Enhancing the Physical Capability of Aerial Robots: From Inspection to Manipulation

Guanqi He

CMU-RI-TR-25-31

April 25, 2025

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Guanya Shi
Sebastian Scherer
Wennie Tabib
Xiaofeng Guo

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my loved ones and amazing robot friends.

Abstract

The growing demand for high-altitude, complex interaction tasks has driven the evolution of Uncrewed Aerial Vehicles (UAVs) from passive perception platforms to systems capable of active physical interaction. This shift has catalyzed the development of Uncrewed Aerial Manipulators (UAMs), which integrate aerial mobility with manipulation capabilities to perform contact-rich tasks such as inspection, repair, and object handling in dynamic environments. In this thesis, we explore how to advance toward general aerial manipulation: enabling a single system to robustly execute a wide range of physical interaction tasks across diverse and challenging scenarios.

To address this challenge, we propose a unified aerial manipulation framework that consists of three key components: (i) a hybrid motion-force control module for dynamic and compliant physical interaction, (ii) a contact-aware trajectory planning module for safe and feasible motion generation under contact constraints, and (iii) an end-effector-centric control interface that separates platform-specific low-level control from high-level decision-making and policy learning. This modular framework aims to enhance system versatility and generality across different manipulation tasks and hardware platforms.

First, we develop a hybrid motion-force controller specifically tailored for aerial manipulation tasks. By explicitly regulating both motion and contact force during flight, this controller enables UAVs to interact compliantly and robustly with their environment, handling disturbances and uncertainties inherent in aerial interaction. This capability provides the foundation for executing complex tasks such as surface contact inspection and manipulation without compromising flight stability.

Second, we extend the scope of aerial manipulation beyond static contacts to dynamic surface interaction. We propose a contact-aware trajectory planner that generates motion plans accounting for varying contact forces and surface motions. Coupled with our hybrid controller, this allows the aerial manipulator to track both force and motion trajectories simultaneously, enabling dynamic tasks such as drawing, sliding, and surface exploration during flight.

Third, we design an end-effector-centric control interface that decouples low-level whole-body control from high-level task planning and policy learning. By focusing control objectives directly at the end-effector, our framework abstracts away platform-specific dynamics, allowing intuitive human teleoperation and facilitating the development of learning-based high-level policies. This design

supports cross-task and cross-platform generalization, paving the way for more scalable and modular aerial manipulation systems.

Through real-world experiments across a variety of tasks, we demonstrate that our unified framework improves the precision, robustness, and generality of aerial manipulation capabilities. We believe this work takes a significant step toward advancing autonomous aerial robots as capable and reliable physical agents in complex real-world environments.

Acknowledgments

I would like to express my heartfelt gratitude to my advisors, Guanya Shi and Sebastian Scherer. Guanya's profound expertise and deep insights into control and learning for robotics have significantly shaped my research over the past two years. Sebastian's extensive understanding of UAV systems and clear vision for impactful research directions have been instrumental in guiding my master's journey.

I am sincerely thankful to my committee members, Wennie Tabib and Xiaofeng Guo, for their invaluable feedback and guidance. Wennie provided tremendous support across various aspects of my projects, from robotic hardware assistance to insightful discussions on ideas. I am especially grateful to Xiaofeng Guo, who I have closely collaborated on numerous projects. Our extensive discussions and dedicated efforts turned many crazy ideas from dreams into reality. Collaborating with Xiaofeng significantly enhanced my research abilities and overall skills, for which I am deeply appreciative.

My deep gratitude also goes to all my collaborators: Jenny Geng, Mohammadreza Mousaei, Jiahe Xu, Luyi Tang, Yuanghang Zhang, Yogita Choudhary, Nikhil Sobanbabu, Tairan He, Chaoyi Pan, Zeji Yi, Wenli Xiao, Chong Zhang, Zhengyi Luo, and many others. I have learned immensely from each of them. A special thank you to Jenny Geng, who has served as a third advisor throughout more than three years of collaboration, consistently guiding my research efforts. I am also especially grateful for many valuable discussions with Tairan He, which inspired me to think about how to push the limits of robotics and reflect on what new capabilities our robots and algorithms can offer.

I would also like to thank my labmates at AirLab and LeCAR Lab for fostering such an incredibly supportive and inspiring lab environment. Our mutual support, both academically and personally, provided countless inspirations and helped me navigate many challenging times.

My thanks also extend to my friends for their constant encouragement, support, and care. I especially thank my girlfriend, Yanwen, whose support, understanding, and patience have provided immense emotional strength throughout this journey.

Finally, I wish to express my deepest gratitude to my parents for their support. Everything I have achieved stands on the foundation they have provided. I dedicate this thesis to them with all my appreciation.

Funding

This work was partly supported by Pennsylvania Infrastructure Technology Alliance (PITA) award and CMU Manufacturing Futures Institute (MFI).

Contents

1 Introduction **1**
1.1 Introduction 1

2 Aerial Interaction with Tactile Sensing **5**
2.1 Introduction 5
2.2 Related Works 5
2.2.1 Aerial Manipulation 5
2.2.2 Tactile Sensing for Manipulation 7
2.3 Methodology 8
2.3.1 The Design of Aerial Manipulator with Tactile Sensing 8
2.3.2 Vision-based Tactile Sensing 9
2.3.3 Sensor Fusion 10
2.3.4 Hybrid Motion-Force Control Based on Tactile Sensing 11
2.4 Experiments 14
2.4.1 Experiment Setup 14
2.4.2 Contact Force Estimation 14
2.4.3 Hybrid Motion-Force Controller Performance 14
2.4.4 Wall Texture Recognition 16
2.5 Conclusion 19

3 Contact-Aware Motion and Force Planning and Control for Aerial Manipulation **21**
3.1 Introduction 21
3.2 Related Works 23
3.3 Problem Formulation and Method Overview 24
3.3.1 Problem Definition 24
3.3.2 System and Pipeline Overview 24
3.3.3 Coordinate Frames and Notations 25
3.3.4 Aerial Manipulator Dynamics 25
3.4 Contact-aware Trajectory Planning 27
3.5 Hybrid Motion and Force Control 29
3.5.1 Contact Wrench Estimation 29
3.5.2 Hybrid Motion-Force Controller 30
3.6 End-Effector and Interface Designs 32

3.6.1 End-Effector Design . . . 32
3.6.2 Aerial Calligraphy Interface . . . 33
3.7 Experiments . . . 33
3.7.1 Experiment Setup . . . 33
3.7.2 Aerial Calligraphy . . . 34
3.7.3 Ablation Studies . . . 36
3.7.4 Speed Comparison . . . 38
3.7.5 Aerial Calligraphy with Touchscreen Interface . . . 38
3.8 Conclusion . . . 39

4 End-Effector-Centric Framework for Versatile Aerial Manipulation 41
4.1 Introduction . . . 41
4.2 Related Works . . . 43
4.2.1 Aerial Manipulation . . . 43
4.2.2 Mobile Manipulation Framework and EE-Centric Interface . . . 44
4.2.3 Teleportation and Imitation Learning . . . 45
4.3 System Overview . . . 46
4.4 Hardware Design . . . 46
4.4.1 Fully-Actuated UAV . . . 46
4.4.2 Manipulator . . . 47
4.4.3 Perception . . . 47
4.5 System Modeling . . . 48
4.5.1 Frames and Notation . . . 48
4.5.2 Fully-Actuated UAV Dynamics . . . 48
4.5.3 Manipulator Kinematics . . . 50
4.5.4 Manipulator Motor Delay . . . 51
4.6 End-Effector-Centric Whole-Body Control with Online Adaptation . . . 52
4.6.1 End-Effector-Centric Model Predictive Controller . . . 52
4.6.2 L1 Online Adaptation . . . 54
4.7 EE-Centric Teleoperation and Policy Learning . . . 55
4.7.1 EE-Centric Aerial Teleoperation . . . 55
4.7.2 EE-Centric Policy Learning . . . 56
4.8 Experiments . . . 56
4.8.1 Experimental Setup . . . 57
4.8.2 Implementation Details . . . 59
4.8.3 Experiment I: Control Performance . . . 59
4.8.4 Experiment II: Aerial Teleoperation . . . 63
4.8.5 Experiment III: Learning from Demonstration . . . 65
4.8.6 Discussion . . . 67
4.9 Conclusion . . . 69

5 Conclusions **71**

Bibliography **73**

List of Figures

2.1 **Fully-actuated UAV integrated with a vision-based tactile sensor:** The tactile sensor captures contact images for wall texture recognition and defect detection, and estimates contact forces, enabling the hybrid motion-force controller to achieve accurate force tracking during aerial interactions. . . 6

2.2 **System overview of aerial interaction with vision-based tactile sensing:** We develop a pipeline that integrates the GelSight tactile sensor with a fully-actuated UAV. The sensor estimates contact forces and recognizes wall textures during flight. The estimated force, fused with indirect measurements from the base force/torque sensor via a Kalman filter, enables accurate external force estimation for hybrid motion-force control. 8

2.3 **Comparison of hybrid motion-force controller performance under different sensing modalities:** The figure compares controller performance when using (a) only the base force/torque (F/T) sensor, (b) only the vision-based tactile sensor, and (c) sensor fusion. After establishing contact, the controller activates (blue-shaded region) and tracks a constant target contact force. 15

2.4 **Wall texture aerial inspection using a vision-based tactile sensor:** We test the recognition method on six textures: (a) no texture (printed paper), (b) wood grain, (c) marble mosaic, (d) quartz stone, (e) diamond vinyl, and (f) foam mat. The UAV makes contact with different textures in a single flight and predicts the surface type online in real-time. 17

2.5 **Confusion matrix for real-world texture recognition during aerial inspection:** This figure presents the confusion matrix for contact texture classification at each timestep during flight experiments. The system achieves an average classification accuracy of 93.4%. 18

3.1 **Aerial writing demonstration using the proposed system:** (a) The aerial manipulator writes a Chinese calligraphy character on the wall. (b) and (c) show additional examples of aerial writing with the words “AIR” and “2024,” respectively. 22

3.2 **Pipeline of the aerial writing system:** Users define target strokes either manually or via a touchscreen interface. The trajectory planning module then generates a dynamically feasible motion-force trajectory, which is tracked by a hybrid motion-force controller. A custom-designed end-effector pen enables the UAV to execute the drawing task. 23

3.3 **Coordinate frames used in the aerial writing tasks:** The frames $\mathcal{W}$, $\mathcal{B}$, $\mathcal{E}$, $\mathcal{S}$, and $\mathcal{C}$ represent the world, vehicle body, end-effector, arm, and contact frames, respectively. 25

3.4 **System overview of the aerial writing platform:** (a) Side view of the UAM system, (b) custom-designed sponge pen used as the end-effector, and (c) calibration curve showing the relationship between stroke linewidth and contact force. 32

3.5 **Performance evaluation over repeated trials of writing the letter "A":** Top row: Final written image. Middle two rows: Time history of end-effector position and contact force tracking performance. Bottom row: visual comparison between the written letter and ground truth image, evaluating the overall writing performance. 34

3.6 **Comparison with the control baseline during aerial writing:** Top: Writing results for letter 'R'. Bottom: Time history of end-effector motion and contact force tracking performance. 35

3.7 **Comparison with the planning baseline during aerial writing:** Top: Writing results for letter 'I'. Bottom: Time history of end-effector motion and contact force tracking performance. 37

3.8 **Effect of velocity profiles on aerial writing performance:** This figure shows the visual results of writing the letter "I" under different end-effector velocity profiles. 37

3.9 **End-to-end aerial writing from user input to execution:** (a) The user draws the target strokes "2024" on a touchscreen interface. (b) The planned trajectory with uniformly spaced timesteps, where denser green waypoints indicate slower planned velocities. (c) The final written result produced by the aerial manipulation system. 39

4.1 **Demonstration of versatile aerial manipulation capabilities:** The proposed framework enables robust execution of diverse tasks, including (a) writing "2025," (b) peg-in-hole insertion, (c) pick-and-place, and (d) light bulb replacement. 42

4.2 **Overview of the end-effector-centric aerial manipulation framework:** The system integrates a UAM platform, an end-effector-centric whole-body MPC, and a high-level policy module that includes both a teleoperation interface and an imitation learning framework using Action Chunk with Transformer (ACT) [89]. The high-level policy—whether from human input or a learned controller—outputs target end-effector states, which are tracked by the MPC to generate motor commands for the UAM. 43

4.3 **UAM hardware system design with key functional components:** The system consists of (1) a fully-actuated hexarotor base, (2) a 4-DoF manipulator, (3) Intel RealSense cameras for vision-based perception and feedback, and (4) an end-effector gripper for object interaction. The coordinate frames shown in the right diagram represent the reference axes used throughout the system. 48

4.4 **End-effector tracking performance on an figure-8 trajectory:** The tracking results show that the baseline without MPC exhibits significant lag, while the baseline without L1 adaptation suffers from static tracking errors caused by model mismatches. 61

4.5 **End-effector tracking error distribution across different methods:** The figure compares the tracking error distributions for three trajectory types using our proposed method and two baseline controllers, highlighting the improved precision of our approach. 61

4.6 **Ablation study on the effect of arm flexibility on MPC performance:** This figure compares the end-effector tracking results with and without adding arm motion penalty. Our method has smoother EE tracking trajectories, highlighting the role of arm flexibility in accurate tracking. 62

4.7 **Comparison of dynamic trajectory tracking performance across methods:** The figure shows tracking results for Figure-8 and Ellipse trajectories using three approaches. Our method consistently achieves the lowest end-effector tracking error in both tasks. 62

4.8 **Relation between arm motion and L1 disturbance estimation:** (a) Joint angles of the manipulator during interaction, and (b) external disturbance τ_{ext} and its L1-based estimation $\hat{\tau}_{\text{ext}}$ along the x, z, and θ_{pitch} axes, showing correlation with arm-induced dynamics. 63

4.9 **Comparison of end-effector command trajectories across teleoperation and policy interfaces:** (a) Command trajectories using the end-effector-centric and full-DoF teleoperation interfaces in a simulated peg-in-hole task. (b) Command trajectories of the learned autonomous policy over 50 test trials using both interfaces in a simulated pick-and-place task. 64

4.10 **Aerial teleoperation manipulation task demonstrations:** (1) Aerial Writing — the UAM writes “2025” on a whiteboard using a marker pen; (2) Rotate Valve — the UAM grasps a handle and completes one full rotation of a valve; (3) Pick-and-Place — the UAM picks up an object and places it in a target location. 64
4.11 **Long-horizon aerial teleoperation for light bulb replacement:** In the first flight, the UAM grasps and unscrews the existing light bulb. In the second flight, it inserts and screws in a new bulb, then presses a button to turn on the light, completing the task across two sequential operations. . . 65
4.12 **Task setup in MuJoCo simulation environment:** The figure illustrates four aerial manipulation tasks used for evaluation: (a) Peg-in-Hole, (b) Rotate the Valve, (c) Pick and Place, and (d) a long-horizon Open and Retrieve task. 66
4.13 **Autonomous aerial manipulation peg-in-hole policy experiment.** The UAM inserts the peg precisely, highlighting both the accuracy of the learned policy and the low-level controller. 68
4.14 **Task scenario randomization for peg-in-hole experiments:** The figure illustrates randomized configurations used during data collection and testing to evaluate the generalization capability of the learned aerial manipulation policy. 68

List of Tables

2.1 Comparison between F/T sensor and GelSight tactile sensor 7
2.2 Notation Definitions 11
2.3 Controller Performance Comparison 16

3.1 Ablation Studies Results 35

4.1 Notation Overview 49
4.2 Controller Parameters 58
4.3 System Identification Results 59
4.4 End-Effector Trajectory tracking performance 60
4.5 Imitation Learning Simulation Success Rate 67

Chapter 1

Introduction

1.1 Introduction

Uncrewed Aerial Vehicles (UAVs) have experienced rapid growth across various applications such as photography, agriculture, and package delivery [9, 25, 55]. Historically, UAV research and application have largely focused on passive observation and avoidance tasks, where the primary goal is to prevent collision with obstacles or other aerial agents. Despite these advancements, many high-altitude, contact-intensive tasks—such as maintenance of elevated structures, inspection of aircraft, and infrastructure repairs—still rely heavily on human intervention, which can be costly, inefficient, and hazardous [57, 66].

Aerial manipulation seeks to bridge this gap by enabling UAVs to perform physical interaction tasks, such as grasping, transporting, assembling, and disassembling objects at elevated locations [44, 49, 53]. Integrating manipulation capabilities with UAVs, thus forming Uncrewed Aerial Manipulators (UAMs), significantly expands their operational scope. However, this integration introduces complex challenges arising from the intricate coupling between aerial mobility and manipulation dynamics, environmental disturbances, and the inherent instability of aerial platforms [58, 64].

This thesis proposes a comprehensive and unified aerial manipulation framework to enhance the versatility, precision, and robustness of UAM systems across diverse tasks and scenarios. The proposed framework comprises three critical components: hybrid motion-force control, contact-aware trajectory planning, and an end-effector-centric control interface.

In Chapter 2, we address the challenge of dynamic and compliant physical interaction by developing a hybrid motion-force controller tailored explicitly for aerial manipulation tasks. Traditional force control approaches [10, 49] typically rely on heavy and expensive Force/Torque (F/T) sensors, which are difficult to integrate on lightweight mobile platforms and often suffer from placement-related latency and accuracy issues [17]. Recently, vision-based tactile sensing such as GelSight [84] has emerged as a promising alternative for fine-grained contact feedback in static settings [1, 41]. In this work, we are the first to deploy a vision-based tactile sensor on a fully-actuated UAV platform, enabling real-time contact state estimation and hybrid force-motion control in dynamic aerial scenarios. Our system supports precise interaction with external environments, facilitating tasks such as surface inspection and wall texture recognition.

In Chapter 3, we expand the aerial manipulation capability beyond static interactions to dynamic and continuous contact tasks. Most prior works either rely on discrete set-point-based contact force control [26, 49, 62], or focus on specific tasks such as cleaning [2] or defect detection [6]. These methods are often insufficient for tasks requiring continuous time-varying force and motion coordination, such as aerial drawing or sliding. To address this gap, we introduce a trajectory planning algorithm that explicitly considers contact dynamics, friction forces, and system feasibility. This capability is demonstrated through the aerial calligraphy task, where our planner and controller jointly enable continuous drawing with stroke thickness modulation, showcasing the system's ability to manage dynamically coupled force and motion tracking.

In Chapter 4, we focus on the modularity and generality of aerial manipulation systems through an end-effector-centric control strategy. Inspired by classic manipulation frameworks [43, 52], end-effector-centric control has gained traction for enabling embodiment-agnostic policy development [15, 46, 68]. In aerial settings, however, the floating base and coupled dynamics between the UAV and manipulator make direct application of such paradigms non-trivial. We propose an end-effector-centric Model Predictive Controller (MPC) for whole-body control, combined with an L1 adaptive component for robustness. Our framework includes a teleoperation system and a learning pipeline that leverages demonstration to train task policies. To our knowledge, this is the first end-effector-centric learning system deployed for real-world aerial manipulation, capable of executing diverse tasks including peg-in-hole insertion, writing, and object placement.

In summary, the main contributions of this thesis are:

- A vision-tactile-enhanced hybrid motion-force controller for dynamic aerial interaction.
- A contact-aware trajectory planner enabling continuous force-motion coordinated tasks such as aerial calligraphy.
- A versatile end-effector-centric whole-body control and policy interface for modular and scalable aerial manipulation.
- A unified framework validated through extensive real-world experiments across a variety of contact-rich aerial tasks.

We believe this work takes a significant step toward advancing general-purpose aerial manipulators by introducing a modular, task-agnostic, and scalable framework that enhances robustness and broadens the deployment capabilities of aerial robotic systems.

Chapter 2

Aerial Interaction with Tactile Sensing

2.1 Introduction

This chapter introduces the hybrid motion-force control system essential for precise and compliant aerial physical interactions. Aerial manipulation inherently requires the simultaneous and precise control of position and force to maintain stable interactions with the environment. Traditional aerial systems largely focus on position stabilization, lacking explicit force regulation capabilities necessary for tasks involving direct physical contact. To address this, we propose a hybrid motion-force controller designed specifically for aerial manipulators. Additionally, we integrate a vision-based tactile sensor (GelSight) directly at the manipulator's end-effector to enhance contact force feedback. This integration significantly improves force sensing accuracy and responsiveness, enabling robust interaction even in the presence of disturbances. Through real-world experiments, we validate our approach in practical tasks, such as precise force tracking and wall texture identification, demonstrating the system's effectiveness in handling complex aerial interaction scenarios.

2.2 Related Works

2.2.1 Aerial Manipulation

Aerial manipulation often requires tracking a reference force during contact with external objects to enable complex tasks such as aerial writing [48, 70] or pushing a target [10]. Prior

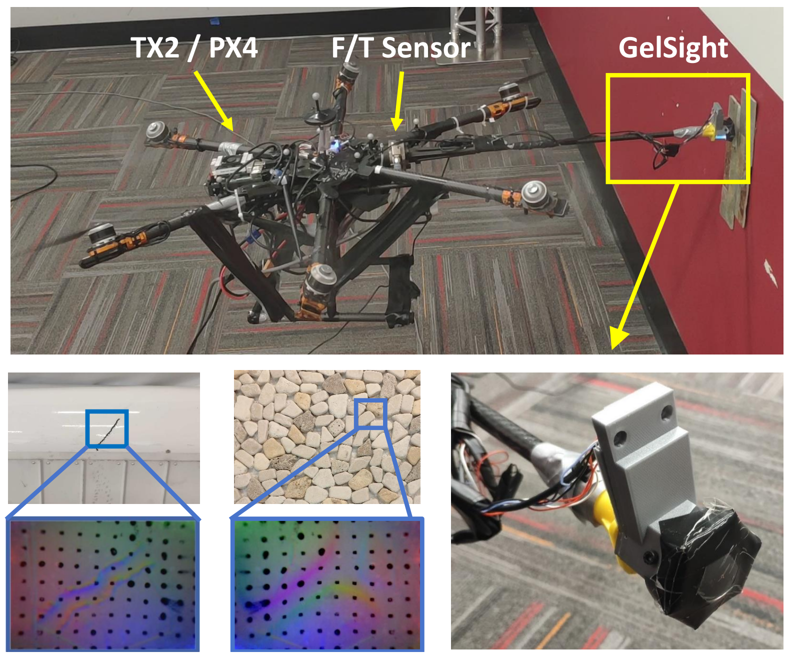

Figure 2.1: **Fully-actuated UAV integrated with a vision-based tactile sensor:** The tactile sensor captures contact images for wall texture recognition and defect detection, and estimates contact forces, enabling the hybrid motion-force controller to achieve accurate force tracking during aerial interactions.

Table 2.1: Comparison between F/T sensor and GelSight tactile sensor

Characteristic	F/T Sensor	GelSight
Weight [g]	255	60
Size $L \times W \times H$ [mm]	$75 \times 75 \times 33$	$65 \times 45 \times 45$
Cost [$]	5000	50
Frequency [Hz]	> 1000	30
Integration	Base (indirect contact)	Tip (direct contact)
Contact image	No	Yes

work includes Bodie et al.[6, 7], who used an F/T sensor and a time-of-flight camera on an omnidirectional aerial manipulator for contact inspection with an axis-selective impedance controller. Similarly, Cuniato et al. mounted an F/T sensor on the root of the end-effector and developed a power-based safety layer when pushing a moving cart [17]. Our previous research [30] integrated an F/T sensor on a fully actuated UAV and developed hybrid motion and force control with an image-based visual servo for wall painting tasks. Traditional methods for force feedback, such as force/torque or force sensing resistor sensors, have limitations like high cost, bulk, or restricted sensing capabilities. We break new ground by being the first to utilize vision-based tactile sensors on aerial manipulators, offering not only precise force measurements but also texture sensing.

2.2.2 Tactile Sensing for Manipulation

Tactile sensing, typically employed for contact force distribution, comes in various forms, such as tactile arrays and small F/T sensors, as well as multi-modal systems like BioTac [20]. Recently, vision-based tactile sensors like GelSight [84] and Digit [47] have advanced the field, particularly in perception and manipulation tasks. These sensors go beyond force distribution to capture contact images, proving useful in diverse perception applications such as slippage detection [82], hardness estimation [83], material identification [85], and object pose estimation [73], etc. It also facilitates various manipulation tasks, such as object grasping [41] and re-grasping [35], pushing [36], cable manipulation [63], etc. However, most of the current applications focus on tabletop settings with quasi-static processes. While a few researchers applied tactile sensing to dynamic scenarios, such as liquid property estimation [39], robot swing [75], etc., the application on mobile robots is rare. Our work brings it to aerial manipulation, a more dynamic context.

