

# Sequential Manipulation Planning for Over-actuated UAMs

Yao Su<sup>1\*</sup>, Jiarui Li<sup>1,2\*</sup>, Ziyuan Jiao<sup>1\*</sup>, Meng Wang<sup>1</sup>, Chi Chu<sup>1,3</sup>, Hang Li<sup>1</sup>, Yixin Zhu<sup>4</sup>, Hangxin Liu<sup>1</sup>

**Abstract**—We investigate the *sequential* manipulation planning problem for unmanned aerial manipulators (UAMs). Unlike prior UAM work that primarily focuses on *one-step* manipulation tasks, *sequential* manipulations require coordinated motions of the floating base, the manipulator, and the object being manipulated, entailing a unified kinematics and dynamics model for motion planning under designated constraints. By leveraging a VKC-based motion planning framework that consolidates components’ kinematics into one chain, the sequential manipulation task of a UAM can be planned as a whole with more coordinated motions. Integrating the kinematics and dynamics models with a hierarchical control framework, we demonstrate, for the first time, an over-actuated UAM achieves a series of new sequential manipulation capabilities in both simulation and experiment.

## I. INTRODUCTION

Combining the agility of unmanned aerial vehicles (UAVs) and the flexibility of manipulators, UAMs can conduct manipulation tasks across rough terrains and in regions unreachable by ground robots [1–3]. Oftentimes, a fully- or even over-actuated UAV serves as the UAMs’ flying vehicle [4–6]; this type of UAVs can track position and orientation independently to provide the UAM with more agile maneuver, achieve a larger reachable workspace, and obtain better dynamic properties compared with traditional multirotors. Existing UAMs leverage a bi-level schema by combining (i) a controller to stabilize the system and track the desired trajectory under forceful contacts with the environment and (ii) a motion planner to produce trajectories satisfying task-related constraints. Such a bi-level schema of UAMs has succeeded in a wide range of aerial manipulation tasks, such as pick-and-place [3, 7], inspection [8, 9], valve operation [10], and door-like articulated object manipulation [11, 12].

To date, UAMs are limited to tasks with **one-step** planning. To endow UAMs with **multi-step** sequential manipulation capability, UAM platform ought to (i) coordinate the motions of its floating base and the manipulator that consists of a series of revolute/prismatic joints, and (ii) effectively produce varied motion patterns at different steps of a sequential task, especially when it interacts with objects with diverse kinematic structures. Developing such a sequential manipulation planning schema for UAMs remains an unexplored topic.

\* Yao Su, Jiarui Li, and Ziyuan Jiao contributed equally to this work.

<sup>1</sup> National Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI). <sup>2</sup> Department of Advanced Manufacturing and Robotics, College of Engineering, Peking University. <sup>3</sup> Department of Automation, Tsinghua University. <sup>4</sup> Institute for Artificial Intelligence, Peking University. Emails: {suyao, lijiarui, jiaoziyuan, wangmeng, chuchi, lihang}@bigai.ai, yixin.zhu@pku.edu.cn, liuhx@bigai.ai

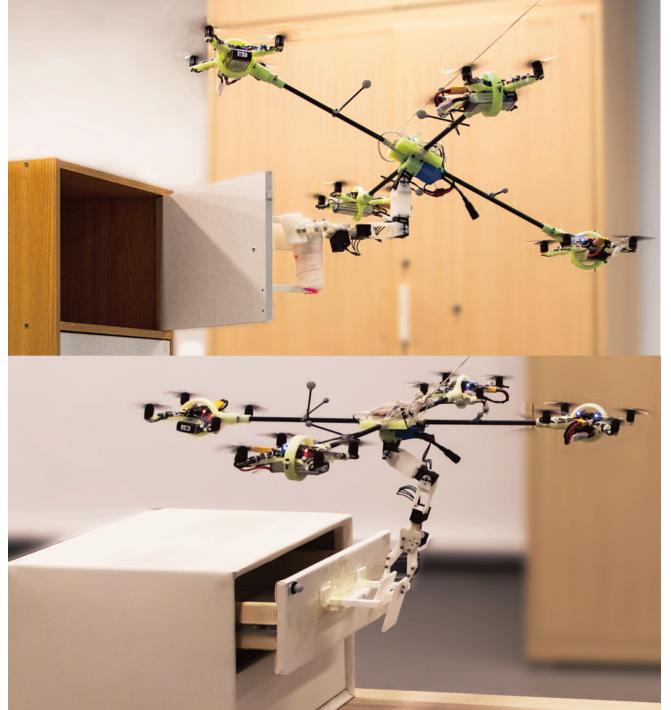


Fig. 1: Two sequential manipulation tasks completed by the UAM. They require an over-actuated UAM platform for more agile motion, a VKC modeling technique for manipulation planning, and an effective hierarchical control algorithm.

Planning sequential manipulation is challenging even for ground mobile manipulators [14–16]. In particular, consolidating the kinematics of the mobile base, the manipulator, and the manipulated object into one kinematics chain—constructing a virtual kinematic chain (VKC)—emerges as an effective means; it plans the mobile manipulator as a whole, yielding more coordinated manipulations [17–20].

