SmoothLander: A Quadrotor Landing Control System with Smooth Trajectory Guarantee Based on Reinforcement Learning

Chenyu Zhao*
Haoyang Wang*
Shenzhen International Graduate School,
Tsinghua University
{zhaocy22,haoyang-22}@mails.tsinghua.edu.cn

Jiaqi Li
Shenzhen International Graduate School,
Tsinghua University
li-jq22@mails.tsinghua.edu.cn

Fanhang Man, Shilong Mu
Shenzhen International Graduate School,
Tsinghua University
{mfh21,msl22}@mails.tsinghua.edu.cn

Wenbo Ding
Shenzhen International Graduate School, Tsinghua University
Pengcheng Laboratory
RISC-V International Open Source Laboratory
Shenzhen, China
ding.wenbo@sz.tsinghua.edu.cn

Xiaoping Zhang
Shenzhen International Graduate School, Tsinghua University
Department of Electrical, Computer and Biomedical Engineering,
Ryerson University
Shenzhen, China
xiaoping.zhang@sz.tsinghua.edu.cn

Xinlei Chen†
Shenzhen International Graduate School, Tsinghua University
Pengcheng Laboratory
RISC-V International Open Source Laboratory
Shenzhen, China
chen.xinlei@sz.tsinghua.edu.cn

ABSTRACT

The landing process of the quadrotors can be affected by the disturbance from the ground effect when approaching the landing surface. Such a disturbance significantly increases the chances of collision and jittering of the quadrotors, thereby posing threats to the safety of both the quadrotors and the mounted equipment. In light of this, we propose SmoothLander, an aerodynamics and reinforcement learning-based control system to stabilize the quadrotors under the influence of the ground effect and control noise. First, we design a landing trajectory for the quadrotor in accordance with aerodynamics. Then we design a reinforcement learning-based command generator to effectively optimize the quadrotor's landing behavior. We evaluate our control system through physical feature-based simulation and in-field experiments. The results show that our method can enable the quadrotor to land more smoothly and stable against control noise than the baseline.

CCS CONCEPTS

• **Applied computing → Aerospace**; • **Computing methodologies → Reinforcement learning**; • **Computer systems organization → Sensors and actuators**.

KEYWORDS

Quadrotor, Ground Effect, Reinforcement learning

ACM Reference Format:
Chenyu Zhao, Haoyang Wang, Jiaqi Li, Fanhang Man, Shilong Mu, Wenbo Ding, Xiaoping Zhang, and Xinlei Chen. 2023. SmoothLander: A Quadrotor Landing Control System with Smooth Trajectory Guarantee Based on Reinforcement Learning. In *Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp/ISWC '23 Adjunct), October 8–12, 2023, Cancún, Mexico.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3544793.3560412

1 INTRODUCTION

As drone technology continues to mature, researchers and industries endue drones with more and more diverse applications and tasks, including environmental sensing[1, 2], surveillance[3, 4], delivery[5, 6], disaster relief[7, 8], etc. The drones, such as quadrotors, can use loaded sensors or software algorithms to accurately locate themselves[9, 10], and then achieve highly stable hovering to execute predetermined tasks. However, the performance of landing for quadrotor is fairly unstable due to the phenomenon of ground effects and the control noise[11], as illustrated in Fig.1. Such a phenomenon causes the quadrotor to experience an extra lift when close to a surface below due to the rebound airflow[12]. While this lift might be helpful to push the quadrotor to take off, it induces disturbance, compromising stability and smoothness during landing. Such disturbances might lead to jittering landings and pose threats to both the quadrotor and the mounted equipment[13].

Researchers have proposed traditional and learning-based methods to reduce the disturbance during low-height flight. Traditional methods include real-time adjustment through linear or self-adaptive control[14, 15], which can effectively compensate for the impact of ground effects[15]. However, due to the fact that adjustment is triggered after experiencing turbulence[16], traditional methods always lead to delayed responses. Moreover, the ground effect is very

*Both authors contributed equally to this research.
†Xinlei Chen is the corresponding author.