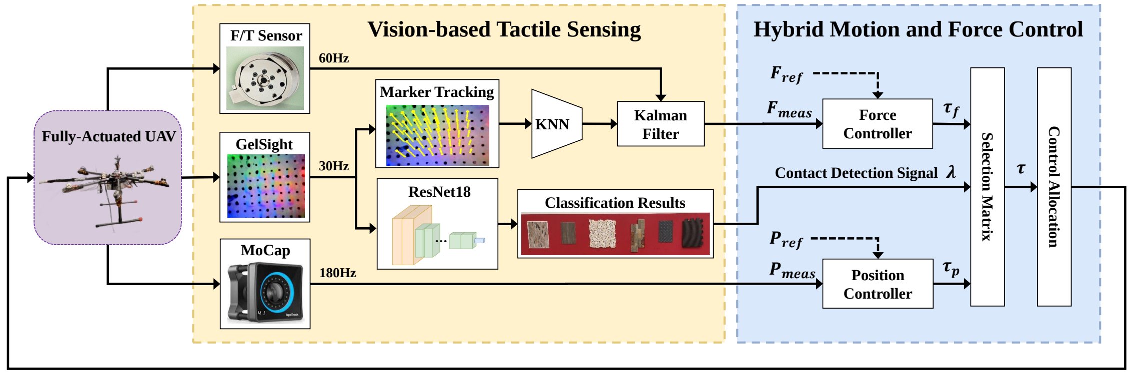

Figure 2.2: **System overview of aerial interaction with vision-based tactile sensing:** We develop a pipeline that integrates the GelSight tactile sensor with a fully-actuated UAV. The sensor estimates contact forces and recognizes wall textures during flight. The estimated force, fused with indirect measurements from the base force/torque sensor via a Kalman filter, enables accurate external force estimation for hybrid motion-force control.

2.3 Methodology

In this section, we introduce our fully actuated UAV system equipped with a vision-based tactile sensor and our method of tactile sensing-based aerial manipulation (Fig. 2.2).

2.3.1 The Design of Aerial Manipulator with Tactile Sensing

Aerial Manipulator Design

To complete the complex aerial manipulation task, we build a fully actuated UAV system based on a hexrotor model (Tarot 960), as shown in Fig. 2.1. Each rotor is tilted at a 30-degree angle, with alternating tilt directions, creating a fully-actuated configuration. We affixed a rigid carbon fiber rod to the front of the vehicle to serve as the manipulator. An ATI Gamma 6-axis F/T sensor is mounted at the base of the manipulator due to its weight (255 g) to balance the system center of mass. The flight control unit is a mRo Pixracer (FMUv4), equipped with a 180 MHz ARM Cortex® M4 processor, an inertial measurement unit (IMU), a barometer, and a gyroscope. Customized PX4 firmware is executed onboard to control the fully actuated platform. Additionally, onboard computation is handled by an Nvidia Jetson TX2, running the ROS Melodic system. For a more comprehensive description of our vehicle's architecture, please refer to our previous work [30, 42].

Onboard Tactile Sensor

In this work, we integrate a vision-based tactile sensor, GelSight, at the tip of the end-effector. GelSight [84] consists of three parts: a dome-shaped soft elastomer gel pad, three RGB LED arrays, and an embedded camera. The external surface of the gel pad is printed with a marker array spaced at 1.7 mm intervals, encompassing approximately 80 markers within the field of view. Upon contact with external objects, the elastomer deforms, reflecting the 3D shape and texture of the contact surface. Concurrently, the printed markers shift, indicating the local force exerted. The RGB LED arrays illuminate the deformed gel pad, and the embedded camera captures images, serving as tactile readings, at a resolution of 320×240 pixels resolution and a rate of 30 Hz. In contrast to traditional force/torque (F/T) sensors, GelSight offers advantages in terms of size, weight, cost, and spatial resolution as shown in Table 2.1 and Fig. 2.1. It doesn't require an extra data acquisition device, which most F/T sensor needs. Those features make it convenient to use on the UAV platform.

2.3.2 Vision-based Tactile Sensing

As previously highlighted, the GelSight sensor image contains two key elements: 1) the RGB image, which reveals the 3D texture of the contact surface, and 2) the printed markers on that surface, the motion of which indicates the local force distribution. In our pipeline, we leverage the motion of these markers to estimate the contact force while utilizing the complete RGB image to deduce the surface texture of the contact point.

Contact Force Estimation

Yuan et al. [84] demonstrated that a Convolutional Neural Network (CNN) can be employed to estimate the 6-axis contact force and torque using the raw RGB image. However, we note that it is computationally expensive and the robustness of the estimation is not sufficient during dynamic contact. Alternatively, we utilize the motion of markers to infer the contact force, as this correlates more directly with the deformation of the elastomer, thereby simplifying the relationship with the contact force and requiring less computation cost. We first track the motion of specific markers. We capture an initial, non-contact tactile image before takeoff to serve as a reference. Subsequently, we use optical flow to track the motion of the central 39 markers since they are always visible even with a large shear force. The two-dimensional motion of these markers forms a 78-dimensional feature vector. Utilizing the k-nearest neighbors algorithm (kNN), we estimate the contact force at each timestep based on this feature set. Training data, complete with labeled ground-truth forces, are used to facilitate this prediction, as further detailed in Section 2.4.2. The kNN algorithm identifies the k closest training points to the current marker motion and uses their ground-truth forces to estimate the present contact force. This non-parametric method, requiring less hyperparameter tuning than parametric methods like Multilayer Perceptrons (MLP), provides sufficient performance in our system.

Contact Texture Detection

We employ the full RGB tactile image from GelSight for contact texture recognition. Specifically, the system determines whether the sensor is in contact with an external surface and, if applicable, identifies its texture. This problem aligns with image classification tasks, treating non-contact as a unique texture category. To address this, we use the ResNet18 structure [32] and retrain the network on our dataset. The network takes GelSight images at each timestep as input and outputs the likelihood for each texture category. To enhance prediction robustness, we also incorporate temporal prediction data, assuming that contact textures change infrequently during contact. We define $p_{k,i}$ as the predicted likelihood of the i^{th} texture at timestep t_k, and $s_{k,i}$ as the cumulative likelihood for that texture up to t_k. The relationship between them is expressed as $s_{k+1,i} = f(s_{k,i}, p_{k,i})$, where we empirically define $f(s_{k,i}, p_{k,i}) = 5s_{k,i} + p_{k,i}$ in this work.

2.3.3 Sensor Fusion

While tactile sensing provides local measurements of force distribution and semantic information, the estimation results from the KNN algorithm may be subject to bias and noise due to sensor wear, aging, and deformation. Conversely, although F/T sensors located at the base of the manipulator measure force indirectly, they can provide the force measurements at a higher frequency compared to the GelSight sensor, which is limited to a sampling rate of 30 Hz. To leverage the strengths of both sensor types, we employ a standard Kalman filter to fuse the force readings from the tactile sensor $\boldsymbol{F}_{tac}$ and the force/torque sensor $\boldsymbol{F}_{ft}$ for more accurate force estimation.

Table 2.2: Notation Definitions

Symbol	Definition
$\mathcal{I}$	inertial (world) frame subscript
$\mathcal{B}$	body-fixed frame subscript
$\mathcal{E}$	end-effector frame subscript
$\mathcal{W}$	contact surface (wall) frame
$\boldsymbol{R}_B^A$	rotation in SO(3) from frame B to frame A

1. *State:* The state of the Kalman filter is defined as the contact force $\boldsymbol{F}_c$ exerted on the end-effector.
2. *Process Model:* We assume the UAV is applying a constant push force against the target surface, and express the process model as:

$$\dot{\boldsymbol{F}}_c = \mathbf{0}_3 \tag{2.1}$$

3. *Measurement Model:* since the force/torque sensor and the tactile sensor measure the contact force separately, the measurement equation is modeled as follows:

$$\begin{bmatrix} \boldsymbol{F}_{ft} \\ \boldsymbol{F}_{tac} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{3\times3} \\ \boldsymbol{I}_{3\times3} \end{bmatrix} \boldsymbol{F}_c + \begin{bmatrix} \boldsymbol{n}_{ft} \\ \boldsymbol{n}_{tac} \end{bmatrix} \tag{2.2}$$

where $\boldsymbol{n}_{ft}, \boldsymbol{n}_{tac} \in \mathbb{R}^3$ are the additive Gaussian white noise in the force/torque and tactile sensor measurements.

We follow the established Kalman filter to update the state and covariance estimation. The estimates are then provided into the hybrid motion-force controller for the computation of UAV's subsequent control commands.

2.3.4 Hybrid Motion-Force Control Based on Tactile Sensing

This section introduces the hybrid motion and force control algorithm for the fully-actuated UAV utilizing the estimation of contact force from sensor fusion as force feedback. We first define the notation and reference frames in Table 2.2.

Aerial Manipulator Dynamics

The dynamic model of the hexarotor aerial manipulator can be derived following our previous work [30]:

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{C}\boldsymbol{v} + \boldsymbol{G} = \boldsymbol{\tau}' + \boldsymbol{\tau_c} \tag{2.3}$$

with inertia matrix $\boldsymbol{M} \in \mathbb{R}^{6\times6}$, centrifugal and Coriolis term $\boldsymbol{C} \in \mathbb{R}^{6\times6}$, gravity wrench $\boldsymbol{G} \in \mathbb{R}^{6}$, control wrench $\boldsymbol{\tau}' \in \mathbb{R}^{6}$, contact wrench $\boldsymbol{\tau}_c \in \mathbb{R}^{6}$ and system twist (linear and angular velocity) $\boldsymbol{v} = \begin{bmatrix}\mathbf{V}^\top & \boldsymbol{\Omega}^\top\end{bmatrix}^\top \in \mathbb{R}^6$ expressed in the body frame.

$$\begin{aligned}
\boldsymbol{M} &= \mathrm{diag}\left(\begin{bmatrix} m\mathbf{I}_{3\times3} & \boldsymbol{J}\end{bmatrix}\right) \\
\boldsymbol{C} &= \mathrm{diag}\left(\begin{bmatrix} m[\boldsymbol{\Omega}]_\times & -\boldsymbol{J}[\boldsymbol{\Omega}]_\times\end{bmatrix}\right) \\
\boldsymbol{G} &= m\mathrm{diag}\left(\begin{bmatrix} \mathbf{R}_{\mathcal{I}}^{\mathcal{B}} & \mathbf{0}_{3\times3}\end{bmatrix}\right)\boldsymbol{g}
\end{aligned} \tag{2.4}$$

where m is the vehicle mass, $\mathbf{J}$ the moment of inertia, $\mathbf{g}$ the gravity. $[*]_\times$ denotes the skew-symmetric matrix associated with vector $*$.

Since the system is fully-actuated, we apply feedback linearization, and the compensated wrench input can be computed as $\boldsymbol{\tau}' = \boldsymbol{C}v - \boldsymbol{G} + \boldsymbol{\tau}$. The dynamics of the system (2.3) becomes

$$\boldsymbol{M}\dot{\boldsymbol{v}} = \boldsymbol{\tau} + \boldsymbol{\tau}_c \tag{2.5}$$

Hybrid Motion and Force Control

Leveraging the advantages of the fully-actuated UAV, the vehicle is capable of controlling its translation and orientation independently, which is different from the coplanar multi-rotors that only generate thrust normal to its rotor plane, leading to completely tilt towards the desired direction. We implement a hybrid motion and force controller in [30], which combines the output of a motion tracking controller $\boldsymbol{\tau}_p$ and a wrench tracking controller $\boldsymbol{\tau}_f$

as follows:

$$\boldsymbol{\tau} = (\mathbf{I}_{6\times 6} - \boldsymbol{\Lambda})\boldsymbol{\tau}_p + \boldsymbol{\Lambda}\boldsymbol{\tau}_f \tag{2.6}$$

$$\boldsymbol{\Lambda} = \text{blockdiag}(\boldsymbol{\Lambda}', \mathbf{0}_{3\times 3}) \in \mathbb{R}^{6\times 6} \tag{2.7}$$

$$\boldsymbol{\Lambda}' = \mathbf{R}_{\mathcal{E}}^{\mathcal{B}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \lambda \end{bmatrix} \tag{2.8}$$

where $\mathbf{R}_{\mathcal{E}}^{\mathcal{B}}$ denotes the rotation matrix from end-effector frame to body frame. Intuitively, the matrix $\boldsymbol{\Lambda}$ selects direct force control commands, and leaves the complementary subspace for the motion control. $\lambda \in \{0, 1\}$ is the force-motion control switch, where $\lambda = 0$ corresponds to the pure motion control, $\lambda = 1$ indicates the force control along the normal direction of the surface. The wrench controller can be activated by tactile contact detection in Sec. 2.3.2.

To alleviate the oscillation and absorb the pushing energy efficiently, we designed an impedance force control scheme to ensure the end-effector can track reference force $\boldsymbol{F}_{ref}$. The interaction force $\boldsymbol{F}_{meas}$ acting on the UAV can be measurements from tactile sensors, measurements from force torque sensors, or output of the Kalman filter. The force-tracking control action can be obtained as

$$\begin{aligned} \boldsymbol{e}_f &= \boldsymbol{F}_{meas} - \boldsymbol{F}_{ref} \\ \boldsymbol{F}_f &= \boldsymbol{F}_{ref} - \boldsymbol{M}_{\mathbf{0}}\boldsymbol{M}_d^{-1}(\boldsymbol{K}_s\boldsymbol{e}_s + \boldsymbol{D}_s\dot{\boldsymbol{e}}_s) + \boldsymbol{K}_{f,p}\boldsymbol{e}_f + \boldsymbol{K}_{f,i}\int \boldsymbol{e}_f dt \\ &= \boldsymbol{F}_{ref} + \boldsymbol{K}_{s,p}\boldsymbol{e}_s + \boldsymbol{K}_{s,d}\dot{\boldsymbol{e}}_s + \boldsymbol{K}_{f,p}\boldsymbol{e}_f + \boldsymbol{K}_{f,i}\int \boldsymbol{e}_f dt \end{aligned} \tag{2.9}$$

with $\boldsymbol{e}_s$ the error between current and operating position, $\boldsymbol{K}_{f,p}$, $\boldsymbol{K}_{f,i} > 0$ the positive definite tunable gain matrices, $\boldsymbol{M}_0$, $\boldsymbol{M}_d$ the actual and desired vehicle mass, respectively. $\boldsymbol{K}_{s,p} = \boldsymbol{M}_0\boldsymbol{M}_d^{-1}\boldsymbol{K}_s$ is the normalized stiffness. $\boldsymbol{K}_{s,d} = \boldsymbol{M}_0\boldsymbol{M}_d^{-1}\boldsymbol{D}_s$ is the normalized damping. Then, the control wrench $\boldsymbol{\tau}_f = \begin{bmatrix} \boldsymbol{u}_f & \mathbf{0}_3 \end{bmatrix}^\top \in \mathbb{R}^6$, where $\boldsymbol{u}_f = \boldsymbol{R}_{\mathcal{W}}^{\mathcal{B}}\boldsymbol{F}_f$ will be passed through Equation (2.8).

As for the motion control, the classical geometric multirotor controller is used to obtain the control wrench $\boldsymbol{\tau}_p$ [42]. Then, the total control wrench $\boldsymbol{\tau}$ is mapped to the rotor speeds of the fully-actuated UAV through the control allocation process.

2.4 Experiments

In this section, we conduct multiple experiments to show the benefit of using the vision-based tactile sensor on the UAV for aerial interaction tasks. Specifically, we perform experiments on force tracking using the hybrid force-motion control, and real-time wall texture recognition on our aerial manipulator based on the tactile feedback.

2.4.1 Experiment Setup

We test our developed system and the algorithm pipeline in the real world. An optitrack Motion capture system is used for localization. For the force tracking and texture recognition experiments, the aerial manipulator is directed to fly against a vertical wall located within the motion capture space, as shown in Fig. 2.1. Repeated experiments were conducted for generalizability.

2.4.2 Contact Force Estimation

We leverage the motion of GelSight markers in each frame to infer contact force. To obtain better calibration and training data, we initially gather data in a controlled tabletop calibration setup and later assess the control performance based on tactile force estimates in flight. In the calibration setup, the GelSight sensor is mounted alongside an F/T sensor, similar to [84]. We manually apply varying forces against the sensor using different surfaces. Both the F/T sensor readings and GelSight marker motions are recorded as training data, accumulating a total of 10,000 data points. We also randomly collect 1,300 data points for model evaluation. During experiments, we utilize these data points and employ a kNN method to estimate contact force, where we set $K = 50$. The contact force estimation Root Mean Square Error (RMSE) is around 0.91 N, which is in a similar error to the work using CNN [84].

2.4.3 Hybrid Motion-Force Controller Performance

In this experiment, we evaluate the performance of our hybrid force-motion controller, which utilizes tactile feedback for force tracking. Upon takeoff, the vehicle is directed to approach a wall and sustain a contact force of 5 N upon making contact. We compare

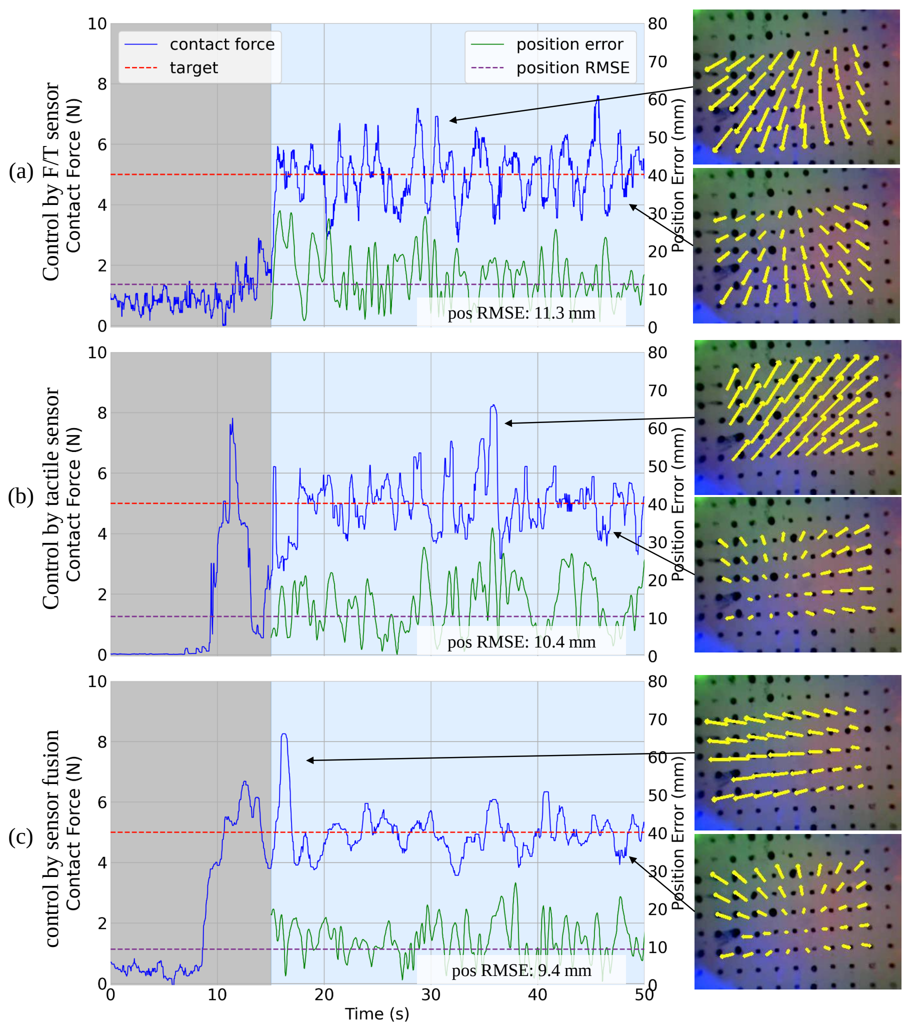

Figure 2.3: **Comparison of hybrid motion-force controller performance under different sensing modalities:** The figure compares controller performance when using (a) only the base force/torque (F/T) sensor, (b) only the vision-based tactile sensor, and (c) sensor fusion. After establishing contact, the controller activates (blue-shaded region) and tracks a constant target contact force.

Table 2.3: Controller Performance Comparison

controller / metrics	Use only F/T sensor	Use only tactile sensor	Use sensor fusion
Force RMSE (N)	0.85	0.95	0.74
Force Overshoot (N)	2.73	3.26	2.95
Force Undershoot (N)	2.76	2.69	1.49
Pos RMSE (mm)	11.28	10.43	**9.42**
Pos Std Dev (mm)	10.99	10.21	**8.46**

performance across three scenarios: (1) relying solely on F/T sensor readings, (2) using only vision-based tactile sensor measurements, and (3) employing sensor fusion that combines both types of measurements.

The controller's performance metrics are outlined in Fig. 2.3 and Table 2.3. Upon making contact with the wall, the hybrid motion-force controller is engaged. We observed that all three test scenarios achieved the target contact force with near-zero steady-state error. Scenarios relying solely on either the F/T sensor or the vision-based tactile sensor showed similar ranges of error variance. However, the scenario employing fused force state estimates exhibited reduced variance. All scenarios experienced some degree of overshoot and undershoot. A closer examination of the corresponding contact images and 6-axis F/T feedback revealed that large shear forces are present during overshoot periods. This implies that force/torque control along other axes may require further optimization.

Given the absence of ground truth for contact force (all sensors introduce some level of measurement error), we also use flight performance during force tracking as an indirect yet meaningful metric. As depicted in Fig. 2.3 and Table 2.3, we specifically examine position tracking error. Notably, using fused force data leads to about a 16% improvement in key performance indicators like RMSE, and standard deviation, compared to only relying on the F/T sensor. In summary, the results strongly indicate that the vision-based tactile sensor can effectively serve as a replacement for, or complement to, traditional F/T sensors, yielding comparable or enhanced flight performance.

2.4.4 Wall Texture Recognition

In this experiment, we evaluate the system's ability to identify wall textures during flight and contact. As shown in Fig. 2.4, we focus on identifying five specific textures: 1) wood

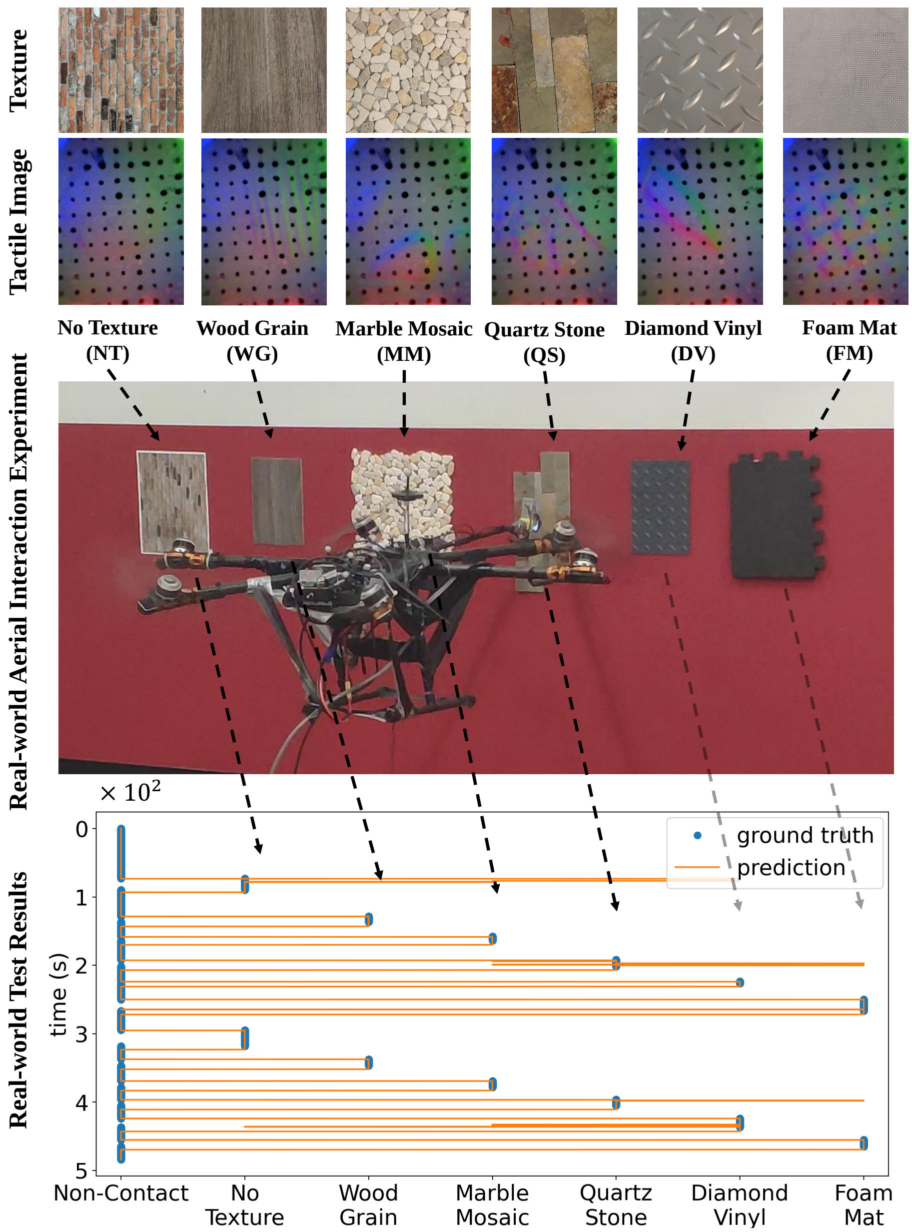

Figure 2.4: **Wall texture aerial inspection using a vision-based tactile sensor:** We test the recognition method on six textures: (a) no texture (printed paper), (b) wood grain, (c) marble mosaic, (d) quartz stone, (e) diamond vinyl, and (f) foam mat. The UAV makes contact with different textures in a single flight and predicts the surface type online in real-time.

