# From Flies to Robots: Inverted Landing in Small Quadcopters with Dynamic Perching

Bryan Habas[1], Bo Cheng[1], *IEEE, Member*

***Abstract*—Inverted landing is a routine behavior among a number of animal fliers. However, mastering this feat poses a considerable challenge for robotic fliers, especially to perform dynamic perching with rapid body rotations (or flips) and landing against gravity. Inverted landing in flies have suggested that optical flow senses are closely linked to the precise triggering and control of body flips that lead to a variety of successful landing behaviors. Building upon this knowledge, we aimed to replicate the flies' landing behaviors in small quadcopters by developing a control policy general to arbitrary ceiling-approach conditions. First, we employed reinforcement learning in simulation to optimize discrete sensory-motor pairs across a broad spectrum of ceiling-approach velocities and directions. Next, we converted the sensory-motor pairs to a two-stage control policy in a continuous optical flow space augmented by ceiling distance measurement. The control policy consists of a first-stage Flip-Trigger Policy, which employs a one-class support vector machine, and a second-stage Flip-Action Policy, implemented as a feed-forward neural network. To transfer the inverted-landing policy to physical systems, we utilized domain randomization and system identification techniques for a zero-shot sim-to-real transfer with emulated optical flow using external motion tracking. As a result, we successfully achieved a range of robust inverted-landing behaviors in small quadcopters, emulating those observed in flies.**

***Index Terms*—Aerial Systems: Mechanics and Control, Biologically-Inspired Robots, Learning and Adaptive Systems, Aerial Systems: Applications**

## SUPPLEMENTARY MATERIAL

**Video:** https://youtu.be/MT0rrtnQ0ME

## I. Introduction

PERCHING is a feat routinely performed by animal fliers with robustness and accuracy, as seen in birds [1]–[3], bees [4]–[9], flies [10]–[12], and bats [13], [14]. With the ability to perch on surfaces that are arbitrarily oriented or moving unpredictably, animal fliers can land in unstructured environments to surveil territory, hitchhike on larger animals for travel to new locations, pollinate plants, or rest.

This research was made possible by the National Science Foundation grants CMMI-2230320 awarded to B.C., as well as the support of the Department of Defense (DoD) through the National Defense Science & Engineering Graduate (NDSEG) Fellowship Program awarded to B.H..

[1]Biological and Robotic Intelligent Fluid Locomotion Lab, Department of Mechanical Engineering, The Pennsylvania State University, University Park, PA 16802, USA. Corresponding to B.C. buc10@psu.edu

Manuscript submitted for review February 29, 2024

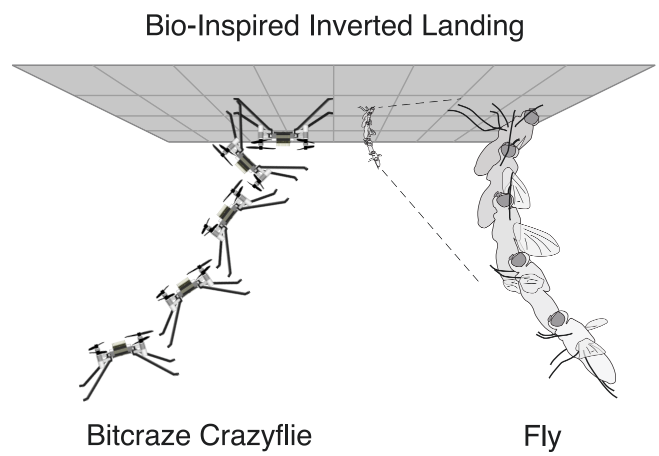


Fig. 1. Bio-inspired inverted landing. An illustration comparing an example of inverted landing in a small quadcopter and a blue-bottle fly [10]. Both an at-scale and a scaled-up version of the landing sequence of the blue-bottle fly are shown.

Achieving similar perching abilities in flying robots is crucial for enabling their fully autonomous operation in unstructured environments [15], with the potential for enabling or augmenting long-term inspections, surveillance, reconnaissance, and the rapid release and retrieval of robots for various missions [16]–[20]. A major drawback of these robot systems over recent years has been their limited battery life, which typically only sustain maximum flight times on the scale of tens of minutes [21]. For this reason, the ability to perch on targeted objects can greatly expand the operational lifetime of quadrotor robots. This capability could be particularly applicable in urban environments, where flat surfaces like walls and ceilings are abundant and many tasks such as camera surveillance, sensor reading, and acting as a radio relay in disaster zones do not require continuous flight [16], [17], [22].

### A. Dynamic Perching and Challenges

Despite the impressive aerial agility and load-carrying capacity exhibited by small aerial robots like quadrotors, they have yet to achieve the perching capabilities observed in animal fliers. Indeed, *dynamic perching*—characterized by rapid maneuvers and subject to velocity constraints—still poses one of the most significant challenges for aerial robots. This skill becomes vital when landing on rapidly moving or inverted (ceiling-like) surfaces, necessitating the flier to accelerate against gravity and rapidly alter its orientation. In such scenarios, fliers must solve a complex distance-velocity-attitude sensing and control problem under stringent time and motor constraints. This requires the small flier to control its 3D linear velocity [5], [23], accurately predict imminent impact [24], [25], and timely adjust its body orientation and landing gear/legs relative to the perching surface through rapid angular maneuvers. These maneuvers must be executed within milliseconds to ensure a touchdown process with proper body inversion and contact force [10].

Challenges associated with perching extend beyond the maneuver itself, involving the limited computational resources available for sensing, control, and planning in small fliers [25], [26]. Achieving accurate onboard sensory estimations within the stringent time constraints of dynamic perching is a formidable task (e.g., a delay of just a few milliseconds could result in a crash), especially in unstructured environments. Aerial robots are also commonly under-actuated, which makes it difficult to execute aggressive maneuvers demanding uncoupled linear and angular acceleration. Moreover, aggressive acrobatic flight necessitates high thrust and extreme angular accelerations that can push the motor limits of the robots, potentially leading to unmodeled behaviors. Therefore, a successful perching strategy must consider these limitations and rely on computationally efficient algorithms in order to replicate the perching behaviors of animal fliers. Importantly, physically-embodied intelligence [27], [28], seen in the landing gear or legs, can help to mitigate computational demands and bolster robustness.

Inverted landing (Figure 1) also presents a significantly greater challenge than landing on horizontal or vertical surfaces due to the fact that gravity works against the robot as it approaches the landing surface. Furthermore, to truly emulate the behavior of flies, the robots must be capable of approaching the surface from a variety of directions [10]. This introduces stringent and variable constraints on both the linear velocity—essential to prevent downward falls and ensure sufficient adhesion upon contact—and the body orientation, which must be accurately controlled to ensure proper contact and avoid collision.

### B. Inverted Landing in Flies

Flies, for example, have successfully solved the dynamic perching problem [29]. Their solution is a sequence of well-coordinated maneuvers, completed in less than 100 ms, that facilitate an inverted landing [10]. To initiate the process, they engage in an upward acceleration, which is followed by a rapid rotation of their body and an extension of their legs; the sequence concludes with the flies executing a leg-assisted body swing, using their fore-legs—now firmly attached to the ceiling—as a pivot point. Moreover, it has been shown that different ceiling-approach directions have led to diverse landing behaviors in terms of axes of rotation and angular rates of body maneuvers [10]. More importantly, the success of these landings can be explained by the timely triggering and proper control of body rotational maneuvers (or flips), which are both strongly correlated with the extent of optical flow that the flies' visual systems can extract.

The landing process in flies involves a combination of both computational and mechanical intelligence. Here, mechanical intelligence refers to the inherent physical capabilities of a system's design that allow it to react passively and flexibly to different situations without active control [30], [31]. During landing, this intelligence is exemplified as flies rapidly and precisely position their legs and tarsi prior to impact (Figure 1), utilizing these appendages to facilitate a robust touchdown process, involving a pendulum-like body swing, dissipation of impact forces and even recovey from failed landings. On the other hand, computational intelligence involves the system's ability to adapt, learn, and make decisions based on sensory feedback and computational processes [32]. In this context, flies, known for their efficient visuomotor mapping, possess direct connections between their visual systems and motor neurons. These connections facilitate rapid motor program selection and timing, enabling reactive aerobatic maneuvers [15]. In particular, the insect visual system is evolved for efficient extraction of optical flow [33], which encodes translational velocity, distance to a surface, and time-to-contact without the need for arduous feature identification and tracking. Visual cues like optical flow are widely used in nature and provide a computationally efficient and effective way for animals of various sizes to follow robust perception-based landing trajectories [1], [4].

The results from inverted landing in flies informed us that diverse landing behavior can emerge from different approach conditions, potentially as the results of a universal policy only based on two types of visual cues: Relative Retinal Expansion Velocity (RREV) and translational optical flow. In addition, it also underscores the importance of the triggering timing of the landing maneuver and suggests that the inverted landing process can be simplified as only the control of timing and the magnitude of the maneuvers [34], [35].

### C. Goal and Contributions of the Current Work

Building upon our preliminary work [25] on the initial data collection for inverted landing with limited experimental testing, this work provides novel contributions and substantial improvements in the following areas:

*(1) Two-Stage General Framework*: We introduce a comprehensive two-stage framework for developing a control policy, applicable to continuous spectrum of flight conditions. The previous work [25] focused solely on using the policy-gradient reinforcement learning for our initial data collection, and it only provides individual sensory-action pairs as a look up table for discrete flight conditions. This paper develops a novel two-stage Flip Trigger Policy and Flip Action Policy, general to arbitrary flight conditions, therefore considerably advances the capability for performing inverted landings in a variety of flight conditions.

*(2) Experimental Validation*: Experimental testing in the previous work [25] is limited to only a single vertical approach condition. This work significantly expands the experimental validation to a large array of flight conditions, which validates the scalability and effectiveness of our the two-stage framework.

*(3) System Identification for Sim to Real Transfer*: We have also improved system identification by precisely estimating the robot's rotational inertia using the bifilar pendulum method, developing a battery behavior compensation model through thrust tests, and incorporating motor speed dynamics into the simulation.

*(4) Simulation Fidelity and Landing Gear Evaluation*: Based on the improved system identification and modeling, we have enhanced the fidelity of our simulation. This improvement has facilitated a thorough reevaluation of different landing gear designs and the development of a framework that examines the key factors influencing landing success.

### D. Organization of the Paper

In Section II, we explore the literature on robotic landing, examining both computational and mechanical intelligence-based methods. The proposed two-stage general control policy for inverted landing, and its training methodology, are presented in detail in Section III. Section IV elaborates on our simulation setup and the experimental testing process. In Section V, we discuss the performance outcomes of our work, the results achieved, and the potential benefits of varying landing gear designs. The final section, Section VI, draws conclusions from our study and suggests avenues for future research. Additional details on system identifications and relevant explanations are provided in Appendix A.

## II. Existing Work on Robot Landing

### A. Computational Intelligence

In the literature of robot perching, a significant number of methods are based on the generation and optimization of target landing trajectories that the robots can follow. These methods offer advantages such as power efficiency, customizable boundary conditions, and the ability to execute aggressive maneuvers [22], [36]. However, they often rely heavily on the provision of real-time external positioning data to the robot or the use of an off-board computer for computing optimal trajectories [22], [37], [38], rendering them less than ideal for rapid dynamic perching and exploring the mechanical intelligence for emergent landing behaviors.

Many robotic perching studies have explored using onboard cameras to compute flight trajectories, such as through state estimation algorithms, to land on vertical surfaces [36], [39]–[41]. While these methods offer advantages over external positioning systems, they tend to experience tracking degradation during aggressive maneuvers and require feature tracking of a predefined landing target, making them less practical in unstructured environments [36]. In contrast, featureless visual-based landing strategies rely on calculating optical flow-based sensor values from monocular cameras, without the need for feature tracking, enabling them to be applied in a wider range of outdoor settings [40], [42]–[44]. Despite their simplicity and computational efficiency, there is still ongoing research into how to optimize these methods for robustness and adaptability to varying environmental conditions, especially in complex terrains.

### B. Mechanical Intelligence

Within the realm of robotic landing research, the design of the physical system plays a pivotal role in enabling robots to attach to a variety of landing surfaces and objects through their mechanical intelligence. This has resulted in a broad range of approaches, with many inspired by nature. These popular methods frequently involve the use of grasping mechanisms that allow the robot to hang or perch on branch-like objects [38], [43], [45]–[51]. Other approaches include the utilization of dry-adhesive materials that mimic gecko skin, enabling the robot to attach to any orientation of exceptionally smooth surfaces [22], [52], [53], as well as novel devices such as suction cups [54], [55], barbed hooks [56], and magnets [57]. The variety of attachment devices offers unique advantages and benefits, making them well-suited for different landing scenarios.

Although there is a growing variety of landing devices in robotics, research analyzing how the geometry of these devices affects landing robustness and landing strategy is still limited. Some authors have partially investigated these effects, such as by using active skids to land on inclined surfaces [58] or by modeling the material properties of landing gear [57], [59], [60], but comprehensive research in this area is still required. Our work contributes to this area by analyzing the effects of varying geometries on highly dynamic landings and impacts for inverted landing scenarios. While our analysis framework and methodology are specifically tailored to this case, it can be easily adapted to evaluate the effects of mechanical intelligence from different landing gear designs on surfaces of varying orientations, such as vertical, horizontal, or inclined.