UbiComp/ISWC '23 Adjunct, October 8–12, 2023, Cancún, Mexico

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/3544793.3560412

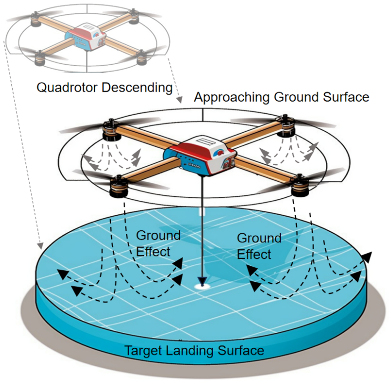

Figure 1: The quadrotor is affected by the ground effect in the landing process

intricate and subject to multiple influence factors including types of quadrotors, landing surface, natural wind, and control noise[11]. It is impractical to model for a large variety of conditions[17]. In the case of learning-based methods, artificial neural networks (ANN) have been applied to predict the ground effect[18]. It was demonstrated that the ANN method significantly outperforms a baseline nonlinear tracking controller in the cases of landing. However, these methods lack stability guarantees[12]. Reinforcement learning (RL) has been successfully applied to a wide range of control applications across various domains. It is a machine learning approach that focuses on an agent learning to make decisions or take actions in an environment to maximize a cumulative reward signal[19]. The application of RL in quadrotor control was first introduced in 2005 when Waslander et al[20] developed a simple linear altitude controller. Also, RL has been employed for controlling quadrotors in landing tasks[21], showcasing exceptional landing performance[22, 23]. However, in-field wind influences and control noise is often neglected when utilizing RL for control.

The problem this paper tries to solve is: **how to control the quadrotors to land smoothly and stably under the interference of the ground effect and control noise?** This is a non-trivial problem and poses significant challenges. To address these challenges, this paper proposes SmoothLander, a reinforcement learning-based control system for compensating for the ground effect and controlling noise during the landing process.

One of the two challenges **(C1)** is that: on the one hand, the quadrotor needs to adjust its thrust of four rotors based on its current state to compensate for the disturbance. On the other hand, the quadrotor may not return directly because the compensating commands do not consider the future ground effect caused by the adjustment. Thus, delayed compensating time is spent, and oscillation unavoidably occurs during this loop process. Additionally, control noise is difficult to precisely measure and filter in real time, further complicating the controller design process. To tackle the challenge **(C1)**, we propose a reinforcement learning-based method for generating safe command sequences in advance under the interference of control noise and ground effect, without delayed adaption. The generated commands account for the future state changes from the combination of ground effect and thrust output. Leveraging the powerful modeling and searching capabilities of reinforcement learning[24, 25], our approach optimizes the quadrotor's control strategy to pre-generate control commands that follow a designed trajectory. This generated trajectory is proven to avoid oscillation and provide continuity, based on the features of a critical damping state in a second-order damping system.

The second challenge **(C2)** is the lack of training data for quadrotors to learn the distribution and influence of ground effect. This challenge arises because the quadrotor has a large state space and a high-dimensional transition matrix, caused by the highly dynamic nature of quadrotors' ground effect, making it hard to collect complete command-state pair data. To address the challenge **(C2)**, the reinforcement learning training environment includes an aerodynamics ground effect model and control noise distribution collected from the real world. The disturbance force influences from this physical GE model and control noise distribution help quadrotors learn how to compensate. We evaluate our method in both physical feature-based simulation and real-world environments. Results demonstrate that SmoothLander achieves smoother and more stable landings than the baseline.

The main contributions of this paper are as follows:

- Design a reinforcement learning-based landing control system, SmoothLander, for quadrotors to ensure smoothness and stability while compensating for the influence of the ground effect and control noise by considering the future interaction with the ground effect, without the oscillation for adjusting.
- Propose a physical feature-based method to generate training data in the reinforcement learning environment for the quadrotor controller to learn the strategy of resisting ground effect.
- Evaluate the effectiveness of the SmoothLander system through both physical feature-based simulation and real-world implementation on quadrotors. The results show that our method can enable the quadrotor to land smoothly and stably with light contacts.

The rest of this paper is organized as follows: Section 2 describes the problem definition. Section 3 represents our controller framework. Section 4 shows our experimental results and evaluation. In the last, Section 5 concludes this paper. We also attach an Appendix Section including several background definitions.

2 PROBLEM DEFINITION

In this section, we first introduce the background definitions of the landing system, including the environment descriptions and the quadrotor states based on dynamic models. Subsequently, we describe our objective for the landing system. Lastly, we formalized the problem of optimizing the landing controller system.

2.1 Key Definitions

Environment and Quadrotors' States: The overall assumed environment of the landing task is based on a free space with a bottom boundary as the ground surface. The thrust coefficient and extra force from ground effect and unknown outside wind, respectively. The control noise mainly contains actuation noise and acceleration noise. The detailed descriptions and definitions are attached in the Appendix section.

