Learning Variable Impedance Control for Aerial Sliding on Uneven Heterogeneous Surfaces by Proprioceptive and Tactile Sensing

Weixuan Zhang, Lionel Ott, Marco Tognon, and Roland Siegwart

Abstract—The recent development of novel aerial vehicles capable of physically interacting with the environment leads to new applications such as contact-based inspection. These tasks require the robotic system to exchange forces with partially-known environments, which may contain uncertainties including unknown spatially-varying friction properties and discontinuous variations of the surface geometry. Finding a solution that senses, adapts, and remains robust against these environmental uncertainties remains an open challenge. This letter presents a learning-based adaptive control strategy for aerial sliding tasks. In particular, the gains of a standard impedance controller are adjusted in real-time by a neural network policy based on proprioceptive and tactile sensing. This policy is trained in simulation with simplified actuator dynamics in a student-teacher learning setup. The real-world performance of the proposed approach is verified using a tilt-arm omnidirectional flying vehicle. The proposed controller structure combines data-driven and model-based control methods, enabling our approach to successfully transfer directly and without adaptation from simulation to the real platform. We achieved tracking performance and disturbance rejection that cannot be achieved using fine-tuned state of the art interaction control method.

Index Terms—Aerial systems: mechanics and control, compliance and impedance control, machine learning for robot control, reinforcement learning.

I. INTRODUCTION

Aerial interaction tasks such as contact-based inspections [1]–[3] require the flying vehicle to slide a sensor along the surface and maintain contact. These surfaces vary in spatial geometry and friction properties. This poses two challenges: Firstly, the surface friction property cannot be directly measured and may change discontinuously if the surface consists of different materials (i.e., heterogeneous surface). In addition, the perception of the surface geometry tends to be impaired by sensor noises, occlusions, and low spatial resolution, especially when the geometry is discontinuous (e.g. steps or holes). Secondly, from the control perspective, the presence of these unknown environment features introduces uncertainties in contact forces. The induced interaction wrench disturbance may cause large abrupt changes in the robot dynamics, which easily destabilizes the system. Existing approaches show interactions with simple continuous surfaces (e.g., planes, cylinders, etc), where the geometry is assumed to be known and friction properties are assumed to be identical everywhere (i.e., homogeneous) [4]–[6]. These approaches also require expert knowledge to perform manual tuning for each environment to achieve optimal performance. How to design a solution that senses, adapts and remains robust to the environmental uncertainties remains an open challenge.

This work presents a method to overcome these challenges. More specifically, a variable impedance controller that uses proprioceptive and tactile sensing to adjust controller gains according to environmental changes.

Fig. 1. A tilt-arm omnidirectional flying vehicle is sliding along surfaces with different friction properties and geometries. The end effector is in contact with a double-sided tape while sliding on a rock like papier-mâché.
An alternative approach to address the aforementioned sensing challenge is to use proprioceptive measurements and tactile sensing to react to changes in the geometry and friction properties of the environment. This type of approach can be found in quadrupedal robots like the MIT Cheetah [12] and Anymal [13]. These robots use either control signals to infer a leg touchdown event or use IMU signals, robot states, and control commands to implicitly infer the surface properties. Compared to visual sensing, these two sensing modalities are more direct in measuring the contact and are more suitable in sensing the discontinuous features of the environment. However, they were never exploited in the field of aerial manipulation to face unknown environments.

From a control perspective, impedance control [14] introduces algorithmic compliance and has shown success for aerial sliding tasks on homogeneous surfaces [4]. Selection of the impedance gains is a trade-off between controller tracking performance and system compliance. Depending on the task and environment, different impedance parameters may perform optimally [15]. A variable impedance controller [16], [17] is thus an attractive solution and has shown promising results in aerial manipulation [18] and [6]. To sense and adapt to the environmental uncertainties during aerial sliding tasks, one can vary the impedance parameters using the previously mentioned two sensing modalities. This nonlinear mapping may also be learned using deep reinforcement learning (RL) techniques [19], which have become a popular tool to generate highly nonlinear and effective control policies using neural networks. Examples combining reinforcement learning with variable impedance control can be found in legged robots [20] and manipulator [21]–[24]. For flying vehicles, learning from simulation instead of real-world is the preferred approach, since a failure typically leads to a crash. In particular, student teacher setups [25] are promising to improve learning efficiency, since a teacher policy can use privileged information from simulation to guide the student policy.

Considering the problem of aerial physical interaction with heterogeneous and uneven surfaces, taking inspiration from different robotic domains, this letter presents a novel control method that combines the benefits of proprioceptive and tactile sensing, variable impedance control and reinforcement learning. A neural network policy is learned to adapt the stiffness gains of a standard impedance controller according to proprioceptive and tactile sensing. The intuition is that these two sensing modalities jointly capture the interaction environmental properties and the learned policy can therefore adjust the controller to be robust against these disturbances. The training of this policy is conducted entirely in simulation in a student-teacher setup. This solution allows for a direct transfer of the learned policy from a simplified closed-loop simulation to an omnidirectional aerial vehicle (O mav as shown in Fig. 1 and described in [26]), significantly improving its robustness and control performance during interaction and outperforming fine-tuned impedance controller.