Contact Texture Ground Truth \ Contact Texture Prediction	No Texture (NT)	Wooden Grain (WG)	Marble Mosaic (MM)	Quartz Stone (QS)	Diamond Vinyl (DV)	Foam Mat (FM)
NT	88.44%	0.00%	3.66%	0.00%	7.91%	0.00%
WG	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%
MM	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%
QS	0.00%	0.00%	0.97%	92.40%	0.00%	6.63%
DV	2.42%	0.00%	15.38%	0.00%	82.20%	0.00%
FM	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%

Figure 2.5: **Confusion matrix for real-world texture recognition during aerial inspection:** This figure presents the confusion matrix for contact texture classification at each timestep during flight experiments. The system achieves an average classification accuracy of 93.4%.

grain, 2) marble mosaic, 3) quartz stone, 4) diamond vinyl, and 5) foam mat. Additionally, we include a sixth texture: a printed imitation of a brick pattern on flat paper. This mimics complex 3D textures while actually being a flat surface, akin to many wallpapers, highlighting the need for tactile feedback beyond simple visual inspection for accurate texture and material perception. We also consider the absence of contact as an additional texture category. To build a robust training dataset, we manually collected samples by pressing the GelSight tactile sensor against these various textures. Each texture contributed 3,000 data points to the training set, which includes all frames where contact was made.

During the flight tests, the aerial manipulator is flown towards various textures affixed to a wall, making contact with approximately 5 N of force for at least 10 seconds. This duration allows the trained classifier to identify and categorize the texture in contact. The manipulator then proceeds to the next target texture. Each texture is engaged twice in this manner. The entire procedure is executed in a continuous flight, without landing the UAV in between tests.

The prediction result is shown in Fig. 2.4 and Fig. 2.5. We note that most of the time, the prediction is accurate with an average accuracy of 93.4%. There are brief inconsistencies in predicting certain textures such as the diamond vinyl, where the sensor momentarily confuses them with marble mosaic texture. These deviations are understandable given the textural similarities with close features. Importantly, the predictions stabilize and self-correct over time, ultimately achieving a post-contact accuracy of 100% for all textures. These results demonstrate that the vision-based tactile sensor can effectively identify 3D textures during aerial contact, suggesting its potential utility in defect detection tasks, such as identifying wing cracks, as depicted in Fig. 2.1.

2.5 Conclusion

This paper develops a fully-actuated aerial platform equipped with a vision-based tactile sensor for aerial manipulation tasks. We design a pipeline that integrates tactile feedback into a hybrid motion-force controller for real-time force tracking during aerial physical interaction. Additionally, we developed an algorithm for wall texture identification based on tactile measurement. Real-world experimental results indicate that our system can either supplement or replace traditional force/torque sensors while maintaining or improving control performance. Utilizing a fused force-state estimate, we achieve a 16% improvement in position tracking error compared to single-sensor configurations. Furthermore, our tactile sensing method demonstrates a 93.4% accuracy rate in real-time texture classification, rising to 100% post-contact. These results highlight the value of incorporating vision-based tactile sensors in aerial manipulation tasks. Future directions include optimizing sensor design and devising a controller that leverages high-dimensional tactile data instead of low-dimensional estimated force/torque values.

Chapter 3

Contact-Aware Motion and Force Planning and Control for Aerial Manipulation

3.1 Introduction

Building upon the control foundations established in Chapter 2, this chapter expands aerial manipulation capabilities to dynamic and continuous surface interactions. Many real-world tasks, such as cleaning, inspection, and aerial drawing, require the continuous and simultaneous regulation of motion and contact forces. Traditional aerial manipulation strategies primarily handle discrete or static contacts, leaving the dynamic scenarios largely unaddressed. To bridge this gap, we propose a novel contact-aware trajectory planning method capable of generating smooth, dynamically feasible trajectories that explicitly consider variable contact constraints. Using aerial calligraphy as a representative application, we demonstrate the capability of our approach to precisely and continuously track time-varying force and motion references, showcasing its suitability for complex dynamic surface tasks.

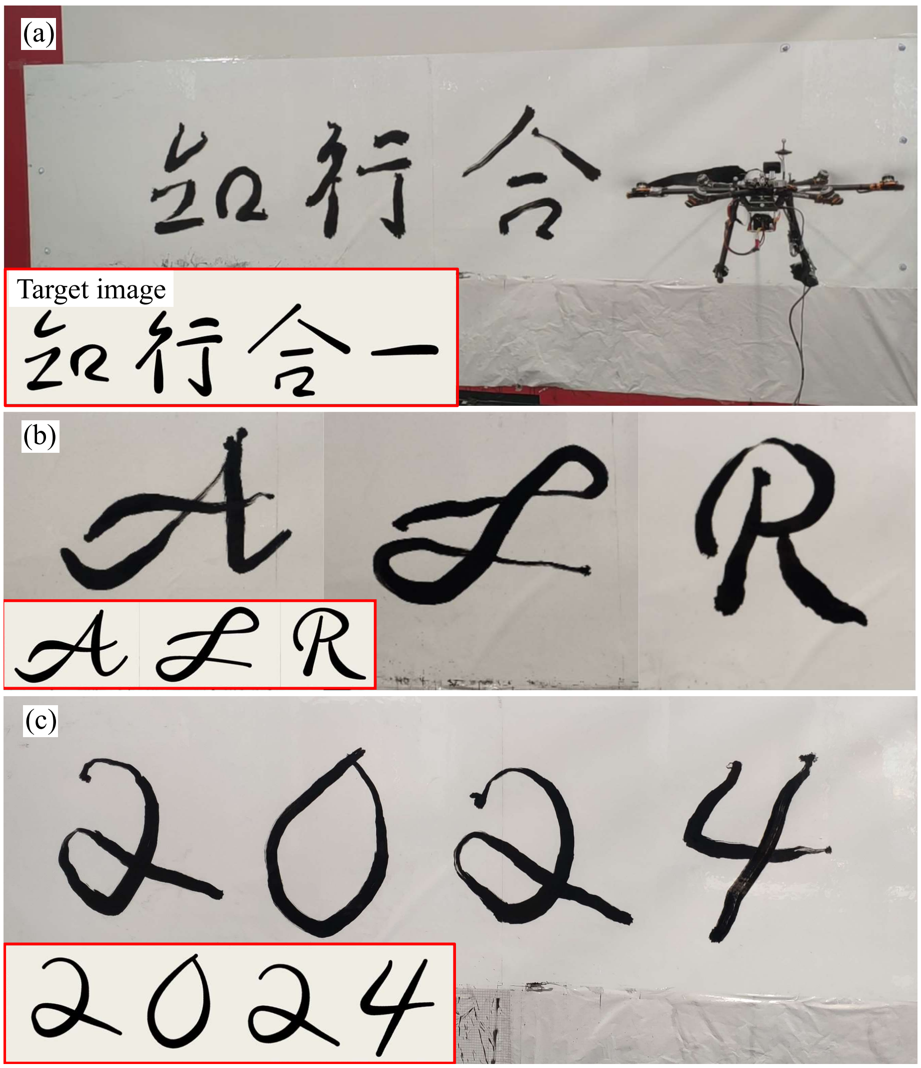

Figure 3.1: **Aerial writing demonstration using the proposed system:** (a) The aerial manipulator writes a Chinese calligraphy character on the wall. (b) and (c) show additional examples of aerial writing with the words “AIR” and “2024,” respectively.

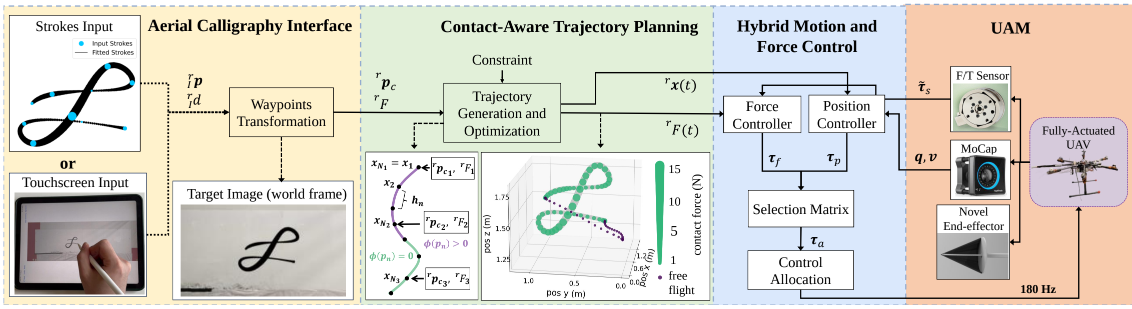

Figure 3.2: **Pipeline of the aerial writing system:** Users define target strokes either manually or via a touchscreen interface. The trajectory planning module then generates a dynamically feasible motion-force trajectory, which is tracked by a hybrid motion-force controller. A custom-designed end-effector pen enables the UAV to execute the drawing task.

3.2 Related Works

In aerial manipulation tasks, UAMs often need to track reference contact force or motion trajectories during contact with external objects. To generate the reference trajectory, the existing works mainly focused on motion planning and treated the interaction force between the UAM and the environment as disturbances [21, 50, 77]. As for contact force estimation, researchers have developed contact mechanics models, such as quasi-static or spring-damper models [10, 87]. Some studies also mounted F/T sensors on UAMs to directly measure the contact force for feedback [7, 59].

Aerial writing, which requires the UAM to control both motion and contact force during the task, is one common task the researcher focuses on. Tzoumanikas et al. [70] developed a hybrid motion-force MPC as the trajectory generator and controller to draw letters with constant linewidth. They obtained the contact force using a linear spring model for vehicle dynamics. Lanegger et al. [48] designed a compliant Gough-Stewart mechanism for line drawing, improving stability and precision with multiple contact points. They equipped three omni-wheels with the end-effector and modeled the whole structure as a three-wheeled omnidirectional ground robot for precise control.

However, among all these works, most of them either track only a reference motion trajectory [59, 70], a set of discrete constant force targets at different locations [7], or time-varying force but without motion [12]. Our goal is to achieve general time-varying contact force and motion tracking to enable various aerial manipulation tasks, such as aerial calligraphy with strokes of varying linewidths. Unlike many studies that consider only normal contact forces, we utilize the full contact model in this work to predict the contact wrench (both normal force and friction force). This model is used in both trajectory planning to generate dynamically feasible trajectories and in hybrid motion-force control with contact wrench compensation. Additionally, our method treats execution time also as a decision variable to be optimized in trajectory optimization [78], improving UAM task efficiency compared to algorithms that predefine the execution time [21, 65].

3.3 Problem Formulation and Method Overview

3.3.1 Problem Definition

This work aims to simultaneously track motion and contact force for UAM during aerial interaction. The problem is defined as follows: control the UAM's end-effector to navigate through a sequence of contact points on a surface with a known shape, while applying the desired normal contact force at each contact point. Specifically, we focus on the aerial calligraphy task. Given a list of waypoints in the image space, each specifying the position and linewidth of a point, the goal is to control the UAM to accurately draw the letter.

3.3.2 System and Pipeline Overview

Our UAM platform is adopted from our previous works [26], which is a fully-actuated hexarotor equipped with a one-link manipulator. An F/T sensor is mounted at the base of the manipulator. The flying calligrapher system pipeline is shown in Fig. 3.2 and consists of four components. First, we develop an aerial calligraphy interface allowing users to specify the target image by either manually defining waypoints or drawing the letter through a touchscreen interface. This module then converts the user input to sparse waypoints in the task space, including the contact point position and magnitude of the normal contact force. Second, based on the sparse waypoints, we develop a contact-aware trajectory planner that generates a dynamically feasible motion-force trajectory for the UAM to track. Subsequently, a hybrid motion-force controller is designed so that the UAM can follow the reference trajectory. Finally, we design a novel end-effector for the UAV to draw the target letters.

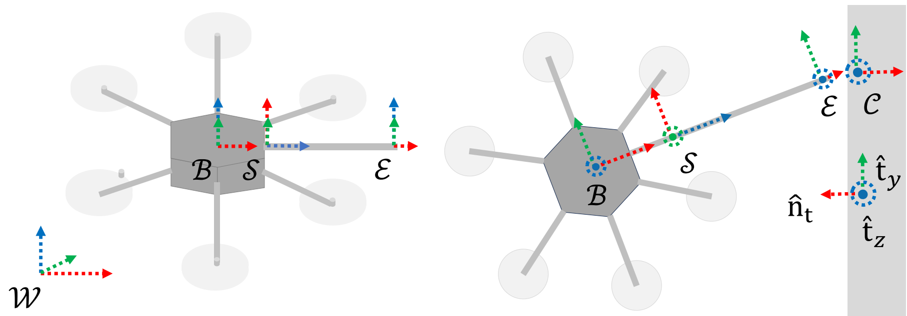

Figure 3.3: **Coordinate frames used in the aerial writing tasks:** The frames $\mathcal{W}$, $\mathcal{B}$, $\mathcal{E}$, $\mathcal{S}$, and $\mathcal{C}$ represent the world, vehicle body, end-effector, arm, and contact frames, respectively.

3.3.3 Coordinate Frames and Notations

We define several coordinate frames, as shown in Fig. 3.3. The world inertial frame $\mathcal{W}$ is an east-north-up frame with the origin at the UAV's takeoff position. The body frame $\mathcal{B}$ originates at the vehicle's center of mass, with axes pointing front, left, and up relative to the vehicle. The end-effector frame $\mathcal{E}$ originates at the end-effector, with axes pointing front, left, and up relative to the vehicle. The arm frame $\mathcal{S}$ originates at the root of the manipulator where the F/T sensor is mounted, with axes aligned to the sensor. The contact frame $\mathcal{C}$ originates at the contact point, with axes aligned to the surface normal (towards inside), left, and up directions of the surface.

Vectors are denoted as bold lowercase symbols. Left-hand subscripts, e.g., ${}_{\mathcal{W}}\boldsymbol{v}$, indicate the coordinate representation in frame $\mathcal{W}$. $\boldsymbol{R}_a^b \in \mathbb{R}^{3\times3}$, $\boldsymbol{T}_a^b \in \mathbb{R}^{3\times1}$, $\boldsymbol{Ad}_{\boldsymbol{T}_a^b} \in \mathbb{R}^{6\times6}$ define the rotation, translation, and wrench (force and torque) transformation for vector from frame b to frame a, respectively [51]. We have: ${}_a\boldsymbol{v} = \boldsymbol{R}_a^b{}_b\boldsymbol{v}$. $\mathbf{0}_{a\times b}$ defines a $\mathbf{0}$ vector in $a \times b$ dimensions. $\boldsymbol{v}_{[i]}$ indicates the i^{th} element of $\boldsymbol{v}$; ${}^r(\cdot)$ denotes the reference quantity.

3.3.4 Aerial Manipulator Dynamics

We define the UAM end-effector position vector in the world frame as ${}_{\mathcal{W}}\boldsymbol{p} = [x, y, z]^\top$, the attitude with rotation matrix representation $\boldsymbol{R}_{\mathcal{W}}^{\mathcal{E}}$ and the angular velocity in the end-effector frame as $\boldsymbol{\omega}$. The pose and twist of the UAM end-effector are defined as $\boldsymbol{q} = [{}_{\mathcal{W}}\boldsymbol{p}, \boldsymbol{R}_{\mathcal{W}}^{\mathcal{E}}]$ and $\boldsymbol{v} = [{}_{\mathcal{W}}\dot{\boldsymbol{p}}^\top, \boldsymbol{\omega}^\top]^\top$.

The dynamic model of the hexarotor aerial manipulator at the end-effector can be derived following our previous work [30].

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{C}\boldsymbol{v} = \boldsymbol{Ad}_{\boldsymbol{T}_{\mathcal{E}}^{\mathcal{B}}}\,{}_{\mathcal{B}}\boldsymbol{\tau}_a + \boldsymbol{Ad}_{\boldsymbol{T}_{\mathcal{E}}^{\mathcal{C}}}\,{}_{\mathcal{C}}\boldsymbol{\tau}_c + \boldsymbol{Ad}_{\boldsymbol{T}_{\mathcal{E}}^{\mathcal{B}}}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{W}}\,{}_{\mathcal{W}}\boldsymbol{g} \tag{3.1}$$

with inertia matrix $\boldsymbol{M} \in \mathbb{R}^{6\times6}$, centrifugal and Coriolis term $\boldsymbol{C} \in \mathbb{R}^{6\times6}$, gravity wrench $\boldsymbol{g} \in \mathbb{R}^6$, contact wrench $\boldsymbol{\tau}_c \in \mathbb{R}^6$, and control wrench $\boldsymbol{\tau}_a = [\boldsymbol{f}_a^\top, \boldsymbol{m}_a^\top]^\top \in \mathbb{R}^6$ from the UAM actuators, where $\boldsymbol{f}_a \in \mathbb{R}^3$ denotes the control force and $\boldsymbol{m}_a \in \mathbb{R}^3$ denotes the torque. Specifically, we have

$$\boldsymbol{M} = \begin{bmatrix} m\boldsymbol{R}_{\mathcal{E}}^{\mathcal{W}} & -m[{}_{\mathcal{E}}\boldsymbol{t}_{\mathcal{B}}^{\mathcal{E}}]_\times \\ m[{}_{\mathcal{E}}\boldsymbol{t}_{\mathcal{B}}^{\mathcal{E}}]_\times\boldsymbol{R}_{\mathcal{E}}^{\mathcal{W}} & \boldsymbol{R}_{\mathcal{E}}^{\mathcal{B}}\boldsymbol{J}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{E}} - m[{}_{\mathcal{E}}\boldsymbol{t}_{\mathcal{B}}^{\mathcal{E}}]_\times[{}_{\mathcal{E}}\boldsymbol{t}_{\mathcal{B}}^{\mathcal{E}}]_\times \end{bmatrix} \tag{3.2}$$

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & -m[\boldsymbol{\omega}]_\times[{}_{\mathcal{E}}\boldsymbol{t}_{\mathcal{B}}^{\mathcal{E}}]_\times \\ \boldsymbol{0}_{3\times3} & [\boldsymbol{\omega}]_\times\boldsymbol{R}_{\mathcal{E}}^{\mathcal{B}}\boldsymbol{J}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{E}} - m[{}_{\mathcal{E}}\boldsymbol{t}_{\mathcal{B}}^{\mathcal{E}}]_\times[\boldsymbol{\omega}]_\times[{}_{\mathcal{E}}\boldsymbol{t}_{\mathcal{B}}^{\mathcal{E}}]_\times \end{bmatrix} \tag{3.3}$$

Here m is the vehicle mass; $\boldsymbol{J}$ is the moment of inertia at the vehicle center of mass in the body frame; ${}_{\mathcal{E}}\boldsymbol{t}_{\mathcal{B}}^{\mathcal{E}}$ is the relative translation pointing from the tip of the end-effector to the vehicle body under the end-effector frame, and $[*]_\times$ is the skew-symmetric matrix associated with vector $*$.

The control wrench comes from the standard UAV rotor thrust and torque. As for the contact wrench, since the one-link manipulator is rigidly attached to the UAV contacting the surface through a point, only reaction force is considered at the contact point without torque, i.e.

$${}_{\mathcal{C}}\boldsymbol{\tau}_c = [\boldsymbol{f}_c^\top, \boldsymbol{0}_{3\times1}^\top]^\top, \quad \boldsymbol{f}_c = \boldsymbol{f}_\perp + \boldsymbol{f}_{/\!/} \tag{3.4}$$

where $\boldsymbol{f}_\perp$ and $\boldsymbol{f}_{/\!/}$ represent the normal force and shear force respectively. Given the contact surface ϕ and the assumption that the end-effector makes a point-contact with the surface, we have the contact constraint $\phi(\boldsymbol{p}) = 0$. The contact axes can be represented by the surface normal vector $\hat{\boldsymbol{n}}_t$, and the in-plane basis vectors $\hat{\boldsymbol{t}}_y - \hat{\boldsymbol{t}}_z$ with

$$\hat{\boldsymbol{n}}_t = \frac{\nabla\phi(\boldsymbol{p})}{\|\nabla\phi(\boldsymbol{p})\|}, \hat{\boldsymbol{t}}_y = \frac{\boldsymbol{g}_{[1:3]} \times \hat{\boldsymbol{n}}_t}{\|\boldsymbol{g}_{[1:3]} \times \hat{\boldsymbol{n}}_t\|}, \hat{\boldsymbol{t}}_z = \frac{\hat{\boldsymbol{t}}_y \times \hat{\boldsymbol{n}}_t}{\|\hat{\boldsymbol{t}}_y \times \hat{\boldsymbol{n}}_t\|} \tag{3.5}$$

Therefore the contact force can be expressed as

$$\begin{aligned}
F &= \|\boldsymbol{f}_\perp\|, \quad \boldsymbol{f}_\perp = F\hat{\boldsymbol{n}}_t \\
\boldsymbol{f}_{/\!/} &= -\mu F \left(\text{sign}(\dot{y})\hat{\boldsymbol{t}}_y + \text{sign}(\dot{z})\hat{\boldsymbol{t}}_z\right) \\
\boldsymbol{\alpha}(\boldsymbol{v}) &= \left[[\hat{\boldsymbol{n}}_t - \mu\text{sign}(\dot{y})\hat{\boldsymbol{t}}_y - \mu\text{sign}(\dot{z})\hat{\boldsymbol{t}}_z]^\top, \boldsymbol{0}_{3\times1}^\top\right]^\top \\
{}_{\mathcal{C}}\boldsymbol{\tau}_c &= F\boldsymbol{\alpha}(\boldsymbol{v})
\end{aligned} \tag{3.6}$$

where F is the magnitude of the normal force. Here, we apply the Coulomb model to estimate the friction force with the friction coefficient μ. The negative sign at the beginning represents that the friction is in the opposite direction of the motion and sign$(\cdot)$ denotes the sign function.

3.4 Contact-aware Trajectory Planning

This section introduces the proposed contact-aware trajectory planning algorithm. It takes M sparse waypoints $({}^r\boldsymbol{p}_{c_i}, {}^rF_i)$, for $i \in 1, \cdots, M$ and contact surface ϕ as input, and outputs the state and action trajectories $\boldsymbol{x}(t)$, $\boldsymbol{\tau}_a(t)$ as the reference for the hybrid motion-force controller, where $\boldsymbol{x}(t) = [\boldsymbol{q}^\top(t), \boldsymbol{v}^\top(t), F(t)]^\top$, $t \in (0, T)$. The waypoints can include both contact points where ${}^rF_i > 0$ and non-contact points where ${}^rF_i = 0$. Notice that our algorithm also optimizes the total time, therefore timestamps are not included in the input waypoints. Different from other works that consider contact reaction wrench as a disturbance, we treat $\boldsymbol{\tau}_c(t)$ as a predictable state in the trajectory planning, and we also output it to the controller.

The trajectory planning algorithm is to find an N-step state and control sequences $\boldsymbol{x}_{1:N}$,

$\boldsymbol{\tau}_{a1:N}$, which can be formulated as an optimization problem:

$$\min_{\substack{\boldsymbol{x}_{1:N},\\ \boldsymbol{\tau}_{a1:N},\\ h_{1:N}}} \sum_{n=1}^{N} \left(\|\boldsymbol{v}_n\|^2{}_{\boldsymbol{W}_v} + \|\boldsymbol{\tau}_{an}\|^2_{\boldsymbol{W}_\tau} + \|\delta\boldsymbol{\tau}_{an}\|^2_{\boldsymbol{W}_{d\tau}} + \gamma \right) h_n \tag{3.7a}$$

$$s.t. \quad \boldsymbol{x}_{n+1} = f_{dyn}(\boldsymbol{x}_n, \boldsymbol{\tau}_{an}, h_n) \tag{3.7b}$$

$$0 < h_n \leq \bar{h} \tag{3.7c}$$

$$\boldsymbol{p}_{N_i} = {}^r\boldsymbol{p}_{ci}, \quad F_{N_i} = {}^rF_i, \ \forall i \in 1, \cdots, M \tag{3.7d}$$

$$\phi(\boldsymbol{p}_n) \begin{cases} = 0, \text{if } {}^rF_{N_i}{}^rF_{N_{i+1}} > 0 \\ > 0, \text{if } {}^rF_{N_i}{}^rF_{N_{i+1}} = 0 \end{cases} \quad \forall n \in [N_i, N_{i+1}] \tag{3.7e}$$

$$g(\boldsymbol{x}_n) = 0 \tag{3.7f}$$

$$\begin{aligned} &\underline{\boldsymbol{\tau}}_a \leq \boldsymbol{\tau}_{an} \leq \bar{\boldsymbol{\tau}}_a, \quad \underline{\dot{\boldsymbol{\tau}}}_a \leq \delta\boldsymbol{\tau}_{an} \leq \bar{\dot{\boldsymbol{\tau}}}_a \\ &\underline{\boldsymbol{v}} \leq \boldsymbol{v}_n \leq \bar{\boldsymbol{v}}, \quad \underline{\dot{F}} \leq \delta F \leq \bar{\dot{F}} \end{aligned} \tag{3.7g}$$

where $\|\boldsymbol{x}\|_{\boldsymbol{W}} := \boldsymbol{x}^\top \boldsymbol{W} \boldsymbol{x}$ denotes the $\boldsymbol{W}$-norm of $\boldsymbol{x}$. The cost function in (3.7a) penalizes the twist $\boldsymbol{v}_n$, control wrench $\boldsymbol{\tau}_{an}$, changing rate of the wrench $\delta\boldsymbol{\tau}_{an}$ and time-interval between two steps h_n. $\boldsymbol{W}_v$, $\boldsymbol{W}_\tau$, $\boldsymbol{W}_{d\tau}$ are three diagonal weighting matrices and γ is the scalar of time penalty.