Several remarkable successes have been recorded in the field of robotic perching on inclined [22], [26] or moving surfaces [61]–[64]. Such methods typically concentrate on the planning, optimization, and tracking of dynamically-feasible trajectories to bring a robot within some empirically constrained states before touchdown [26], [61], [65]. Despite their successes, these methods pose significant computational challenges when applied to rapid dynamic perching in small robots and are not ideally equipped to leverage mechanical intelligence during the touchdown process. Another limitation lies in their reliance on empirically-tuned constraints on terminal states that are specific to a particular type of landing, as well as on computationally-intensive state estimation of both the robot and the perching target, often facilitated by external motion tracking systems. Additionally, traditional optical-flow-based methods like tau-theory [1], which rely on zero-velocity contact, are inadequate for fostering diverse landing behaviors on inverted surfaces. With these successes in mind, our approach to solving the particularly challenging inverted landing problem enhances the understanding of dynamic landings, addressing aspects of success that are often overlooked in other works. Additionally, this work proposes a versatile framework that can be adapted for landing on surfaces of various orientations.

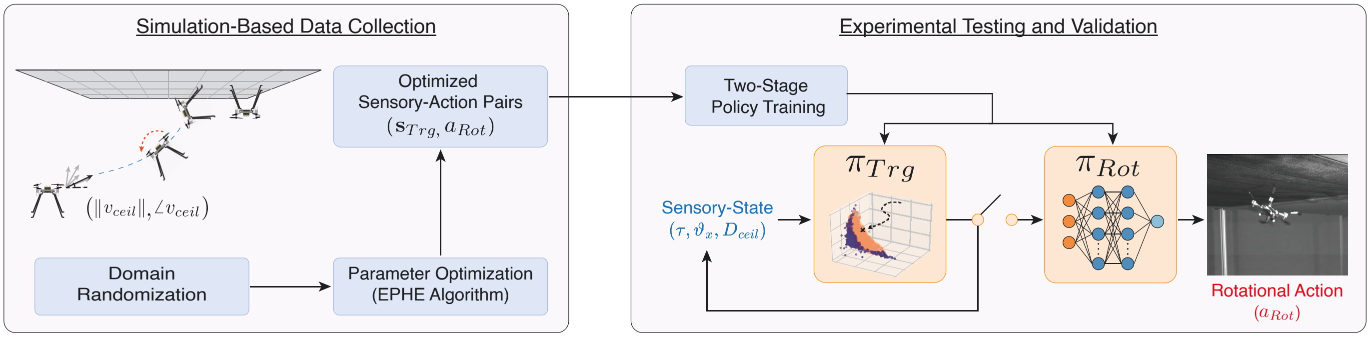


Fig. 2. Block diagram illustrating the data collection process through simulation, the training of our two-stage policy using the collected data, and the implementation of the policy in experimental tests.

## III. Method: Two-stage, General Control Policy for Inverted-Landing

### A. Overview

The training of our generalized control policy for inverted landings begins with simulating various ceiling-approach conditions; similar to those observed in flies (see Simulation and Experimental Setup). Using model-free reinforcement learning, we collected discrete sensory-motor action pairs, specifically triggering sensory states ($\mathbf{s}_{Trg}$) within the augmented optical-flow space ($\mathbb{OF}_a$), and the corresponding motor action for the rotational maneuver—a constant pitch torque ($a_{Rot}$). The pitch axis was selected to complement the robot's forward motion as a roll maneuver would yield similar results with sideways motion due to the robot's symmetry. Flipping around other axes, such as the diagonal, would also reduce the robot's maneuverability and negatively impact landing performance. This approach led to optimized motor control, maximizing the success rate of inverted landings under our ceiling-approach conditions [25].

To transform the discrete sensory-motor pairs into a generalized policy, we developed a two-stage control policy. In the first stage, we identified a cluster within $\mathbb{OF}_a$ where the discrete sensory states $\mathbf{s}_{Trg}$ linked with successful landings (exceeding a defined threshold of success rate). We used this cluster to construct a continuous boundary function around the region, distinguishing optimal triggering states from sub-optimal or failed ones, giving rise to a *Flip Trigger Policy* ($\pi_{Trg}$). This policy allowed the robot agent to detect when inside the optimal sensory region to initiate body angular maneuvers (or flips). To model the $\pi_{Trg}$ (or the boundary region of $\mathbf{s}_{Trg}$ in $\mathbb{OF}_a$), we used a One-Class Support Vector Machine (OC-SVM). In stage two, we utilized supervised learning to train a simple neural network that forms the *Flip Action Policy* $\pi_{Rot}$, a continuous sensory-motor map that generalizes discrete sensory-motor pairs. $\pi_{Rot}$ provides the body-moment command, controlling the rotational maneuver in a feedforward fashion. Thus, under this control policy, if the robot identifies its location within the $\mathbb{OF}_a$ region specified by the optimized $\pi_{Trg}$ policy, it generates a body rotational moment according to $\pi_{Rot}$ to execute an inverted landing. An overview of our two-stage policy can be seen in Figure 2.

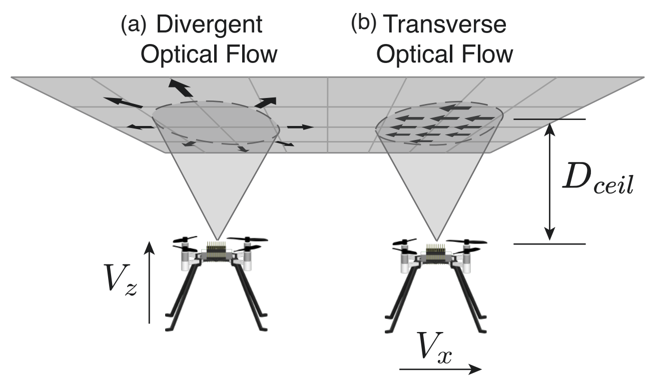


Fig. 3. (a) Diagram illustrating divergent optical flow: as the quadrotor nears the ceiling, observed points radiate outward. (b) Diagram showcasing transverse optical flow, where feature points move horizontally across the field of view during the robot's translation beneath the ceiling.

### B. Augmented-Optical Flow Space

The sensory states $\mathbf{s}_{Trg} \in \mathbb{OF}_a \subseteq \mathbb{R}^3$ used as inputs for our two-stage control policy, consist of two emulated visual sensory cues: a divergent optical flow-related term, time-to-contact ($\tau$), and a transverse optical flow term ($\vartheta_x$). These are augmented with a robot-to-surface distance metric ($D_{ceil}$) to prevent the sensory-space from being under-defined [25]. The variable $\tau$ encodes the time until the robot reaches a surface, assuming a constant velocity and heading. Studies show that in flies and pigeons, $\tau$ provides predictive information about when to initiate a rapid angular maneuver, converting a potential collision into a successful perch [1], [10]. As a visual cue, $\tau$ is the mathematical inverse of divergent optical flow, or the Relative Retinal Expansion Velocity ($RREV$), which represents how rapidly objects expand within the camera's Field of View (FoV) as the robot approaches an object (see Figure 3a). $\tau$ has been shown to directly encode the environment's affordance [66] (i.e., comprehending the robot's potential dynamic fit within the environment or perception-action [23]). Furthermore, various works support the utilization of this term in $\tau$-theory, where the time derivative remains constant within the animal system, and a single parameter value determines the entire landing trajectory [1].

While $\tau$ can be derived directly from onboard visual inputs without the need to measure the relative position or velocity between the robot and the landing target, in this study, it was emulated using external motion tracking data as per the following equation:

$$\tau = \frac{D_{ceil}}{V_z}. \quad (1)$$

Where $D_{ceil}$ represents the distance to the ceiling, and $V_z$ denotes the vertical velocity of the system.

Moreover, the transverse optical flow term, denoted as ($\vartheta_x$), encodes the relative distance and horizontal velocity of the robot with respect to the landing surface. This term is determined by both the horizontal velocity of the robot ($V_x$) and its distance relative to the landing surface ($D_{ceil}$), refer to Figure 3b [4], [35]. $\vartheta_x$ is anticipated to correlate directly with the robustness of the robot's landing and the resulting body-moment required for the flip action. The value of $\vartheta_x$ can be emulated using external motion tracking data through the following equation:

$$\vartheta_x = \frac{V_x}{D_{ceil}}. \quad (2)$$

Finally, the authors' previous research indicated that the reliance on purely the two aforementioned optical flow terms was insufficient to fully define the policy-space [25]. Specifically, the $\mathbf{s}_{Trg}$ regions corresponding to successful and failed landings couldn't be separated within the optical flow space $\mathbb{OF}(\tau, \vartheta_x) \subseteq \mathbb{R}^2$. To overcome this obstacle, it was necessary to augment the optical flow space with an additional parameter such as $D_{ceil}$ or the robot's airspeed. Although insects can sense their airspeed through their antennae, it is challenging for small robotic fliers to achieve the same. Hence, we chose $D_{ceil}$ as the augmentation parameter. This measurement can be potentially derived using a laser distance sensor or estimated by fusing the system's onboard accelerometer measurements with the time derivative of $\tau$ [67]. Consequently, the triggering sensory state $\mathbf{s}_{Trg} \in \mathbb{OF}_a\left(\tau, \vartheta_x, D_{ceil}\right) \subseteq \mathbb{R}^3$ forms the input tuple for our generalized control policy.

Note that there is mathematical equivalence between $V_x$, $V_z$, and $D_{ceil}$ and the augmented optical flow space variables $(\tau, \vartheta_x, D_{ceil})$ as outlined in Equations 1 and 2. Our preference for the latter arises from the practical advantages of estimating these variables directly from onboard sensors. Although our experiments use a Vicon motion capture system to emulate optical flow, direct estimation of time-to-contact and transverse optical flow is feasible using onboard cameras. Several methods enable efficient calculation of these variables using simple monocular cameras and featureless-based optical flow algorithms [67]–[69]. Additionally, optical flow space can be augmented by variables other than $D_{ceil}$, such as motor thrust commands, which does not require additional onboard sensors. In other words, augmented optical flow spaces can be designed in various ways to achieve comparable landing performance and to enhance the system's adaptability. This adaptability makes it well-suited for iterative development towards fully autonomous landings with onboard feedback systems.

### *C. Reinforcement Learning-Based Optimization of Sensory-Action Pairs for Variable Ceiling Approach Conditions*

Instead of directly obtaining a control policy applicable to any arbitrary ceiling-approach condition, we determined the optimal sensory-motor action pairs $(\mathbf{s}_{Trg}, a_{Rot})$ individually for each simulated ceiling-approach condition. For each learning trial, characterized by a constant velocity magnitude $\|\mathbf{v}_{ceil}\|$ and flight angle relative to the horizon $\angle\mathbf{v}_{ceil}$, we used policy gradient Reinforcement Learning (RL) to optimize the optimal trigger timing threshold, $\tau_{cr}$, and a corresponding body-rotation motor-action, $a_{Rot}$, that maximized the reward function related to successful landings (as detailed in the following section). This body-rotation action was applied by adjusting the fore/aft motor thrusts to induce an angular moment about the robot's pitch axis. To fully cover the space of desired velocity conditions, the learning process was repeated three times for each $\|\mathbf{v}_{ceil}\| \in [1.5 - 3.5]$ m/s and $\angle\mathbf{v}_{ceil} \in [30^\circ - 90^\circ]$, incremented by $0.1$ m/s and $3.75^\circ$, respectively. We recorded the complete set of optimized $(\mathbf{s}_{Trg}, a_{Rot})$ pairs for all approach conditions, as well as the corresponding landing success rates.

In this optimization process, our RL approach utilized the EM-based Policy Hyper Parameter Exploration (EPHE) algorithm [70]. We chose this algorithm for its fast convergence and adaptive learning rate, which reduces the need for extensive hyperparameter tuning. In our application, the EPHE algorithm determines an optimal set of parameters by generating a series of parameterized Gaussian distributions $\boldsymbol{O} = [O_{\tau_{cr}}, O_{Rot}]$ with $\boldsymbol{O} = \mathcal{N}(\boldsymbol{\mu}, I\boldsymbol{\sigma}^{\mathbf{2}})$, defined by the vectors $\boldsymbol{\mu} = [\mu_{\tau_{cr}}, \mu_{Rot}]$ and $\boldsymbol{\sigma} = [\sigma_{\tau_{cr}}, \sigma_{Rot}]$. These vectors are then optimized through interaction with the environment until the distributions approximate a deterministic value.

The detailed RL process is as follows: Each landing trial began with the quadrotor in an initial hovering state, after which it was programmed to follow a constant velocity trajectory with a specified speed $\|\mathbf{v}_{ceil}\|$ and flight angle $\angle\mathbf{v}_{ceil}$ that leads to collision with the ceiling surface. This velocity trajectory was maintained throughout the landing trial. Following this, the agent performed several rollouts where, for each rollout, a policy $\boldsymbol{\theta} = [\tau_{cr}, a_{Rot}]$ was sampled from the current distribution $\boldsymbol{O}$ and used to execute a landing maneuver. To do so, the quadrotor observed the current $\tau$ value until it dropped below the $\tau_{cr}$ threshold set by the sampled policy, at which point it executed the body flip motor-action, a constant pitch torque ($a_{Rot}$), dictated by the same policy parameters. The rollout terminated based on whether the landing was successful or a timeout threshold was reached. At this juncture, a reward for the rollout was calculated before the process was repeated for a new rollout and sampled policy. After N rollouts were performed, the best K rewards and policy parameters were used to update the original policy distribution $\boldsymbol{O}$; this constituted a complete episode. This process was repeated for 15 episodes or until there was convergence to a nearly deterministic set of optimized parameters which maximized the probability of a successful landing for a given set of velocity conditions. In each learning trial, convergence to a deterministic policy typically occurred within 100 rollouts.

