[1] 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.

[2] 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.