(3.7b) represents the discrete-time dynamics of the system (3.1), derived using RK4 discretization. Since timestep h_n is also an optimization variable, we impose an upper limit $\bar{h}$ on the time interval for each step (3.7c) to ensure dynamic accuracy. The value of $\bar{h}$ is determined based on the trajectory length and the total number of steps N, and is fine-tuned during the experiments.

(3.7d) ensures the UAM tracks waypoints by imposing the vehicle sequentially pass through the i^{th} waypoint at step N_i. Due to the upper limit for the time interval h_n, the optimization may become infeasible if there are insufficient steps between waypoints. Conversely, too many steps will cause a significant computational burden. We pre-allocate the step-index N_i in proportion to the distance between adjacent waypoints to reduce the optimization time while ensuring feasibility. We set $N_1 = 1$ and $N = N_M$.

(3.7e) ensures the end-effector contacts the target surface during the segment between two contact waypoints and ensures the end-effector does not make contact with the surface during other segments.

(3.7f) provides the task-related constraint. Here, in the aerial calligraphy, to maintain the end-effector perpendicular to the wall for better drawing performance, the desired UAM orientation is constructed by (3.5)

$$g(\boldsymbol{x}) := \boldsymbol{R}_{\mathcal{W}}^{\mathcal{B}} \ominus {}^{r}\boldsymbol{R}_{\mathcal{W}}^{\mathcal{B}}, \quad {}^{r}\boldsymbol{R}_{\mathcal{W}}^{\mathcal{B}}(\boldsymbol{p}) = [\hat{\boldsymbol{t}}_z, \hat{\boldsymbol{t}}_y, -\hat{\boldsymbol{n}}_t] \tag{3.8}$$

where $\ominus$ denotes the subtraction operation in the $SO(3)$ manifold. Additional safety and feasibility constraints for vehicle control wrench, velocity and contact force are added as (3.7g), where $\underline{\boldsymbol{a}}$ and $\bar{\boldsymbol{a}}$ represents the lower limit and upper limit of the variable $\boldsymbol{a}$.

Since the time intervals obtained from problem (3.7a) may be non-uniform, we apply linear interpolation to the planning results to ensure the control and state sequences are consistent with the controller frequency. Given $\boldsymbol{x}_{1:T}, \boldsymbol{\tau}_{a1:T}$, $h_{1:T}$, for $t \in [T_j, T_{j+1}]$, and $T_j = \sum_{n=1}^{j} h_n$,

$$\boldsymbol{x}(t) = \lambda \boldsymbol{x}_j + (1-\lambda)\boldsymbol{x}_{j+1}, \ \boldsymbol{\tau}_a(t) = \lambda \boldsymbol{u}_j + (1-\lambda)\boldsymbol{\tau}_{aj+1}$$

with $\lambda = (t - T_j)/(T_{j+1} - T_j)$. We also calculate the model predicted contact wrench $\boldsymbol{\tau}_c(t)$ from (3.6) for later use. In summary, the planning algorithm outputs ${}^{r}\boldsymbol{x}(t)$, ${}^{r}\boldsymbol{\tau}_a(t)$, and ${}^{r}\boldsymbol{\tau}_c(t)$ for the hybrid motion-force controller.

3.5 Hybrid Motion and Force Control

This section discusses the design of the hybrid motion-force controller. Based on our previous work [26], we enhance the controller by incorporating contact wrench models, including the dynamics of contact force, friction force, and friction-induced torque.

3.5.1 Contact Wrench Estimation

Unlike the free-flight UAV, the primary challenge of UAM is to compensate the contact wrench $\boldsymbol{\tau}_c$ during aerial interaction. We estimate it by fusing the sensor readings into the analytic model.

As described in (3.4), ${}_{\mathcal{C}}\boldsymbol{\tau}_c$ consists of both the normal and shear forces. Since we have an F/T sensor at the origin of frame $\mathcal{S}$, it allows us to estimate ${}_{\mathcal{C}}\boldsymbol{\tau}_c$. Specifically, for the aerial calligraphy task, given the planned trajectory ${}^{r}\boldsymbol{x}$, we obtain ${}_{\mathcal{C}}^{r}\boldsymbol{\tau}_c = {}^{r}F\,{}^{r}\boldsymbol{\alpha}({}^{r}\boldsymbol{v})$. On the other hand, F/T sensor provides the measurement ${}_{\mathcal{S}}\tilde{\boldsymbol{\tau}}_s$ for the wrench at the origin of frame $\mathcal{S}$ (denote as ${}_{\mathcal{S}}\boldsymbol{\tau}_s$). In addition, it is obvious to see that the relation of the ${}_{\mathcal{S}}\boldsymbol{\tau}_s$ and the contact wrench can be expressed as ${}_{\mathcal{B}}\boldsymbol{\tau}_c = \boldsymbol{Ad}_{\boldsymbol{T}_{\mathcal{B}}^{\mathcal{S}}}\,{}_{\mathcal{S}}\boldsymbol{\tau}_s$. Therefore, the contact normal force can be estimated by fusing the sensor measurement with the contact force model

$$\hat{F} = \left[\boldsymbol{Ad}_{\boldsymbol{T}_{\mathcal{B}}^{\mathcal{C}}}\,{}^{r}\boldsymbol{\alpha}\right]^{+} \boldsymbol{Ad}_{\boldsymbol{T}_{\mathcal{B}}^{\mathcal{S}}}\,{}_{\mathcal{S}}\tilde{\boldsymbol{\tau}}_s \tag{3.9}$$

where $[\cdot]^{+}$ denotes the pseudo-inverse of a matrix. Furthermore, under hover or zero tilt condition of the fully-actuated UAM and given the specific flat vertical wall, $\boldsymbol{R}_{\mathcal{B}}^{\mathcal{C}} \approx \boldsymbol{I}$. Thus, the estimate can be simplified as $\hat{F} = [\boldsymbol{R}_{\mathcal{B}}^{\mathcal{S}}\,{}_{\mathcal{S}}\boldsymbol{\tau}_{s[1:3]}]_{[1]}$. The corresponding contact wrench estimates can be obtained as

$$_{\mathcal{C}}\hat{\tau}_c = \hat{F}\boldsymbol{\alpha}({}^{r}\boldsymbol{v}) \tag{3.10}$$

Here, we use reference velocity ${}^{r}\boldsymbol{v}$ instead of measured velocity for robustness considerations.

3.5.2 Hybrid Motion-Force Controller

The fully-actuated UAM is able to control the translation and orientation independently. We develop a contact-aware hybrid motion-force controller based on our previous work [26] for the UAM to track the planned trajectory ${}^{r}\boldsymbol{x}(t)$.

The control wrench $\boldsymbol{\tau}_a$ combines the output of the motion tracking controller $\boldsymbol{\tau}_p$ and wrench tracking controller $\boldsymbol{\tau}_f$ as follows:

$$_{\mathcal{B}}\boldsymbol{\tau}_a = \boldsymbol{R}_{\mathcal{B}}^{\mathcal{C}}((\mathbf{I}_{6\times 6} - \boldsymbol{\Lambda})\boldsymbol{R}_{\mathcal{C}}^{\mathcal{B}}\,{}_{\mathcal{B}}\boldsymbol{\tau}_p + \boldsymbol{\Lambda}_{\mathcal{C}}\boldsymbol{\tau}_f) \tag{3.11}$$

where $\boldsymbol{\Lambda} = \text{blockdiag}[\delta, 0, 0, 0, 0, 0] \in \mathbb{R}^{6\times 6}, \delta \in \{0, 1\}$. Intuitively, the matrix $\boldsymbol{\Lambda}$ selects direct force control commands and leaves the complementary subspace for the motion control. δ is the force motion control switch based on ${}^{r}F$. δ is set to 0 when ${}^{r}F = 0$, corresponding to the pure motion control, and δ is set to 1 when ${}^{r}F > 0$, indicating the force control along the normal direction of the surface.

For motion control, a PID controller is designed to directly control the end-effector motion.

$$
\begin{aligned}
{}_{\mathcal{B}}\boldsymbol{\tau}_p = & {}_{\mathcal{B}}^{r}\boldsymbol{\tau}_a + \boldsymbol{Ad}_{\boldsymbol{T}_{\mathcal{B}}^{\mathcal{C}}}({}_{\mathcal{C}}^{r}\boldsymbol{\tau}_c - {}_{\mathcal{C}}\hat{\boldsymbol{\tau}}_c) \\
& - \boldsymbol{K}_q\boldsymbol{e}_q - \boldsymbol{K}_v e_v - \boldsymbol{K}_i \int \boldsymbol{e}_q dt
\end{aligned} \tag{3.12}
$$

where

$$
\begin{aligned}
& \boldsymbol{e}_q = [\boldsymbol{e}_p^\top, \boldsymbol{e}_R^\top]^\top, \ \boldsymbol{e}_v = [\dot{\boldsymbol{e}}_p^\top, \boldsymbol{e}_\omega^\top]^\top, \ \boldsymbol{e}_p = \boldsymbol{R}_{\mathcal{B}}^{\mathcal{W}}({}_{\mathcal{W}}\boldsymbol{p} - {}_{\mathcal{W}}{}^{r}\boldsymbol{p}) \\
& \boldsymbol{e}_R = \frac{1}{2}({}^{r}\boldsymbol{R}_{\mathcal{W}}^{\mathcal{B}\,\top}\boldsymbol{R}_{\mathcal{W}}^{\mathcal{B}} - \boldsymbol{R}_{\mathcal{W}}^{\mathcal{B}}\,{}^{r}\boldsymbol{R}_{\mathcal{W}}^{\mathcal{B}\,\top})^\vee, \ \boldsymbol{e}_\omega = \boldsymbol{\omega} - {}^{r}\boldsymbol{\omega}
\end{aligned} \tag{3.13}
$$

${}_{\mathcal{B}}^{r}\boldsymbol{\tau}_a$ and ${}_{\mathcal{C}}^{r}\boldsymbol{\tau}_c$ come from the planning. Here $(\cdot)^\vee$ is to extract a vector from a skew-symmetric matrix and $\boldsymbol{K}_q, \boldsymbol{K}_v, \boldsymbol{K}_i$ are the positive definite tunable controller gain matrices. Here ${}_{\mathcal{C}}\hat{\boldsymbol{\tau}}_c$ is the key novelty of the proposed controller that we incorporate contact wrench models.

As for wrench tracking, we design an impedance controller to ensure that the end-effector maintains the desired wrench. Given the contact force reference F^r and the estimated contact normal force (3.9), the controller is designed as:

$$
\begin{aligned}
e_f &= \hat{F} - {}^{r}F, \quad e_x = \boldsymbol{e}_{q[1]} \\
F_f &= {}^{r}F - m_0 m_d^{-1}(K_e e_x + D_e \dot{e}_x) \\
&\quad - K_{f,p} e_f - K_{f,i} \int e_f dt \\
&= {}^{r}F - K_{e,p} e_x - K_{e,d} \dot{e}_x - K_{f,p} e_f - K_{f,i} \int e_f dt
\end{aligned} \tag{3.14}
$$

with m_0, m_d the actual and desired vehicle mass, respectively, $K_{e,p}$ is the normalized stiffness, $K_{e,d}$ is the normalized damping. Then, the control wrench ${}_{\mathcal{C}}\boldsymbol{\tau}_f = [F_f, \boldsymbol{0}_{1\times 5}]^\top$ and ${}_{\mathcal{W}}\boldsymbol{\tau}_p$ will be passed through (3.11) to get the total control wrench. Finally, ${}_{\mathcal{B}}\boldsymbol{\tau}_a$ is distributed to each rotor through the control allocation process.

Notice that $\hat{\boldsymbol{\tau}}_c$ in (3.12), and $\hat{F}$ in (3.9) take the contact force, including the normal force, friction force and the induced torque into account. Therefore, the designed controller is a contact-aware hybrid motion-force controller.

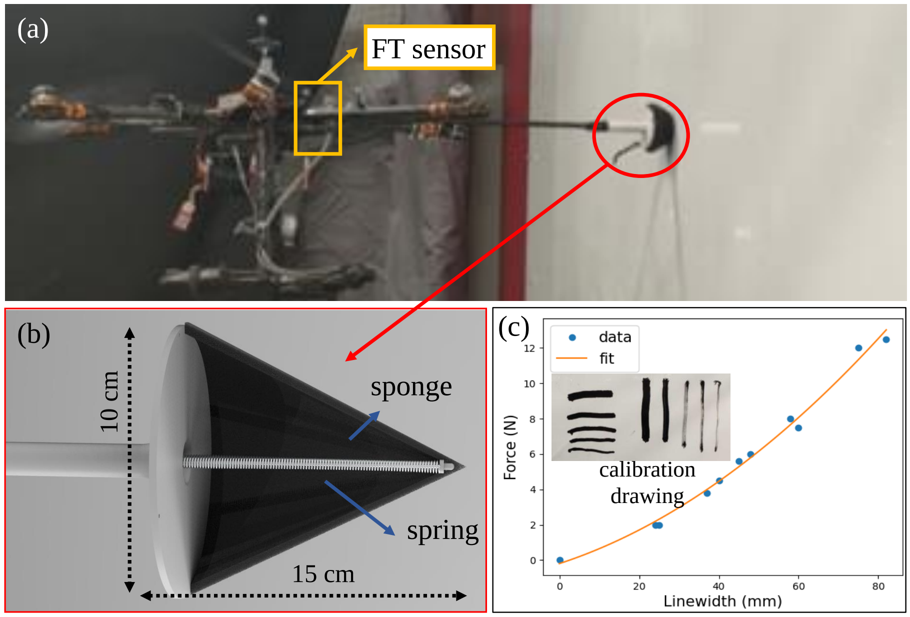

Figure 3.4: **System overview of the aerial writing platform:** (a) Side view of the UAM system, (b) custom-designed sponge pen used as the end-effector, and (c) calibration curve showing the relationship between stroke linewidth and contact force.

3.6 End-Effector and Interface Designs

In this section, we introduce the end-effector and user interface design for the aerial calligraphy task.

3.6.1 End-Effector Design

Aerial calligraphy requires precise tracking of target linewidths, which can be achieved through contact force tracking. Most existing pens cannot produce varied linewidths, and specialized tools like the Chinese Calligraphy Brush need extra steps to align bristles for quality writing. In general, the pen for aerial calligraphy must satisfy these requirements: 1) Linewidth must correlate positively with contact force; 2) Changes in linewidth should be noticeable with varying force; 3) The correlation should be easily adjustable; 4) Writing should remain symmetrical during force application and release; 5) The pen should efficiently retain ink. We design a novel end-effector that meets these requirements.

Fig. 3.4 illustrates our pen design, consisting of two primary components: the sponge and the spring support. The cone-shaped sponge ensures isotropic properties and effectively absorbs and retains ink. A spring encircles a metal rod within the sponge, with adjustable stiffness and preload to fine-tune the force-linewidth curve. We dilute black ink with isopropyl alcohol (IPA) to enhance adhesion to the operating surface, which is a dry-erase whiteboard paper.

3.6.2 Aerial Calligraphy Interface

The interface allows users to input strokes from open-source datasets [45] or through manual design. We also develop a touchscreen interface, implemented as an iPad application, which enables direct drawing of target strokes by capturing waypoints and the pressure applied during drawing. For each user input target strokes s_i, including waypoints ${}_{\mathcal{I}}\boldsymbol{p}_i$ (position) in image space $\mathcal{I}$ and ${}_{\mathcal{I}}d_i$ (linewidth), the interface processes it using Cubicspline fitting with constraints such as minimal curvature, and maximum and minimum linewidth to ensure feasibility. The target contact force ${}^{r}F_i$ is then calculated from the pre-calibrated force-linewidth relationship. The position ${}^{r}\boldsymbol{p}_{ci}$ is obtained by transforming ${}_{\mathcal{I}}\boldsymbol{p}$ to frame $\mathcal{W}$. For multi-stroke letters, we add the non-contact point at the endpoints of each stroke to enable stroke switching. Subsequently, the developed contact-aware trajectory planning algorithm generates the reference motion-force trajectory $\boldsymbol{x}(t), \boldsymbol{\tau}_a(t)$, which the UAM tracks to draw the target letter. In the real realization, we run the optimization for each free flight period and continuous contact period and then stitch them together.

3.7 Experiments

In this section, we conduct experiments to demonstrate the effectiveness of our method. We perform aerial calligraphy on several letters from different languages and investigate the impact of various system modules and writing speeds on overall performance. More writing results are available on our project website.

3.7.1 Experiment Setup

We use a customized fully-actuated hexarotor UAV with onboard computation handled by an Nvidia Xavier AGX running ROS Noetic. System localization is achieved using

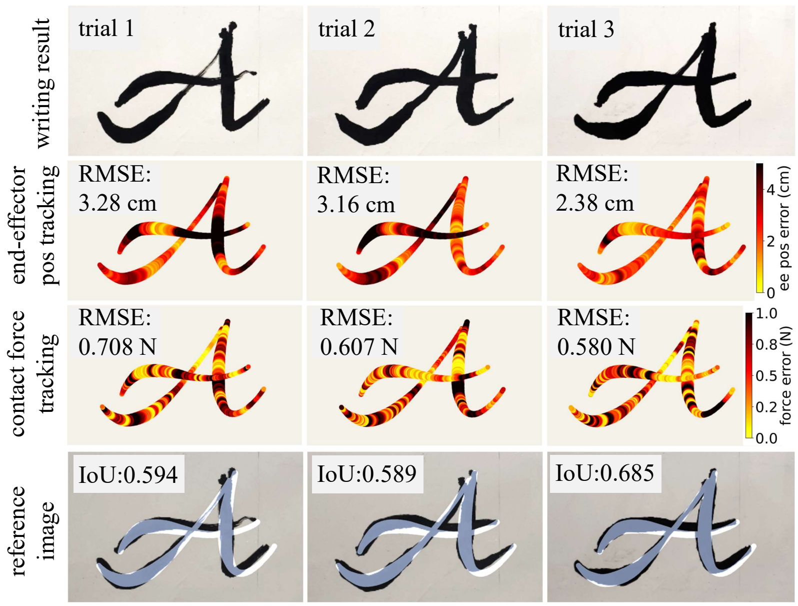

Figure 3.5: **Performance evaluation over repeated trials of writing the letter "A":** Top row: Final written image. Middle two rows: Time history of end-effector position and contact force tracking performance. Bottom row: visual comparison between the written letter and ground truth image, evaluating the overall writing performance.

an Optitrack Motion capture system. For aerial calligraphy, the newly designed pen (Sec. 3.6.1) is mounted as the end-effector, and a dry-erase whiteboard paper is attached to a vertical wall for drawing. The friction coefficient is pre-measured as $\mu = 0.4$. The force-linewidth relationship is calibrated by drawing lines with varying contact forces and fitting a polynomial curve, as shown in Fig. 3.4. Trajectory planning and optimization is performed using CasADI [4] with IPOPT [74] solver, initialized with a 0.1s time interval and linearly interpolated states between the waypoints, and hover thrusts. In real applications, we control the vehicle base for implementation convenience.

3.7.2 Aerial Calligraphy

The aerial manipulator writes three English letters: 'A', 'I', and 'R' in a custom typeface and font size. We manually define the sparse waypoints with point locations and local linewidths. The maximum reference velocity is set to $20\ \mathrm{cm/s}$, the maximum acceleration to $20\ \mathrm{cm/s^2}$; and the maximum contact force change rate to $2\ \mathrm{N/s}$. We conducted three trials for each letter to test repeatability.

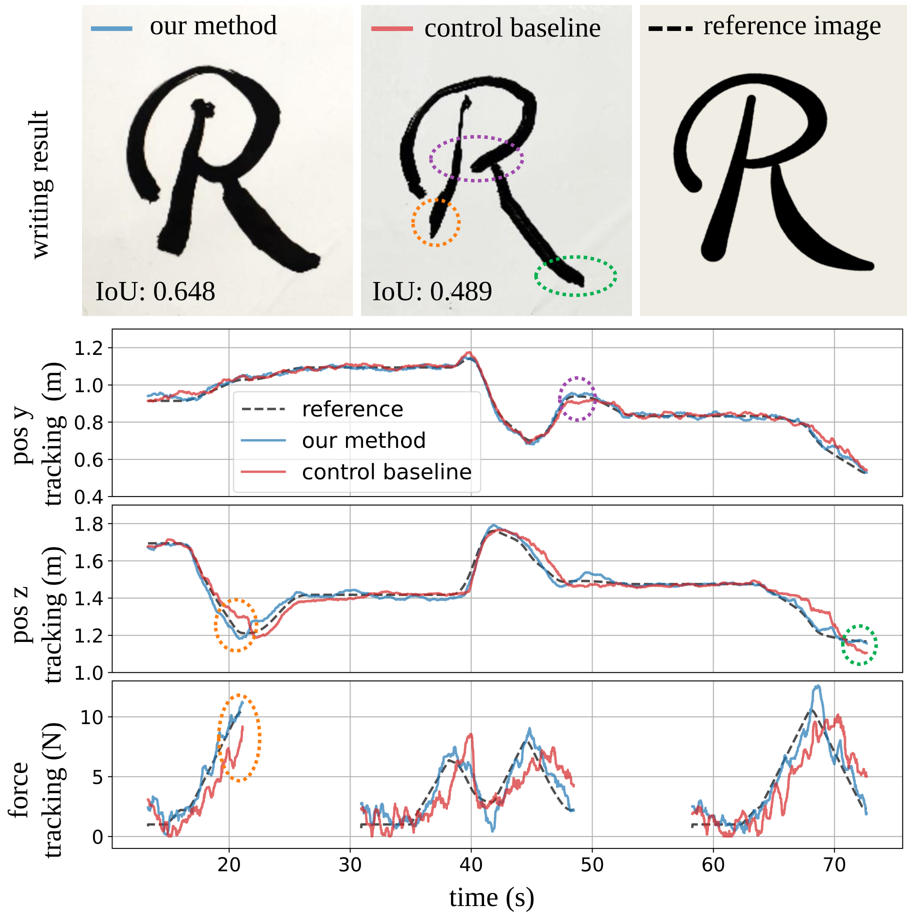

Figure 3.6: **Comparison with the control baseline during aerial writing:** Top: Writing results for letter 'R'. Bottom: Time history of end-effector motion and contact force tracking performance.

Table 3.1: Ablation Studies Results

	Our Method	Control Baseline	Planning Baseline
EE Pos RMSE (cm)	$\mathbf{2.93 \pm 0.40}$	5.47 ± 0.74	4.40 ± 0.14
Base Pos RMSE (cm)	$\mathbf{2.48 \pm 0.39}$	4.07 ± 0.93	3.68 ± 0.65
Force RMSE (N)	$\mathbf{0.72 \pm 0.11}$	1.89 ± 0.07	1.04 ± 0.28
IoU	$\mathbf{0.59 \pm 0.09}$	0.43 ± 0.12	0.49 ± 0.06

Fig. 3.5 shows the visual performance of the written letter. The UAM tracks motion and force trajectories well, with the written letters closely matching the target shapes and linewidths. Challenges arise when the UAM changes direction or at the start and end points of strokes due to the non-homogeneous property of the sponge pen during the establishment and detachment of contact. Smaller circular sections, such as the bottom-left curve of the letter 'A', or the bottom-left part of the letter 'I' (see Fig. 3.7), also present difficulties. This is presumably due to the rapid changing of friction mode (static or sliding friction) and friction force (both magnitude and direction) in these areas which requires a more accurate friction model. Despite these issues, the controller's performance is consistent across trials.

To quantitatively evaluate the performance, we calculate the Root Mean Squared Error (RMSE) by comparing the reference trajectory, shown in Tab. 3.1. The RMSE of the base position tracking is around 2.5 cm and of end-effector (ee) position tracking is around 2.9 cm. The contact force RMSE is around 0.7 N, which is better than the previous work that tracked constant contact force [26]. This improvement is attributed to the developed contact-aware motion and force planning and control, which compensates for unmodeled behavior. We also use Intersection over Union (IoU) to evaluate the overall visual performance. We transform both the target and written letters into binary images, treating the letter as the target region at the pixel level. Our method achieves an average IoU of 0.59, with an average intersection ratio of 0.79 and an average union ratio of 1.35.

We also test our system on a more complex scenario: Chinese idiom calligraphy. As shown in Fig. 3.1, these characters feature more strokes, intricate shapes, and sharper turns. Our system still successfully wrote these characters with tracking and visual errors comparable to previous tests, highlighting the robustness of our approach across diverse writing tasks.