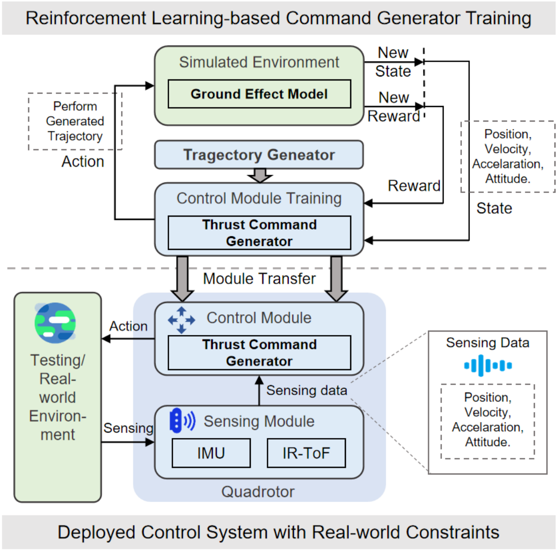

Figure 2: System Overview

2.2 Problem Formulation

In order to achieve a stable quadrotor landing, the trajectory and controller design are crucial factors for ensuring fast, smooth, and stable flight[26]. A desired trajectory $\mathbf{p_d}$ is designed to achieve a stable landing, followed by the design of a stabilized controller that compensates for disturbances caused by ground effect and control noise. The controller enables the quadrotor to follow the desired path during the landing task. **The optimization objective** of the system is to minimize the distance between the real-time position and the position along the desired trajectory, while ensuring the velocity and acceleration of the quadrotors meeting specific constraints during the entire landing process. **The main challenge** comes from uncertain mapping between actuation $\mathbf{u}$ and acceleration $\mathbf{a}$ under disturbances. We have the mathematical formulation of the stabilized controller problem as:

$$\arg\min_{\mathbf{u}} \|\mathbf{p}_{init} + \int_0^T (\mathbf{v}_{init} + \int_0^T \mathbf{a}\, dt)dt - \mathbf{p}_d(t)\|_2 \tag{1}$$

$$s.t. \quad \mathbf{a}_{min} \leq \mathbf{a} \leq \mathbf{a}_{max}$$

$$\mathbf{v}_{min} \leq \mathbf{v} = \mathbf{v}_{init} + \int_0^T \mathbf{a}\, dt \leq \mathbf{v}_{max}$$

where $\mathbf{p}_d(t)$ and $\mathbf{v}_{init}$ are desired position of quadrotor's flight at time t and initial velocity, respectively. Acceleration $\mathbf{a}$ and velocity $\mathbf{v}$ are limited by the hardware constrain and the purpose of safety and smooth flight.

3 SYSTEM DESIGN

In this section, the components and design details of the control system will be introduced. The system comprises three main components: the *Trajectory Generator*, the *Ground Effect Modeling*, and the *Thrust Command Generator*, as illustrated in Fig.2. The Trajectory Generator generates desired trajectories for quadrotors to follow, reducing the chances of oscillations and collisions. Ground Effect Modeling provides sufficient training data for quadrotors to learn the nature of ground effect within the reinforcement learning environment. The Thrust Command Generator generates control commands to ensure that the quadrotor follows the designed trajectory while also compensating for ground effect and control noise, thereby ensuring smooth landing flight.

3.1 Trajectory Generator

Limited by the constraints of kino-dynamics for quadrotors, the trajectory must satisfy the requirements of a smooth and stable landing without oscillation, which can be guaranteed by a well-designed pre-defined trajectory. Therefore it should also be designed for quadrotors to achieve without extra help.

The motion of a quadrotor is subject to the constraints imposed by its kino-dynamics and physical hardware design[27], which necessitates that the trajectory's smoothness and stability adhere to these limitations. Specifically, to ensure smoothness and stability, the position and velocity sequence must maintain continuity without any glitches and ensure soft contact with the ground. In terms of operational constraints, the velocity and acceleration must not exceed the drone's performance limit. To ensure safety, a deceleration buffer is necessary for the ending period to avoid collisions, oscillations, or heavy impact. To achieve these features, we refer to the critical damped state of a second-order damping system, which is a widely used design in control systems. The second-order damping system has three primary states: overdamping, underdamping, and critical damping. The first state takes a long time to return to the equilibrium position, and the second state oscillates around the desired position. In contrast, the critical damped state ensures smoothness and a relatively quick return to the final stable position without any oscillations. The smoothness, stability, and quick response of the critical damped state align with the requirements of the quadrotor landing trajectory. Thereby, following the model of the critical damped state in a second-order damping system, we design such trajectory $\mathbf{p}_d = [p_{d_x}, p_{d_y}, p_{d_z}]^\top \in \mathbb{R}^3$, as:

$$\mathbf{p}_d(t) = e^{(-Ct)}(1 + Ct)(\mathbf{p}_{init} - \mathbf{p}_{end}) + \mathbf{p}_{end}, \ t \in \mathbb{R}^+, \tag{2}$$

where $\mathbf{p}_{init} = [p_{init_x}, p_{init_y}, p_{init_z}]^\top \in \mathbb{R}^3$ is the initial position and $\mathbf{p}_{end} = [p_{end_x}, p_{end_y}, p_{end_z}]^\top \in \mathbb{R}^3$ is the destination of the UAV. Time t starts counting from 0 at the beginning of the task process and C controls the velocity of the flight.