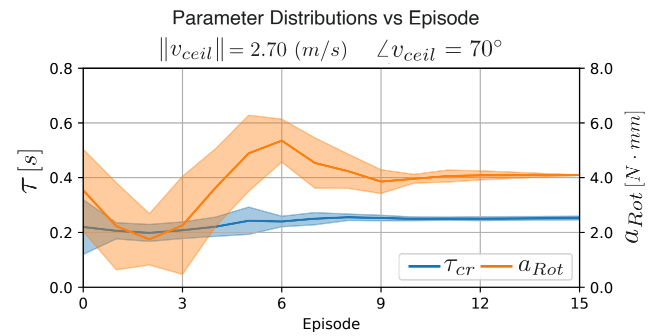


Fig. 4. The figure illustrates the convergence of $\tau_{cr}$ and $a_{Rot}$ values obtained through reinforcement learning in simulation. These values are used to establish an optimal sensory-motor pair. The shaded regions indicate a distribution range of $\mu_i \pm 2\sigma_i$ for samples taken in each episode.

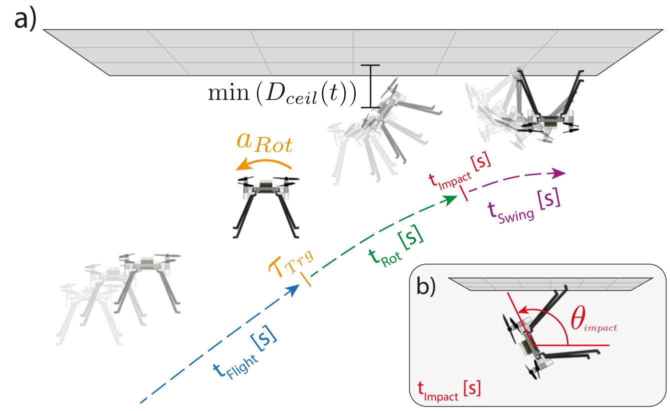


Fig. 5. a) Depiction of the landing process, highlighting the variables used in the reward function. b) Illustrative representation of the impact orientation variable $\theta_{impact}$

Figure 4 provides an example of such convergence, illustrating the convergence behavior for a landing trial with a velocity magnitude of 2.70 m/s and a flight angle of 70 degrees relative to the horizon.

### D. Design of Reward Function

In the area of reinforcement learning, constructing an effective reward function is crucial for successful learning, swift convergence, and the integrity of the agent's learned policy. Applying the principles of curriculum learning to our reward function design allowed for consistent and reliable policy convergence. Here, each desirable behavioral trait for successful inverted landing incrementally increases the total reward, thereby guiding the system towards mastering more complex behaviors.

These behavioral traits and their associated reward functions were characterized to: minimize the quadrotor's distance to the ceiling as depicted in (3); initiate the flip maneuver prior to surface contact as per (4); adjust the impact angle to ensure initial contact is made by the fore-legs as depicted in (5); and fine-tune the preceding conditions to guarantee the achievement of reliable inverted landing as demonstrated in (6) [71]. Illustrative understanding of the variables used in the provided reward functions are shown in Figure 5.

The constants $(c_0, c_1)$, present in equations (3) and (4), serve to normalize their corresponding reward functions and adjust the breadth of their clipped zones. By tuning these elements, the reward functions can direct the system to converge within a practical array of values, avoiding an overly specific fixation on a single value. This principle is notably applied in (4), which advocates for the flip maneuver to fall within the typical trigger interval found for this robotic system $\tau_{cr} \in [0.15, 0.25]$ [s], without overly dictating the timing of the maneuver [25]. Notably, $\tau_{cr}$ values larger than this range typically failed to achieve contact with the landing surface, while values below this range did not allow sufficient time for the robot to rotate into a suitable landing orientation.

A penalty factor of $r_{legs} \leftarrow r_{legs}/3$ modifies equation (6) in instances of propeller or body contact. This adjustment serves to deter any resulting policy that might induce structural harm to the quadrotor. Consequently, to compute the total reward for the entire episode, the individual rewards were proportionally weighted and summed accordingly, $r = 0.05 \cdot r_D + 0.1 \cdot r_\tau + 0.2 \cdot r_\theta + 0.65 \cdot r_{legs}$. By weighting the terms as such, the system initially focuses on learning to trigger the maneuver closer to the ceiling using the easier-to-achieve smaller weight rewards. Subsequently, the larger weighted reward components then guide the system to refine the impact conditions.

$$r_D = clip\left(\frac{1}{\min(D_{ceil}(t))}, 0, c_0\right) \cdot \frac{1}{c_0}, \tag{3}$$

$$r_\tau = clip\left(\frac{1}{|\tau_{trg} - 0.2|}, 0, c_1\right) \cdot \frac{1}{c_1}, \tag{4}$$

$$r_\theta = \begin{cases} \frac{|\theta_{impact}|}{120°} & 0° \le |\theta_{impact}| < 120° \\ 1.0 & 120° \le |\theta_{impact}| \le 180° \end{cases}, \tag{5}$$

$$r_{legs} = \begin{cases} 1.0 & N_{legs} = 3 \mid\mid 4 \\ 0.5 & N_{legs} = 1 \mid\mid 2 \\ 0 & N_{legs} = 0 \end{cases}. \tag{6}$$

### E. Develop Two-Stage, General Control Policy in Continuous Domain

Next we transformed the optimized $(\mathbf{s}_{Trg}, a_{Rot})$ pairs to a general control policy within a continuous domain, applicable to a wider range of approach scenarios. This generalized policy first identified the bounds of the triggering state region, i.e. the region of $\mathbf{s}_{Trg} \in \mathbb{OF}_a(\tau, \vartheta_x, D_{ceil})$ for which triggering the flip maneuver can lead to a successful landing. Subsequently, it generated the optimal rotation action based on the specific triggering state $\mathbf{s}_{Trg}$. This two-stage approach leads to our control policy consisting of the *Flip Trigger Policy* ($\pi_{Trg}$) and the *Flip Action Policy* ($\pi_{Rot}$). In crafting this policy, we employed a combination of unsupervised and supervised machine learning methods. Notably, the current approach exemplifies a grey-box approach, allows for direct identification of policy region for triggering, emergent policy behaviors, and landing success rate, therefore allowing us to examine the robustness

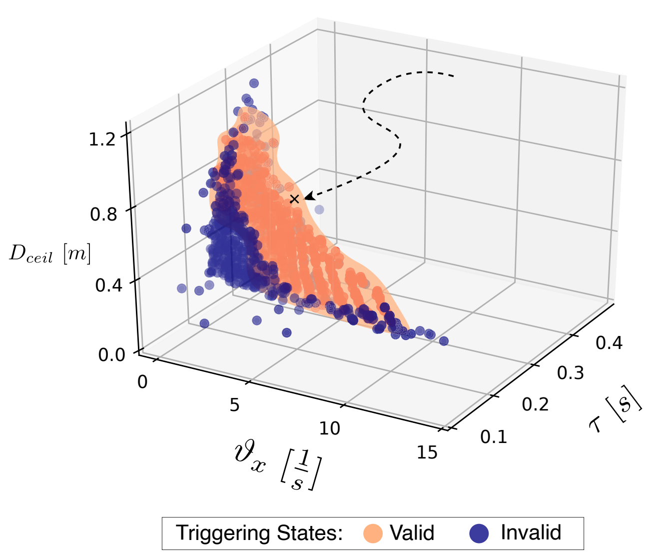


Fig. 6. The decision boundary formed by the One-Class SVM, implemented for the *Flip Trigger Policy*, is shown. This boundary distinguishes between valid and invalid triggering states for inverted landings, consistent with the policy $\pi_{Trg}(\mathbf{s}_{Trg})$. A representative trajectory is also depicted, indicating the exact timing of the rotational maneuver trigger upon intersecting with the boundary region.

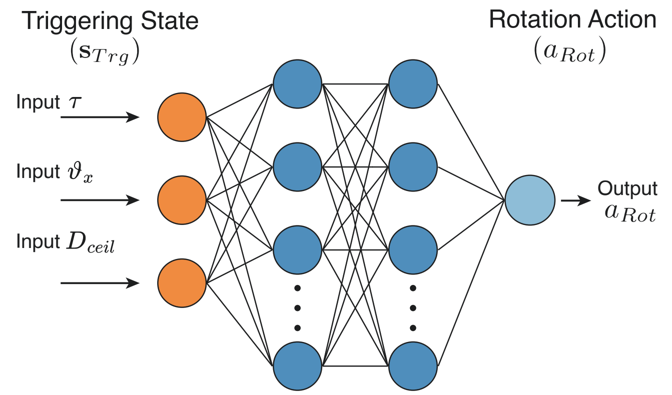


Fig. 7. Depiction of feed-forward neural network used to determine body-rotation action ($a_{Rot}$) from the set of triggering sensory states ($\mathbf{s}_{Trg}$).

of landing. This trait is also essential for future work using trajectory planning to fly the robots into this policy region for successful landing. This stands in contrast to the conventional black-box methods utilized in Deep RL algorithms, which directly generate both the triggering and rotation actions for inverted landing without insight to its inner workings [71].

*1) Flip Trigger Policy (One-Class SVM):* Since every $(\mathbf{s}_{Trg}, a_{Rot})$ pair corresponds to a specific success rate, we can identify this optimized cluster by setting a success rate threshold of $80\%$. Consequently, the *Flip Trigger Policy* can be articulated as a closed boundary function in the $\mathbb{OF}_a$ space, indicative of this specific cluster. We then define $\pi_{Trg}(\mathbf{s}_{Trg})$ to output a binary value when the current sensory-state lies within this region, triggering the flip maneuver accordingly.

Owing to the highly nonlinear nature of this region, we employed unsupervised machine learning, particularly outlier detection algorithms, to shape this boundary function. These algorithms are a subset of multi-class clustering techniques, focusing on separating the data into a single class and evaluating whether new data lies inside or outside the established group. Noteworthy algorithms suitable for this application include Isolation Forest [72], Local Outlier Factor [73], Robust Principal Component Analysis [74], and One-Class Support Vector Machines [75]–[77]. In this context, we opted for utilizing a One-Class Support Vector Machine (OC-SVM) due to its superior efficiency in decision-making calculations following the initial training.

OC-SVMs bear similarities to standard Support Vector Machines (SVMs) in that they project training data into a higher dimensional space using a non-linear kernel function, then use a hyper-plane to separate this high-dimensional data. However, OC-SVMs deviate by focusing on a single class and optimizing the hyper-plane to separate the training data from the origin in the hyper-space, thereby creating a non-linear decision function around the data cluster in the lower-dimensional training space. This leads to a binary output from the trained model: a positive number if a data point falls within the defined cluster, expressed by the equation:

$$\pi_{Trg}(\mathbf{s}_{Trg}) = sgn\left(\sum_{i=1}^{n} \alpha_i K(\mathbf{s}_{Trg}, \mathbf{s}_i)\right) + \rho \tag{7}$$

or a negative number if the data point is outside the learned cluster. In this equation, $\alpha_i$ denotes the coefficients for the support vectors generated during training, $K$ signifies the Radial Basis Function (RBF) kernel, $\mathbf{s}_{Trg}$ refers to the sensory-state vector under test, $\mathbf{s}_i$ indicates the learned support vectors, and $\rho$ is the bias value of the hyper-plane. The model also incorporates two hand-tuned parameters: the $\gamma$ value within the RBF kernel, affecting the decision boundary's smoothness, and a $\nu$ term that establishes the confidence threshold for data classification within the cluster. Visualization of the learned triggering policy region ($\pi_{Trg}(\mathbf{s}_{Trg}) > 0$), the collection of discrete sensory states $\mathbf{s}_{Trg}$ separated by a landing rate threshold of $80\%$, and a generalized flight trajectory can be seen in Figure 6.

*2) Flip Action Policy (Feed-Forward Neural Network):* Upon triggering, a control action denoted as $a_{Rot}$ is applied to the robot, generating the rotational-maneuver through a self-induced moment about its pitch-axis ($b_y$). This action, $a_{Rot}$, is computed as a function of $\mathbf{s}_{Trg}$ according to $a_{Rot} = \pi_{Rot}(\mathbf{s}_{Trg})$ [25]. This function serves to generalize the discrete $(\mathbf{s}_{Trg}, a_{Rot})$ pairs into the continuous sensory-space $\mathbb{OF}_a$. To determine $\pi_{Rot}$, we utilized supervised machine learning techniques, specifically training a feed-forward neural network, using $(\mathbf{s}_{Trg}, a_{Rot})$ pairs achieving over an 80% landing success rate as both inputs and outputs. The neural network (Figure 7) features two hidden layers with ten nodes each, optimized for computation speed and size, and utilized the Elu activation function. Throughout the training process, the input data was normalized to zero mean and unit variance, and a Mean-Squared Error loss function was employed to quantify the discrepancy between the training and predicted values.

## IV. Simulation and Experimental Setup

### A. Simulation Setup

Our simulation environment, designed using the Robot Operating System (ROS) and the Gazebo physics environment, was implemented on the Ubuntu 20.04 platform. This environment hosted a quadrotor model based on the Bitcraze Crazyflie 2.1, preserving the physical dimensions and flight parameters of the real-world quadrotor. It also included an implementation of Lee et al.'s Geometric Tracking Controller [78], manually tuned for our system to regulate flight, track trajectories, and update sensory state estimates at 100 Hz.