Our contributions are as follows:

- A learning-based solution for aerial sliding tasks that senses, adapts, and remains robust against challenging interaction environment uncertainties in surface geometry and friction properties.
- An approach to address sim-to-real transfer by including a closed-loop controller to suppress model uncertainty, which allows for learning from simplified actuator dynamics.

The above contributions have been validated experimentally using an Omav with a rigid single-body end effector for the task of sliding along surfaces with different friction properties and uneven geometry.

II. PRELIMINARIES

In this section we provide a brief overview of the models used to represent robot dynamics and environment interactions before describing the basics of impedance control.

A. Robot Dynamics

It is assumed that the flying vehicle has a single-body end effector rigidly attached to its body (see Fig. 2). The robot is modeled as a single rigid body and its dynamics are expressed using Newton-Euler method in free flight and interaction are given by the following equation

\[ M\ddot{\mathbf{v}} + C\dot{\mathbf{v}} + g = \mathbf{w}_{\text{act}} + \mathbf{w}_{\text{dist}}, \]

where \( M \in \mathbb{R}^{6\times 6} \) is the symmetric positive definite inertia matrix and \( C \in \mathbb{R}^{6\times 6} \) contains the centrifugal and Coriolis terms, and \( g \in \mathbb{R}^{6} \) is the gravity. The generalized velocity \( \dot{\mathbf{v}} \in \mathbb{R}^{6} \) represents the center of mass velocity and body rates of the system. The generalized acceleration vector are denoted as \( \ddot{\mathbf{v}} \). The terms \( \mathbf{w}_{\text{act}} \) and \( \mathbf{w}_{\text{dist}} \in \mathbb{R}^{6} \) are both stacked force and torque vectors acting on the system generated by rotor actuation and disturbance sources (e.g., contact or wind disturbances), respectively.
B. Interaction With the Environment

When the robot is sliding along the surface with its end effector, the force disturbance \( f_{\text{dist}} \) has three sources: 1) environmental aerodynamic effects, e.g., ground effects, wall effects, and wind gusts, 2) actuation modeling errors, and 3) the contact force \( f_{\text{con}} \). Both 1) and 2) are assumed to be negligible as they are at least an order of magnitude smaller than that of the contact force [27], [28]. The treatment of wind gust disturbance is referred to future work.

During interaction flights, the robot is assumed to have a single contact point with the uneven surface at \( C \) (Fig. 2). A local contact frame \( \mathcal{F}_T \) is attached to the contact point such that its \( x \)-axis is normal to the tangent plane at \( C \). The contact force acting on the end effector expressed in the body frame \( \mathcal{F}_B \) is modeled as follows:

\[
f_B = f_B^C = R_T^B (F_{\perp} n_T^f + F_{\parallel} n_f^f),
\]

where \( F_{\perp} \) is the scalar normal force and \( F_{\parallel} \) is the scalar friction force. The coordinate transformation matrix \( R_T^B \) transforms a vector from \( \mathcal{F}_T \) to \( \mathcal{F}_B \). The unit normal vector \( n_T^f \) is perpendicular to the tangent plane at \( C \) and \( n_f^f \) is the sliding force direction parallel to the tangent plane at \( C \). The relative orientation \( R_T^B \) is assumed to be partially known due to imperfect map. In addition, the end effector and the contact force creates a torque around the vehicle’s center of mass with the lever arm length denoted as \( l \).

When the end effector is sliding with nonzero velocity on the surface, a Coulomb friction model is assumed, i.e.:

\[
F_{\parallel} = \mu(p_C) F_{\perp},
\]

where \( \mu(p_C) \) is the friction coefficient that can vary spatially across the surface, depending the position of contact \( p_C \).

C. Impedance Controller With Constant Gains

An impedance controller with constant control gains is a common approach used for aerial sliding tasks [26] and used in this letter as the baseline approach.

Given a desired sliding path on the surface, a reference pose trajectory is designed based on the given surface map. This reference trajectory consists of a desired center of mass position which results in an end effector position that is always behind the sliding surface by a constant distance \( \delta \in \mathbb{R} \) (Fig. 2), also denoted as the penetration level. For the attitude part of the trajectory, the vector along the tool arm should align with the contact frame \( z \)-axis.

Given this desired reference trajectory, an impedance controller with constant control gains has the following form:

\[
w_{\text{act}} = C \ddot{v} + g + (M_M^{-1} - I_6) w_{\text{dist}}
M_M^{-1}(-M_M \ddot{v}_{\text{ref}} + D_M \dot{e}_v + K_M \dot{e}_s + K_M \dot{e}_s),
\]

with \( \dot{e}_s \in \mathbb{R}^6 \), containing the position and attitude tracking error as shown in [26]. \( M_M, D_M, K_M, K_M \in \mathbb{R}^{6 \times 6} \) are the desired inertia, damping, and stiffness matrices, respectively.