It can be proved that $\mathbf{p}_d$ is continuous and smooth based on the continuity and smoothness of its first-order and second-order derivatives. Furthermore, this trajectory guarantees that the velocity and acceleration of the quadrotor remain bounded, as demonstrated by the maximum values of its derivatives. To prevent the quadrotor from receiving motion commands[28] that exceed its hardware design capabilities, we can constrain its motion by selecting an appropriate value for the constant C.

3.2 Ground Effect Modeling

The ground effect refers to the upward force created by the reflection of the downward wind generated by the quadrotor's rotors. During the landing process, this effect can be undesirable and its magnitude is strongly dependent on several factors, including the distance to the ground, the hardware configuration, and the current actuation of the rotors. The changing state of the quadrotor can cause variations in the magnitude of the disturbance forces induced by the ground effect. To predict and compensate for this effect, it is crucial to have a precise model that provides prior knowledge to the quadrotor for self-adjustment. To address this issue, we adopt a model of the ground effect on the quadrotor based on[29], which is derived from the physical parameters of quadrotors and experimental results. By decoupling the interaction between the quadrotor and the ground effect, this model can help to mitigate the impact of the ground effect during the landing process. The model is:

$$\begin{aligned} T(n, p_z) = Kn^2/[1 - (D/4p_z)^2 - D^2 p_z/\sqrt{(d^2 + 4p_z^2)^3} \\ - (D^2/2)(p_z/\sqrt{(2d^2 + 4p_z^2)^3}\ ) \\ - 2D^2(p_z/\sqrt{(b^2 + 4p_z^2)^3})I_b], \\ K = \frac{k_T}{\rho D^4}, \end{aligned} \tag{3}$$

where T is the actual thrust containing the ground effect, ρ is the air density, D is the radius of the propeller, d is the distance between the closed two rotor axes, b is the distance between two diagonal rotor axes, and I_b is the empirical body lift coefficient. Additionally, k_T The denominator of the model contains additional terms beyond the first two terms to account for flow recirculation and the central body raise. The lift generated by the model considers the airflow velocity at the body's central position and modifies its impact using I_b. This model has been validated experimentally on a real-world testbench[11].

3.3 Thrust Command Generator

Designing a controller to generate stabilized thrust output for quadrotors while dealing with control noise and coupled ground effect using traditional methods is challenging. Although using reinforcement learning can satisfy these requirements, it introduces new challenges, such as generating safe actions and simulating physical ground effect[30, 31]. To address these challenges, we adopt a changeable action space and physical feature-based ground effect model described in section 3.2 for our command generator.

Action Space: The action space for the thrust command generator is limited by the quadrotor's hardware design and must be within a safe region to prevent instability caused by excessive actuation commands or high velocities. Moreover, the velocity of the quadrotor must change slowly to ensure stability and prevent oscillations, and the velocity should be low when approaching the final destination to reduce the risk of collision or exceeding the hover position. To meet these requirements, the action space is constructed by adjusting based on the previous state.

State Space: The quadrotor is confined to a motion region within a free space Ω that includes a rigid ground boundary. The quadrotor's states are defined as $\mathbf{p}$, $\mathbf{v}$, $\mathbf{a}$, and $\boldsymbol{\omega}$.

Reward: The reward function is denoted as $r\left(\mathbf{u}t, (\mathbf{p}t, \mathbf{v}t, \mathbf{a}t, \boldsymbol{\omega}_t)\right)$, to minimize the difference between the actual positions and the desired positions for the quadrotor to closely follow the desired trajectory, defined in Eq(1).

4 EVALUATION

In this section, we test our method in both simulation and the real world, aiming to evaluate its effectiveness and performance. For simulation, our method is compared with a non-linear tracking controller in one-dimensional z-axis. We also implement our methods on Crazyflie 2.1 in the real world to test the performance.

Algorithm 1 Learning a policy for the control of quadrotors based on reinforcement learning algorithm

```
Randomly initialize a model $\pi$
for epoch=1:$M$ do
    randomly initialize $\mathbf{p}_t$, $\mathbf{v}_t$, $\mathbf{a}_t$,$\boldsymbol{\omega}_t$;
    for $t = 0 : T - 1$ do
        Quadrotor executes an action $\mathbf{u}_t = \pi(\mathbf{p}_t)$;
        Calculate $\mathbf{f_u}$, $\mathbf{f_w}$, $\boldsymbol{\tau}_\mathbf{u}$, $\boldsymbol{\tau}_\mathbf{w}$;
        Calculate $\mathbf{a}_{t+1}$, $\mathbf{v}_{t+1}$;% according to Eq(5)
        Calculate $\mathbf{p_{t+1}}$;% according to Eq(5)
        $\mathbf{r}_{t+1} = R\left(\mathbf{p}_t, \mathbf{u}_t, \mathbf{p}_{t+1}\right)$
        Update the model $\boldsymbol{\pi}$
        Update the state, $\mathbf{p}_{t+1}$
    end for
end for
```

4.1 Experimental Setup

Evaluation Metrics: The objective of our SmoothLander is to land with a smooth and stable trajectory with interference from GE and control noise. Therefore, the error between the quadrotor's actual trajectory and its desired trajectory describes the performance of the SmoothLander controller in compensating for the ground effect.