3.7.3 Ablation Studies

We conducted an ablation study to investigate the benefits of the proposed trajectory planning and hybrid motion-force control modules. We compared system performance under three scenarios: 1) our proposed approach; 2) without the friction and contact force model in the controller (Control Baseline); and 3) without contact-aware trajectory planning (Planning Baseline), using equal distance sampling of target waypoints for reference trajectory instead, which only passes ${}^{r}\boldsymbol{q}$ into the controller. The results are shown in Tab.

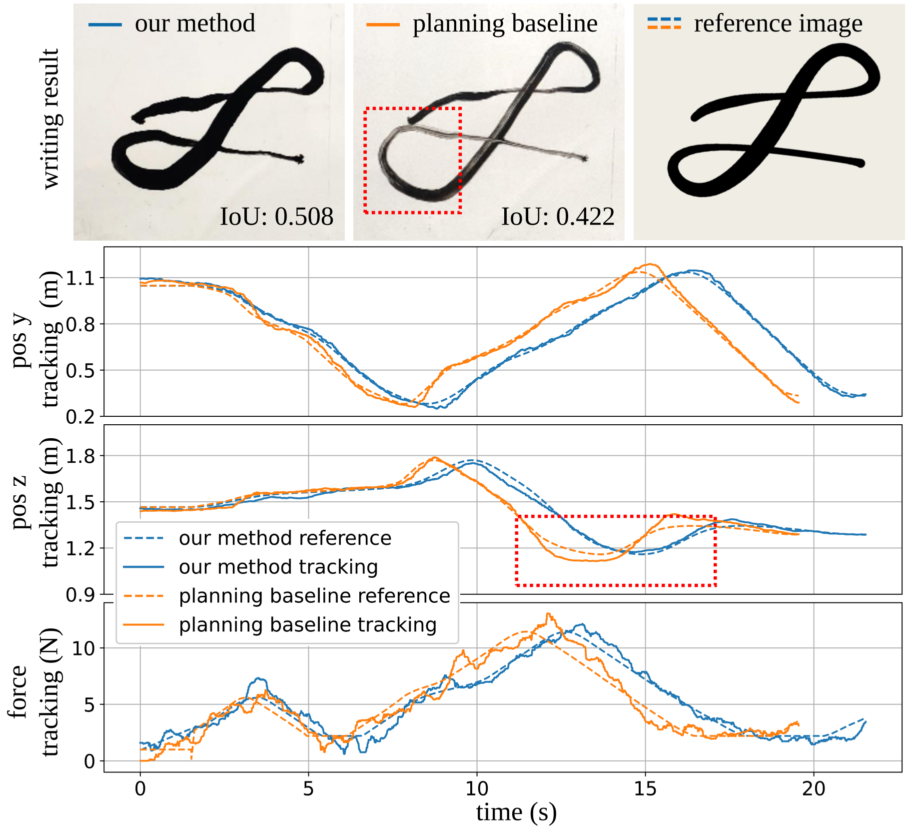

Figure 3.7: **Comparison with the planning baseline during aerial writing:** Top: Writing results for letter ‘I’. Bottom: Time history of end-effector motion and contact force tracking performance.

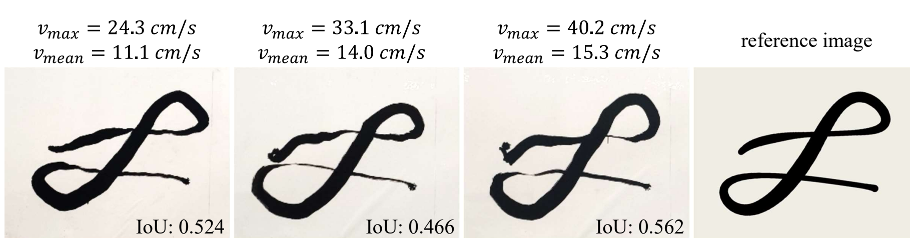

Figure 3.8: **Effect of velocity profiles on aerial writing performance:** This figure shows the visual results of writing the letter “I” under different end-effector velocity profiles.

3.1.

Fig. 3.6 shows the comparison with scenario 2). It is clear that without a proper contact force model, the contact force trajectory tracking is less accurate, leading to inconsistent stroke widths and occasional loss of contact. Without a proper friction force model, the written letter shapes are distorted, particularly at the ends of strokes and junctions, such as the vertical line and junction of the letter ‘R’.

Fig. 3.7 shows the comparison with scenario 3). Without proper dynamic constraints, the linear interpolated reference trajectory makes sharp turns at corners, resulting in discontinuous velocity references that are unfeasible for the vehicle. This causes overshoots at corners, such as the bottom-left corner of the letter ‘I’. In addition, testing the Planning Baseline with a slower speed resulted in even worse performance due to frequent switching between static and sliding friction, introducing more complex friction dynamics.

3.7.4 Speed Comparison

To demonstrate the effects of the velocity on writing accuracy, we perform the experiments with three different velocity profiles whose maximum speed is 24.3 cm/s, 33.1 cm/s, and 40.2 cm/s. Fig. 3.8 shows the visual results. The proposed method achieves a final writing IoU of 0.562 at a velocity of 40 cm/s. To the best of our knowledge, this is significantly faster than all related works [70]. The main restriction on achieving higher velocities is the calibrated contact force change speed, limited by hardware and the contact model. Given that sliding at a slow speed introduces extra challenges, there is an optimal velocity range, highlighting the necessity of our trajectory planning algorithm.

3.7.5 Aerial Calligraphy with Touchscreen Interface

We demonstrate the convenience of our touchscreen interface for a complete aerial writing demo. A human user writes ‘2024’ on the touchscreen. We then preprocess the input waypoints to make them feasible for our system. We regenerate the target image and render it for user verification. The UAM then performs contact-aware trajectory planning and hybrid motion-force control to draw the letters. Fig. 3.9 shows the user’s raw input, the regenerated target image, and the written letters by the UAM. The touchscreen provides a more intuitive way for users to input targets.

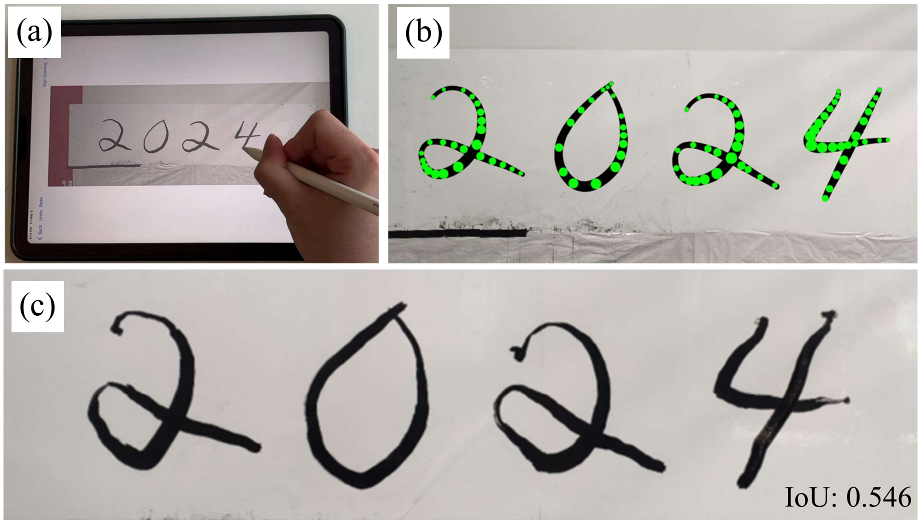

Figure 3.9: **End-to-end aerial writing from user input to execution:** (a) The user draws the target strokes “2024” on a touchscreen interface. (b) The planned trajectory with uniformly spaced timesteps, where denser green waypoints indicate slower planned velocities. (c) The final written result produced by the aerial manipulation system.

3.8 Conclusion

In this letter, we propose a general framework to simultaneously track time-varying contact force in the surface’s normal direction and motion trajectories on tangential surfaces. Specifically, we propose an aerial calligraphy system. Our pipeline imports predefined target strokes or user drawings from a touchscreen, generates a dynamically feasible trajectory, and controls the UAM to write the target calligraphy. Experiments demonstrate that our method can draw diverse letters with effective motion and force tracking. The IoU of the final drawing is 0.59, and the end-effector position tracking RMSE is 2.9 cm. Future work includes integrating the touchscreen as an online teleoperation interface.

Chapter 4

End-Effector-Centric Framework for Versatile Aerial Manipulation

4.1 Introduction

To achieve greater generality and versatility in aerial manipulation, this chapter proposes an end-effector-centric (ee-centric) control and policy learning framework. Although previous chapters introduced task-specific control and planning strategies, real-world manipulation often requires handling diverse tasks without significant system reconfiguration. Drawing inspiration from classical robotic manipulation paradigms, we develop an ee-centric Model Predictive Control (MPC) approach that abstracts low-level platform dynamics from high-level task planning and policy execution. Complementing this control architecture, we present a teleoperation interface and an imitation learning pipeline, enabling seamless human demonstration and subsequent autonomous policy generation. Experimental results validate the system's versatility by executing a variety of manipulation tasks—such as pick-and-place operations, peg-in-hole insertion, and light bulb replacement—without task-specific controller adjustments.

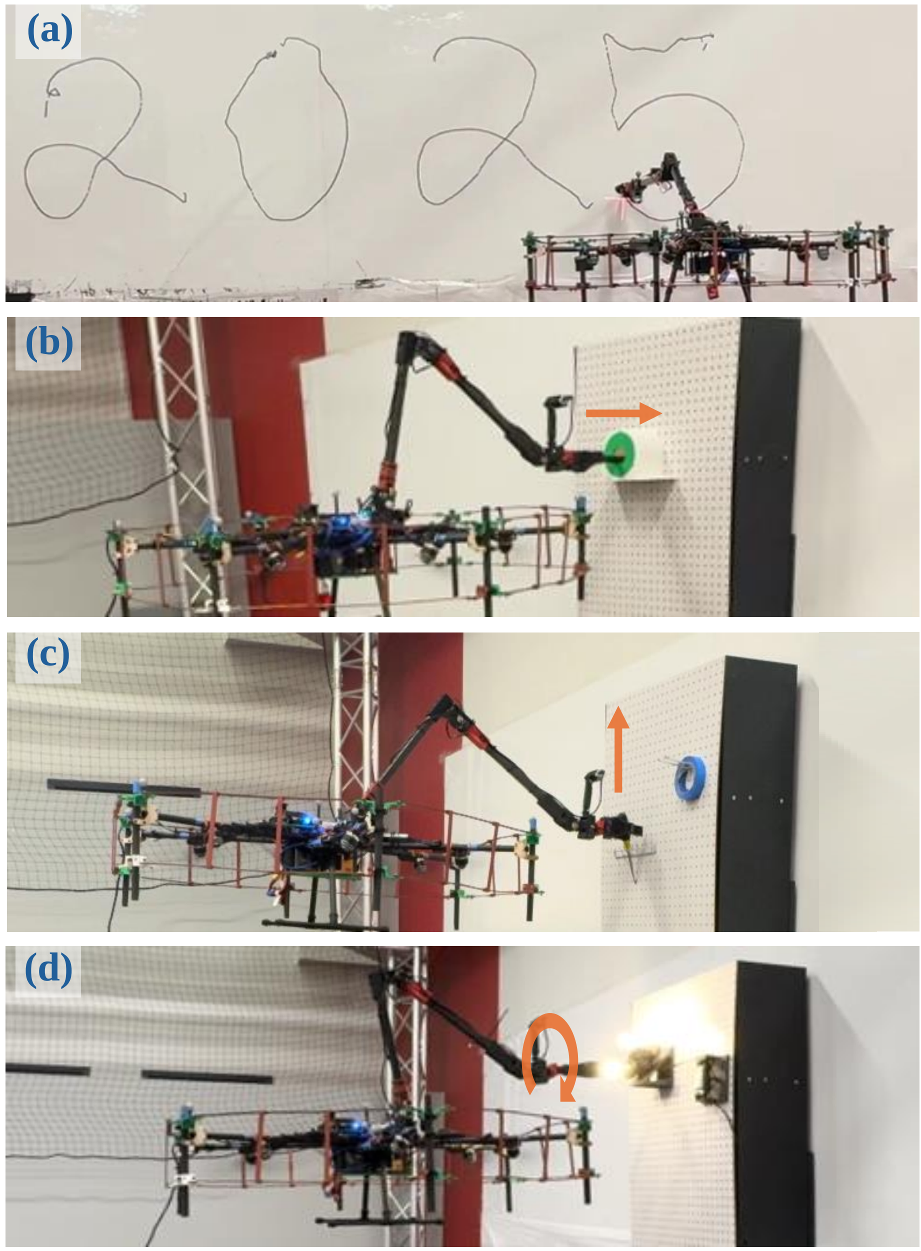

Figure 4.1: **Demonstration of versatile aerial manipulation capabilities:** The proposed framework enables robust execution of diverse tasks, including (a) writing “2025,” (b) peg-in-hole insertion, (c) pick-and-place, and (d) light bulb replacement.

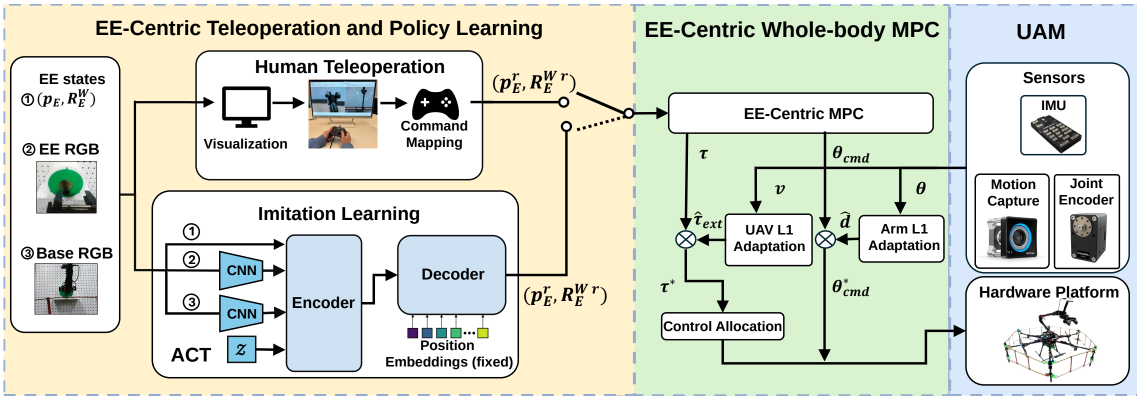

Figure 4.2: **Overview of the end-effector-centric aerial manipulation framework:** The system integrates a UAM platform, an end-effector-centric whole-body MPC, and a high-level policy module that includes both a teleoperation interface and an imitation learning framework using Action Chunk with Transformer (ACT) [89]. The high-level policy—whether from human input or a learned controller—outputs target end-effector states, which are tracked by the MPC to generate motor commands for the UAM.

4.2 Related Works

4.2.1 Aerial Manipulation

There have been many research efforts exploring aerial manipulation for various kinds of tasks [57]. Based on the motion primitives they require, common aerial manipulation tasks can be categorized into: 1) Aerial Interaction, which requires maintaining contact with external objects, for tasks such as inspection [6, 7, 27], aerial writing [30, 48, 70] or pushing a target [10]. Researchers mostly developed a point-contact arm, such as a rigid rod, and proposed the hybrid motion-force control framework, although achieving high-precision tracking performance, struggled to handle scenarios requiring grasping; 2) Aerial Grasping [54], where previous works mainly focused on designing different custom end-effectors, such as claw [60] or soft gripper [22]. Some work also showed amazing results, achieving high-speed grasping, or grasping moving objects [71], but sacrificed payload capacity or precision due to specialized hardware designs. 3) Aerial Insertion. Typical work includes [61] where they proposed a specific hole searching policy for bolt screwing tasks, and [76] where they achieved mm-level peg-in-hole task; 4) Manipulate articulated objects such as doors [65], or valves [11]. In general, although different works have shown success on different specific tasks, the specific system design and algorithm development make the same hardware and algorithm hard to deploy to different tasks, reducing its potential for practical long-horizon versatile aerial manipulation tasks. In our work, we target these four types of aerial manipulation tasks, developing a versatile framework to handle all of them.

4.2.2 Mobile Manipulation Framework and EE-Centric Interface

Combining a whole-body tracking controller with a high-level policy has shown promise in mobile manipulation systems such as humanoids. [33] proposed a framework that consists of a robust humanoid whole-body controller with a high-level policy, either an autonomous agent like GPT-4o or an imitation learning policy learned from teleoperation. [23] developed a system that consists of a transformer-based low-level control and imitation learning for humanoids. [34] discussed the interface between high-level policy and low-level controller, and proposed a unified controller for humanoid whole-body control supporting versatile interfaces. Beyond the explicit whole-body motion, some work extends the interface to latent variables and extends the whole-body control to a vision-motor manipulation policy. [86] proposed a hierarchical framework that consists of the understanding module, a pre-trained large visual-language model running in low-frequency, and the execution module, a visual-based action policy running in high-frequency. Gemini robotics [67], Helix [19], and Isaac GR00T N1 [5] also adopted similar structures.

Defining tasks or commands using ee-centric approaches is widely adopted in general manipulation fields, as it is more intuitive and can be cross-embodiment. For example, Båberg et al. [13] developed a teleoperation interface to enable full control of the end-effector pose. The Universal Manipulation Interface [15, 29] demonstrates a data-collection and policy-learning framework that allows direct skill transfer from in-the-wild human demonstrations to multiple robot embodiments. Their system employs a hand-held gripper and carefully designed hardware-agnostic policies, showcasing the potential for ee-centric solutions in multi-platform scenarios. Similarly, other mobile manipulation strategies, such as N2M2 [37, 38] and HarmonicMM [80], reduced the operator's burden by extracting feasible base motions from end-effector trajectories. However, these works generally remain limited to ground robots. Although several aerial manipulation studies have adopted end-effector-centric methods [8, 28], they primarily focused on developing controllers to track specified end-effector trajectories without a systematic framework that tackles various tasks comprehensively.

In our work, we propose to adopt the two-layer hierarchical system for aerial manipulation and propose a unified framework with the ee-centric interface for versatile aerial manipulation tasks.

4.2.3 Teleportation and Imitation Learning

Developing a robust and practical autonomous aerial manipulation policy is extremely challenging due to complex real-world environments and high precision and safety requirements. Moreover, policies are typically designed to handle specific tasks and lack the generality to handle unexpected conditions. Therefore, teleoperation, which takes human effort into the loop for policy development, attracts researchers' interest as a practical solution. For example, [3] developed the UAM with a fully actuated UAV with a 0 DoF arm and controlled the end-effector directly by teleoperation, but their method is highly coupled with the specific UAM design, and the system struggles with versatile tasks due to the workspace limitation. In most structured UAM teleoperation works, users control each DoF separately [81] or explicitly switch to different modes during different phases [16]. Both of these increase the human teleoperator's burden and require the teleoperator to have a rich understanding of the specific hardware system. Moreover, even if incorporated with human input, the previous works still have not shown vast versatility in different tasks and scenarios.

Recently, imitation learning (IL) has demonstrated significant potential for autonomous policy learning due to its high data efficiency, straightforward framework, and outstanding performance. Recent progress in both systems, such as ALOHA [89] and mobile ALOHA [24], and algorithms, such as ACT [89] and diffusion policy [14], have significantly facilitated the success of long-horizon, contact-rich, complex manipulation tasks. However, there is no precedent to incorporate such IL-based policy into aerial manipulation fields due to the lack of a mature demonstration collection system, such as a well-developed teleoperation system, and the lack of a proven framework. In this work, we develop an intuitive teleoperation system using the ee-centric framework. It also helps to collect human demonstration data, enabling us to develop an imitation learning-based policy for autonomous aerial manipulation.

4.3 System Overview

Our aerial manipulation system is designed to enable precise and versatile operations. The system incorporates an end-effector-centric (ee-centric) interface to decouple high-level decision-making from low-level control, increasing the framework's versatility. As shown in Fig. 4.2, our system consists of an aerial manipulator platform, an ee-centric whole-body MPC, and an ee-centric policy module. The hardware platform consists of a fully-actuated hexarotor and a 4 DoF robotic arm. The platform has a large enough workspace and wrench space for different tasks. A motion capture system and onboard IMUs are used for drone state estimation. The joint encoders are used to get arm joint angles, and the end-effector states are then calculated based on forward kinematics. The ee-centric whole-body MPC reads the end-effector state target from high-level policy and generates the reference trajectory and reference control for both the UAV and the robotic arm. An L1 online adaptation control term is designed to further improve the tracking performance. The UAV control commands are then sent to the control allocation to generate the motor commands for the UAV to execute. At the most high-level, the ee-centric policy module gets current observations and generates the target end-effector states online without the need to consider the specific platform jointly. We developed two high-level policy modules. The first is the ee-centric teleoperation interface that allows human users to directly control the end-effector pose. The second is an imitation learning framework, where we adopt Action Chunking with Transformers (ACT) [90], to learn autonomous policies from human teleoperation demonstrations. The following sections will introduce the developed modules accordingly.

4.4 Hardware Design

4.4.1 Fully-Actuated UAV

The foundation of the system is a fully-actuated hexarotor UAV capable of independently generating six-dimensional forces and torques, based on our previous works [27, 28, 30]. This capability allows precise control of position and orientation, which is essential for executing complex aerial manipulation tasks. The robust design ensures stability in dynamic environments while ensuring high-precision end-effector tracking. We use Tarot680 as the drone base, 6 KDE 4215XF motors with 12-inch 2-blade propellers as our driving force, LiPo batteries for on-board power supply, an Intel NUC for on-board computation, and a customized PX4 autopilot for low-level flight control and information processing.

4.4.2 Manipulator

The UAV integrates a 4-DOF robotic manipulator optimized for versatile and precise task execution. The arm features three pitch joints and one roll joint, driven by Dynamixel XM540 and XM430 servos. Its configuration allows high-precision operations. The system achieves whole-body manipulation capabilities by combining the UAV's actuation with the manipulator, enhancing task execution across diverse scenarios. The manipulator includes a modular end-effector, allowing interchangeable tools for specific tasks. For instance, a two-finger gripper with replaceable tips enables precision handling, while a circular gripper is ideal for changing light bulbs.

4.4.3 Perception

In this work, we mainly focus on the indoor environment, and we include a brief discussion about outdoor environment applications in Appendix B. We use both the motion capture system and the PX4 onboard IMU for drone state estimation. The motion capture system provides the UAV's position and can be replaced with other localization methods, such as SLAM [88]. The manipulator arm joint angles are estimated by the joint encoders, and the end-effector states are online calculated based on forward kinematics.

To further improve drone perception for teleoperation and autonomous policy development, we equip the aerial manipulator with two RealSense RGBD cameras. One camera is mounted on the UAV base to capture a broad view of the entire workspace, while the other is positioned near the end-effector to deliver detailed close-up views of the target area. This dual-camera setup ensures teleoperators and vision-based policies can maintain precise control and situational awareness during complex manipulation tasks.

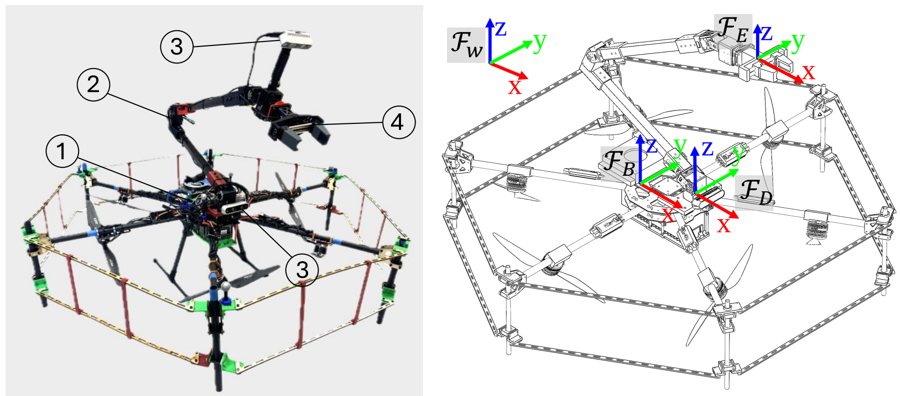

Figure 4.3: **UAM hardware system design with key functional components:** The system consists of (1) a fully-actuated hexarotor base, (2) a 4-DoF manipulator, (3) Intel RealSense cameras for vision-based perception and feedback, and (4) an end-effector gripper for object interaction. The coordinate frames shown in the right diagram represent the reference axes used throughout the system.

4.5 System Modeling

4.5.1 Frames and Notation

The frames depicted in the Fig 4.3 are defined as follows: $\mathcal{F}_W$ is an inertial world frame with its z-axis opposite to the gravity vector, ensuring $\hat{z}_W$ points upward. $\mathcal{F}_B$ is rigidly attached to the UAV's body at its center of gravity, with axes (x_B, y_B, z_B) aligned with the UAV body frame. $\mathcal{F}_D$ is the manipulator base frame, centered at the attachment point of the manipulator on the UAV. The transformation from $\mathcal{F}_B$ to $\mathcal{F}_D$ is defined by a constant translation $\mathbf{p}_D \in \mathbb{R}^3$ and a fixed orientation $\boldsymbol{R}_B^D \in SO(3)$. $\mathcal{F}_E$ is the end-effector frame with axes (x_E, y_E, z_E), where x_E is aligned with the roll axis of the 4th joint of the manipulator, while y_E remains horizontal. Other symbols in the paper are listed in Table 4.1 for the convenience of the following discussion.