Plugging (4) into (1) results in

\[
M_M \ddot{e}_v + D_M \dot{e}_v + K_M \dot{e}_s = w_{\text{dist}}.
\]

This implies that an impedance controller shapes the closed-loop system as a second-order system. Note that there will be inevitable pose error due to the contact wrench acting on the flying vehicle. The wrench command \( u_{\text{act}} \) is then allocated through a chosen mapping and a saturation function to individual actuator commands (for more details see [26]).

III. METHODOLOGY

A. Problem Statement

The goal is to enable a flying vehicle to accurately follow a trajectory planned based on an imperfect map while remaining stable and staying in contact with an uneven surface which has unknown discontinuities in geometry and unknown friction properties \( \mu(p_C) \). Given a task-space reference trajectory, we assume the robot has access to contact wrench measurements \( w_{\text{meas}} \) via a force torque sensor, and it is controlled by an impedance controller with a gain-adjusting policy \( \pi_\theta \), which is parametrized by \( \theta \). To achieve the above goal, we propose a strategy to find a deterministic policy \( \pi_\theta \) that adjusts the controller’s gains to fulfill the following criteria: 1) minimize the tracking error \( ||e_v||^2 + ||e_s||^2 \) where \( || \cdot || \) denotes the Euclidean norm; 2) ensure \( f_{\text{con}} > 0 \); 3) ensure the platform stability.

B. Variable Impedance Learning Controller

The proposed approach adds a control gain adaptation policy to the standard impedance controller (4). The policy adapts the impedance controller gains based on the proprioceptive measurements and the tactile feedback via the adaptive unit depicted in Fig. 3. In particular, the desired stiffness \( K_{\text{des}}(z) \in \mathbb{R}^{6 \times 6} \) is a function of the \( m \)-dimensional measurements \( z \in \mathbb{R}^m \):

\[
K_{\text{des}}(z) = K_{\text{min}} + (K_{\text{max}} - K_{\text{min}}) \text{diag}(\pi(z))
\]

where \( m \) is the dimension of the measurements, \( K_{\text{min}} \) and \( K_{\text{max}} \) are diagonal positive semidefinite matrices, and \( \pi : \mathbb{R}^m \rightarrow \mathbb{R}^6 \) where \( 0 < \pi(\cdot) < 1 \) with vector 0 and 1 of dimension 6. These constraints on the mapped value make sure the adaptive gains have lower and upper bounds. The lower bound \( K_{\text{min}} \) ensures a minimum tracking performance while the upper bound \( K_{\text{max}} \) prevents the system from instabilities caused by actuator saturation and system delay. These limits are both derived empirically.

---

**Figure 3** Variable impedance learning controller. It augments the control strategy depicted in [26] by adding a control gain adaptation policy, which takes as input the state, control error and the wrench measurements from the wrench sensor and map them to impedance gain.
through experiments. For brevity, in the following we use $\pi$ instead of $\pi_{\theta}$.

To obtain a damped second-order system, we impose a fixed relationship between $D_{\text{des}}$ and $K_{\text{des}}(z)$. The desired damping $D_{\text{des}}$ is varied with the square root of the diagonal components of $K_{\text{des}}(z)$:

$$D_{\text{des}} = 2\zeta \sqrt{K_{\text{des}}(z)},$$

(7)

where $\zeta$ is a damping ratio. While we assume that the desired stiffness and damping can be well tracked, the desired inertia is in practice challenging to track as it requires an accurate actuation control [4]. We therefore set $M_{\text{des}}$ equal to $M$ and the adaptation of $M_{\text{des}}$ is deferred to future works.

With $M = M_{\text{des}}$ and (7) inserted into (4), the following adaptive controller command can be obtained,

$$w_{\text{act}} = M \dot{v}_{\text{ref}} - 2\zeta \sqrt{K_{\text{des}}(z)} \ddot{e}_v - K_{\text{des}}(z) \dot{e}_s + Cv + g.$$  

(8)

With (8) plugged into (1), the closed-loop error dynamics are shaped as a second-order system,

$$M_{\text{des}} \ddot{e}_v + 2\zeta \sqrt{K_{\text{des}}(z)} \dot{e}_v + K_{\text{des}}(z) \dot{e}_s = w_{\text{dist}}.$$  

(9)

Note that changing the stiffness $K_{\text{des}}(z)$ affects the interaction wrench $w_{\text{dist}}$. To see this, consider (9) at steady state $\dot{e}_v = \ddot{e}_v = 0$ and projected along the surface normal direction. We obtain

$$k_{\perp}(z) \delta = F_{\perp},$$

(10)

where $k_{\perp}(z)$ is the position stiffness gain in the surface normal direction. Assuming the end effector is in contact with the surface, which results in a constant position tracking error $\delta$, $F_{\perp}$ is thus proportional to $k_{\perp}(z)$. This is the intuition of why we adapt the stiffness gain to counteract surface disturbances.