Baseline: We compare our method with a nonlinear tracking controller that regards $\mathbf{f_w} \equiv 0$ in Eq(5). It tries to maintain a composite variable $\mathbf{s} = 0$, which

$$\mathbf{s} = \dot{\mathbf{e}}_p + \Gamma \mathbf{e_p} = \dot{\mathbf{p}} - \mathbf{v}_r \tag{4}$$

where $\mathbf{e_p} = \mathbf{p} - \mathbf{p_d}$ is the tracking error, $\mathbf{v}_r$ is a reference velocity, Γ is a positive definite matrix. Thus, tracking the trajectory can be transformed to just maintaining the velocity $\mathbf{v}$ to follow $\mathbf{v}_r = \dot{\mathbf{p}}_d - \Gamma \mathbf{e}_p$.

Experimental Setup: For both the simulation and real-world experiment, the environment is an open-air ground without any natural winds. The platform used is a nano-quadrotor, Crazyflie 2.1[32, 33], equipped with an infrared ToF distance sensor for flight height data collection. The command input is the actuation signal $\mathbf{u}$ generated from the Thrust Command Generator. To build a close-to-real model for simulation and real-world tests, we measured Crazyflie's mass m, rotor radius D, the distance between rotor axes d, diagonal length b, as well as air pressure ρ. Also, we tested the thrust constant k_T, actuation noise $\boldsymbol{\sigma_u}$, and acceleration noise $\boldsymbol{\sigma_a}$ from the real world. The control command rate, the attitude update rate, and the position estimation rate are all the same at 100Hz. The environment of both The detailed experiment process is described in the next subsection.

4.2 System Performance

Overall Landing Performance: In this study, we conduct a landing simulation to evaluate the performance of our trajectory tracking method. To assess the filtering of unwanted disturbances from the ground effect, we carry out 100 tests and compare the mean trajectories of the quadrotor using our method with those using a

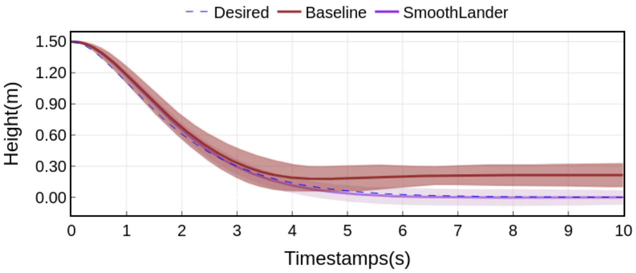

Figure 3: Mean trajectories of landing simulations. The lines indicate the mean trajectories of 100 simulations with the control noise of 0.01 for both acceleration and actuation noise variance. The shadowed areas show trajectory variance. Quadrotor using SmoothLander is nearly following the desired trajectory while using baseline struggles above the ground.

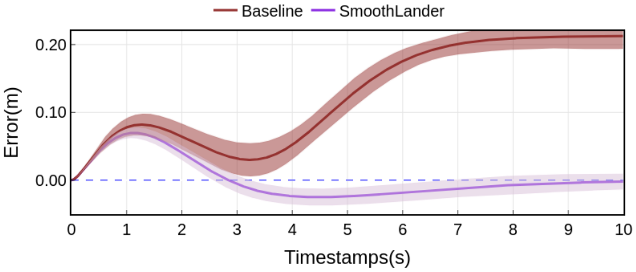

Figure 4: Mean errors compared with the desired trajectory are shown as lines and variances are shown with shadow area. SmoothLander always has lower errors and variances, which prove its stability and capability when resisting GE.

baseline method. The quadrotor first ascends to an initial height of $p_z = 1.5(m)$ and then hovers at this height before initiating the landing process along the desired trajectory $\mathbf{p}_d$ with an initial zero-velocity, ultimately coming to a stop at ground level. As shown in Fig.3, our Smoothlander model successfully brings the quadrotor to a low-velocity contact with the ground, while the baseline method struggles to approach the ground under the influence of the upward force. This demonstrates the efficacy of our reinforcement learning model in predicting the impact of the ground effect and generating compensated control commands to follow $\mathbf{p}d$. In addition, we compare the mean absolute errors (MAE) between the simulated trajectory $\mathbf{p}$ and the desired trajectory $\mathbf{p}d$, as shown in Fig.4. The results indicate that SmoothLander has smaller mean errors and smaller standard variances, indicating high-precision control and less fluctuation during the landing process.