4.5.2 Fully-Actuated UAV Dynamics

A fully-actuated UAV is adopted as the base of the aerial manipulator, which can generate six-dimensional force and torque independently. Let the generalized position of the UAV be represented as $\boldsymbol{q} = [\boldsymbol{p}, \boldsymbol{R}_B^W]$, where $\boldsymbol{p} \in \mathbb{R}^3$ represents the position of the UAV in the global

Table 4.1: Notation Overview

Symbol	**Dimension**	**Description**
m	$\mathbb{R}$	Mass
$\boldsymbol{J}$	$\mathbb{R}^{3\times 3}$	Inertia tensor
$\boldsymbol{p}$, $\boldsymbol{R}_B^W$	$\mathbb{R}^3$, $SO(3)$	UAV position expressed in $\mathcal{F}_W$ and orientation between $\mathcal{F}_B$ and $\mathcal{F}_W$
$\boldsymbol{v}$	$\mathbb{R}^3$	Generalized velocity, expressed in $\mathcal{F}_W$
$\boldsymbol{\tau}$	$\mathbb{R}^6$	Control wrench
$\boldsymbol{\tau}_{\text{ext}}$	$\mathbb{R}^6$	External disturbance wrench
$\boldsymbol{g}_W$	$\mathbb{R}^6$	Gravity vector in $\mathcal{F}_W$
$\boldsymbol{p}_E$, $\boldsymbol{R}_E^W$	$\mathbb{R}^3$, $SO(3)$	End-effector position expressed in $\mathcal{F}_W$ and orientation between $\mathcal{F}_E$ and $\mathcal{F}_W$
$\boldsymbol{\theta}$	$\mathbb{R}^4$	Current joint angles
$\boldsymbol{\theta}_{\text{cmd}}$	$\mathbb{R}^4$	Commanded joint angles
$\boldsymbol{d}$	$\mathbb{R}^4$	Joint servo disturbance
$\boldsymbol{\zeta}$	$\mathbb{R}^{3\times 4}$	DH parameter with the i^{th} joint component $[\theta_i, l_i, a_i, \alpha_i]^T$
$\boldsymbol{\beta}$	$\mathbb{R}$	Joint motor delay constant

coordinate frame, and $\boldsymbol{R}_B^W \in SO(3)$ represents its orientation. The generalized velocity is denoted as $\boldsymbol{v} = [\dot{\boldsymbol{p}}, \boldsymbol{\omega}]$, where $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity.

The UAV dynamics can be formulated using Newton-Euler equations for rigid body motion as follows:

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{C}\boldsymbol{v} + \boldsymbol{g} = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{cxt}} \tag{4.1}$$

with inertia matrix $\boldsymbol{M} \in \mathbb{R}^{6\times6}$, centrifugal and Coriolis term $\boldsymbol{C} \in \mathbb{R}^{6\times6}$, gravity wrench $\boldsymbol{g} \in \mathbb{R}^6$, control wrench $\boldsymbol{\tau} \in \mathbb{R}^6$ from the UAM actuators, and unknown external wrench $\boldsymbol{\tau}_{ext} \in \mathbb{R}^6$ from model mismatch and manipulator interaction.

Specifically, we have

$$\boldsymbol{M} = \operatorname{diag}\left(\begin{bmatrix} m\boldsymbol{I}_{3\times3} & \boldsymbol{J} \end{bmatrix}\right), \tag{4.2}$$

$$\boldsymbol{C} = \operatorname{diag}\left(\begin{bmatrix} m[\boldsymbol{\omega}]_\times & -\boldsymbol{J}[\boldsymbol{\omega}]_\times \end{bmatrix}\right), \tag{4.3}$$

$$\boldsymbol{g} = m \operatorname{diag}\left(\begin{bmatrix} \boldsymbol{R}_B^W & \boldsymbol{0}_{3\times3} \end{bmatrix}\right) \boldsymbol{g}_W. \tag{4.4}$$

where m is the vehicle mass, $\boldsymbol{J}$ is the moment of inertia at the vehicle center of mass in the body frame, $\boldsymbol{g}_W = [0, 0, g, 0, 0, 0]^\top$ is the gravitational acceleration in $\mathcal{F}_W$, and $[*]_\times$ is the skew-symmetric matrix associated with vector $*$.

4.5.3 Manipulator Kinematics

In this work, the manipulator employs servos as joint actuators, which cannot accurately control joint torques directly. Therefore, only the kinematics of the manipulator is considered in the system modeling. The interaction between the manipulator and the fully-actuated UAV is treated as a disturbance and is compensated in real-time using L1 adaptive control.

We use the standard Denavit-Hartenberg (DH) convention [18] to model the forward kinematics of our 4-DoF robotic arm. Under the DH formulation, the adjacent frame transformation $\boldsymbol{T}_i^{i-1}$ is characterized by four parameters θ_i, d_i, a_i, α_i, where the first one is the joint angle and the last three are pre-identified and fixed during robot movement. Define DH parameter $\zeta_i = [\theta_i, l_i, a_i, \alpha_i]^\top \in \mathbb{R}^4$. The frame transformation from end-effector frame to arm base body frame can be written as

$$\boldsymbol{T}_E^D(\boldsymbol{\theta};\boldsymbol{\zeta}) = \prod_{i=1}^{4} \boldsymbol{T}_i^{i-1}(\theta_i;\zeta_i) \tag{4.5}$$

Then the transformation from world frame to end-effector frame can be computed as follows:

$$\boldsymbol{T}_E^W = \begin{bmatrix} \boldsymbol{R}_E^W & \boldsymbol{p}_E^W \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} = \boldsymbol{T}_B^W \cdot \boldsymbol{T}_D^B \cdot \boldsymbol{T}_E^D \tag{4.6}$$

where $\boldsymbol{T}_B^W$ can be obtained from UAV odometry and $\boldsymbol{T}_D^B$ is a fixed transformation between the manipulator base and the UAV.

The accurate DH parameters are obtained through system identification. We collect motion data of the manipulator using a motion capture system and compute the DH parameters via least squares regression. The detailed parameter values are presented in Table 4.3.

4.5.4 Manipulator Motor Delay

The servo motor dynamics are approximated as first-order systems to account for command-to-state delay. For the 4-DoF manipulator, the relationship between commanded joint angles $\boldsymbol{\theta}_{\text{cmd}} \in \mathbb{R}^4$ and actual joint angles $\boldsymbol{\theta} \in \mathbb{R}^4$ is governed by:

$$\text{diag}(\boldsymbol{\beta})\dot{\boldsymbol{\theta}} + \boldsymbol{\theta} = \boldsymbol{\theta}_{\text{cmd}} + \boldsymbol{d} \tag{4.7}$$

where $\boldsymbol{\beta} \in \mathbb{R}$ are the joint-specific time delay constants, and $\boldsymbol{d} \in \mathbb{R}^4$ is the unknown disturbance in servo control. This formulation captures the transient response characteristics of each actuator. The motor delay coefficients $\boldsymbol{\beta}$ are identified alongside the DH parameters using least squares regression, with the results presented in Table 4.3.

4.6 End-Effector-Centric Whole-Body Control with Online Adaptation

Given the over-actuated nature of our system and the users' primary focus on the end-effector motion, we employ model predictive control to regulate the end-effector trajectory. This approach enables whole-body coordination between the manipulator and the fully-actuated UAV, ensuring precise and efficient end-effector motion control. As discussed in previous sections, the complex interaction between the manipulator and the UAV is treated as the disturbance in the modeling stage, which introduces uncertainty in the nominal model used in the whole-body MPC. To mitigate the disturbances and model uncertainties, we integrate the L1 adaptive controller, ensuring robust disturbance compensation and accurate tracking performance. The diagram of the control algorithm is illustrated in Fig. 4.2.

4.6.1 End-Effector-Centric Model Predictive Controller

In the following, we describe the whole-body MPC framework used to optimize the end-effector reference trajectory.

For the MPC formulation, we define the following state and control variables:

$$\boldsymbol{x} := \begin{bmatrix} \boldsymbol{p}_E & \boldsymbol{R}_E^W & \boldsymbol{v} & \boldsymbol{\theta} \end{bmatrix} \quad \boldsymbol{u} := \begin{bmatrix} \boldsymbol{\tau} & \boldsymbol{\theta}_{\text{cmd}} \end{bmatrix} \tag{4.8}$$

We use the following error functions for the position of the end-effector, the UAM velocity, the wrench control input, and the manipulator joint angle, respectively:

$$\boldsymbol{e}_p = \boldsymbol{p}_E - \boldsymbol{p}_E^r \tag{4.9a}$$

$$\boldsymbol{e}_R = \frac{1}{2}\left(\boldsymbol{R}_E^{W\,r\top}\boldsymbol{R}_E^W - \boldsymbol{R}_E^{W\top}\boldsymbol{R}_E^{W\,r}\right)^\vee \tag{4.9b}$$

$$\boldsymbol{e}_v = \boldsymbol{v} - \boldsymbol{v}^r \tag{4.9c}$$

$$\boldsymbol{e}_\theta = \boldsymbol{\theta} - \boldsymbol{\theta}^r \tag{4.9d}$$

$$\boldsymbol{e}_u = \boldsymbol{u} - \boldsymbol{u}^r \tag{4.9e}$$

where $(*)^\vee$ is the vee-operator that extracts a vector from a skew-symmetric matrix $*$ and $(\cdot)^r$ represents the reference state values. The reference signals $\boldsymbol{p}_E^r$ and $\boldsymbol{R}_E^{W\,r}$ are provided by a high-level teleoperation command or derived from an imitation learning policy. Manipulator default joint angle $\boldsymbol{\theta}^r$ is pre-selected, and we define the reference control $\boldsymbol{u}^r = [\mathbf{0}_6, \hat{\boldsymbol{\theta}}]$, where $\hat{\boldsymbol{\theta}}$ is the current joint states.

The MPC formulation minimizes a cost function over a finite time horizon H while subject to system dynamics and constraints:

$$\boldsymbol{u}_{\text{opt}} = \arg\min_{\boldsymbol{u}} \left\{ L_e(\boldsymbol{x}_H, \boldsymbol{x}_H^r) + \sum_{i=1}^{H} L_r(\boldsymbol{x}_n, \boldsymbol{x}_n^r, \boldsymbol{u}_n) \right\} \tag{4.10a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{n+1} = \boldsymbol{f}_{dyn}(\boldsymbol{x}_n, \boldsymbol{\tau}_n) \tag{4.10b}$$

$$\boldsymbol{x}_0 = \hat{\boldsymbol{x}}, \quad \boldsymbol{x}_n \in \mathcal{X} \tag{4.10c}$$

$$\boldsymbol{u}_{\text{lb}} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\text{ub}} \tag{4.10d}$$

Eq. (4.10a) defines the optimization objective, where H represents the discrete prediction horizon. The stage and terminal costs, L_r and L_e, are quadratic functions of the tracking errors, given by $\boldsymbol{e}_i^\top \boldsymbol{Q}_i \boldsymbol{e}_i$, where $\boldsymbol{e}_i \in \{\boldsymbol{e}_p, \boldsymbol{e}_R, \boldsymbol{e}_v, \boldsymbol{e}_\theta, \boldsymbol{e}_u\}$. The gain matrices $\boldsymbol{Q}_i$ are positive definite and tuned experimentally to balance precision and robustness.

Eq. (4.10b) enforces the discrete-time system dynamics, incorporating the fully actuated UAV dynamics Eq. (4.1), manipulator kinematics Eq. (4.6), and joint servo dynamics Eq. (4.7). The continuous system dynamics are discretized using a fourth-order Runge-Kutta (RK4) integration scheme to maintain numerical stability and accuracy. Disturbances $\boldsymbol{\tau}_{\text{ext}}$ and $\boldsymbol{d}$ are ignored in the MPC formulation and solving process, but will be handled in the following Section 4.6.2 via online L1 adaptation.

Eq. (4.10c) introduces state constraints, where $\hat{\boldsymbol{x}}$ represents the latest state estimate. The feasible state space $\mathcal{X}$ is defined by: 1) Self-collision avoidance constraints: ensuring that the manipulator does not collide with the UAV structure. 2) Environment collision constraints: preventing the UAV contact with external obstacles. 3) Safety constraints: including velocity limits and joint angle restrictions to ensure safe operation.

Eq. (4.10d) imposes actuation limits on the aerial manipulator, where $\boldsymbol{u}_{\text{lb}}$ and $\boldsymbol{u}_{\text{ub}}$ define the lower and upper bounds of the control inputs.

The end-effector centric whole-body MPC formulation is a general framework that can adapt to various vehicle types and can extend to systems with multiple end-effectors. The control inputs, UAV dynamics, and arm kinematics can be tailored to specific vehicle and manipulator configurations, ensuring flexibility across different aerial manipulation systems.

4.6.2 L1 Online Adaptation

As discussed in previous sections, the complex interaction between the manipulator and the UAV is treated as the disturbance, which introduces uncertainty in the nominal model used in the whole-body MPC. Such model uncertainties are typically bounded, and prior knowledge about their characteristics is usually available [31, 58]. To mitigate the disturbances and model uncertainties, we integrate the L1 adaptive controller, ensuring robust disturbance compensation and accurate tracking performance.

We adopt the L1 adaptive controller from [40, 79] in both the fully-actuated UAV motion control and the manipulator joint angle tracking control, to compensate the disturbance $\boldsymbol{\tau}_{\text{ext}}$ in Eq. (4.1) and $\boldsymbol{d}$ in Eq. (4.7).

The adaptation law is designed by

$$\boldsymbol{M}\dot{\hat{\boldsymbol{v}}} + \boldsymbol{C}\hat{\boldsymbol{v}} + \boldsymbol{g} = \boldsymbol{\tau} + \hat{\boldsymbol{\tau}}_{\text{ext}} + \boldsymbol{A}_v(\hat{\boldsymbol{v}} - \boldsymbol{v}) \tag{4.11a}$$

$$\hat{\boldsymbol{\tau}}'_{\text{ext}} = -(e^{\boldsymbol{A}_v dt} - \boldsymbol{I}_{6\times 6})^{-1}\boldsymbol{A}_v e^{\boldsymbol{A}_v dt}(\hat{\boldsymbol{v}} - \boldsymbol{v}) \tag{4.11b}$$

$$\hat{\boldsymbol{\tau}}_{\text{ext}} \leftarrow \text{low pass filter}(\hat{\boldsymbol{\tau}}_{\text{ext}}, \hat{\boldsymbol{\tau}}'_{\text{ext}}) \tag{4.11c}$$

where $\hat{\boldsymbol{v}} \in \mathbf{R}^6$ denotes the estimated UAV velocity, $\boldsymbol{A}_v$ is a Hurwitz matrix, dt is the discretization step length, and $\hat{\boldsymbol{\tau}}_{\text{ext}} \in \mathbb{R}^6$ encapsulates the unknown wrench disturbances. Here Eq. (4.11a) is a velocity estimator and Eq. (4.11b) and Eq. (4.11c) update and filter the disturbance $\hat{\boldsymbol{\tau}}_{\text{ext}}$.

Thus, the total UAV wrench control command $\boldsymbol{\tau}^*$ is computed as

$$\boldsymbol{\tau}^* = \boldsymbol{\tau}_{\text{mpc}} + \hat{\boldsymbol{\tau}}_{\text{ext}} \tag{4.12}$$

Similarly, the L1 adaptive controller for the manipulator joint angles is formulated to compensate for dynamic disturbances and model uncertainties:

$$\text{diag}(\boldsymbol{\beta})\dot{\hat{\boldsymbol{\theta}}} + \hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_{\text{cmd}} + \hat{\boldsymbol{d}} + \boldsymbol{A}_d(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}), \tag{4.13a}$$

$$\hat{\boldsymbol{d}}' = -(e^{\boldsymbol{A}_d dt} - \boldsymbol{I}_{4\times 4})^{-1}\boldsymbol{A}_d e^{\boldsymbol{A}_d dt}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}), \tag{4.13b}$$

$$\hat{\boldsymbol{d}} \leftarrow \text{low pass filter}(\hat{\boldsymbol{d}}, \hat{\boldsymbol{d}}'). \tag{4.13c}$$

where $\hat{\boldsymbol{\theta}}$ is the estimated joint state, and the disturbance term $\hat{\boldsymbol{d}} \in \mathbb{R}^4$. Thus, the final joint control command $\boldsymbol{\theta}^*$, incorporating the adaptive disturbance compensation, is given by:

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}_{\text{cmd}} + \hat{\boldsymbol{d}}. \tag{4.14}$$

4.7 EE-Centric Teleoperation and Policy Learning

As we mentioned, our framework enables the decoupling between the high-level policy and low-level controller, with the ee-centric interface serving as the sole connection between them. This allows the policy to be embodiment-agnostic, eliminating the need to consider low-level tracking control. In this section, we introduce two aerial manipulation systems we developed based on this framework: the ee-centric aerial teleoperation system and the imitation-learning-based autonomous aerial manipulation system.

4.7.1 EE-Centric Aerial Teleoperation

We developed an aerial manipulation teleoperation system with the ee-centric interface, allowing the operator to focus solely on controlling the target end-effector pose, as if they have complete control of a freely moving hand in 3D space.

Robotic teleoperation requires bidirectional communication between the user and the robot. For the user-to-robot command, we developed a gamepad program so that the user can control the end-effector position $\boldsymbol{p}_E{}^r$ and orientation $\boldsymbol{R}_E^{W\,r}$ with buttons and joysticks.

For robot-to-user communication, different from tabletop and mobile manipulation settings, users in aerial manipulation settings often lack direct visual access to the workspace, making it necessary to rely on onboard perception systems. In our work, we address this limitation by providing real-time visualization of RGB images captured from cameras mounted on both the end-effector and the base of the UAM. These images are displayed on monitors for continuous user observation. Further enhancing teleoperation efficacy, we have found it crucial to also visualize the user's inputs directly. To this end, we render the commanded target end-effector pose trajectories in real-time within 3D world frame plots. This dual approach of visual feedback not only improves spatial awareness but also significantly enhances user performance in teleoperation tasks.

4.7.2 EE-Centric Policy Learning

To establish the autonomous aerial manipulation policy for versatile tasks, we develop an ee-centric policy learning framework based on imitation learning. Specifically, we adopt Action Chunk with Transformer (ACT) as the network structure [89]. ACT utilizes a Conditional Variational Autoencoder (CVAE) where the encoder compresses action sequences and joint observations into a latent style variable. The transformer-based decoder generates action sequences from the latent variable (only during training and set to be the mean of the prior during testing), current joint observations, and encoded image features. The action chunking mitigates compounding errors and enhances the model's ability by predicting multiple future actions at once.

In this work, the ACT policy as well as policy observation and action are defined as follows:

$$\boldsymbol{a}_{t:t+K} = \pi_{\varphi}(\boldsymbol{o}_t) \tag{4.15}$$

$$\boldsymbol{o}_t = \left\{I_E,\ I_B,\ \boldsymbol{p}_E,\ \boldsymbol{R}_E^W\right\}_t \tag{4.16}$$

$$\boldsymbol{a}_t = \left\{\boldsymbol{p}_E^r,\ \boldsymbol{R}_E^{W^r}\right\}_t \tag{4.17}$$

where π denotes the ACT policy and φ is the network parameter. I_B, I_E are RGB images from the base camera and the end-effector camera, each with 640×480 resolution. K is the chunking size. $\boldsymbol{p}_E$, $\boldsymbol{R}_E^W$, $\boldsymbol{p}_E^r$, $\boldsymbol{R}_E^{W^r}$ denote the current and target UAM position and orientation, respectively. We use ResNet-18 as the backbone to encode the RGB images before inputting them into the transformer encoder. The flowchart of the algorithm implementation is illustrated in Fig. 4.2.

4.8 Experiments

To validate the effectiveness of our proposed framework, we conduct a series of experiments focusing on end-effector trajectory tracking, aerial teleoperation, and policy learning for autonomous aerial manipulation[1]. We first assess whether the proposed whole-body MPC with L1 adaptation approach enhances trajectory tracking accuracy. Additionally, we

[1]Please check out our project page for more visualization videos: https://lecar-lab.github.io/flying_hand/.

evaluate how the ee-centric interface facilitates intuitive and precise teleoperation, reducing operator burden and improving task execution in a series of teleoperation tasks. Finally, we investigate whether the high-quality teleoperation demonstrations can be leveraged to train imitation learning-based policies for autonomous aerial manipulation in both simulation and real-world environments.

4.8.1 Experimental Setup

Trajectory Tracking Task Setup

To show the effectiveness of our proposed method in end-effector trajectory tracking tasks, we perform a comparison between our control methods against two baseline approaches:

- **w.o. MPC**: This baseline replaces the ee-centric MPC with the Direct Force Feedback Control(DFFC) method from [56], which directly controls the end-effector acceleration based on the current reference pose. However, it lacks a prediction horizon to account for future trajectories.
- **w.o. L1**: This baseline excludes the L1 adaptive component, leaving disturbances from UAV and manipulator interactions and modeling uncertainties uncompensated during control execution.

We conduct experiments with three types of reference trajectories for the end-effector, each lasting 60 seconds. The setpoint trajectory requires the aerial manipulator to keep the end-effector hovering at a fixed position $\boldsymbol{p}_E = [0.0, 0.0, 1.3]$. The ellipse trajectory requires tracking a sinusoidal trajectory defined as $\boldsymbol{p}_E = [0.5\sin(0.3t), 0.0, 1.4 + 0.2\sin(0.3t + 0.75)]$. The figure-8 trajectory requires tracking a trajectory $\boldsymbol{p}_E = [0.1 + 0.6\sin(0.3t), 0.0, 1.35 + 0.25\sin(0.6t)]$. The maximum velocity in the reference trajectory is about 0.2 m/s. The pitch, row, and yaw attitude of the end-effector are all fixed at zero during the tracking. Root Mean Square Error (RMSE) is used as the tracking performance evaluation criterion. Each trajectory is repeated three times to compute the mean and standard deviation.

Aerial Manipulation Task Setup

We conducted a series of experiments to evaluate the capabilities and applications of our aerial manipulation system. We select different typical tasks from each category we

Table 4.2: Controller Parameters

Horizon Length T	2.5 s
Horizon Steps N	100
State Cost $\boldsymbol{Q}_p$	$\mathrm{diag}(12, 12, 12)$
Rotation Cost $\boldsymbol{Q}_R$	$\mathrm{diag}(10, 10, 10)$
Velocity Cost $\boldsymbol{Q}_v$	$\mathrm{diag}(0.1, 0.1, 0.1)$
Joint Angle Cost $\boldsymbol{Q}_\theta$	$\mathrm{diag}(0.1, 0.1, 0.1)$
Control Cost $\boldsymbol{Q}_u$	$\mathrm{diag}(0.03, 0.03, 0.03, 0.1, 0.1, 0.1)$

discussed in Sec. 4.2.1, including:

- **Aerial Writing**: Drawing a target shape (the digit “2025”) on a vertical wall, with an overall size of approximately $3\mathrm{m} \times 0.8\mathrm{m}$. This task required precise specification and tracking of the end-effector pose trajectory while maintaining stable contact with the surface.
- **Aerial Peg-in-Hole**: Inserting a 20 mm diameter pole into a 50 mm diameter hole positioned around 150 cm above the ground.
- **Rotate Valve**: Manipulating the articulated valve by grasping its handle and rotating it along a 20 cm diameter circle, emulating industrial valve manipulation.
- **Aerial Pick and Place**: Grasping and placing various objects with different shapes and sizes, including the screwdriver, pen, tape, and glue bottle.
- **Unmount Light Bulb**: Grasping a mounted light bulb and unscrewing it from the socket.
- **Mount Light Bulb**: A long horizon task that requires a sequence of motion, including inserting a light bulb into a socket, screwing it in, and subsequently turning it on by pressing the button.

For different tasks, different end-effectors are adopted, including the parallel jaw gripper for the pick and place and peg-in-hole task, a passive elastic claw for grasping the light bulb, and a bucket-shaped gripper for rotating the valve.

Table 4.3: System Identification Results

Mass Matrix $\boldsymbol{M}$	$\mathrm{diag}(0.105, 0.121, 0.101, 0.025, 0.011, 0.013)$
Motor Delay β	(0.66, 0.68, 0.81, 0.85)
Joint 1 DH Param $\boldsymbol{\zeta}_1$	$d_1 = 0.0$, $a_1 = 0.363$, $\alpha_1 = 0.10$
Joint 2 DH Param $\boldsymbol{\zeta}_2$	$d_2 = 0.050$, $a_2 = 0.441$, $\alpha_2 = -0.10$
Joint 3 DH Param $\boldsymbol{\zeta}_3$	$d_3 = 0.0$, $a_3 = 0.007$, $\alpha_3 = -1.578$
Joint 4 DH Param $\boldsymbol{\zeta}_4$	$d_4 = 0.076$, $a_4 = 0.200$, $\alpha_4 = 0.0$

4.8.2 Implementation Details

The optimal control problem in the ee-centric MPC is implemented using ACADOS [72] with a 25ms discretisation step and a 2.5s constant prediction horizon, running in 100 Hz, and other controller parameters are listed in Table 4.2. The control output is executed in a receding horizon style, where at each iteration, only the first control input $\boldsymbol{u}_0$ is applied to the system.