The simulation incorporated a ceiling plane as the landing surface, with custom plugins added to model the attachment joint as a ball joint and to record the quadrotor's impact angle upon landing. We also mirrored the mechanical properties of the quadrotor's legs: the hip joint was simulated as a revolute joint and the spring constant and the damping ratio were set at $K = 0.08 \frac{N \cdot mm}{rad}$ and $\zeta = 0.25$ respectively, to emulate the underdamped nature of the physical legs.

Furthermore, our simulation operated at a step size of 0.001 seconds to maintain precise control over the model's behavior. The precision and dependability of our simulation were confirmed in tandem with the system identification steps delineated in Appendix I.

### B. Simulation to Real-World Transfer

Executing RL in physical systems, especially those characterized by high-speed collisions, presents challenges due to substantial time and monetary investments. Furthermore, the considerable disparity between simulations and real-world outcomes complicates sim-to-real policy transfers. To bridge this gap, our approach consisted of robustly developing a policy in simulation and transferring it to the real-world environment via a zero-shot method. Essential to this was the accurate modeling of the quadcopter in the simulation to enhance the congruence between the simulated and physical models. This was achieved through precise measurement of the quadcopter's rotational inertia, modeling the motor-speed dynamics as a first-order system determined via system identification, and incorporating a battery compensation algorithm in the physical to ensure consistent thrust values throughout battery discharge.

In addition, to bolster policy transfer and resilience, we employed domain randomization during data collection [79]. This involved modulating parameters during learning to enhance the policy's adaptability to environment variations, effectively harnessing diverse simulation data for improved physical performance. Specifically, we randomized inertial parameters, allowing the agent exposure to a spectrum of model variations to avert overfitting. During reinforcement learning data acquisition, both system mass ($m$) and body inertia about the flip-axis ($I_{yy}$) were varied at the onset of each rollout, drawing from Gaussian distributions centered on their base values. With standard deviations set at $\sigma_m = 0.5$ [g] and $\sigma_{Iyy} = 1.5 \cdot 10^{-6}$ [kg m$^2$], this strategy enhanced simulation diversity and fostered the creation of more robust policies.

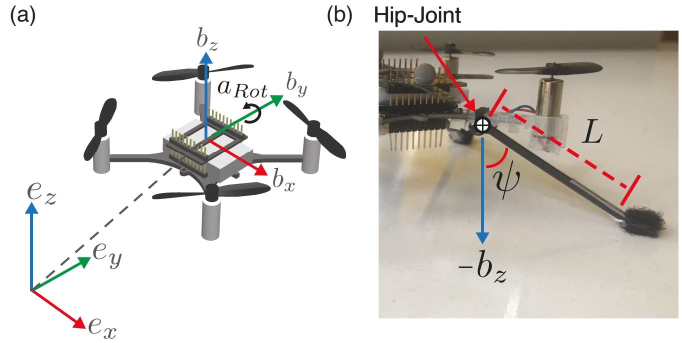


Fig. 8. (a) Illustration of the quadrotor's coordinate system relative to the global frame. The body-rotation action ($a_{Rot}$), is realized as a moment about the $b_y$ axis, generated by the front motors. (b) Diagram depicting the parameters for the leg angle ($\psi$) and leg length ($L$), along with the location of the hinge joint.

### C. Landing Gear Designs

Successful dynamic perching largely depends on achieving an appropriate touchdown that brings the robot to a stable landing posture relative to the landing surface. For small aerial robots or animals, this touchdown process can be highly dynamic and involve various forms of soft to hard collisions [27]. These landings generally require the robot or animal to have at least one foot firmly planted on the substrate, which is then followed by a rapid body swing or oscillation, culminating in all feet perching on the substrate [10], [25], [80]. Unlike the process of tracking a pre-planned or optimized trajectory, this dynamic touchdown process emerges rapidly from the immediate physical states and properties at the robot-surface interface [81]. Therefore, the success of a touchdown process, and consequently the landing, can be significantly influenced by the configurations of the landing gear.

TABLE I
Leg Design Configurations

| Leg Design | Angle $\psi$ (deg) | Length $L$ (mm) |
|---|---|---|
| Narrow-Short | 5° | 50 |
| Narrow-Long | 5° | 75 |
| Semi Narrow-Short | 30° | 50 |
| Semi Narrow-Long | 30° | 75 |
| Wide-Short | 60° | 50 |
| Wide-Long | 60° | 75 |

In this work, we explored the influence of various landing gear configurations on inverted landing performance through simulation. We then verified the most effective design via experimental trials where our designs incorporated adhesive feet, capable of fully adhering to the ceiling in a ball-joint manner under minimal contact, and a hip joint with behavior reminiscent of a torsional spring and damper system (Figure 8b). We tested this model experimentally using a VELCRO™ pad to establish a connection between the landing gear feet and the ceiling and determined that there was clear alignment with our simulation. The leg designs were parameterized by their length ($L$) and the angle they formed with the body's $-b_z$ axis ($\psi$) (Figure 8b). Our design configurations are detailed in Table I.

### D. Inverted Landing Sequence

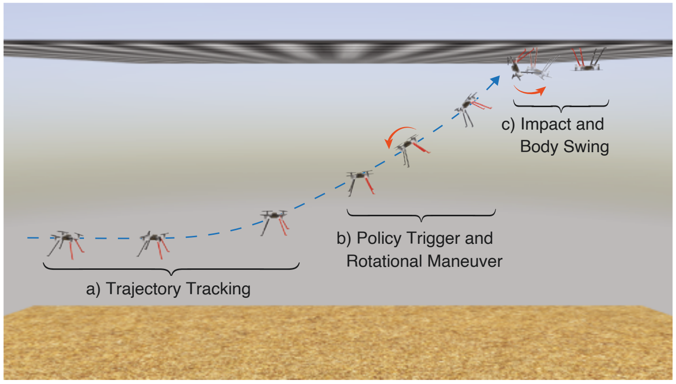


Fig. 9. Visualization of our inverted landing sequence, where the quadrotor follows an initial approach trajectory, initiates and executes the landing maneuver, and then impacts and swings into a secure landing position.

The landing sequence in our inverted landing framework consists of three stages: *1) Trajectory Tracking*: Starting from a hover, the quadcopter follows a pre-defined trajectory, tailored to maintain a specific flight speed and angle according to the desired ceiling-approach conditions. *2) Policy Trigger and Rotational Maneuver*: While progressing along this trajectory, the quadrotor continuously monitors its state in $\mathbb{OF}_a$ via emulated sensor data from an external positioning system. Upon meeting the triggering condition defined by $\pi_{Trg}(\mathbf{s}_{Trg})$, it executes a rotational maneuver action as dictated by $\pi_{Rot}(\mathbf{s}_{Trg})$. *3) Impact and Body Swing*: The robot's fore-legs then make first contact and adhere to the landing surface, leading to a pendulum-like body swing. Landing success is then evaluated based on the robots stabilized landing condition, if there was body/propeller contact, and the number of legs secured to the surface. Depiction of these stages are illustrated in Figure 9.

### E. Physical Experiment Setup

In our experiments for physical inverted landing, we utilized the Crazyflie 2.1 nano-sized quadrotor, augmented with four upgraded 19,000 KV brushed DC motors. This compact aerial platform was chosen for its superior maneuverability, supported by an open-source framework, and its enhanced collision resistance due to its small size and the simplicity of part replacement. The quadrotor was equipped with 3D printed legs, designed according to the Semi Narrow-Long configuration (Table I). For communication, we employed the Crazyswarm ROS package [82], establishing a connection between various ROS nodes and the Crazyflie Real Time Protocol embedded in the system's firmware.

As well, we used a Vicon motion capture system to provide real-time position and orientation data, streamed to the Crazyflie at a rate of 100 Hz. This data facilitated the creation of flight trajectories and the simulation of optical flow sensor data, in anticipation of incorporating on-board sensors in future studies.

The experiments were conducted using a portable test cage with dimensions of 1.5m width $\times$ 2.5 m length $\times$ 2.1m height. Velcro pads were fitted onto the ceiling surface to enable the robot's attachment. To capture the flight trajectory in the space near the triggering state, we positioned four Vicon cameras at the same height as the ceiling landing surface. An additional twelve cameras were positioned above to secure accurate tracking and positioning data throughout flight trajectory prior to triggering (Figure 10).

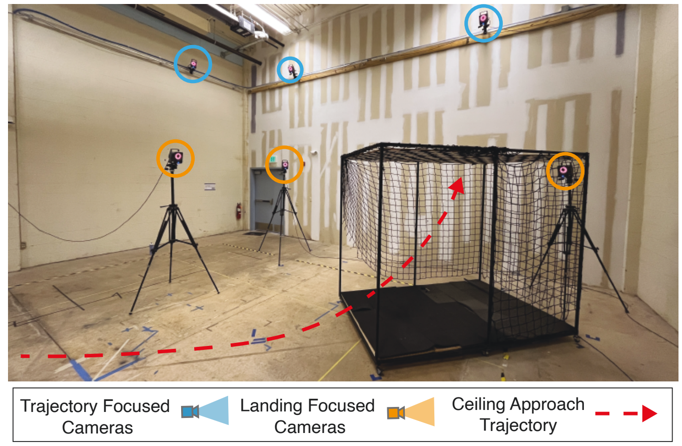


Fig. 10. Experimental setup showing an example flight trajectory, the positions of motion capture cameras guiding the flight, and cameras ensuring accurate estimation of triggering state values near the landing surface.

During experimental testing, both $\pi_{Trg}$ and $\pi_{Rot}$, previously trained in the simulation, were uploaded onto the Crazyflie. The quadrotor would begin its flight trajectory outside the test area, accelerating to achieve the predetermined flight speed $\|\mathbf{v}_{ceil}\|$ and flight angle $\angle\mathbf{v}_{ceil}$. This would set it on a collision course with the test surface and prompt it to execute the two-stage landing control policy according to $\pi_{Trg}$ and $\pi_{Rot}$. We designed these flight trajectories to mirror those used in the simulation, allowing us to test various ceiling-approach conditions. The behaviors of each inverted landing trial were recorded, with the process being repeated across a range of feasible flight velocities and angles within our testing environment.

## V. Validation of the Two-Stage Inverted Landing Policy in Simulated Environments

### A. Training Generalized Inverted Landing Control Policy via Optimized Sensory-Action Pairs

Utilizing the EPHE algorithm, we focused on collecting optimized sensory-action pairs $(\mathbf{s}_{Trg}, a_{Rot})$ for inverted landing in a simulated environment. During this data collection stage, the magnitude of the ceiling-approach velocity ($\|\mathbf{v}_{ceil}\|$) was incrementally varied from 1.5 to 3.5 m/s, in steps of 0.1 m/s, and the flight angle relative to the horizon ($\angle\mathbf{v}_{ceil}$) was adjusted from 30° to 90° in increments of 3.75°. To enhance the robustness of our following sim-to-real transfer, we also introduced domain randomization by varying the robot's inertial parameters at the beginning of each landing attempt.

The results of this data collection stage demonstrated a range of outcomes in inverted landing success, which we further categorized based on specific characteristics observed

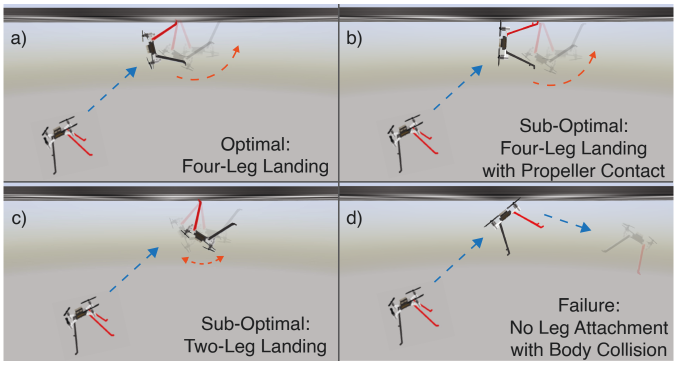


Fig. 11. Classifications of Inverted Landing Performance. (a) Optimal: Successful four-leg landing, no contact with body or propellers. (b) Sub-Optimal: Four-leg landing, but with body or propellers making contact. (c) Sub-Optimal: Incomplete two-leg landing, may include body contact. (d) Failure: Body collision, no leg attachment achieved.

during the landings. Here, *Optimal* inverted landings were identified by a distinct sequence: initially, the robot's fore-legs contacted the ceiling, followed by a successful swing that led to the hind-legs touching down, resulting in all four legs adhering to the surface without the body or propellers making contact (see Fig. 11a).

A category of *Sub-Optimal* landings were also noted. These landings were often similar to optimal ones, but involved initial contact with either the body or the propellers before stabilizing into a four-leg landing position (Fig. 11b). Another sub-optimal scenario involved the robot failing to complete the body swing maneuver, resulting in the hind-legs not contacting the ceiling and only the two fore-legs adhering to the ceiling surface (Fig. 11c). This subset included cases with and without propeller or body contact. Finally, the most unsuccessful attempts occurred when the robot collided with the ceiling without any contact between the legs and the landing surface (Fig. 11d). For a more comprehensive understanding, detailed visualizations of these landing classifications are available in the accompanying video.