### C. Reinforcement Learning of Gain Adaptation Policy

We assume that the mapping $\pi$ can be modeled as a discrete-time continuous Markov decision process (MDP). An MDP is defined by a state space $S$, an action space $A$, a scalar reward function $R$, and the transition probability $P$ that dictates the stochastic system dynamics. A learning agent selects an action $a$ from its policy $\pi$ and receives a reward $r$. The objective of the RL framework is to find an optimal policy $\pi^*$ that maximizes the discounted sum of rewards over an infinite time horizon:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau(\pi)} \sum_{t=0}^{\infty} \gamma^t r[t]$$

(11)

where $\gamma \in (0, 1)$ is the discount factor, and $\tau(\pi)$ is the trajectory distribution under policy $\pi$, with $t$ denoting the discretized time indices. The reward $r[t]$ at $k$ is

$$r[t] = -l_e R||e_R[t]||^2 - l_p ||e_p[t]||^2 - l_d ||d[t]||^2 - l_o ||\omega[t]||^2 - l_a \frac{||a[t]||}{||a[t]|}|| - \frac{||a[t-1]||}{||a[t-1]||}.$$  

(12)

$l_*$ with subscripts $*$ denotes the corresponding weight chosen such that all individual reward terms are at the same order of magnitude. $e_R[t]$ and $e_p[t]$ denotes the attitude and translational tracking error. $d[t] = (p_R^T + p_{\text{end}}^T - p_{\text{end}}^T) \cdot e_R^T$ denotes the scalar distance along the $z$-axis of the local contact frame between the end effector and the surface and $\cdot$ denotes the dot product between two vectors. The purpose of this term is to make sure that the policy keeps the end effector in contact with the surface. $\omega[t]$ denotes the angular velocity. We penalize large angular velocities to avoid instabilities. The action $a[t]$ is the output of the policy $\pi(z)$ at time $t$. The associated term is to make sure that the control inputs are smooth. The loss components are designed to reduce the tracking error while keeping contact with the surface without causing discontinuities in the actions and thus the actuator commands. These terms are in line with the problem statement in Section III-A. Note that since the simulation of each individual actuator dynamics is omitted. The energy consumption of the flying vehicle can only be indirectly inferred and is therefore not included in the reward function.

The policy $\pi$ is a fully connected neural network with three hidden layers of 32 units, its activation functions being leaky ReLu, and its last layer being a Sigmoid layer which guarantees to map to a bounded interval. For training we use the off-the-shelf RL algorithm proximal policy optimization (PPO) [29], a policy gradient algorithm that has been demonstrated to work for variable impedance control in contact tasks with a manipulator [30].

### D. Simulation Using Simplified Dynamics

To allow for efficient evaluation and training of the policy $\pi$, a simplified dynamics simulation is used. The flying vehicle is simulated as a single rigid body and the simulation of the individual actuator dynamics are approximated collectively as a single process. A saturation function on the wrench command is implemented, the output of which is delayed and set as external force and torque directly acting on the robot. Both the saturation threshold and the system delay are a conservative estimate of the empirically obtained actuation limits. This ensures that the actuator limits are well respected and the closed-loop system behaves like a delayed second-order system as designed. The inertia and mass are obtained from CAD. Although the actuator dynamics are simplified, special attention is paid to identify the correct center of mass position and the relative position of the end effector in the body frame ($r_{\text{com}}$ and $r_{\text{end}}$ in Fig. 2). They together determine the induced torque disturbance from a given contact force, which is essential for the simulation to learn the correct disturbance rejection strategy.

To simulate the interaction environment, surfaces that have different friction coefficients are generated and concatenated together. Therefore, when the robot’s end effector slides across the border between two surfaces with different friction properties, it experiences discontinuous changes in interaction forces. Furthermore, each surface can have a different height that leads to an uneven surface as a whole.

---

1The same material was used for the end effector throughout this letter. Thus, for the sake of brevity, we only talk about surface friction coefficients when it would be more accurate to talk about friction pairs between the end effector and the surface.
The teacher serves as a guidance and is equipped with an NUC i7 (see (10)) and therefore the torque disturbance, \(-\pi_\text{torque disturbance}\). We remark that the stability using RL may be further refined using RL. Noisy observations (especially the interaction wrench measurements) are first-order low-pass filtered before they are input to the student policy. To have a smooth change in the stiffness gain \(K_{\text{des}}(z)\), the output of the student policy is also low-pass filtered before they are used to compute \(K_{\text{des}}(z)\).

### E. Learning From Simulation

To efficiently learn an optimal policy that determines the adaptive stiffness \(K_{\text{des}}(z)\) (see (6)), a student teacher learning approach [25] is deployed.