Since both the L1 adaptive controller and the MPC controller require accurate system modeling $\boldsymbol{f}_{dyn}$ to achieve effective control performance, we perform system identification to estimate uncertain parameters shown in Table 4.3. We excite the system with two types of motions. First, arm motion-only trajectories are executed while keeping the UAV stationary to calibrate the DH parameters $\boldsymbol{\zeta}$ and joint servo delay $\boldsymbol{\beta}$. These trajectories ensure that the manipulator kinematic parameters and joint motor dynamics accurately reflect the actual manipulator motion response. Second, UAV free-flight trajectories are conducted to identify the drone dynamics described in Eq. (4.1).

4.8.3 Experiment I: Control Performance

Table 4.4 shows the comparison of our approach against w.o. MPC and w.o. L1 baselines in three types of reference end-effector trajectories. The results show that our proposed method achieves the lowest tracking error, with approximately 1 cm in hover and 4 cm during motion. In contrast, the baseline w.o. L1 exhibits 1.3 cm and 6.5 cm, respectively, while the baseline w.o. MPC performs the worst, with 2 cm in hover and 8 cm in motion. As shown in Fig. 4.4, compared with our method (blue), the baseline w.o. L1 (green) exhibits overshoot in the X and Z axes and bias in Y, indicating that the L1 controller

Table 4.4: End-Effector Trajectory tracking performance

RMSE (cm)	Setpoint	Ellipse	Figure-8
Our Method	**1.00** $\pm$ **0.11**	**3.98** $\pm$ **0.41**	**4.62** $\pm$ **0.58**
w.o. L1 Adaptation	1.33 $\pm$ 0.16	6.67 $\pm$ 0.42	6.28 $\pm$ 0.50
w.o. MPC	2.07 $\pm$ 0.19	8.52 $\pm$ 0.79	7.35 $\pm$ 0.51

effectively mitigates both transient and steady-state errors caused by model uncertainties. The baseline w.o. MPC (orange) suffers from significant motion lag, as DFFC fails to account for trajectory feedforward. Fig. 4.5 depicts the error distribution for all trajectories using the proposed methods and two baselines, showing that our method achieves the smallest and most centered error, whereas the L1 baseline exhibits steady-state errors and the MPC baseline displays a broader error spread due to dynamic lag. These results confirm the effectiveness of our proposed control scheme.

To further analyze the contribution of the L1 adaptive control, specifically its effectiveness in handling model mismatch and interaction disturbance, we visualized the base external wrench τ_{ext} measured from motion capture by numerical differentiation and the estimated wrench $\hat{\tau}_{ext}$ from L1 adaptation. The interaction force is within 5 N and the torque within 1.4 Nm. Fig. 4.8 shows disturbances along the base x (red), z (blue) and θ_{pitch} (green), respectively. The disturbances and model uncertainties primarily arise from arm motions, inaccurate thrust gain, and hovering thrust bias. Shaded areas indicate obvious interaction torque in base pitch resulting from arm motion. The results demonstrate that the L1 adaptive controller accurately compensates for base disturbances, effectively mitigating the impact of model mismatch.

We also investigate the contribution of arm flexibility to end-effector tracking performance by increasing the arm control cost 5 times. Fig. 4.6a *Slow-Arm* results show a 35% larger end-effector tracking error when arm flexibility is restricted, with a more oscillating end-effector trajectory (Fig. 4.6c) compared to the MPC with flexible arm (Fig. 4.6b). This aligns with the intuition that higher arm flexibility improves end-effector tracking performance, as the arm typically responds faster than the drone base.

Despite the high tracking performance of the proposed methods, Fig. 4.7 reveals that tracking error increases at lower altitudes (around 1m), likely due to unmodeled ground and wall effect disturbances. Additionally, oscillations in the end-effector trajectory are

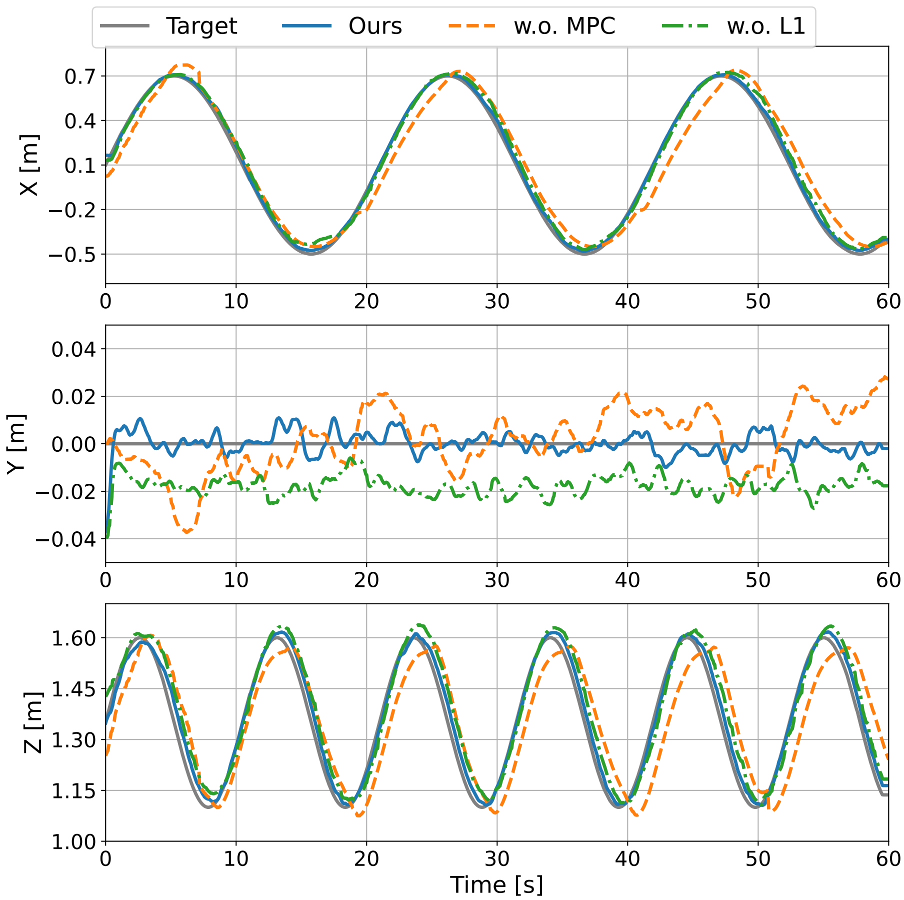

Figure 4.4: **End-effector tracking performance on an figure-8 trajectory:** The tracking results show that the baseline without MPC exhibits significant lag, while the baseline without L1 adaptation suffers from static tracking errors caused by model mismatches.

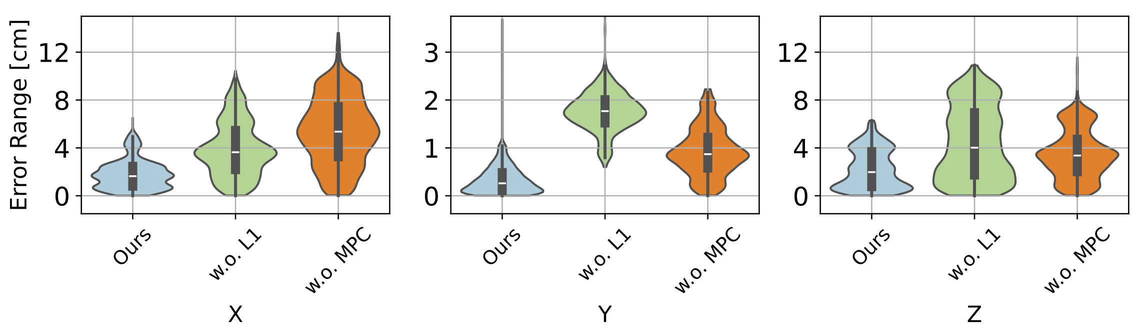

Figure 4.5: **End-effector tracking error distribution across different methods:** The figure compares the tracking error distributions for three trajectory types using our proposed method and two baseline controllers, highlighting the improved precision of our approach.

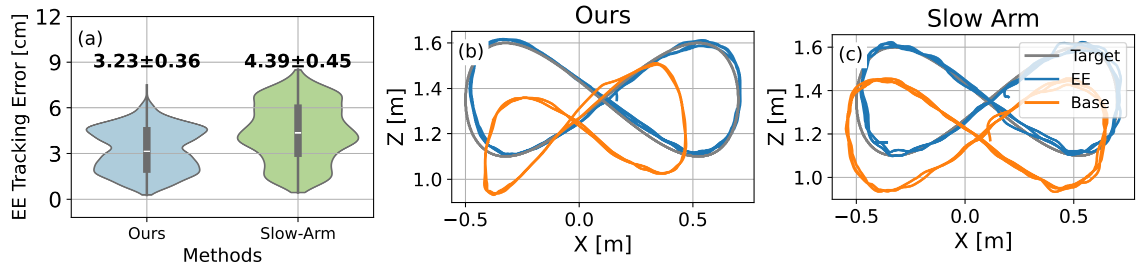

Figure 4.6: **Ablation study on the effect of arm flexibility on MPC performance:** This figure compares the end-effector tracking results with and without adding arm motion penalty. Our method has smoother EE tracking trajectories, highlighting the role of arm flexibility in accurate tracking.

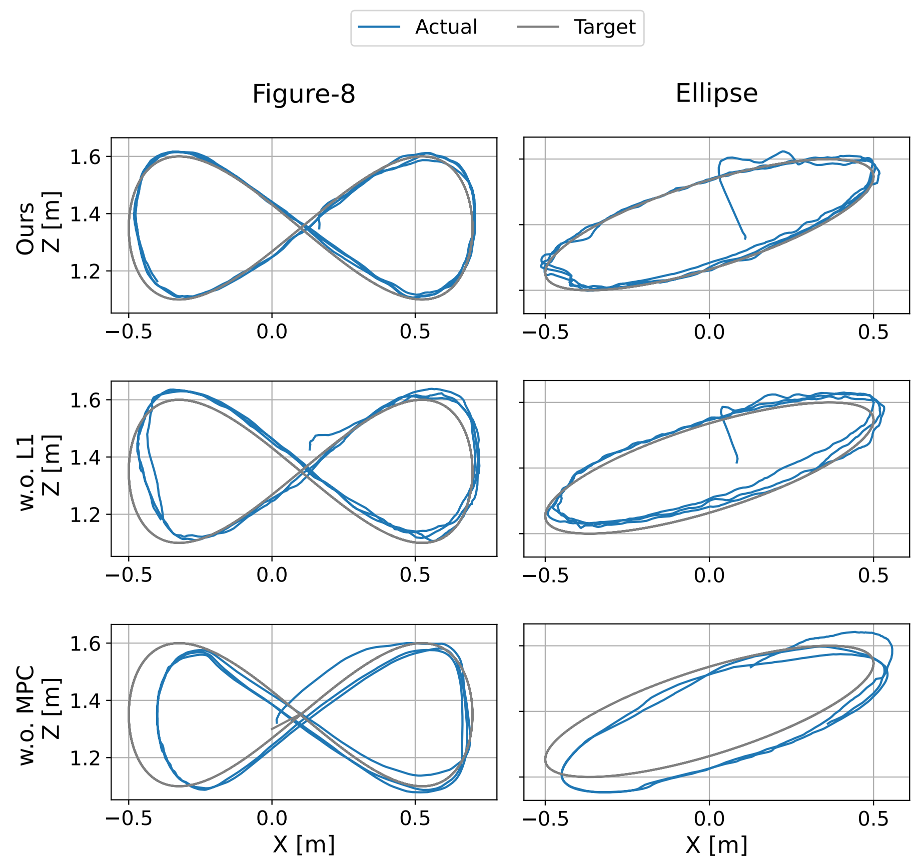

Figure 4.7: **Comparison of dynamic trajectory tracking performance across methods:** The figure shows tracking results for Figure-8 and Ellipse trajectories using three approaches. Our method consistently achieves the lowest end-effector tracking error in both tasks.

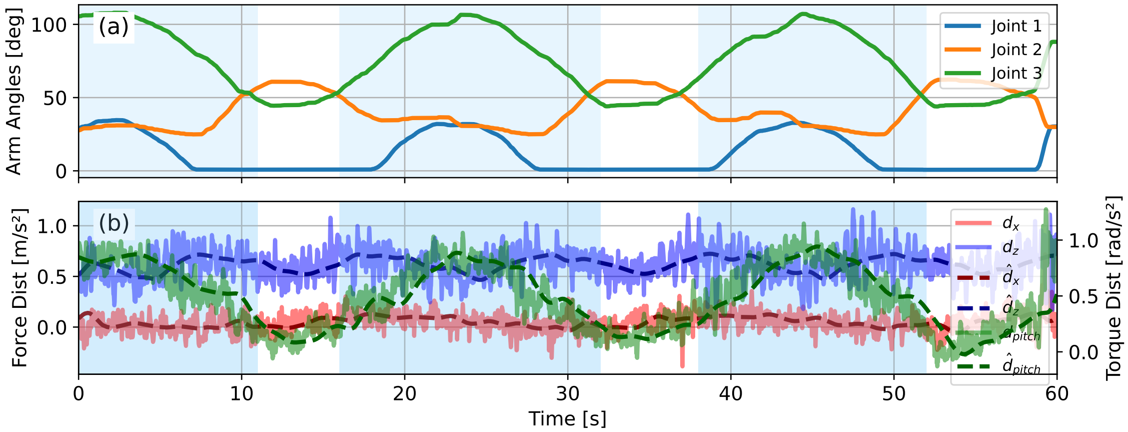

Figure 4.8: **Relation between arm motion and L1 disturbance estimation:** (a) Joint angles of the manipulator during interaction, and (b) external disturbance τ_{ext} and its L1-based estimation $\hat{\tau}_{\text{ext}}$ along the x, z, and θ_{pitch} axes, showing correlation with arm-induced dynamics.

observed across all methods. We notice servo backlash (around $0.5°$ dead zone), which results in a 2 cm control dead zone in the end-effector task space, limiting tracking precision during fast UAV maneuvers. Further improvements can be achieved through more accurate system modeling and higher-precision hardware to enhance tracking accuracy.

4.8.4 Experiment II: Aerial Teleoperation

First, we demonstrate the effectiveness and versatility of the proposed teleoperation system by targeting aerial writing, rotating the valve, aerial pick and place, unmount, and mount light bulb tasks. As shown in Fig. 4.10 and Fig. 4.11, human teleoperators can easily achieve all aerial manipulation tasks with little learning and operation cost. One key success factor we attribute is the ee-centric interface, which reduces human effort and improves the quality of teleoperation data for future policy learning.

In a subsequent experiment, we evaluate the benefits of directly controlling the end-effector's pose using our framework against controlling each degree of freedom (DoF) for UAVs and robotic arms [81] in a simulated peg-in-hole task. The teleoperation command trajectories are illustrated in Fig. 4.9a. Direct control of the end-effector allowed operators to issue more fluid command trajectories, significantly enhancing the precision of end-effector movements and decreasing the time required to complete the task.

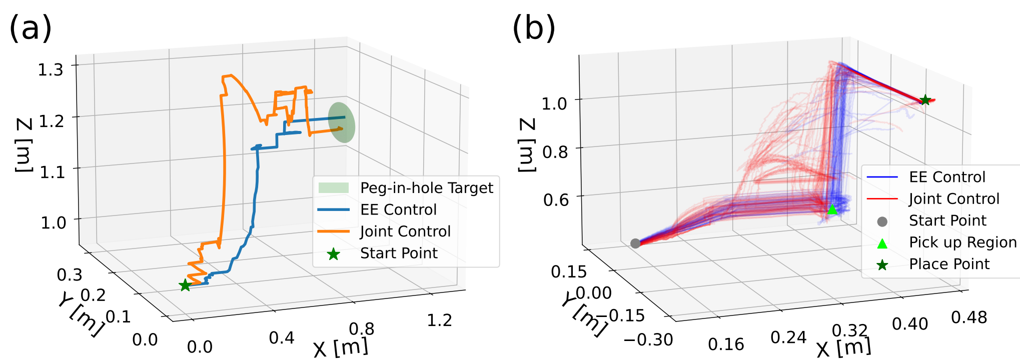

Figure 4.9: **Comparison of end-effector command trajectories across teleoperation and policy interfaces:** (a) Command trajectories using the end-effector-centric and full-DoF teleoperation interfaces in a simulated peg-in-hole task. (b) Command trajectories of the learned autonomous policy over 50 test trials using both interfaces in a simulated pick-and-place task.

Figure 4.10: **Aerial teleoperation manipulation task demonstrations:** (1) Aerial Writing — the UAM writes “2025” on a whiteboard using a marker pen; (2) Rotate Valve — the UAM grasps a handle and completes one full rotation of a valve; (3) Pick-and-Place — the UAM picks up an object and places it in a target location.

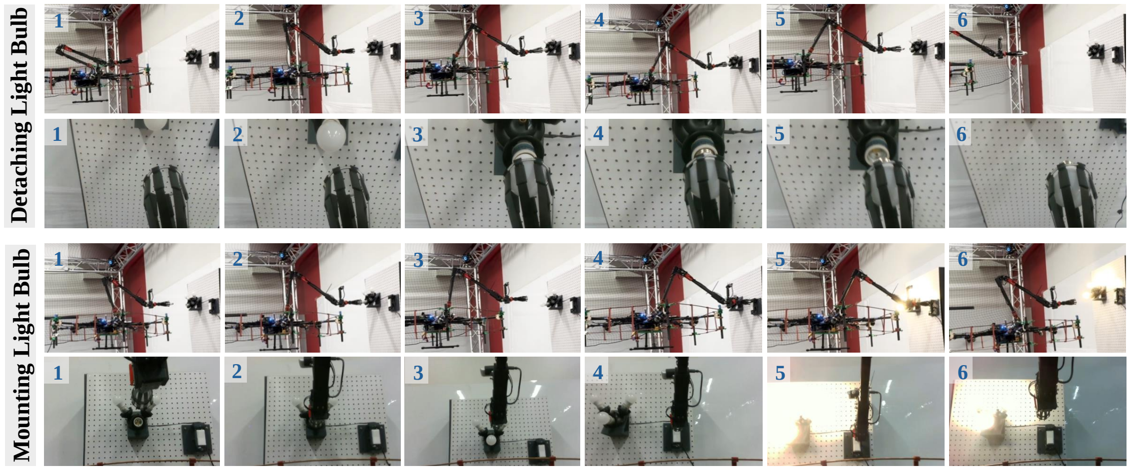

Figure 4.11: **Long-horizon aerial teleoperation for light bulb replacement:** In the first flight, the UAM grasps and unscrews the existing light bulb. In the second flight, it inserts and screws in a new bulb, then presses a button to turn on the light, completing the task across two sequential operations.

4.8.5 Experiment III: Learning from Demonstration

Simulation Experiments

We first demonstrate our learning from demonstration framework in Mujoco [69] simulator with four tasks: (i) *peg-in-hole*, (ii) *rotate valve*, (iii) *pick and place* and (iv) *open and retrieve*, as shown in Fig. 4.12 (a). To collect demonstrations, we use a scripted policy for each task. Every episode of the scripted policy lasts about 12 seconds for each task. We collect 50 episodes for each task. Note that with our ee-centric interface, we do not consider any joint configuration when collecting demonstrations, which allows us to efficiently collect smooth demonstrations without tediously adjusting each joint position to complete the task. Our ACT policy for each task in the simulation is trained with the action chunk size of 100 and limited 5000 epochs. More details on implementation are included in Appendix A. After training, we choose the policy with the least validation loss to perform 50 evaluation trials. The evaluation result is shown in Table 4.5.

To show the advantage of learning from an ee-centric demonstration compared to a joint space demonstration, we use the same demonstration trajectory but change the observation and action to be in UAM configuration space, i.e., the UAV position and orientation, and

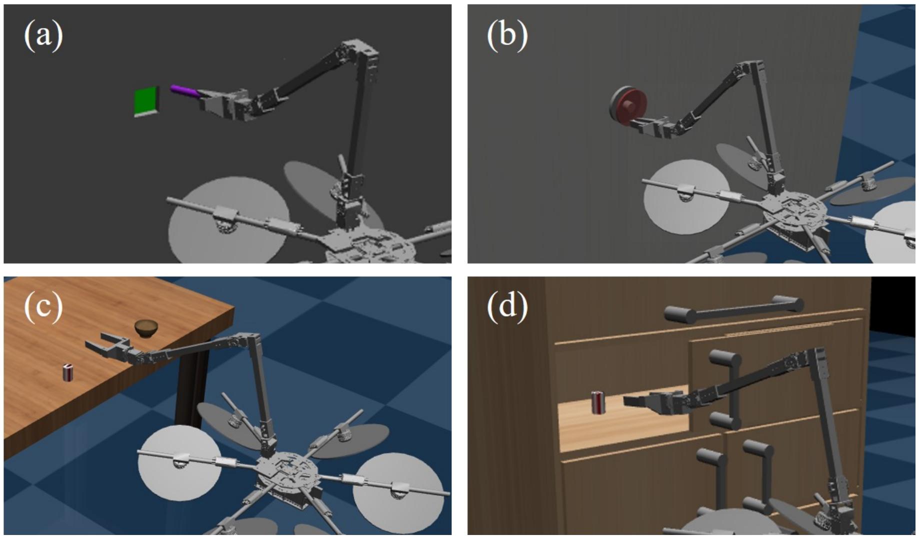

Figure 4.12: **Task setup in MuJoCo simulation environment:** The figure illustrates four aerial manipulation tasks used for evaluation: (a) Peg-in-Hole, (b) Rotate the Valve, (c) Pick and Place, and (d) a long-horizon Open and Retrieve task.

each joint angle of the robotic arm. After that, we train a joint space ACT policy with the same training setting as the ee-centric ACT policy, except that the end-effector pose in the observation and action space is replaced by the drone base pose and full manipulator joint angles. Their success rate comparison is summarized in Table 4.5. It shows that with limited 5000 training epochs, ee-centric policy outperforms the joint space policy in challenging tasks including *pick and place*, *peg in hole*, and *open and retrieve*. As illustrated in Fig. 4.9b, the ee-centric policy targets the pickup region and the place point more precisely, while the joint space policy is more prone to generate wrong targets.

Overall, the experimental results reveal several complementary insights:

- **Geometric Precision Advantage**: Our ee-centric policy achieves 2.5× higher success rate in geometrically sensitive *peg in hole* task, directly benefiting from task-space supervision that eliminates the accumulated end-effector error from the joint space.
- **Multi-Skill Composition**: In the *open and retrieve* task, our ee-centric policy achieves 2× higher success rate than the joint space policy, which demonstrates its inherent advantages in multi-skill decomposition and execution.

Table 4.5: Imitation Learning Simulation Success Rate

	Rotate Valve	Pick & Place	Peg in Hole	Open & Retrieve
Joint Space	50/50	38/50	9/50	8/50
EE	50/50	**48/50**	**23/50**	**17/50**

Real-world Experiments

We adopt the aerial peg-in-hole task to demonstrate our capability to derive an autonomous policy from human demonstrations for aerial manipulation in the real world. The task configurations are illustrated in Fig. 4.13 and Fig. 4.14.

We collected 25 episodes of demonstration data via human teleoperation, varying the hole's horizontal position, with each episode taking approximately 2 minutes, culminating in a total of around 50 minutes of operational data and about 2 hours of wall-clock time. The data is downsampled to 10 Hz, and the action chunk size is empirically set to 100 during the training process. After training through 100,000 epochs, the policy with the least validation loss is selected. We tested with random unseen horizontal hole positions and the learned policy successfully completed 4 out of 5 real-world peg-in-hole tests, i.e., 80% successful rate. The UAM pushed the peg forward to the edge of the hole and didn't insert it inside successfully. These results underline the potential of learning-based approaches in aerial manipulation under our developed framework, while also highlighting the need to develop robust recovery policies, especially for aerial manipulation.

4.8.6 Discussion

The experiments demonstrate our framework in end-effector trajectory tracking, aerial teleoperation, and policy learning for autonomous aerial manipulation. The precise end-effector control framework demonstrated superior end-effector tracking accuracy with minimal error. The high-precision control enables efficient, user-friendly aerial teleoperation, allowing human operators to perform multiple complex tasks, which also helps high-quality demonstration data collection. Leveraging the ee-centric framework, advanced high-level policies such as imitation learning can be easily incorporated into aerial manipulation, which further expands the development of this field.