Focusing on achieving purely optimal inverted landings (four-leg adhesion with no body/propeller contact), we analyzed the optimal-class landing success rate across various ceiling-approach conditions. All subsequent data will be presented based on this criteria. The collected data for the Semi-Narrow Long configuration is presented in Velocity-space ($V_x$, $V_z$, $D_{ceil}$) as shown in Fig. 12a. In this data, we observed that a higher velocity magnitude ($\|\mathbf{v}_{ceil}\|$) generally led to increased success rates and encouraged the triggering of the rotational maneuver further away from the ceiling. Additionally, landings with more angled approaches, indicated by a higher $V_x$, yielded higher success rates compared to more vertical approaches, which are characterized by a higher $V_z$. For a clearer visualization, the same data is smoothed and presented in Fig. 14a, which displays the landing success rates using polar coordinates ($\|\mathbf{v}_{ceil}\|$, $\angle\mathbf{v}_{ceil}$).

Additionally, the raw dataset of collected $\mathbf{s}_{Trg}$ values is also depicted in the augmented optical flow space $\mathbb{OF}_a$, alongside their corresponding landing success rates (Fig. 12b). In this $\mathbb{OF}_a$ space, a clear distinction between high and low

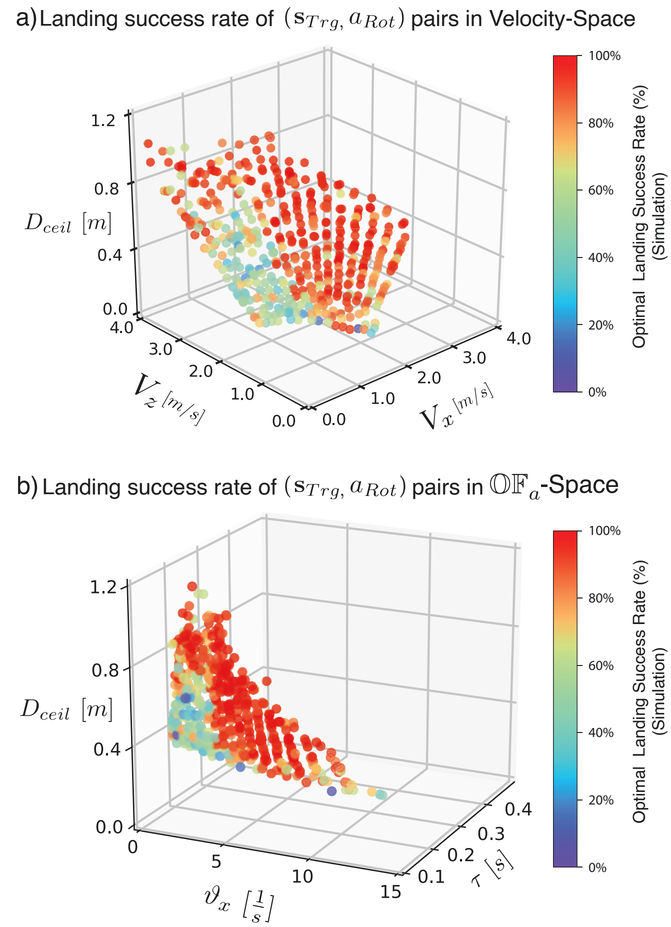


Fig. 12. a) Visualization of sensory-action pairs in Velocity-space, highlighting the correlation between higher horizontal velocities and increased inverted landing success in the simulation. b) Visualization of sensory-action pairs in $\mathbb{OF}_a$-space, demonstrating a distinct separation between states leading to high and low landing success.

landing success rates is evident, allowing for the definition of an enclosed boundary function through setting a success rate threshold (the basis for $\pi_{Trg}$). As well, the relationship between $\mathbf{s}_{Trg}$ and $a_{Rot}$ is illustrated in a side-view via Figure 13, where the data in $\mathbb{OF}_a$ is color-coded based on the magnitude of $a_{Rot}$ (the relation that is approximated by $\pi_{Rot}$). This representation highlights a correlation where triggering at lower time-to-contact values ($\tau$) is generally associated with higher magnitudes of $a_{Rot}$.

In our study, collecting the full dataset for a single leg configuration involved approximately 115,000 simulation-based landing attempts, yielding around 1,000 optimized ($\mathbf{s}_{Trg}, a_{Rot}$) pairs. We established a success criterion for purely optimal landings at an 80% success rate threshold, leading to the identification of about 300 pairs. These pairs, indicative of optimal four-leg landings without body or propeller contact, formed a distinct cluster in the $\mathbb{OF}_a$ space and were crucial for training our generalized two-stage policy. The first stage, the flip triggering policy ($\pi_{Trg}$), generates a boundary function around this subset of training pairs (illustrated by the shaded area in Fig. 6), dictating the precise moment to initiate the rotational maneuver. The second stage, the rotational maneuver policy ($\pi_{Rot}$), then calculates the necessary body moment based on the triggering state.

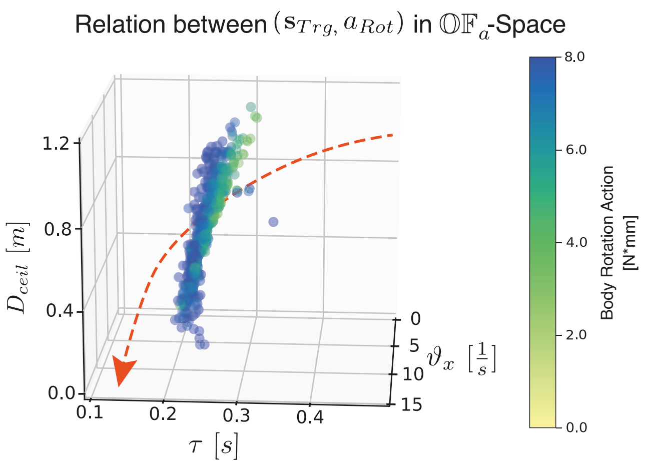


Fig. 13. Illustration of the full collection of state-action pairs, where an increase in the rotational moment is observed as the triggering time-to-contact ($\tau_{Trg}$) decreases.

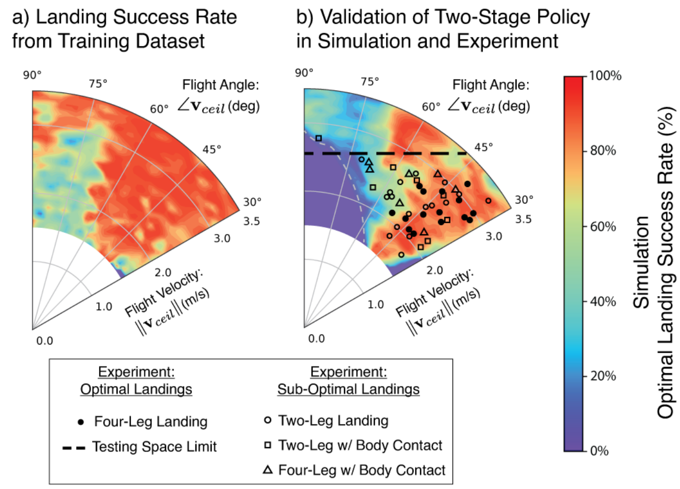


Fig. 14. (a) Polar plot showcasing the overall inverted landing capabilities derived from the simulated training dataset specific to the Narrow-Long leg configuration. (b) A comparison of simulation-based results with experimental outcomes for our formulated two-stage policy, showing the vertical velocity limitation due to the experimental setup, indicated by a dashed line.

### *B. Validation of the Generalized Inverted Landing Policy in Simulation*

Upon completion of training our generalized two-stage inverted landing control policy, we conducted validation tests in both simulated and experimental environments. In the simulation, the policy was tested under flight conditions similar to those used for collecting the initial sensory-action pairs. The results of which indicated that at lower ceiling-approach angles (between 30° and 65°), our generalized control policy (Fig. 14b) performed comparably to the optimal discrete sensory-action pairs (Fig. 14a). However, in near-vertical approach conditions (from 65° to 90°), we observed moderate performance degradation. This result was expected due to the prevalence of successful sensory-action pairs at lower angles compared to the more vertical approaches, as shown in Figure 14a. Consequently, this led to a training bias in the two-stage control policy towards lower angle approaches, further illustrated in background plot of Fig. 14b.

### *C. Impact of Leg Design and Flight Conditions on Quadrotor Inverted Landing Efficacy*

Additionally, we simulated various leg configurations and investigated their impact on landing success across flight conditions; the details of the configurations tested are presented in Table I. Furthermore, for data collection we applied the same parameter optimization approach previously described, focusing on a range of flight velocity magnitudes ($\|\mathbf{v}_{ceil}\|$) and directions ($\angle \mathbf{v}_{ceil}$) consistent with our initial data collection methodology shown in Section V.A.

Our study revealed significant variations in landing success based on the leg configuration. Specifically, the Semi-Narrow configurations demonstrated higher success at lower flight angles (30 deg to 65 deg), as illustrated in Figure 15b,e. However, their effectiveness declined markedly in near-vertical flight conditions (above 65 deg). In contrast, the Narrow configurations excelled in more vertical flights (above 50 deg) but were less successful at angles below 50 deg, as shown in Figure 15a,d. The Wide configurations generally underperformed in comparison, except at very low or extremely vertical angles (Figure 15c,f). Across these configurations, we also observed a trend where higher flight velocity magnitudes correlated with increased inverted landing success. Additionally, shorter leg lengths generally exhibited less success across all flight conditions compared to their longer counterparts, as depicted in Figure 15a-c.

The diversity in quadrotor inverted landing behaviors can primarily be attributed to the balance of four key factors: 1) *Swing Distance about Fore-leg Contact*; 2) *Gravitational Contribution to the Swing*; 3) *Sufficiency of Momentum Transfer*; 4) *Impact Window Size*. These factors are significantly influenced by the body angle at impact, the quadrotor's leg geometry, and the flight conditions when the landing policy is triggered. This complex interaction leads to varying inverted landing capabilities across different configurations.

Viewing through this framework lens, it can be seen that minimizing the *Swing Distance about Fore-leg Contact* significantly enhances inverted landing success. Here the impact angle and the quadrotor's leg geometry directly determine the necessary swing distance for the hind legs to make contact with the surface. Figure 16a illustrates this, where, for a given body angle at impact, the Narrow-Long geometry necessitates a smaller swing than the Wide-Long configuration to achieve the desired landing state, thereby reducing the travel distance and energy required.

Furthermore, maximizing the *Gravitational Contribution to the Swing* significantly enhances landing success. This effect is influenced by the quadrotor's impact angle and leg geometry, which can result in the Center of Mass (CoM) being lower at the final state than at the time of impact. Figure 16a illustrates this for the Narrow-Long leg configuration, where gravity aids

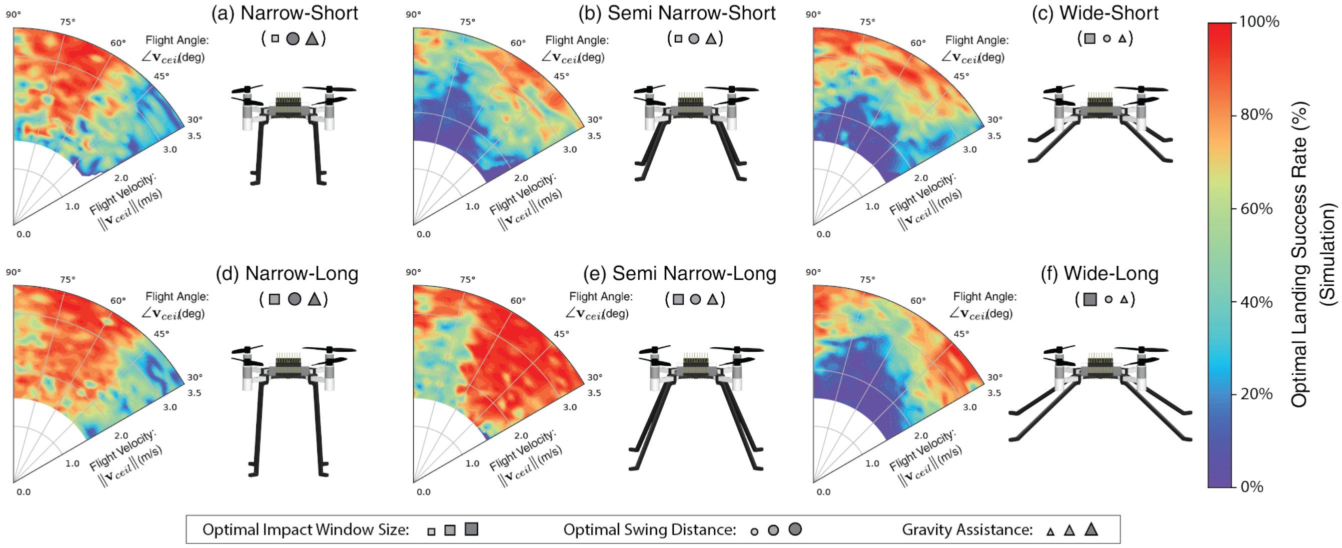


Fig. 15. Polar plots indicating the landing success rate found in simulation for various leg design configurations. The radial axis corresponds to the flight velocity magnitude ($\|\mathbf{v}_{ceil}\|$), while the angular axis represents the flight angle ($\angle\mathbf{v}_{ceil}$). Data was gathered for parameters where $\|\mathbf{v}_{ceil}\|$ ranges from 1.5 to 3.5 m/s and $\angle\mathbf{v}_{ceil}$ spans from 30° to 90°.