Fig. 4 provides an overview of this approach: Firstly, we design a teacher with access to privileged (ground-truth) information to dynamically select the desired stiffness in the variable impedance controller. Then a policy is learned to emulate the teacher and may be further improved using RL. The policy can be directly transferred to real-world without any additional sim-to-real adaptation.

The intuition behind the student teacher learning is that the teacher has access to the privileged information which makes it much easier to design or train in an RL setting. We can also embed empirical tuning experience or other adaptive variable impedance strategies into the teacher. This is helpful for challenging tasks, as we find out empirically a direct reinforcement learning always lead to instabilities of the flying vehicle and prevents successful learning.

1) **Teacher Design:** The teacher \(\pi_t\) serves as a guidance policy for the student policy. It makes use of the privileged information \(z_{\text{priv}}\). Compared to \(z\), it contains additional information from the simulation, to which the student policy does not have access in real deployment.

In this work, we employ either a simple handcrafted policy \(\pi_t\) or a neural network learned from simulation using reinforcement learning as the teacher. Upon a rough surface, the handcrafted policy decreases the translational stiffness gain to reduce the normal force \(F_n\) (see (10)) and therefore the torque disturbance, while the angular stiffness gain is also increased to better reject the torque disturbance. For reinforcement learning of the teacher, the teacher is learned from scratch using the privileged information, including friction coefficient at contact point and the surface normal vector in the vicinity of the contact.

2) **Student Learning:** The control gain adaptation policy \(\pi^*(z)\) is bootstrapped via supervised learning with the following loss function

\[
\pi^* = \arg \min_\pi \|\pi_t(z_{\text{priv}}) - \pi(z)\|^2. \tag{13}
\]

where the feature \(z\) contains control signal, state estimate, IMU signals and interaction wrench measurements, with the first three signals being proprioceptive sensing and the last one being tactile sensing. They indirectly provide the information about the interaction between the environment and the robot.

Training data is collected by rolling out the simulation using the teacher. For each rollout, the robot first approaches the surface, gets into contact and starts sliding following the desired trajectory. The policy \(\pi^*(z)\) can be further refined using RL.

3) **Data Processing:** Noisy observations (especially the interaction wrench measurements) are first-order low-pass filtered before they are input to the student policy. To have a smooth change in the stiffness gain \(K_{\text{des}}(z)\), the output of the student policy is also low-pass filtered before they are used to compute \(K_{\text{des}}(z)\).

### F. Remark

1) **Stability:** We remark that the stability using RL may be ensured using the concept of passivity [31], or Lyapunov methods [32]. This is left as future work. Instead we discuss here practical measures to face possible causes of the instability. Those are mainly twofold: actuator saturation due to the rapid changes in the control gains or high gains and low gains which is incapable of stabilizing this open-loop unstable system. Thus we implemented the following strategies: 1) Lower and upper bounds on the stiffness gain as shown in (6); 2) Slew rate limit on the gains are empirically determined and only applied in deployment as precaution; 3) The output of the policy \(\pi^*_s(z)\) is filtered for a smooth control signal in training and deployment.

2) **Sim-to-Real Transfer:** The sim-to-real gap, i.e., the mismatch between simulation and reality, is a challenging problem when learning from simulation, and can limit the transferability of the learned policy. Frequent causes of sim-to-real gaps are inaccurate modeling of the actuator dynamics and delays in the system [33]. This is especially a problem for end-to-end learning approaches. Without feedback controller in the loop the learning procedure relies on the accuracy of the open-loop dynamics simulation and the sim-to-real gap can diverge exponentially. However, given a well-designed feedback controller (e.g., the one one presented in Fig. 3), which shapes the system to a desired second-order system (9) and suppresses model uncertainty, the gap between the reality and the simulation is kept small. This is particularly advantageous for a complicated system like the Omav, where a large amount of training data is required for an accurate model learning of the whole body dynamics (18 actuators and 6 degrees of freedom). As a comparison, a million samples are required for the modeling of a single one degree of freedom actuator of quadrupedal walking robot [33]. What is worse, if the Omav crashes or if its configuration changes, training data needs to be recollected again for an accurate model learning. Our approach does not require such a meticulous effort.

### IV. Experiments

#### A. Experimental Setup

The experimental set-up and the platform are shown in Figs. 1 and 6. The experiments are carried out at the indoor aerial robotic testbed of the Autonomous System Lab, ETH Zurich. The Omav weighs 4.5 kg and is equipped with an NUC i7 computer and a PixHawk flight controller. For a more complete
A motion capture system provides pose estimates for both the robot and the whiteboard/rock at a challenging environment typically seen in real applications. A pole end effector, measures the interaction wrench. There are three interaction environments for real experiments: sand paper on a flat white board to investigate interaction with heterogeneous surface, a step of 2 cm on a white board (Fig. 6) to investigate interaction with discontinuous surface geometry, and finally, a rock-like structure (Fig. 1) which combines both traits to form a challenging environment typically seen in real applications. A motion capture system provides pose estimates for both the robot and the whiteboard/rock at 100 Hz. The robot is unaware of their surface properties and the local unevenness on the surface. The task trajectory is to follow a straight line trajectory parallel to the gravity vector with zero pitch and roll while sliding on the vertical surface. The penetration level $\delta$ is set to 0.07 m, which leaves sufficient margin to ensure contact under the state estimate uncertainty.