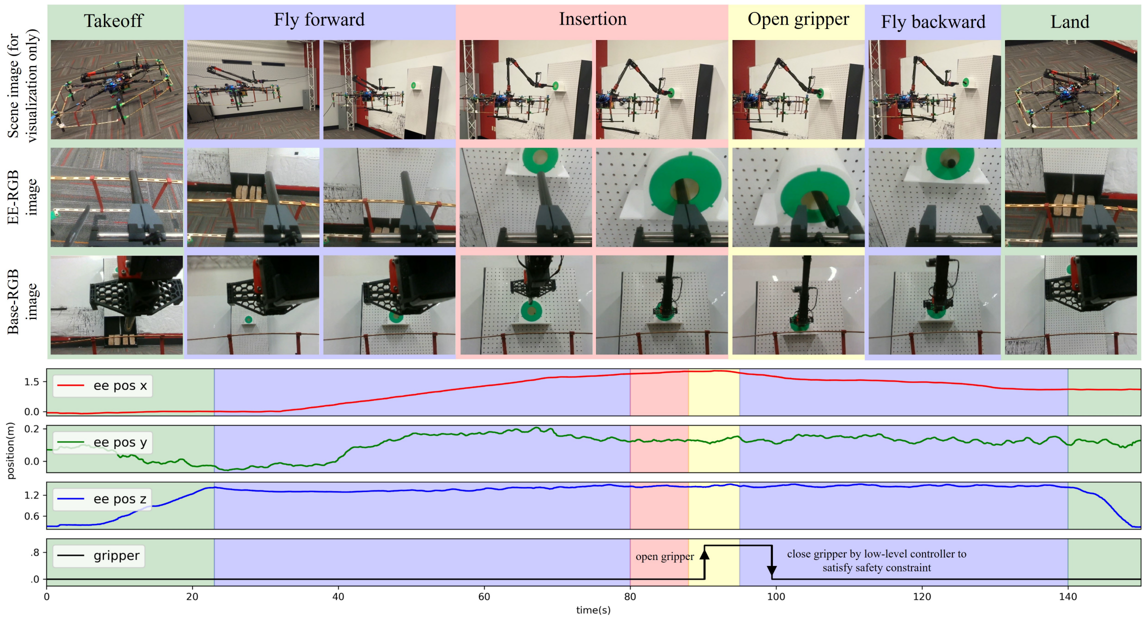

Figure 4.13: **Autonomous aerial manipulation peg-in-hole policy experiment.** The UAM inserts the peg precisely, highlighting both the accuracy of the learned policy and the low-level controller.

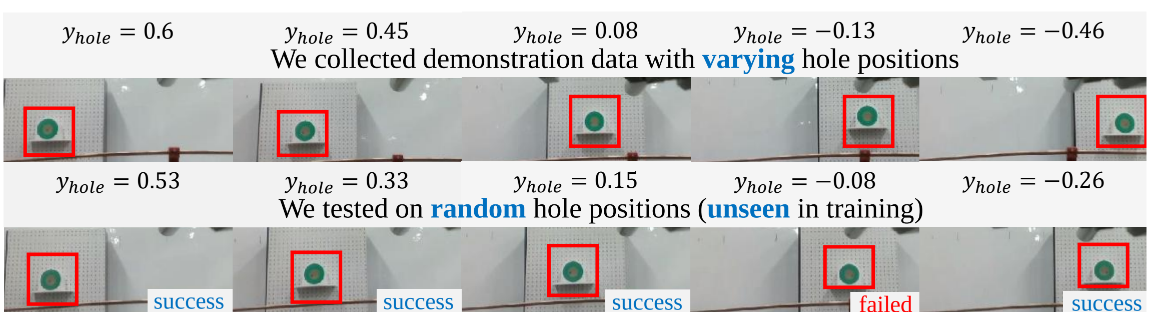

Figure 4.14: **Task scenario randomization for peg-in-hole experiments:** The figure illustrates randomized configurations used during data collection and testing to evaluate the generalization capability of the learned aerial manipulation policy.

4.9 Conclusion

This work presents a unified end-effector-centric aerial manipulation framework for versatile aerial manipulation tasks. Our system includes a versatile hardware platform that consists of a fully actuated UAV and a 4-DOF manipulator, an ee-centric whole-body MPC to ensure precise end-effector tracking, and an ee-centric high-level policy, where we developed both an intuitive teleoperation system and an imitation learning-based autonomous system. Through extensive real-world experiments, we demonstrated the system's versatility across various aerial manipulation tasks, including writing, peg-in-hole, pick-and-place, valve rotating, and light bulb replacement. More importantly, we demonstrate how this modular and standardized ee-centric framework effectively decouples the high-level policy from the low-level controller, which enables seamless integration of existing standard high-level policy modules from the broader manipulation community, such as teleoperation and imitation learning, into the field of aerial manipulation. The proposed framework achieved high precision, adaptability, and robust performance, making it a significant step toward standardizing aerial manipulation within the broader manipulation field. Future work will extend the framework's applicability to outdoor environments, incorporate onboard perception for obstacle avoidance, and further improve the end-effector tracking performance.

Chapter 5

Conclusions

This thesis presents a unified aerial manipulation framework designed to advance the capabilities of Uncrewed Aerial Manipulators (UAMs) toward general and versatile physical interaction tasks. We propose an architecture comprising three key components: a hybrid motion-force controller for compliant and robust aerial interaction, a contact-aware trajectory planner for generating dynamically feasible contact-rich motion, and an end-effector-centric interface that decouples platform-specific low-level control from high-level task planning and policy learning.

Through the development and real-world validation of this framework, we demonstrate significant improvements in control precision, system robustness, and task generality across a diverse set of aerial manipulation tasks. The hybrid motion-force controller enables compliant and dynamic interaction with contact surfaces, extending aerial manipulation beyond static tasks to dynamic surface trajectories. The contact-aware planner ensures safety and feasibility under contact constraints, while the end-effector-centric control interface facilitates intuitive human teleoperation and supports learning-based high-level policies. Importantly, the modular structure of the framework lays a foundation for scalable, cross-task, and cross-platform aerial manipulation systems, bridging the gap between task-specific aerial manipulation research and broader general-purpose manipulation approaches.

While the results achieved in this work represent a meaningful step forward, several promising directions remain for future exploration:

From Lab to Field: Future work will focus on deploying the aerial manipulation system in real-world outdoor environments, handling challenges such as GPS-denied localization, low-latency teleoperation, and environmental disturbances.

Enhancing Control Accuracy: Further research will aim to develop more robust and accurate models of contact dynamics, enabling the system to maintain precise interaction performance even under complex external disturbances.

Simplifying and Scaling the Platform: Efforts will also be directed toward designing a more modular and easily deployable hardware platform, supported by open-source hardware and software releases, to facilitate broader adoption and collaborative development within the robotics community.

We believe that the proposed unified framework, along with the identified future directions, provides a strong foundation for advancing aerial manipulation from controlled laboratory settings to real-world, autonomous, and intelligent aerial physical interaction systems.

Bibliography

[1] Arpit Agarwal, Abhiroop Ajith, Chengtao Wen, Veniamin Stryzheus, Brian Miller, Matthew Chen, Micah K Johnson, Jose Luis Susa Rincon, Justinian Rosca, and Wenzhen Yuan. Robotic defect inspection with visual and tactile perception for large-scale components. 1.1

[2] Muhannad Alkaddour, Mohammad A Jaradat, Sara Tellab, Nidal Sherif, Muhammad Alvi, Lotfi Romdhane, and Khaled S Hatamleh. Novel design of lightweight aerial manipulator for solar panel cleaning applications. *IEEE Access*, 2023. 1.1

[3] Mike Allenspach, Nicholas Lawrance, Marco Tognon, and Roland Siegwart. Towards 6dof bilateral teleoperation of an omnidirectional aerial vehicle for aerial physical interaction. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9302–9308. IEEE, 2022. 4.2.3

[4] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 2018. 3.7.1

[5] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. 4.2.2

[6] Karen Bodie, Maximilian Brunner, Michael Pantic, Stefan Walser, Patrick Pfändler, Ueli Angst, Roland Siegwart, and Juan Nieto. An omnidirectional aerial manipulation platform for contact-based inspection. *arXiv preprint arXiv:1905.03502*, 2019. 1.1, 2.2.1, 4.2.1

[7] Karen Bodie, Maximilian Brunner, Michael Pantic, Stefan Walser, Patrick Pfändler, Ueli Angst, Roland Siegwart, and Juan Nieto. Active interaction force control for contact-based inspection with a fully actuated aerial vehicle. *IEEE Transactions on Robotics*, 37(3):709–722, 2020. 2.2.1, 3.2, 4.2.1

[8] Karen Bodie, Marco Tognon, and Roland Siegwart. Dynamic end effector tracking with an omnidirectional parallel aerial manipulator. *IEEE Robotics and Automation Letters*, 6(4):8165–8172, 2021. 4.2.2

[9] Rogerio Bonatti, Wenshan Wang, Cherie Ho, Aayush Ahuja, Mirko Gschwindt, Efe Camci, Erdal Kayacan, Sanjiban Choudhury, and Sebastian Scherer. Autonomous aerial cinematography in unstructured environments with learned artistic decision-making. *Journal of Field Robotics*, 37(4):606–641, 2020. 1.1

[10] Maximilian Brunner, Livio Giacomini, Roland Siegwart, and Marco Tognon. Energy tank-based policies for robust aerial physical interaction with moving objects. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2054–2060. IEEE, 2022. 1.1, 2.2.1, 3.2, 4.2.1

[11] Maximilian Brunner, Giuseppe Rizzi, Matthias Studiger, Roland Siegwart, and Marco Tognon. A planning-and-control framework for aerial manipulation of articulated objects. *IEEE Robotics and Automation Letters*, 7(4):10689–10696, 2022. 4.2.1

[12] Jeonghyun Byun, Junha Kim, Dohyun Eom, Dongjae Lee, Changhyeon Kim, and H Jin Kim. Image-based time-varying contact force control of aerial manipulator using robust impedance filter. *IEEE Robotics and Automation Letters*, 2024. 3.2

[13] Fredrik Båberg, Yuquan Wang, Sergio Caccamo, and Petter Ögren. Adaptive object centered teleoperation control of a mobile manipulator. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 455–461, 2016. doi: 10.1109/ICRA.2016.7487166. 4.2.2

[14] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023. 4.2.3

[15] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots, 2024. URL `https://arxiv.org/abs/2402.10329`. 1.1, 4.2.2

[16] Andre Coelho, Yuri Sarkisov, Xuwei Wu, Hrishik Mishra, Harsimran Singh, Alexander Dietrich, Antonio Franchi, Konstantin Kondak, and Christian Ott. Whole-body teleoperation and shared control of redundant robots with applications to aerial manipulation. *Journal of Intelligent & Robotic Systems*, 102:1–22, 2021. 4.2.3

[17] Eugenio Cuniato, Nicholas Lawrance, Marco Tognon, and Roland Siegwart. Power-based safety layer for aerial vehicles in physical interaction using lyapunov exponents. *IEEE Robotics and Automation Letters*, 7(3):6774–6781, 2022. 1.1, 2.2.1

[18] Jacques Denavit and Richard S Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. 1955. 4.5.3

[19] Figure AI. Helix: A vision-language-action model for generalist humanoid control. `https://www.figure.ai/news/helix`, February 2025. Accessed: 2025-04-23. 4.2.2

[20] Jeremy A Fishel and Gerald E Loeb. Sensing tactile microvibrations with the biotac—comparison with human sensitivity. In *2012 4th IEEE RAS & EMBS international conference on biomedical robotics and biomechatronics (BioRob)*, pages 1122–1127. IEEE, 2012. 2.2.2

[21] Joshua Fishman and Luca Carlone. Control and trajectory optimization for soft aerial manipulation. In *2021 IEEE Aerospace Conference (50100)*, pages 1–17. IEEE, 2021. 3.2

[22] Joshua Fishman, Samuel Ubellacker, Nathan Hughes, and Luca Carlone. Dynamic grasping with a" soft" drone: From theory to practice. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4214–4221. IEEE, 2021. 4.2.1

[23] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024. 4.2.2

[24] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation, 2024. URL `https://arxiv.org/abs/2401.02117`. 4.2.3

[25] Junyi Geng and Jack W Langelaan. Cooperative transport of a slung load using load-leading control. *Journal of Guidance, Control, and Dynamics*, 43(7):1313–1331, 2020. 1.1

[26] Xiaofeng Guo, Guanqi He, Mohammadreza Mousaei, Junyi Geng, Guanya Shi, and Sebastian Scherer. Aerial interaction with tactile sensing. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1576–1582. IEEE, 2024. 1.1, 3.3.2, 3.5, 3.5.2, 3.7.2

[27] Xiaofeng Guo, Guanqi He, Mohammadreza Mousaei, Junyi Geng, Guanya Shi, and Sebastian Scherer. Aerial interaction with tactile sensing. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1576–1582. IEEE, 2024. 4.2.1, 4.4.1

[28] Xiaofeng Guo, Guanqi He, Jiahe Xu, Mohammadreza Mousaei, Junyi Geng, Sebastian Scherer, and Guanya Shi. Flying calligrapher: Contact-aware motion and force planning and control for aerial manipulation. *IEEE Robotics and Automation Letters*, 2024. 4.2.2, 4.4.1

[29] Huy Ha, Yihuai Gao, Zipeng Fu, Jie Tan, and Shuran Song. Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers, 2024. URL `https://arxiv.org/abs/2407.10353`. 4.2.2

[30] Guanqi He, Yash Jangir, Junyi Geng, Mohammadreza Mousaei, Dongwei Bai, and Sebastian Scherer. Image-based visual servo control for aerial manipulation using a fully-actuated uav. In *2023 IEEE/RSJ International Conference on Intelligent Robots*

and Systems (IROS), pages 5042–5049. IEEE, 2023. 2.2.1, 2.3.1, 2.3.4, 2.3.4, 3.3.4, 4.2.1, 4.4.1

[31] Guanqi He, Yogita Choudhary, and Guanya Shi. Self-supervised meta-learning for all-layer dnn-based adaptive control with stability guarantees. *arXiv preprint arXiv:2410.07575*, 2024. 4.6.2

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2.3.2

[33] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024. 4.2.2

[34] Tairan He, Wenli Xiao, Toru Lin, Zhengyi Luo, Zhenjia Xu, Zhenyu Jiang, Jan Kautz, Changliu Liu, Guanya Shi, Xiaolong Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. *arXiv preprint arXiv:2410.21229*, 2024. 4.2.2

[35] Francois R Hogan, Maria Bauza, Oleguer Canal, Elliott Donlon, and Alberto Rodriguez. Tactile regrasp: Grasp adjustments via simulated tactile transformations. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2963–2970. IEEE, 2018. 2.2.2

[36] Francois R Hogan, Jose Ballester, Siyuan Dong, and Alberto Rodriguez. Tactile dexterity: Manipulation primitives with tactile feedback. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 8863–8869. IEEE, 2020. 2.2.2

[37] Daniel Honerkamp, Tim Welschehold, and Abhinav Valada. Learning kinematic feasibility for mobile manipulation through deep reinforcement learning. *IEEE Robotics and Automation Letters (RA-L)*, 2021. 4.2.2

[38] Daniel Honerkamp, Tim Welschehold, and Abhinav Valada. N^2m^2: Learning navigation for arbitrary mobile manipulation motions in unseen and dynamic environments. *IEEE Transactions on Robotics*, 2023. doi: 10.1109/TRO.2023.3284346. 4.2.2

[39] Hung-Jui Huang, Xiaofeng Guo, and Wenzhen Yuan. Understanding dynamic tactile sensing for liquid property estimation. *arXiv preprint arXiv:2205.08771*, 2022. 2.2.2

[40] Kevin Huang, Rwik Rana, Alexander Spitzer, Guanya Shi, and Byron Boots. Datt: Deep adaptive trajectory tracking for quadrotor control. In *Conference on Robot Learning*, pages 326–340. PMLR, 2023. 4.6.2

[41] Shubham Kanitkar, Helen Jiang, and Wenzhen Yuanl. Poseit: A visual-tactile dataset of holding poses for grasp stability analysis. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 71–78. IEEE, 2022. 1.1, 2.2.2

[42] Azarakhsh Keipour. *Physical interaction and manipulation of the environment using aerial robots*. PhD thesis, Carnegie Mellon University, 2022. 2.3.1, 2.3.4

[43] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987. 1.1

[44] Suseong Kim, Seungwon Choi, and H Jin Kim. Aerial manipulation using a quadrotor with a two dof robotic arm. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4990–4995. IEEE, 2013. 1.1

[45] Shaunak Kishore. Make me a hanzi. `https://github.com/skishore/makemeahanzi`, 2018. 3.6.2

[46] Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *Journal of machine learning research*, 22(30):1–82, 2021. 1.1

[47] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3): 3838–3845, 2020. 2.2.2

[48] Christian Lanegger, Marco Ruggia, Marco Tognon, Lionel Ott, and Roland Siegwart. Aerial layouting: Design and control of a compliant and actuated end-effector for precise in-flight marking on ceilings. *Proceedings of Robotics: Science and System XVIII*, page p073, 2022. 2.2.1, 3.2, 4.2.1

[49] Dongjae Lee, Hoseong Seo, Dabin Kim, and H Jin Kim. Aerial manipulation using model predictive control for opening a hinged door. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1237–1242. IEEE, 2020. 1.1

[50] Hyeonbeom Lee, Hyoin Kim, and H Jin Kim. Planning and control for collision-free cooperative aerial transportation. *IEEE Transactions on Automation Science and Engineering*, 15(1):189–201, 2016. 3.2

[51] KM Lynch. *Modern Robotics*. Cambridge University Press, 2017. 3.3.3

[52] Matthew T Mason. Toward robotic manipulation. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):1–28, 2018. 1.1

[53] Daniel Mellinger, Quentin Lindsey, Michael Shomin, and Vijay Kumar. Design, modeling, estimation and control for aerial grasping and manipulation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2668–2673. IEEE, 2011. 1.1

[54] Jiawei Meng, Joao Buzzatto, Yuanchang Liu, and Minas Liarokapis. On aerial robots with grasping and perching capabilities: A comprehensive review. *Frontiers in*

Robotics and AI, 8:739173, 2022. 4.2.1

[55] UM Rao Mogili and BBVL Deepak. Review on application of drone systems in precision agriculture. *Procedia computer science*, 133:502–509, 2018. 1.1

[56] Gabriele Nava, Quentin Sablé, Marco Tognon, Daniele Pucci, and Antonio Franchi. Direct force feedback control and online multi-task optimization for aerial manipulators. *IEEE Robotics and Automation Letters*, 5(2):331–338, 2020. doi: 10.1109/LRA.2019.2958473. 4.8.1

[57] Anibal Ollero, Marco Tognon, Alejandro Suarez, Dongjun Lee, and Antonio Franchi. Past, present, and future of aerial robotic manipulators. *IEEE Transactions on Robotics*, 38(1):626–645, 2021. 1.1, 4.2.1

[58] Michael O'Connell, Guanya Shi, Xichen Shi, Kamyar Azizzadenesheli, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics*, 7(66):eabm6597, 2022. 1.1, 4.6.2

[59] Lazar Peric, Maximilian Brunner, Karen Bodie, Marco Tognon, and Roland Siegwart. Direct force and pose nmpc with multiple interaction modes for aerial push-and-slide operations. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 131–137. IEEE, 2021. 3.2

[60] William RT Roderick, Mark R Cutkosky, and David Lentink. Bird-inspired dynamic grasping and perching in arboreal environments. *Science Robotics*, 6(61):eabj7562, 2021. 4.2.1

[61] Micha Schuster, David Bernstein, Paul Reck, Salua Hamaza, and Michael Beitelschmidt. Automated aerial screwing with a fully actuated aerial manipulator. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3340–3347, 2022. doi: 10.1109/IROS47612.2022.9981979. 4.2.1

[62] Micha Schuster, David Bernstein, Paul Reck, Salua Hamaza, and Michael Beitelschmidt. Automated aerial screwing with a fully actuated aerial manipulator. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3340–3347. IEEE, 2022. 1.1

[63] Yu She, Shaoxiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. Cable manipulation with a tactile-reactive gripper. *The International Journal of Robotics Research*, 40(12-14):1385–1401, 2021. 2.2.2

[64] Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural lander: Stable drone landing control using learned dynamics. In *2019 international conference on robotics and automation (icra)*, pages 9784–9790. IEEE, 2019. 1.1

[65] Yao Su, Jiarui Li, Ziyuan Jiao, Meng Wang, Chi Chu, Hang Li, Yixin Zhu, and Hangxin Liu. Sequential manipulation planning for over-actuated unmanned aerial

manipulators. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6905–6911. IEEE, 2023. 3.2, 4.2.1

[66] Alejandro Suarez, Victor M Vega, Manuel Fernandez, Guillermo Heredia, and Anibal Ollero. Benchmarks for aerial manipulation. *IEEE Robotics and Automation Letters*, 5(2):2650–2657, 2020. 1.1

[67] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025. 4.2.2

[68] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 1.1

[69] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109. 4.8.5

[70] Dimos Tzoumanikas, Felix Graule, Qingyue Yan, Dhruv Shah, Marija Popovic, and Stefan Leutenegger. Aerial manipulation using hybrid force and position nmpc applied to aerial writing. *arXiv preprint arXiv:2006.02116*, 2020. 2.2.1, 3.2, 3.7.4, 4.2.1

[71] Samuel Ubellacker, Aaron Ray, James M. Bern, Jared Strader, and Luca Carlone. High-speed aerial grasping using a soft drone with onboard perception. *npj Robotics*, 2(1):5, 2024. ISSN 2731-4278. doi: 10.1038/s44182-024-00012-1. URL `https://doi.org/10.1038/s44182-024-00012-1`. 4.2.1

[72] Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Jonathan Frey, Niels van Duijkeren, Andrea Zanelli, Branimir Novoselnik, Thivaharan Albin, Rien Quirynen, and Moritz Diehl. acados – a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation*, 2021. 4.8.2

[73] Maria Bauza Villalonga, Alberto Rodriguez, Bryan Lim, Eric Valls, and Theo Sechopoulos. Tactile object pose estimation from the first touch with geometric contact rendering. In *Conference on Robot Learning*, pages 1015–1029. PMLR, 2021. 2.2.2

[74] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 2006. 3.7.1

[75] Chen Wang, Shaoxiong Wang, Branden Romero, Filipe Veiga, and Edward Adelson. Swingbot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5633–5640. IEEE, 2020. 2.2.2

[76] Meng Wang, Zeshuai Chen, Kexin Guo, Xiang Yu, Youmin Zhang, Lei Guo, and

Wei Wang. Millimeter-level pick and peg-in-hole task achieved by aerial manipulator. *IEEE Transactions on Robotics*, 2023. 4.2.1

[77] Meng Wang, Zeshuai Chen, Kexin Guo, Xiang Yu, Youmin Zhang, Lei Guo, and Wei Wang. Millimeter-level pick and peg-in-hole task achieved by aerial manipulator. *IEEE Transactions on Robotics*, 40:1242–1260, 2024. doi: 10.1109/TRO.2023.3338956. 3.2

[78] Zhepei Wang, Xin Zhou, Chao Xu, and Fei Gao. Geometrically constrained trajectory optimization for multicopters. *IEEE Transactions on Robotics*, 38(5):3259–3278, 2022. 3.2

[79] Zhuohuan Wu, Sheng Cheng, Kasey A. Ackerman, Aditya Gahlawat, Arun Lakshmanan, Pan Zhao, and Naira Hovakimyan. L1adaptive augmentation for geometric tracking control of quadrotors. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1329–1336, 2022. doi: 10.1109/ICRA46639.2022.9811946. 4.6.2

[80] Ruihan Yang, Yejin Kim, Rose Hendrix, Aniruddha Kembhavi, Xiaolong Wang, and Kiana Ehsani. Harmonic mobile manipulation, 2024. URL `https://arxiv.org/abs/2312.06639`. 4.2.2

[81] Grigoriy A Yashin, Daria Trinitatova, Ruslan T Agishev, Roman Ibrahimov, and Dzmitry Tsetserukou. Aerovr: Virtual reality-based teleoperation with tactile feedback for aerial manipulation. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 767–772. IEEE, 2019. 4.2.3, 4.8.4

[82] Wenzhen Yuan, Rui Li, Mandayam A Srinivasan, and Edward H Adelson. Measurement of shear and slip with a gelsight tactile sensor. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 304–311. IEEE, 2015. 2.2.2

[83] Wenzhen Yuan, Mandayam A Srinivasan, and Edward H Adelson. Estimating object hardness with a gelsight touch sensor. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 208–215. IEEE, 2016. 2.2.2

[84] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017. 1.1, 2.2.2, 2.3.1, 2.3.2, 2.4.2

[85] Wenzhen Yuan, Yuchen Mo, Shaoxiong Wang, and Edward H Adelson. Active clothing material perception using tactile sensing and deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4842–4849. IEEE, 2018. 2.2.2

[86] Jianke Zhang, Yanjiang Guo, Xiaoyu Chen, Yen-Jen Wang, Yucheng Hu, Chengming Shi, and Jianyu Chen. Hirt: Enhancing robotic control with hierarchical robot transformers. *arXiv preprint arXiv:2410.05273*, 2024. 4.2.2

[87] Weixuan Zhang, Lionel Ott, Marco Tognon, and Roland Siegwart. Learning variable impedance control for aerial sliding on uneven heterogeneous surfaces by proprioceptive and tactile sensing. *IEEE Robotics and Automation Letters*, 7(4):11275–11282, 2022. 3.2

[88] Shibo Zhao, Hengrui Zhang, Peng Wang, Lucas Nogueira, and Sebastian Scherer. Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8729–8736. IEEE, 2021. 4.4.3

[89] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. (document), 4.2, 4.2.3, 4.7.2

[90] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL `https://arxiv.org/abs/2304.13705`. 4.3