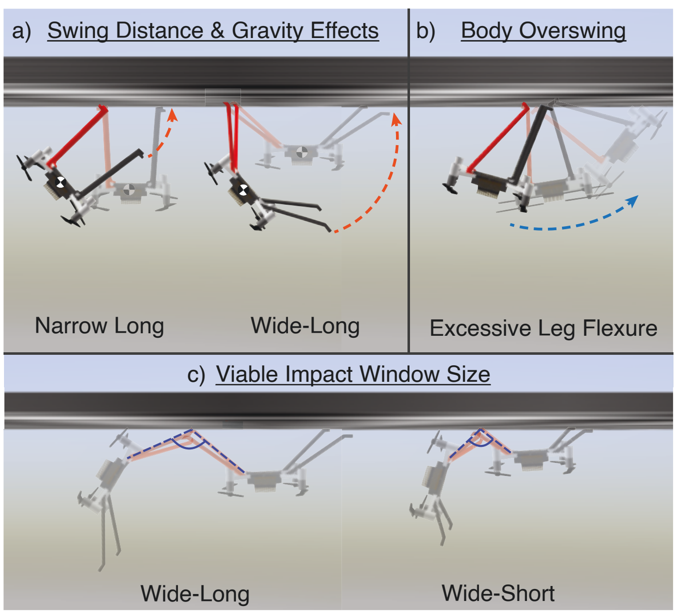


Fig. 16. a) For the same body orientation at impact, quadrotors with narrower leg designs require a shorter swing distance to make contact with the landing surface. b) Leg joints with excessive flexure can lead to overswing and body contact with the landing surface. c) Shorter-legged designs typically lead to a narrower range of viable impact conditions compared to their longer leg counterparts.

in achieving the desired state with minimal effort. Conversely, configurations like the Wide-Long, also shown in Figure 16a, may require the CoM to be higher in the final state than at impact, necessitating additional energy for a successful body-swing.

Balancing the *Sufficiency of Momentum Transfer* is crucial for inverted landing success, even when other factors are unfavorable. This aspect depends on the body angle at impact, leg geometry, and the quadrotor's flight conditions. Here, properly aligning the impact angle to convert translational momentum into rotational momentum allows the robot to cover the necessary swing distance and overcome any adverse gravitational effects. Failure to adequately coordinate this can lead to a two-leg landing due to insufficient rotational momentum. Additionally, some combinations of flight conditions and leg geometry may not provide enough translational momentum for a successful four-leg inverted landing. Our supplemental video further illustrates the importance of momentum management, swing distance, and gravitational effects for achieving inverted landing success

Additionally, the leg configuration's geometry directly affects the *Impact Window Size*, determining the range of viable impact angles that ensure only the legs, and not the body, contact the ceiling, as shown in Figure 16c. Shorter-legged designs (compared to their longer counterparts) typically offer a narrower range of viable impact windows, increasing the need for precision in impact conditions. In systems with significant noise, like ours, this increased sensitivity can significantly affect the robustness and success rate of inverted landings.

The direct effects of these concepts are evident in the results presented and in Figure 15. For example, the Semi-Narrow configurations (Figure 15b,e) are notably effective under lower flight angles, and capable of striking a balance between manageable swing distances, gravitational contributions, and sufficient momentum transfer, unlike when under more vertical flight conditions. On the other hand, the Narrow configurations (Figure 15a,d) perform well across a wide range of vertical flight conditions, benefiting from their design, which results in smaller swing distances and gravitational assistance. Their reduced success at lower flight angles, however, was attributed to excessive leg flexure leading to an overswing behavior and propeller contact, as depicted in Figure 16b. This issue, linked to the stiffness of the leg joints, will be a focus of future research, with the anticipation that stiffer joints could enhance

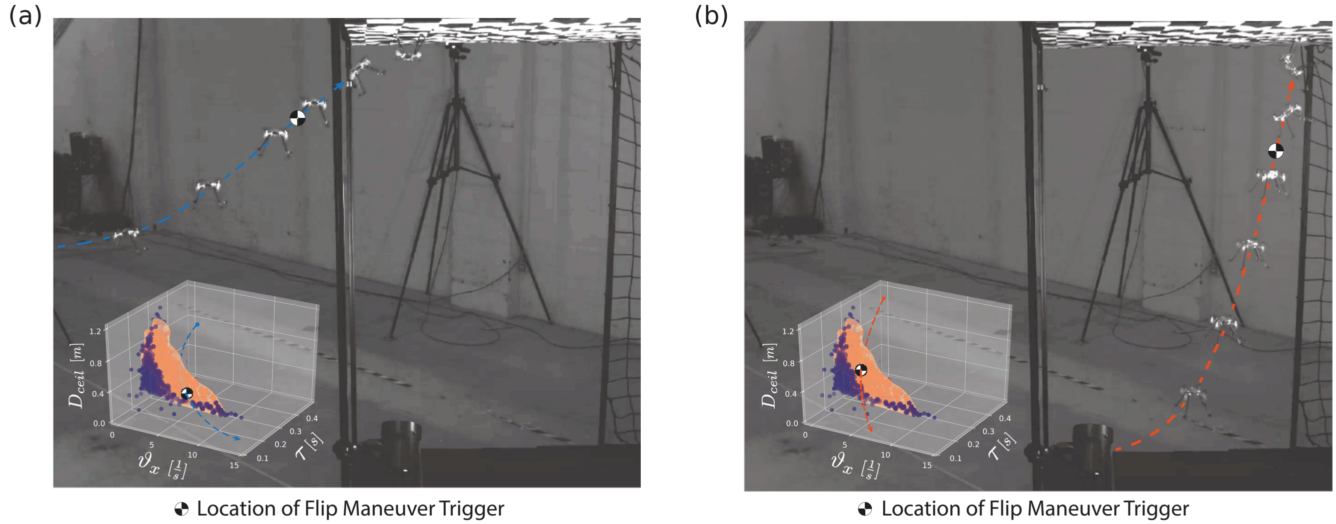


Fig. 17. (a) Experimental inverted landing sequence for the flight condition (2.50 m/s, 37°) resulting in a four-leg successful landing. The trajectory through $\mathbb{OF}_a$ space is depicted, with emphasis on the intersection point with the boundary region in both Cartesian and $\mathbb{OF}_a$ spaces. (b) A corresponding depiction for the flight condition (2.60 m/s, 70°), leading to a two-leg landing.

performance at lower flight angles.

Moreover, the Wide configurations, with their significantly larger swing distances due to leg geometry, and detrimental gravitational effects, demand higher translational velocities in flight for successful landings. While sufficient momentum transfer can be achieved in very low or extremely vertical flight angles, it becomes challenging in intermediate cases. Identifying and modelling the specific aspects which determine if a sufficient balance between swing distance, gravitational effects, and efficient momentum transfer factors can be achieved will also be the topic of future research. Finally, across all leg angles (i.e., Narrow, Semi-Narrow, Wide), the shorter leg configurations have a smaller *Impact Window Size*, which led to reduced performance across flight conditions (Figure 16a-c), compared to their longer leg counterparts (Figure 16d-f).

## VI. Validation of the Two-Stage Inverted Landing Policy in Experiments

In transitioning from simulation to experimental application and to validate the trained two-stage policy, we employed a zero-shot sim-to-real transfer approach. This validation was conducted using a Bitcraze Crazyflie 2.1 quadrotor, the same model our simulation was based on. As well, the simulated Semi-Narrow Long leg configuration yielded the overall best landing performance in our range of experimentally viable test conditions, therefore this leg configuration was used for our validation work.

For experimental testing, we selected ceiling-approach conditions with lower flight angles, primarily in the range of 30 to 65 degrees, which correlated with higher landing success rates in simulation and where our control policy demonstrated comparable performance to the collected sensory-action pairs. It is important to note that due to limitations such as the robot's thrust capacity and the available space, achieving high vertical velocity approaches were difficult in our experiments. This limitation is quantitatively illustrated as a dashed-line in Figure 14b., the region above which where the experimental tests because piratically infeasible. Approach conditions in this region typically resulted in lower success rates in simulation and were not chosen for experimental validation. In addition, we chose only to experimentally test the best-performing leg design among those tested in the simulation (shown in Figure 15) because the fast and dynamic perching experiments can be very costly, where a failed landing would result in the quadcopter crashing into the ceiling and/or falling to the ground, and therefore damaging the blades, frame or even motors of the quadcopter.

The results of our testing revealed a mixture of both optimal (i.e., characterized by four-leg touchdown and no body/propeller contact) and sub-optimal (i.e., two-leg landings or four-leg landings with body/propeller contact) landing outcomes; as illustrated by the foreground icons in Figure 14b. With classification of these conditions accomplished by observation through a high-speed camera. Notably, the simulation-based data using in training our control policy consisted primarily of optimal-class landings outcomes, so the prevalence of sub-optimal landings indicate degradation of our two-stage policy when transferring from our simulation-based environment to our physical experiment environment.

To this end, optimal landings were predominantly observed at flight angles between 30 and 45 degrees; an example of such a landing is illustrated in Fig. 17a. Conversely, sub-optimal landings were present across all tested conditions, often overlapping with optimal landings without a distinct separation. Figure 17b shows an example of such a sub-optimal landing. Notably, all flight conditions intersecting the triggering boundary function ($\pi_{Trg}$) achieved some level of success, resulting in either optimal or sub-optimal landings, with no direct failures recorded. Flight conditions outside the boundary function limits, represented in Fig. 14b as the dark purple area inside the grey-dashed line, were not tested as they fell beyond the feasible range of our control policy, posing a high risk of damage to the robot.

These experimental tests showed reduced performance compared to our simulation results, likely due to several factors. A major discrepancy stemmed from inaccuracies in the motor model, affecting both flight dynamics and body-rotation characteristics, and in the simulation of the landing gear/hinge joint. Specifically, the current model fails to fully capture the deformation behavior of the 3D printed landing gear. To overcome these issues, future work should expand use of domain randomization to encompass a wider range of system variabilities. Additionally, improving model fidelity and refining the reward function to emphasize robustness could significantly increase the success rate of experimental inverted landings.

## VII. Conclusion and Future Work

In this study, we aimed at achieving inverted landing in robotic fliers and explored the concept of dynamic perching, a skill mastered by many species birds, bees, and bats, yet a significant challenge in robotics. Our methodology centers on a biologically inspired, two-stage control policy that reflects the adept sensory-motor mapping observed previously in flies and utilizes augmented optical flow cues. It is formulated from sensory-action pairings identified through parameter optimization in simulation and further refined using various machine learning techniques. The policy translates emulated augmented optical-flow data into motor control actions in terms of initiating (in the first stage) and modulating (in the second stage) the rotational maneuvers that are critical for successful inverted landing. Our approach, distinct from traditional full-state trajectory-based methods, promotes a range of emergent behaviors through computationally-efficient reactive motor control. The effectiveness of this comprehensive policy, proven in both simulated and real-world environments, highlights its potential in advancing dynamic robotic perching.

Additionally, our study of varying leg geometries highlights the complex relationship between design, control, and function, emphasizing the importance of mechanical intelligence. Through testing various configurations, it is clear that leg geometry significantly influences the success and robustness of inverted landings. Factors like swing distance around fore-leg contact, and efficiency of momentum transfer at impact, which are crucial in determining inverted landing success, are directly influenced by the parameters of leg design. In our results, we found that narrower leg configurations lead to shorter swing distances, enhancing inverted landing robustness, particularly in vertical flight scenarios compared to more horizontal flight. As well, achieving a sufficiently high velocity is essential to bolster landing capabilities and ensure effective contact.

In conclusion, our research represents significant progress in emulating the perching capabilities of nature's most adept fliers. The integration of computational and mechanical intelligence is crucial in this pursuit. As the field of aerial robotics advances, emulating the perching expertise of birds and insects is becoming more attainable. Future work will focus on several key objectives: directly estimating sensory inputs from the onboard cameras and the robot's IMU; refining our two-stage control methodology to enhance the synergy between computational strategies and physical design; and integrating the identification of landing targets through computer vision methodologies with the generation of optimal approach trajectories via onboard motion planning and trajectory optimization. Together, these advancements will enable fully autonomous quadrotor landings on inverted surfaces. By deepening our understanding of these elements and unraveling their interdependence, we will move closer to achieving the autonomous robust perching capabilities in small aerial robots that will enable them to perch and fly reliably and safely in urban or natural environments. As well as performing further applications like prolonged inspection under bridges or serving as communication relays in cluttered disaster zones.

## Acknowledgments

We would like to extend our sincere gratitude to Zafar Anwar for his invaluable assistance with slow-motion video recording, and the OpenAI GPT-4 model for contributing to the editing process of this manuscript.

## Appendix A
## System Identification

### A. Inertia Estimation

Given our reliance on a zero-shot sim-to-real transfer method, the success of our two-stage control policy and the execution of inverted landings are strongly tied to the simulation's accuracy, thus emphasizing the need for precise rotational inertia estimation. To this end, we applied the bifillar pendulum method [83], [84] to measure the rotational inertia about the quadrotor's three principal axes. The technique involved suspending the quadrotor using two vertical strings and introducing a minor rotational displacement, thereby inducing oscillation about the vertical axis, as illustrated in Figure 18. Using the onboard gyro sensor, we measured the oscillation period and calculated the average time between peaks. The moment of inertia was then estimated using the following equation:

$$I_{est} = \frac{mg\ (D\ T_{avg})^2}{L\ (4\pi)^2}, \tag{8}$$

where $mg$ is the weight of the quadrotor in Newtons, $D$ is the distance between the strings, $L$ is the string length, and $T_{avg}$ is the average oscillation period. We repeated this procedure for each axis of the system. The results for the standard Crazyflie, as well as our modified system with attached legs, are presented in Table II.