For the simulation we use RaiSim [34], a cross-platform multi-body physics engine for robotics. During training in the simulation, for each rollout, the robot approaches the surface and starts sliding along the surface for 15 seconds, which emulates a task trajectory from the real-world experiments. During each rollout, the Omav reaches a speed of 0.2 m/s and slides across surfaces with different friction coefficients. For each interaction environment, we set up a different interaction environment in the simulation and learn a different policy to treat each problem separately. Find a single policy that tackles different environments can be studied in the context of continual learning and is planned for future work.

To train the teacher or student policy using RL, a hundred simulation instances are spawned with slightly perturbed vehicle and environment properties. In each epoch, these instances are simultaneously simulated to obtain a batch of rewards for stochastic gradient policy optimization. The policy was trained on a single NVIDIA 3060Ti GPU, which takes from 2 to 8 hours depending on the interacting environment.

Since there is only a straight line to follow, the dimension of the measurement vector $z$ is reduced to six: filtered pitch velocity, pitch error, position control error in the sliding direction, linear velocity, friction force filtered, normal force filtered. The output to the action space are the linear gain in the surface normal direction and the angular gain in the pitch direction, i.e. the axis of torque disturbance. The rest of the controller gains is kept constant.

B. Sim-to-Real Transfer

Fig. 5 demonstrates the policy transfer from simulation to reality. It compares four experiments completing the same sliding task with different controllers (baseline or our approach) and different set-ups (simulation or real experiment). The plots are aligned using the measured force impact when the Omav enters from free flight into contact with the whiteboard. For evaluation, the Omav must slide across three concatenated surfaces: the first is the white board with low friction, then a sand paper with high friction, and finally the whiteboard again. The friction coefficients are empirically estimated through force torque sensor measurements, with which three surfaces are generated in the simulation that replicate the experimental evaluation set-up. The approach in evaluation is a policy $\pi^*$ defined in the (13) (without further refinement using RL) and a baseline impedance controller with constant control gains. The policy is trained in a simulation environment with six different surface friction coefficients (0.05, 0.15, 0.25, 0.45, 0.55, 0.62) that covers a range of friction coefficients. During collection of data using the teacher, each surface is randomly assigned one of the six friction coefficients to robustify the learned policy.

Fig. 5 demonstrates that the learned adaptive control strategy can be successfully transferred from simulation to the real-world. Given the same controller (either baseline or our approach), the simulation and the real-world experiments result in a close similarity of the respective robot state (the pitch angle) and controller gains (the angular stiffness gain $k_\theta$ and translational stiffness gain $k_l$). The pitch angle is shown since it has the most obvious correlation with the surface friction coefficient given the baseline controller.

C. Sliding Across a Heterogeneous Flat Surface

Fig. 5 also showcases the performance of the regressed student policy on heterogenous surfaces. In this subsection, we only evaluate the real experimental data. While sliding on the surface, the contact force unavoidably leads to a tilted angle of the Omav. With the baseline controller, the vehicle tilts on average 4.9° after encountering a high friction surface (from 199 s to 202 s), whereas with our approach, the tilt of the vehicle only increases to about 1.3° on average. This shows that the Omav is able to keep a almost constant tilt angle when sliding across different surfaces, thus improve the attitude tracking performance and reducing the chance of a crash at the transition of difference surfaces.
D. Sliding Over a Surface With Discontinuous Geometry

The experiment (Fig. 6) presented in this section aims to investigate the ability to reject the disturbance caused by surface discontinuity and remain stable (criteria 3 in Section III-A). For the real experiment, a piece of foam is taped to the whiteboard and creates a step of about 2 cm along the sliding trajectory. During sliding, the step blocks the end effector, which leads to an increase of the pitch angle. The end effector then detaches from the surface and the sudden disappearance of the contact force presents as a large disturbance to the robot.

Our approach is trained as follows: Initially an even surface with randomly generated friction coefficients as described in Section IV-C is used to bootstrap the policy. Then a piece of uneven surface with 1 cm steps is created by setting each neighbouring surface to have a height difference of 1 cm. The policy is refined and trained for 900 epochs. The height difference is then changed to 2 cm with another 700 epochs of training. Training was terminated early when the reward stopped increasing for 300 epochs.

The behaviour exhibited by our approach when a step is encountered is to adapt the control gain to be more compliant. Compared to the baseline approach, this leads to less oscillations in pitch velocity when the Omav’s end effector slips and is out of contact (the yellow region in Fig. 6).