Then we implement Smoothlander and baseline in the real world on a Crazyflie 2.1, which only controls the vertical power output with equal four rotors' thrust. The experiment is conducted identically to the simulation, with the same initial settings. The quadrotor using the baseline controller stops descending before reaching the final height, but Smoothlander contacts the ground steadily at last, as is shown in Fig.5. However, due to the high data variance of the low-cost sensors and the variable power output associated with the battery, the performance of the system is somewhat suppressed.

Control Noise Reduction Performance: We compare the performance of two methods under different levels of actuation noise and acceleration noise in a simulation. The mean cumulative position errors resulting from 100 simulations with varying noise variances are shown in Fig.6. As the variances increase, both methods experience growing mean errors due to the increasing uncertainty of the actions' effects. However, Smoothlander consistently exhibits smaller cumulative errors than the baseline, indicating its superior ability to filter control noise. This ability is likely attributed to the training process, during which our reinforcement learning model learns the noise distribution by interacting with the environment.

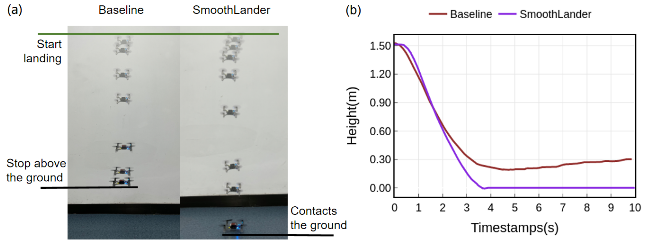

Figure 5: Landing performance in real-world experiments. (a)Landing Crazyflie. (b)Trajectories of landing. The quadrotor using baseline does not land on the ground as last, but using SmoothLander it does.

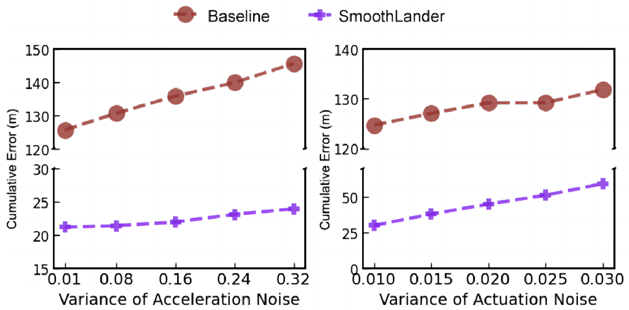

Figure 6: Trajectory errors with different noise variance. Although errors increase as the variance rises in both methods, SmoothLander always has much lower errors.

5 CONCLUSION

This paper proposes a landing control system, SmoothLander, designed to achieve smooth and stable flight even in the presence of ground effect interference and control noise. The system employs reinforcement learning to overcome the disturbance force from the ground effect and reduce control noise, thereby enabling more precise flight. Results from both physical feature-based simulations and in-field implementation tests demonstrate that SmoothLander outperforms the baseline in terms of landing performance.

6 ACKNOWLEDGMENTS

This paper was supported by the National Key R&D program of China (2022YFC3300703), Guangdong Innovative and Entrepreneurial Research Team Program (2021ZT09L197), Shenzhen 2022 Stabilization Support Program (WDZC20220811103500001), and Tsinghua Shenzhen International Graduate School Cross-disciplinary Research and Innovation Fund Research Plan (JC20220011).

REFERENCES

[1] Norman Kerle, Francesco Nex, Markus Gerke, Diogo Duarte, and Anand Vetrivel. Uav-based structural damage mapping: A review. *ISPRS international journal of geo-information*, 9(1):14, 2019.

[2] Haidong Zhang, Lingqing Wang, Ting Tian, and Jianghai Yin. A review of unmanned aerial vehicle low-altitude remote sensing (uav-lars) use in agricultural monitoring in china. *Remote Sensing*, 13(6):1221, 2021.

[3] Xinlei Chen, Aveek Purohit, Shijia Pan, Carlos Ruiz, Jun Han, Zheng Sun, Frank Mokaya, Patric Tague, and Pei Zhang. Design experiences in minimalistic flying sensor node platform through sensorfly. *ACM Transactions on Sensor Networks (TOSN)*, 13(4):1–37, 2017.

[4] Andrey V Savkin and Hailong Huang. A method for optimized deployment of a network of surveillance aerial drones. *IEEE Systems Journal*, 13(4):4474–4477, 2019.

[5] Dane Bamburry. Drones: Designed for product delivery. *Design Management Review*, 26(1):40–48, 2015.

[6] Xuecheng Chen, Haoyang Wang, Zuxin Li, Wenbo Ding, Fan Dang, Chengye Wu, and Xinlei Chen. Deliversense: Efficient delivery drone scheduling for crowdsensing with deep reinforcement learning. In *Adjunct Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2022 ACM International Symposium on Wearable Computers*, pages 403–408, 2022.