*1) Thrust-Battery Compensation:* Maintaining the desired thrust values from the quadrotor's motors throughout the entire flight is essential for the accuracy and consistency of our experiments. This includes ensuring the thrusts used in experimental settings align with those in our simulation. To address this, we introduced a battery compensation algorithm that guarantees consistent thrust values through PWM modulation—irrespective of fluctuations in battery voltage.

TABLE II
MEASURED BODY MOMENT OF INERTIA

| | Crazyflie 2.1 - Stock | Crazyflie 2.1 - NL Legs |
|---|---|---|
| Mass [g] | 30.0 | 38.1 |
| $I_{xx}$ [$10^{-6}$ kg m$^2$] | 12.19 | 27.93 |
| $I_{yy}$ [$10^{-6}$ kg m$^2$] | 14.55 | 30.46 |
| $I_{zz}$ [$10^{-6}$ kg m$^2$] | 23.55 | 47.12 |

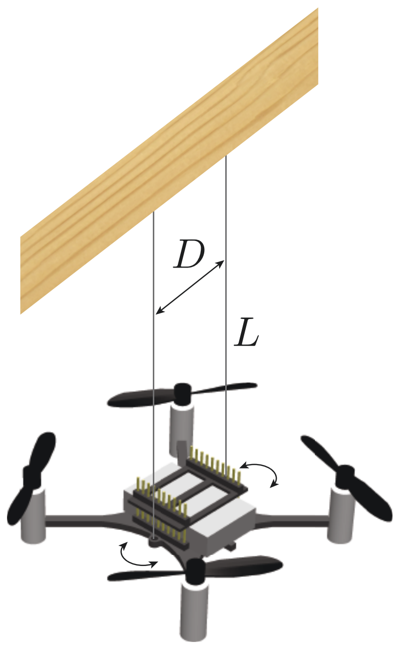


Fig. 18. Bifillar pendulum setup used to estimate rotational inertia about the quadrotor's primary axes. $D$ represents the distance between mounting points and $L$ represents the length of the string.

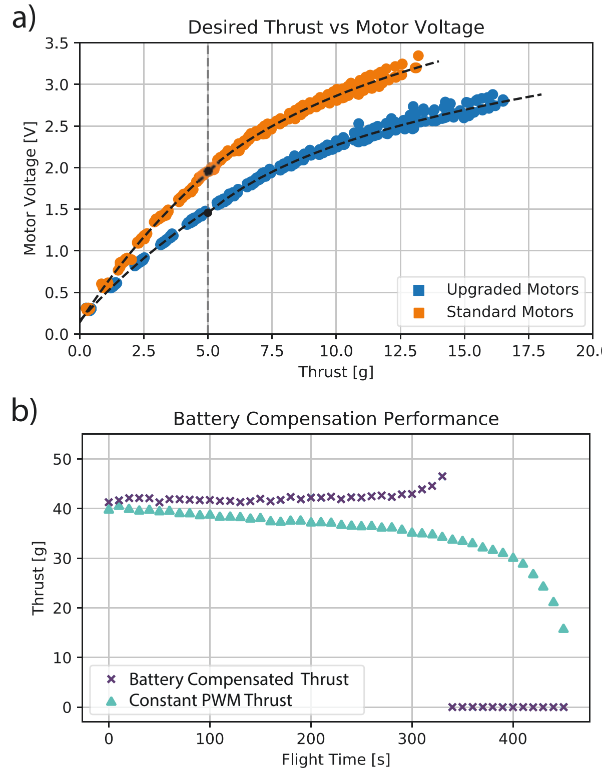


Fig. 19. (a) Plot illustrating the relationship between desired motor thrust and the necessary motor voltage ($V_{motor}$) for both standard and upgraded motors. (b) Plot depicting the efficacy of our upgraded motors using the battery compensation algorithm in comparison to a constant PWM strategy.

The battery compensation algorithm we developed, an enhancement of Bitcraze's open-source framework, is specifically designed to provide stable thrust during flight, counterbalancing the effects of battery depletion. Traditional brushed DC motors, like the ones originally equipped on the Crazyflie, generate thrust by applying a fraction of the available voltage to the motors via a PWM signal. However, as the battery voltage decreases over time, a constant PWM signal can lead to thrust inconsistency. To mitigate this, our algorithm modulates the PWM signal, maintaining a consistent voltage applied to the motors despite decreasing battery voltage, thereby ensuring consistent thrust.

To address fluctuations in the Crazyflie's battery voltage affecting motor performance, we established a regression-based approach to discern the voltage required for given thrust commands. Using a predictive curve, we modeled the voltage-thrust relationship, translating controller thrust values to the appropriate motor voltage.

To derive this curve, the Crazyflie quadrotor was powered by an external supply at a fixed voltage and to negate ground effect interference, the quadrotor was suspended over an airspace and anchored to a scale. We then systematically varied PWM values, logging the consequent thrust, supply voltage, and measured onboard voltage. The motor voltage, calculated as $V_{motor} = V_{onboard} \cdot \frac{PWM_{cmd}}{PWM_{Max}}$, was then plotted against the measured thrusts. The resultant data was then fitted to a logarithmic curve, described by the equation:

$$V_{motor} = a \cdot ln(f_{thrust} - b) + c. \tag{9}$$

For improved precision, especially at higher thrust values, we bifurcated the curve into two regions (refer to Figure 19a). In practice, the resultant PWM command is computed from the current battery voltage, desired voltage, and the requested motor thrust. This computation ensures constant thrust throughout the flight.

The efficacy of our algorithm is validated in Figure 19b. Here, we contrast the thrust values generated by our battery compensation and PWM modulation algorithm with those resulting from a constant PWM value over the life of a charged batttery. Both the standard Crazyflie 2.1 motors and our upgraded BetaFPV 7x16mm 19,000 KV motors underwent this procedure. The parameters identified through this process are detailed in Table III.

*2) Motor-Speed Dynamics:* To further enhance the precision of our simulation, we incorporated rotor acceleration dynamics by modeling the motor thrusts as a first-order system. This is an improvement on previous work by the authors which assumed an instantaneous motor response to thrust [25], neglecting the decay behavior of rear motors that relies solely on air drag forces for slowing down. The inclusion

TABLE III
BATTERY COMPENSATION PARAMETERS

| | $V_{motor} = a \cdot ln(b \cdot f_{thrust}) + c$ | | |
|---|---|---|---|
| Thrust $\leq$ 5g | $a$ | $b$ | $c$ |
| Stock Motors | 0.618 | 1.394 | 0 |
| Upgraded Motors | 1.285 | 1.512 | 0 |
| Thrust $>$ 5g | $a$ | $b$ | $c$ |
| Stock Motors | 2.097 | -4.464 | 5.636 |
| Upgraded Motors | 3.230 | -5.469 | 5.979 |

of this behavior in our model greatly improved the accuracy between our simulation and physical experiments.

For a precise representation of motor behavior, we established time-constants ($\tau_{up}/\tau_{down}$) to model the first-order system's speed-up and slow-down dynamics. To do this we built a custom tachometer, comprising an IR detection sensor and an Arduino micro-controller, to record the motor's time profile for varying speed changes. Using the conversion term from Forster et al. [85], we obtained the corresponding thrust values from propeller speeds. Then, an exponential curve was fitted to these profiles (Figure 20), yielding the speed-up and slow-down time constants ($\tau_{up} = 0.05$ and $\tau_{down} = 0.16$).

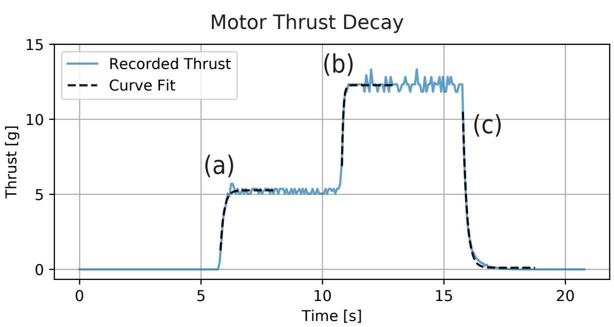


Fig. 20. Figure depicting the quadrotor motor's acceleration and deceleration characteristics, modeled as a first-order system. The curve fit time constants are: (a) $\tau_{up} = 0.06s$; (b) $\tau_{up} = 0.05s$; (c) $\tau_{down} = 0.16s$ .

## REFERENCES


[1] D. N. Lee, M. N. Davies, P. R. Green, and F. . Van Der Weel, "Visual control of velocity of approach by pigeons when landing," *Journal of experimental biology*, vol. 180, no. 1, pp. 85–104, 1993.

[2] W. R. Roderick, D. D. Chin, M. R. Cutkosky, and D. Lentink, "Birds land reliably on complex surfaces by adapting their foot-surface interactions upon contact," *Elife*, vol. 8, p. e46415, 2019.

[3] A. C. Carruthers, A. L. Thomas, S. M. Walker, and G. K. Taylor, "Mechanics and aerodynamics of perching manoeuvres in a large bird of prey," *The Aeronautical Journal*, vol. 114, no. 1161, pp. 673–680, 2010.

[4] M. V. Srinivasan, S.-W. Zhang, J. S. Chahl, E. Barth, and S. Venkatesh, "How honeybees make grazing landings on flat surfaces," *Biological cybernetics*, vol. 83, no. 3, pp. 171–183, 2000.

[5] E. Baird, N. Boeddeker, M. R. Ibbotson, and M. V. Srinivasan, "A universal strategy for visually guided landing," *Proceedings of the National Academy of Sciences*, vol. 110, no. 46, pp. 18 686–18 691, 2013.

[6] P. Tichit, I. Alves-dos Santos, M. Dacke, and E. Baird, "Accelerated landings in stingless bees are triggered by visual threshold cues," *Biology letters*, vol. 16, no. 8, p. 20200437, 2020.

[7] ——, "Accelerated landing in a stingless bee and its unexpected benefits for traffic congestion," *Proceedings of the Royal Society B*, vol. 287, no. 1921, p. 20192720, 2020.

[8] P. Goyal, J. L. van Leeuwen, and F. T. Muijres, "Bumblebees land rapidly by intermittently accelerating and decelerating towards the surface during visually guided landings," *iScience*, p. 104265, 2022.

[9] P. Goyal, G. de Croon, J. van Leeuwen, and F. Muijres, "Flight control model of landing maneuver in bumblebees," in *Zoology 2019*, 2019.

[10] P. Liu, S. P. Sane, J.-M. Mongeau, J. Zhao, and B. Cheng, "Flies land upside down on a ceiling using rapid visually mediated rotational maneuvers," *Science advances*, vol. 5, no. 10, p. eaax1877, 2019.

[11] S. Balebail, S. K. Raja, and S. P. Sane, "Landing maneuvers of houseflies on vertical and inverted surfaces," *PloS one*, vol. 14, no. 8, p. e0219861, 2019.

[12] A. Borst, "How do flies land?" *BioScience*, vol. 40, no. 4, pp. 292–299, 1990.

[13] D. N. Lee, J. A. Simmons, P. A. Saillant, and F. Bouffard, "Steering by echolocation: a paradigm of ecological acoustics," *Journal of Comparative Physiology A*, vol. 176, no. 3, pp. 347–354, 1995.

[14] D. K. Riskin, J. W. Bahlman, T. Y. Hubel, J. M. Ratcliffe, T. H. Kunz, and S. M. Swartz, "Bats go head-under-heels: the biomechanics of landing on a ceiling," *Journal of Experimental Biology*, vol. 212, no. 7, pp. 945–953, 2009.

[15] J.-C. Zufferey, *Bio-inspired flying robots: experimental synthesis of autonomous indoor flyers*. Epfl Press, 2008.

[16] A. Restas *et al.*, "Drone applications for supporting disaster management," *World Journal of Engineering and Technology*, vol. 3, no. 03, p. 316, 2015.

[17] B. Mishra, D. Garg, P. Narang, and V. Mishra, "Drone-surveillance for search and rescue in natural disaster," *Computer Communications*, vol. 156, pp. 1–10, 2020.

[18] S. J. Kim and G. J. Lim, "Drone-aided border surveillance with an electrification line battery charging system," *Journal of Intelligent & Robotic Systems*, vol. 92, no. 3, pp. 657–670, 2018.

[19] J. Seo, L. Duque, and J. Wacker, "Drone-enabled bridge inspection methodology and application," *Automation in Construction*, vol. 94, pp. 112–126, 2018.

[20] J. Irizarry, M. Gheisari, and B. N. Walker, "Usability assessment of drone technology as safety inspection tools," *Journal of Information Technology in Construction (ITcon)*, vol. 17, no. 12, pp. 194–212, 2012.

[21] K. Cesare, R. Skeele, S.-H. Yoo, Y. Zhang, and G. Hollinger, "Multi-uav exploration with limited communication and battery," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 2230–2235.

[22] J. Thomas, M. Pope, G. Loianno, E. W. Hawkes, M. A. Estrada, H. Jiang, M. R. Cutkosky, and V. Kumar, "Aggressive flight with quadrotors for perching on inclined surfaces," *Journal of Mechanisms and Robotics*, vol. 8, no. 5, p. 051007, 2016.