Inspired by VKC, we extend the whole-body sequential manipulation from ground robots to UAMs. Here, “whole-body” refers to the unification of the trajectory planning for the floating base and the motion planning for the manipulator. First, we devise a novel UAM [21–23] by integrating a 4-degree-of-freedom (DoF) manipulator with an over-actuated UAV that can be easily replicated by composing four modular quadcopters. Next, through a dedicated nullspace-based control allocation framework [21, 22], this new UAM platform possesses high thrust efficiency, can achieve arbitrary attitudes control, and is robust against controller sampling frequency and measurement noise. Finally, after inserting *virtual* linkages and joints and abstracting the object being manipulated by its kinematic structure, we derive the (virtual) kinematics and dynamics of this new UAM and solve

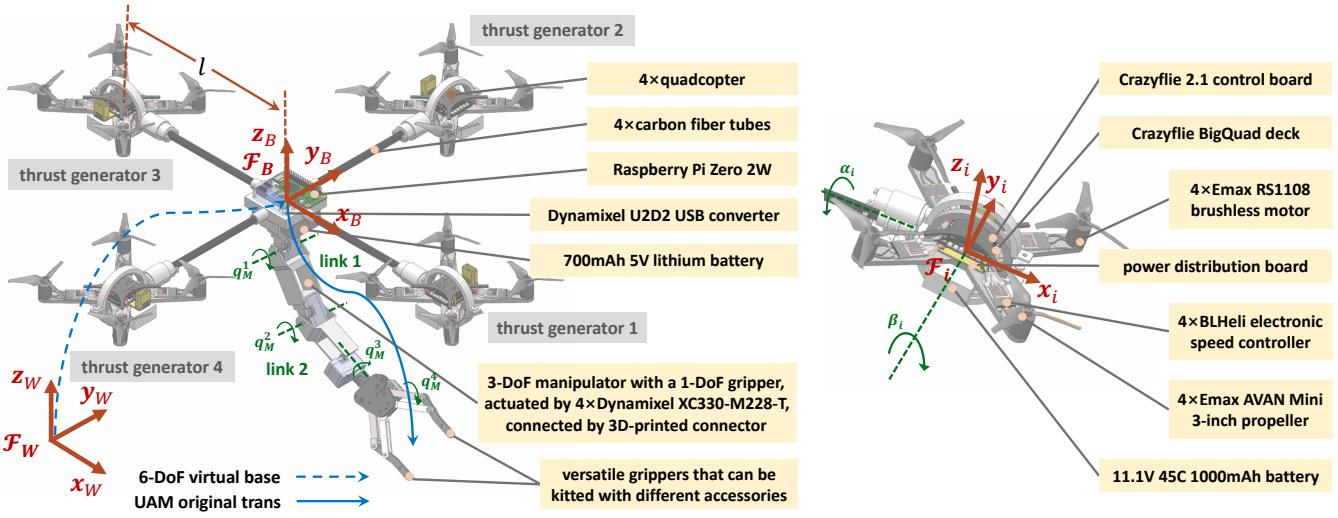


Fig. 2: **Hardware design and attached coordinate frames of our over-actuated UAM platform.** The flying vehicle consists of four omnidirectional thrust generators [13]. Each thrust generator has a 2-DoF passive gimbal mechanism and a quadcopter for over-actuation. The manipulator has three serial rotational DoFs and a parallel gripper.

the corresponding motion planning problems on the VKC via trajectory optimization [17, 18]. A hierarchical control architecture is further developed.

Together, our new UAM platform with VKC-based planning framework and a hierarchical control architecture demonstrate various sequential aerial manipulation tasks involving multiple steps in simulations and physical experiments. Fig. 1 illustrates an example of relocating an object into a closed drawer and a closed cabinet, requiring six manipulation steps. For the first time, this task demonstrates the plausibility and potential of planning sequential aerial manipulation tasks for UAMs.

This paper makes three contributions: (i) We put forward a novel mechanical design of an over-actuated UAM and derive the dynamics model of the system. (ii) We devise a manipulation planning and hierarchical control framework for our UAM. (iii) We demonstrate the entire pipeline’s sequential manipulation capabilities in both simulations and experiments.

The remainder of the paper is organized as follows. Sec. II presents the hardware design of our UAM platform. Secs. III to V describe the system dynamics, manipulation planning, and control framework of this platform, respectively. Simulation and experimental results are summarized in Secs. VI and VII, respectively. Sec. VIII concludes the paper.

## II. HARDWARE DESIGN

As shown in Fig. 2, our UAM platform consists of an over-actuated flying vehicle and a 4-DoF robotic manipulator connected at the bottom; it weights 1.21 kg with a maximum payload of 3 kg. Due to limited computing power, the platform’s controller runs on a remote PC that sends commands to the platform wirelessly.