E. Sliding Across a Challenging Surface

A challenging rock-like papier-mâché surface (Fig. 1) is set up to compare our approach with baseline controllers. The surface of this rock is uneven and heterogeneous. Double-sided tapes (high friction) and plastic surface with lubricant (low friction) are added to the surface to emulate heterogeneous surface in extreme situations. Two trajectories (trajectory red and blue as the color in Fig. 7 indicates) were tested for more variety. For the training environment, we use procedurally generated terrain maps that have similar surface variations to the rock. During rollout, the friction coefficients are randomly selected from six friction coefficients (0.1, 0.15, 0.45, 0.6, 0.75, 0.9) every four seconds. Note that the evaluation and training environments are distinct and demonstrated the generalizability of our approach.

As shown in Fig. 7, our approach consistently outperforms the baseline controller with nine combinations of low, middle and high angular and translational gain. These different combinations represent a tuning process that is often practiced in real-world. Each data point represents the average tracking performance over the sliding of the same trajectory for the three times with the same controller. Several interesting observations can be made: firstly, among the baseline approaches, a good tracking performance is generally achieved by high stiffness controller but the converse is not true. It is observed in the experiment that a high gain controller ($k_l = 20$ and $k_a = 100$ for trajectory blue) can lead to instability. However, we never experienced instability issues with our policy throughout the experiments. This indicates that our approach adapts the controller to be more compliant when necessary. Secondly, for each trajectory, the optimal set of constant gains are different. For example, to achieve best tracking performance in pitch, trajectory red and trajectory blue have different optimal gains (upper left in the plot). This means that in reality, the engineer have to tune the parameters for each specific trajectory and surface to gain optimal performance. However, our approach always performs well for both trajectories. The tracking data points of our approach are both located in the lower left of the plot, which means good tracking performance in both position and orientation. This implies that our policy adapts to the different surface unevenness and varying friction coefficients. An time history plot of partial inputs and outputs of the policy in plot Fig. 8 provides an intuition on the adaptiveness of our policy. Note that when the pitch error increases from 194 s to 195.5 s, the angular gain $k_a$ is also increased to maximum to reduce the tracking error. However, around 195.5, the end effector detaches from the surface and causes oscillations on pitch rate, the angular gain is decreased to be more compliant and the vehicle remains stable.
V. CONCLUSION

This letter presented an approach that senses, adapts and remains robust against disturbances caused by discontinuous surface variations in geometry and friction during aerial sliding tasks for fully actuated flying vehicles. When the environmental property changes, an adaptation policy adjusts the control gains of a standard impedance controller to reject these disturbances. Experimental results demonstrated that the policy learned in simulation can be directly transferred to the aerial vehicle without adaptation. The learned policy is able to slide on a challenging rock-like surface and outperform state-of-art interaction controllers.

ACKNOWLEDGMENT

The authors would like to thank Nicholas Lawrence for thoroughly going through the manuscript and Eugenio Cuniato for the fruitful discussions and help with experiments.