[23] F. Kendoul, "Four-dimensional guidance and control of movement using time-to-contact: Application to automated docking and landing of unmanned rotorcraft systems," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 237–267, 2014.

[24] F. Van Breugel and M. H. Dickinson, "The visual control of landing and obstacle avoidance in the fruit fly drosophila melanogaster," *Journal of Experimental Biology*, vol. 215, no. 11, pp. 1783–1798, 2012.

[25] B. Habas, B. AlAttar, B. Davis, J. W. Langelaan, and B. Cheng, "Optimal inverted landing in a small aerial robot with varied approach velocities and landing gear designs," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2003–2009.

[26] J. Mao, G. Li, S. Nogar, C. Kroninger, and G. Loianno, "Aggressive visual perching with quadrotors on inclined surfaces," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5242–5248.

[27] M. Kovac, "Learning from nature how to land aerial robots," *Science*, vol. 352, no. 6288, pp. 895–896, 2016.

[28] R. Pfeifer, M. Lungarella, and F. Iida, "Self-organization, embodiment, and biologically inspired robotics," *science*, vol. 318, no. 5853, pp. 1088–1093, 2007.

[29] W. G. Hyzer, "Flight behavior of a fly alighting on a ceiling," *Science*, vol. 137, no. 3530, pp. 609–610, 1962.

[30] M. Sitti, "Physical intelligence as a new paradigm," *Extreme Mechanics Letters*, vol. 46, p. 101340, 2021.

[31] A. Khaheshi and H. Rajabi, "Mechanical intelligence (mi): a bioinspired concept for transforming engineering design," *Advanced Science*, vol. 9, no. 32, p. 2203783, 2022.

[32] R. Pfeifer and J. Bongard, *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006.

[33] A. Borst, “Drosophila’s view on insect vision,” *Current biology*, vol. 19, no. 1, pp. R36–R47, 2009.

[34] M. V. Srinivasan, S. Zhang, and J. S. Chahl, “Landing strategies in honeybees, and possible applications to autonomous airborne vehicles,” *The Biological Bulletin*, vol. 200, no. 2, pp. 216–221, 2001.

[35] J. S. Chahl, M. V. Srinivasan, and S.-W. Zhang, “Landing strategies in honeybees and applications to uninhabited airborne vehicles,” *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 101–110, 2004.

[36] J. Mao, S. Nogar, C. Kroninger, and G. Loianno, “Robust active visual perching with quadrotors on inclined surfaces,” *arXiv preprint arXiv:2204.02458*, 2022.

[37] K. Hang, X. Lyu, H. Song, J. A. Stork, A. M. Dollar, D. Kragic, and F. Zhang, “Perching and resting—a paradigm for uav maneuvering with modularized landing gears,” *Science Robotics*, vol. 4, no. 28, p. eaau6637, 2019.

[38] W. Chi, K. Low, K. H. Hoon, and J. Tang, “An optimized perching mechanism for autonomous perching with a quadrotor,” in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 3109–3115.

[39] J. A. Dougherty and T. Lee, “Monocular estimation of ground orientation for autonomous landing of a quadrotor,” *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 6, pp. 1407–1416, 2016.

[40] M. T. Alkowatly, V. M. Becerra, and W. Holderbaum, “Bioinspired autonomous visual vertical control of a quadrotor unmanned aerial vehicle,” *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 2, pp. 249–262, 2015.

[41] S. Yang, S. A. Scherer, and A. Zell, “An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle,” *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1, pp. 499–515, 2013.

[42] H. Das, K. Sridhar, and R. Padhi, “Bio-inspired landing of quadrotor using improved state estimation,” *IFAC-PapersOnLine*, vol. 51, no. 1, pp. 462–467, 2018.

[43] C. Luo, L. Yu, and P. Ren, “A vision-aided approach to perching a bioinspired unmanned aerial vehicle,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 3976–3984, 2017.

[44] F. Kendoul, “Four-dimensional guidance and control of movement using time-to-contact: Application to automated docking and landing of unmanned rotorcraft systems,” *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 237–267, 2014.

[45] J. Thomas, J. Polin, K. Sreenath, and V. Kumar, “Avian-inspired grasping for quadrotor micro uavs,” in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 55935. American Society of Mechanical Engineers, 2013, p. V06AT07A014.

[46] J. Thomas, G. Loianno, K. Daniilidis, and V. Kumar, “Visual servoing of quadrotors for perching by hanging from cylindrical objects,” *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 57–64, 2016.

[47] P. Yu and K. Wong, “An implementation framework for vision-based bat-like inverted perching with bi-directionalthrust quadrotor,” *International Journal of Micro Air Vehicles*, vol. 14, p. 17568293211073672, 2022.

[48] L. Bai, H. Wang, X. Chen, J. Zheng, L. Xin, Y. Deng, and Y. Sun, “Design and experiment of a deformable bird-inspired uav perching mechanism,” *Journal of Bionic Engineering*, vol. 18, no. 6, pp. 1304–1316, 2021.

[49] C. E. Doyle, J. J. Bird, T. A. Isom, J. C. Kallman, D. F. Bareiss, D. J. Dunlop, R. J. King, J. J. Abbott, and M. A. Minor, “An avian-inspired passive mechanism for quadrotor perching,” *IEEE/ASME Transactions On Mechatronics*, vol. 18, no. 2, pp. 506–517, 2012.

[50] H. Hsiao, J. Sun, H. Zhang, and J. Zhao, “A mechanically intelligent and passive gripper for aerial perching and grasping,” *IEEE/ASME Transactions on Mechatronics*, 2022.

[51] H. Zhang, E. Lerner, B. Cheng, and J. Zhao, “Compliant bistable grippers enable passive perching for micro aerial vehicles,” *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 5, pp. 2316–2326, 2020.

[52] A. Kalantari, K. Mahajan, D. Ruffatto, and M. Spenko, “Autonomous perching and take-off on vertical walls for a quadrotor micro air vehicle,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4669–4674.

[53] Q. He, X. Xu, Z. Yu, K. Huo, Z. Wang, N. Chen, X. Sun, G. Yin, P. Du, Y. Li *et al.*, “Optimized bio-inspired micro-pillar dry adhesive and its application for an unmanned aerial vehicle adhering on and detaching from a ceiling,” *Journal of Bionic Engineering*, vol. 17, no. 1, pp. 45–54, 2020.

[54] H. Hsiao, F. Wu, J. Sun, and J. Zhao, “A novel passive mechanism for flying robots to perch onto surfaces,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1183–1189.

[55] H. W. Wopereis, T. Van Der Molen, T. Post, S. Stramigioli, and M. Fumagalli, “Mechanism for perching on smooth surfaces using aerial impacts,” in *2016 IEEE international symposium on safety, security, and rescue robotics (SSRR)*. IEEE, 2016, pp. 154–159.

[56] M. T. Pope, C. W. Kimes, H. Jiang, E. W. Hawkes, M. A. Estrada, C. F. Kerst, W. R. Roderick, A. K. Han, D. L. Christensen, and M. R. Cutkosky, “A multimodal robot for perching and climbing on vertical outdoor surfaces,” *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 38–48, 2016.

[57] Z. Huang, S. Li, J. Jiang, Y. Wu, L. Yang, and Y. Zhang, “Biomimetic flip-and-flap strategy of flying objects for perching on inclined surfaces,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5199–5206, 2021.

[58] J. Kim, M. C. Lesak, D. Taylor, D. J. Gonzalez, and C. M. Korpela, “Autonomous quadrotor landing on inclined surfaces using perception-guided active asymmetric skids,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7877–7877, 2021.

[59] X. Ni, Q. Yin, X. Wei, P. Zhong, and H. Nie, “Research on landing stability of four-legged adaptive landing gear for multirotor uavs,” *Aerospace*, vol. 9, no. 12, p. 776, 2022.

[60] D. J. Dunlop and M. A. Minor, “Modeling and simulation of perching with a quadrotor aerial robot with passive bio-inspired legs and feet,” *ASME Letters in Dynamic Systems and Control*, vol. 1, no. 2, 2021.

[61] S. Liu, W. Dong, Z. Wang, and X. Sheng, “Hitchhiker: A quadrotor aggressively perching on a moving inclined surface using compliant suction cup gripper,” *arXiv preprint arXiv:2203.02304*, 2022.

[62] T. Baca, P. Stepan, V. Spurny, D. Hert, R. Penicka, M. Saska, J. Thomas, G. Loianno, and V. Kumar, “Autonomous landing on a moving vehicle with an unmanned aerial vehicle,” *Journal of Field Robotics*, vol. 36, no. 5, pp. 874–891, 2019.

[63] H.-T. Zhang, B.-B. Hu, Z. Xu, Z. Cai, B. Liu, X. Wang, T. Geng, S. Zhong, and J. Zhao, “Visual navigation and landing control of an unmanned aerial vehicle on a moving autonomous surface vehicle via adaptive learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5345–5355, 2021.

[64] A. Paris, B. T. Lopez, and J. P. How, “Dynamic landing of an autonomous quadrotor on a moving platform in turbulent wind conditions,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9577–9583.

[65] J. L. Paneque, J. R. Martínez-de Dios, A. Ollero, D. Hanover, S. Sun, A. Romero, and D. Scaramuzza, “Perception-aware perching on power-lines with multirotors,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3077–3084, 2022.

[66] J. J. Gibson, *The ecological approach to visual perception: classic edition*. Psychology Press, 2014.

[67] P. Chirarattananon, “A direct optic flow-based strategy for inverse flight altitude estimation with monocular vision and imu measurements,” *Bioinspiration & biomimetics*, vol. 13, no. 3, p. 036004, 2018.

[68] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[69] B. K. Horn, Y. Fang, and I. Masaki, “Hierarchical framework for direct gradient-based time-to-contact estimation,” in *2009 IEEE Intelligent Vehicles Symposium*. IEEE, 2009, pp. 1394–1400.

[70] J. Wang, E. Uchibe, and K. Doya, “Em-based policy hyper parameter exploration: application to standing and balancing of a two-wheeled smartphone robot,” *Artificial Life and Robotics*, vol. 21, no. 1, pp. 125–131, 2016.

[71] B. Habas, J. W. Langelaan, and B. Cheng, “Inverted landing in a small aerial robot via deep reinforcement learning for triggering and control of rotational maneuvers,” *arXiv preprint arXiv:2209.11043*, 2022.

[72] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[73] Z. Cheng, C. Zou, and J. Dong, “Outlier detection using isolation forest and local outlier factor,” in *Proceedings of the conference on research in adaptive and convergent systems*, 2019, pp. 161–168.

[74] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: identifying density-based local outliers,” in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[75] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, “Support vector method for novelty detection,” *Advances in neural information processing systems*, vol. 12, 1999.

[76] J. Ma and S. Perkins, “Time-series novelty detection using one-class support vector machines,” in *Proceedings of the International Joint*

*Conference on Neural Networks, 2003.*, vol. 3. IEEE, 2003, pp. 1741–1745.
[77] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," in *Proceedings of the ACM SIGKDD workshop on outlier detection and description*, 2013, pp. 8–15.
[78] T. Lee, M. Leok, and N. H. McClamroch, "Nonlinear robust tracking control of a quadrotor uav on se (3)," *Asian Journal of Control*, vol. 15, no. 2, pp. 391–408, 2013.
[79] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.
[80] C. Evangelista, P. Kraft, M. Dacke, J. Reinhard, and M. V. Srinivasan, "The moment before touchdown: landing manoeuvres of the honeybee Apis mellifera," *Journal of Experimental Biology*, vol. 213, no. 2, pp. 262–270, 2010.
[81] W. R. Roderick, M. R. Cutkosky, and D. Lentink, "Touchdown to take-off: At the interface of flight and surface locomotion," *Interface Focus*, vol. 7, no. 1, 2017.
[82] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3299–3304.
[83] G. A. Garcia, A. R. Kim, E. Jackson, S. S. Keshmiri, and D. Shukla, "Modeling and flight control of a commercial nano quadrotor," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2017, pp. 524–532.
[84] M. R. Jardin and E. R. Mueller, "Optimized measurements of unmanned-air-vehicle mass moment of inertia with a bifilar pendulum," *Journal of Aircraft*, vol. 46, no. 3, pp. 763–775, 2009.
[85] J. Förster, "System identification of the crazyflie 2.0 nano quadrocopter," B.S. thesis, ETH Zurich, 2015.

**Bryan Habas** is currently pursuing his Ph.D. at The Pennsylvania State University, under the mentorship of Professor Cheng in the BioRob-InFL laboratory. He earned his Master of Science in Mechanical Engineering from the same institution in 2021. Bryan's research endeavors encompass a broad spectrum of areas including aerial robotics, control systems, applied machine learning, and the simulation of dynamic systems.

**Bo Cheng (Member, IEEE)** received the B.S. degree in automation from Zhejiang University, Hangzhou, China, in 2006, the M.S. degree in mechanical engineering from the University of Delaware, Newark, DE, USA, in 2009, and the Ph.D. degree in mechanical engineering from Purdue University, West Lafayette, IN, USA, in 2012. He is currently an Associate Professor with the Department of Mechanical Engineering, Pennsylvania State University, University Park, PA, USA. His research interests include bioinspired robotics, micro aerial vehicles, robot control and learning, insect and hummingbird flight, and bioinspired fluid dynamics.