[7] Sudipta Chowdhury, Adindu Emelogu, Mohammad Marufuzzaman, Sarah G Nurre, and Linkan Bian. Drones for disaster response and relief operations: A continuous approximation model. *International Journal of Production Economics*, 188:167–184, 2017.

[8] Haoyang Wang, Xuecheng Chen, Yuhan Cheng, Chenye Wu, Fan Dang, and Xinlei Chen. H-swarmloc: Efficient scheduling for localization of heterogeneous mav swarm with deep reinforcement learning. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 1148–1154, 2022.

[9] Carlos Ruiz, Xinlei Chen, Lin Zhang, and Pei Zhang. Collaborative localization and navigation in heterogeneous uav swarms: Demo abstract. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 324–325, 2016.

[10] Xinlei Chen, Carlos Ruiz, Sihan Zeng, Liyao Gao, Aveek Purohit, Stefano Carpin, and Pei Zhang. H-drunkwalk: Collaborative and adaptive navigation for heterogeneous mav swarm. *ACM Transactions on Sensor Networks (TOSN)*, 16(2):1–27, 2020.

[11] Pedro Sanchez-Cuevas, Guillermo Heredia, and Anibal Ollero. Characterization of the aerodynamic ground effect and its influence in multirotor control. *International Journal of Aerospace Engineering*, 2017, 2017.

[12] Antonio Matus-Vargas, Gustavo Rodriguez-Gomez, and Jose Martinez-Carranza. Ground effect on rotorcraft unmanned aerial vehicles: A review. *Intelligent Service Robotics*, 14(1):99–118, 2021.

[13] Chris Dinelli, John Racette, Mario Escarcega, Simon Lotero, Jeffrey Gordon, James Montoya, Chase Dunaway, Vasileios Androulakis, Hassan Khaniani, Sihua Shao, et al. Configurations and applications of multi-agent hybrid drone/unmanned ground vehicle for underground environments: A review. *Drones*, 7(2):136, 2023.

[14] Kirill V Rozhdestvensky. *Aerodynamics of a lifting system in extreme ground effect.* Springer Science & Business Media, 2000.

[15] Daewon Lee, H Jin Kim, and Shankar Sastry. Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter. *International Journal of control, Automation and systems*, 7:419–428, 2009.

[16] Manmohan Sharma and Indrani Kar. Control of a quadrotor with network induced time delay. *ISA transactions*, 111:132–143, 2021.

[17] Bara J Emran and Homayoun Najjaran. A review of quadrotor: An underactuated mechanical system. *Annual Reviews in Control*, 46:165–180, 2018.

[18] M Habibnia and J Pascoa. Ann assisted flow modeling and analysis for a cyclorotor in ground effect. *Aerospace Science and Technology*, 95:105495, 2019.

[19] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[20] Steven Lake Waslander, Gabriel M Hoffmann, Jung Soon Jang, and Claire J Tomlin. Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3712–3717. IEEE, 2005.

[21] Jemin Hwangbo, Inkyu Sa, Roland Siegwart, and Marco Hutter. Control of a quadrotor with reinforcement learning. *IEEE Robotics and Automation Letters*, 2(4):2096–2103, 2017.

[22] Riccardo Polvara, Massimiliano Patacchiola, Sanjay Sharma, Jian Wan, Andrew Manning, Robert Sutton, and Angelo Cangelosi. Toward end-to-end control for uav autonomous landing via deep reinforcement learning. In *2018 International conference on unmanned aircraft systems (ICUAS)*, pages 115–123. IEEE, 2018.

[23] Jiawei Wang, Teng Wang, Zichen He, Wenzhe Cai, and Changyin Sun. Towards better generalization in quadrotor landing using deep reinforcement learning. *Applied Intelligence*, 53(6):6195–6213, 2023.

[24] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[25] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[26] Roberto Zanasi, C Guarino Lo Bianco, and Alberto Tonielli. Nonlinear filters for the generation of smooth trajectories. *Automatica*, 36(3):439–448, 2000.

[27] Ross E Allen and Marco Pavone. A real-time framework for kinodynamic planning in dynamic environments with application to quadrotor obstacle avoidance. *Robotics and Autonomous Systems*, 115:174–193, 2019.

[28] Carlos Ruiz, Xinlei Chen, and Pei Zhang. Hybrid and adaptive drone identification through motion actuation and vision feature matching. In *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 327–328, 2017.

[29] Pedro Sanchez-Cuevas, Guillermo Heredia, and Anibal Ollero. Multirotor aerodynamic effects in aerial manipulation. *Aerial Robotic Manipulation: Research, Development and Applications*, pages 67–82, 2019.

[30] Tommaso Mannucci, Erik-Jan van Kampen, Cornelis De Visser, and Qiping Chu. Safe exploration algorithms for reinforcement learning controllers. *IEEE transactions on neural networks and learning systems*, 29(4):1069–1081, 2017.