REFERENCES

[1] M. Tognon et al., “A truly-redundant aerial manipulator system with application to push-and-slide inspection in industrial plants,” IEEE Robot. Automat. Lett., vol. 4, no. 2, pp. 1846–1851, Apr. 2019.
[2] M. Á. Trujillo, J. R. Martinez-de Dios, C. Martin, A. Viguaria, and A. Ollero, “Novel aerial manipulator for accurate and robust industrial NDT contact inspection: A new tool for the oil and gas inspection industry,” Sensors, vol. 19, no. 6, 2019, Art. no. 1305.
[3] R. Watson et al., “Dry coupled ultrasonic non-destructive evaluation using an over-actuated unmanned aerial vehicle,” IEEE Trans. Automat. Sci. Eng., early access, Jul. 26, 2021, doi: 10.1109/TASE.2021.3094966.
[4] M. Ryll et al., “6D interaction control with aerial robots: The flying end-effector paradigm,” Int. J. Robot. Res., vol. 38, no. 9, pp. 1045–1062, 2019.
[5] D. Tzoumanikas, F. Graule, Q. Yan, D. Shah, M. Popovic, and S. Leutenegger, “Aerial manipulation using hybrid force and position NMPC applied to aerial writing,” in Proc. Robot.: Sci. Syst., 2020.
[6] K. Bodie et al., “Active interaction force control for contact-based inspection with a fully actuated aerial vehicle,” IEEE Trans. Robot., vol. 37, no. 3, pp. 709–722, Jun. 2021.
[7] E. Shahiriari, L. Johannsmeier, E. Jensen, and S. Haddadin, “Power flow regulation, adaptation, and learning for intrinsically robust virtual energy tanks,” IEEE Robot. Automat. Lett., vol. 5, no. 1, pp. 211–218, Jan. 2020.
[8] S. Jung, T. C. Hsia, and R. G. Bonitza, “Force tracking impedance control of robot manipulators under unknown environment,” IEEE Trans. Control Syst. Technol., vol. 12, no. 3, pp. 474–483, May 2004.
[9] W. Amanhoud, M. Khoramshahi, M. Bonnesoeur, and A. Billard, “Force adaptation in contact tasks with dynamical systems,” in Proc. IEEE Int. Conf. Robot. Automat., 2020, pp. 6841–6847.
[10] D. Lee, H. Seo, I. Jang, S. J. Lee, and H. J. Kim, “Aerial manipulator pushing a movable structure using a DOB-based robust controller,” IEEE Robot. Automat. Lett., vol. 6, no. 2, pp. 723–730, Apr. 2021.
[11] A. Suarez, G. Heredia, and A. Ollero, “Physical-virtual impedance control in ultralightweight and compliant dual-armed aerial manipulators,” IEEE Robot. Automat. Lett., vol. 3, no. 3, pp. 2553–2560, Jul. 2018.
[12] D. J. Hyun, S. Seok, J. Lee, and S. Kim, “High-speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the mit cheetah,” Int. J. Robot. Res., vol. 33, no. 11, pp. 1417–1445, 2014.
[13] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning quadrupedal locomotion over challenging terrain,” Sci. Robot., vol. 5, no. 47, 2020, Art. no. eabc5986.
[14] N. Hogan, “Impedance control: An approach to manipulation,” in Proc. IEEE Amer. Control Conf., 1984, pp. 304–313.
[15] D. S. Walker, J. K. Salisbury, and G. Niemeyer, “Demonstrating the benefits of variable impedance to telerobotic task execution,” in Proc. IEEE Int. Conf. Robot. Automat., 2011, pp. 1348–1353.
[16] R. Ikeura and H. Inooka, “Variable impedance control of a robot for cooperation with a human,” in Proc. IEEE Int. Conf. Robot. Automat., 1995, vol. 3, pp. 3097–3102.
[17] F. J. Abu-Dakka and M. Saveriano, “Variable impedance control and learning a review,” Front. Robot. AI, vol. 7, 2020, Art. no. A590681.
[18] A. Y. Mersha, S. Stramigoli, and R. Carloni, “Variable impedance control for aerial interaction,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2020, pp. 3435–3440.
[19] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, Cambridge, MA, USA: MIT Press, 2018.
[20] M. Bogdanovic, M. Khadiv, and L. Righetti, “Learning variable impedance control for contact sensitive tasks,” IEEE Robot. Automat. Lett., vol. 5, no. 4, pp. 6129–6136, Oct. 2020.
[21] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, “Learning variable impedance control,” Int. J. Robot. Res., vol. 30, no. 7, pp. 820–833, 2011.
[22] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, and K. Harada, “Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach,” Appl. Sci., vol. 10, no. 19, 2020, Art. no. 6923.
[23] C. C. Beltran-Hernandez et al., “Learning force control for contact-rich manipulation tasks with rigid position-controlled robots,” IEEE Robot. Automat. Lett., vol. 5, no. 4, pp. 5709–5716, Oct. 2020.
[24] L. Roveda et al., “Model-based reinforcement learning variable impedance control for human-robot collaboration,” J. Intell. Robot. Syst., vol. 100, no. 2, pp. 417–433, 2020.
[25] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, “Learning by cheating,” in Proc. Conf. Robot. Learn., 2020, vol. 100, pp. 61–68.
[26] K. Bodie et al., “An omnidirectional aerial manipulation platform for contact-based inspection,” in Proc. Robot.: Sci. Syst. XV, 2019.
[27] A. Garofano-Soldado, G. Heredia, and A. Ollero, “Aerodynamic interference in confined environments with tilted propellers: Wall effect and corner effect,” in Proc. IEEE Aerial Robot. Syst. Physically Interacting Environ., 2021, pp. 1–8.
[28] W. Zhang, M. Brunner, L. Ott, M. Kamel, R. Siegwart, and J. Nieto, “Learning dynamics for improving control of overactuated flying systems,” IEEE Robot. Automat. Lett., vol. 5, no. 4, pp. 5283–5290, Oct. 2020.
[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017, arXiv:1707.06347.
[30] R. Martin-Martín, M. A. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg, “Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2019, pp. 1010–1017.
[31] E. Spyrikos-Papastavridis, P. R. N. Childs, and J. S. Dai, “Passivity preservation for variable impedance control of compliant robots,” IEEE/ASME Trans. Mechatronics, vol. 25, no. 5, pp. 2342–2355, Oct. 2020.
[32] S. A. Khader, H. Yin, P. Falco, and D. Kragic, “Stability-guaranteed reinforcement learning for contact-rich manipulation,” IEEE Robot. Automat. Lett., vol. 6, no. 1, pp. 1–8, Jan. 2021.
[33] J. Hwangbo et al., “Learning agile and dynamic motor skills for legged robots,” Sci. Robot., vol. 4, no. 26, 2019, Art. no. eaau5872.
[34] J. Hwangbo, J. Lee, and M. Hutter, “Per-contact iteration method for solving contact dynamics,” IEEE Robot. Automat. Lett., vol. 3, no. 2, pp. 895–902, Apr. 2018. [Online]. Available: www.raisim.com