[31] William Koch, Renato Mancuso, Richard West, and Azer Bestavros. Reinforcement learning for uav attitude control. *ACM Transactions on Cyber-Physical Systems*, 3(2):1–21, 2019.

[32] Wojciech Giernacki, Mateusz Skwierczyński, Wojciech Witwicki, Paweł Wroński, and Piotr Kozierski. Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering. In *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 37–42. IEEE, 2017.

[33] Xinlei Chen, Aveek Purohit, Carlos Ruiz Dominguez, Stefano Carpin, and Pei Zhang. Drunkwalk: Collaborative and adaptive planning for navigation of microaerial sensor swarms. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 295–308, 2015.

[34] Md Shah Alam and Jared Oluoch. A survey of safe landing zone detection techniques for autonomous unmanned aerial vehicles (uavs). *Expert Systems with Applications*, 179:115091, 2021.

7 APPENDIX

Environment Description: Consider a bounded free region Ω, with height $H \in \mathbb{R}^{+}$, length $\mathrm{L} \in \mathbb{R}$, and width $\mathrm{W} \in \mathbb{R}$, which is the quadrotor's flight space. The quadrotor can take off, land, and fly in this region above the floor. $\Sigma = \{h, l, w \mid h = 0, l \in L, w \in W\}$ represents a rigid floor surface where quadrotors can stay, but violent impacts or incline landing to the floor may cause a hazardous collision.

States of Quadrotors: As a flight agent, the quadrotor has states containing global position $\mathbf{p} = [p_x, p_y, p_z]^\top \in \mathbb{R}^3$, velocity $\mathbf{v} \in \mathbb{R}^3$, body angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ and attitude rotation matrix R$\in SO(3)$. Then we can describe the following dynamics:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v}, \dot{\mathbf{v}} = \mathbf{a}, \\ m\mathbf{a} &= m\mathbf{g} + R\mathbf{f_u} + \mathbf{f_w}, \\ R &= RM(\boldsymbol{\omega}), \\ \mathbf{J}\dot{\boldsymbol{\omega}} &= \mathbf{J}\boldsymbol{\omega} \times \boldsymbol{\omega} + \boldsymbol{\tau_u} + \boldsymbol{\tau_w}, \end{aligned} \tag{5}$$

where $\mathbf{a}$ is the acceleration of the drone's movement, m and $\mathbf{g} = [0,\ 0,\ -g]$ are mass and gravity acceleration vector, respectively. $M(\cdot)$ indicates skew-symmetric mapping. $\mathbf{f_u} = [0, 0, T]^\top$ and $\mathbf{f_w}$ is the reflection force from four rotors thrust and unknown outside wind. $\mathbf{u} = [n_1^2, n_2^2, n_3^2, n_4^2]^T$ are actuation signal, while n_1, n_2, n_3, n_4 are motor rotation speeds. Accordingly, $\boldsymbol{\tau_u}$ and $\boldsymbol{\tau_w}$ are the torques from four rotors and outside wind. The thrust $T = H_0\mathbf{u}$ determines the motion of quadrotor, with

$$H_0 = \begin{bmatrix} k_T & k_T & k_T & k_T \\ 0 & k_T l_r & 0 & -k_T l_r \\ -k_T l_r & 0 & k_T l_r & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix}, \tag{6}$$

where k_T is thrust coefficient and l_r is the length of rotor arm, and c_Q represents torque coefficient. The critical factors in the UAV stable landing problem are $\mathbf{f_w}$ and $\boldsymbol{\tau_a}$, the unknown wind disturbance from GE. Disturbance force $\mathbf{f_w} = [f_{w,x}, f_{w,y}, f_{w,z}]^\top$ and disturbance torques $\boldsymbol{\tau_a} = [\tau_{a,x}, \tau_{a,y}, \tau_{a,z}]^\top$ comes from complicated aerodynamics interactions between quadrotors and environment, especially to the ground. In general, the larger the rotor output power and the closer the drone is to the ground, the more effect the disturbance will have[34].

To be detailed, control noise existed in many processes, such as actuation noise $\boldsymbol{\sigma_a}$ in the process from control signal to rotor actual thrust output and acceleration noise $\boldsymbol{\sigma_{\mathbf{u}}}$ caused by measurement and computing precision, two main components, with

$$\begin{aligned} \mathbf{u_m} &= \mathbf{u} + \boldsymbol{\sigma_{\mathbf{u}}}, \\ \mathbf{a_m} &= \mathbf{a} + \boldsymbol{\sigma_a}, \end{aligned} \tag{7}$$

where $\mathbf{u_m}$ is the measured actuation signal and $\mathbf{a_m}$ is the measured acceleration. In reality, these noises may cause unpredictable effects on the landing task.