### A. Flying Vehicle

The flying vehicle’s central frame is a rigid body made of a resin block fixed with four carbon-fiber tubes. Each tube

is connected to an omnidirectional thrust generator with two added passive DoFs to a generic quadcopter through a 3D-printed gimbal mechanism; see Fig. 2. Each quadcopter comprises a Crazyflie 2.1 control board, a Bitcraze’s BigQuad Deck, and a power distribution board connected to an 11.1v Lithium battery. Four Emax RS1108 brushless motors actuate 3-inch propellers with a maximum thrust force of  $t_{\max} = 2.6N$ . Their speeds are controlled by an Electronic Speed Controller (ESC). This flying vehicle has been fully verified in prior work [22, 23], capable of independently tracking 6-DoF position and attitude trajectory and achieving arbitrary attitude rotations with high thrust efficiency.

### B. Robotic Manipulator

The robotic manipulator is installed at the bottom of the flying vehicle. It comprises three serial rotational DoFs and a parallel gripper. Four Dynamixel XC330-M228-T motors are utilized to actuate the manipulator, and a Raspberry Pi Zero (RPi Zero) and a Dynamixel U2D2 converter are equipped on the flying vehicles to receive the control command wirelessly; RPi Zero then send these signals to control the motors. The manipulator subsystem is powered by a 5 V battery. The system’s physical properties are tabulated in Tab. I.

## III. DYNAMICS MODELLING

The complete dynamics model of the UAM platform is too complex for controller design, while dividing it into two decoupled subsystems—the arm and the flying vehicle—introduces severe disturbance to the platform. As a result, we simplify the flying vehicle’s dynamics by concentrating on compensating the gravity torque introduced by the shift of center of mass (CoM) when the manipulator is in motion.

### A. Platform Configuration and Notation

Fig. 2 illustrates related coordination frames. The world frame and the UAM’s body frame are denoted as  $F_W$  and  $F_B$ , respectively. We define the body frame’s position as  $\mathbf{p} = [x, y, z]^T$ , the attitude in the roll-pitch-yaw convention

TABLE I: **Physical Parameters of the UAM platform.**  $m_0$  and  $I_0$  denote the mass and inertia matrix of the flying vehicle’s mainframe, respectively.  $m_i$  and  $I_i$  denote each thrust generator’s mass and inertia matrix, respectively.  $m_M^j$  and  $I_M^j$  denote the mass and inertia matrix of the manipulator link  $j$ , respectively.

Group	Parameter	Value
flying vehicle	$m_B^0/kg$	0.168
	$m_B^i/kg$	0.222
	$\text{diag}(I_B^0)/kg \cdot cm^2$	[0.30 0.30 0.60]
	$\text{diag}(I_B^i)/kg \cdot cm^2$	[2.23 2.84 4.51]
	$l/m$	0.21
	$t_{max}/N$	2.6
manipulator	$m_M^1/kg$	0.044
	$m_M^2/kg$	0.040
	$m_M^3/kg$	0.043
	$\text{diag}(I_M^1)/kg \cdot cm^2$	[0.22 0.21 0.04]
	$\text{diag}(I_M^2)/kg \cdot cm^2$	[0.22 0.19 0.06]
	$\text{diag}(I_M^3)/kg \cdot cm^2$	[0.82 0.80 0.15]
others	gripper range/mm	4 – 35
	remote PC control rate/Hz	100
	quadcopter control rate/Hz	500
	manipulator control rate/Hz	500
	communication delay/ms	20

as  $\boldsymbol{\theta} = [\phi, \theta, \psi]^T$ , and the angular velocity in  $\mathcal{F}_B$  as  $\boldsymbol{\omega} = [p, q, r]^T$  [24]. Frame  $\mathcal{F}_i$  is attached to the  $i$ th thrust generator’s center. We combine the flying vehicle’s pose and velocity as  $\mathbf{q}_B = [{}^W\mathbf{p}^T, {}^B\boldsymbol{\theta}^T]^T$  and  $\dot{\mathbf{q}}_B = [{}^W\mathbf{v}^T, {}^B\boldsymbol{\omega}^T]^T$ . Let  $\mathbf{q}_M \in \mathbb{R}^{4 \times 1}$  be the manipulator’s joint angles.

### B. Flying Vehicle Dynamics

The dynamics model of the flying vehicle is simplified as:

$$\ddot{\mathbf{q}}_B = \begin{bmatrix} \frac{1}{m} {}^W\mathbf{R} & 0 \\ 0 & {}^B\mathbf{J}(\mathbf{q}_M)^{-1} \end{bmatrix} \mathbf{u} + \begin{bmatrix} g\hat{\mathbf{z}} \\ {}^B\boldsymbol{\tau}_g(\mathbf{q}_M) - {}^B\boldsymbol{\omega} \times {}^B\mathbf{J}(\mathbf{q}_M) {}^B\boldsymbol{\omega} \end{bmatrix}, \quad (1)$$

where  $g$  is the gravitational acceleration,  $m$  the whole platform’s total mass,  $\mathbf{J}$  the whole platform’s inertia matrix,  ${}^B\boldsymbol{\tau}_g$  is the gravitational torque due to the displacement of its CoM from the geometric center, and  $\hat{\mathbf{z}} = [0, 0, 1]^T$  is the unit vector in the vertical direction in the world frame. Of note,  $\mathbf{J}$  and  ${}^B\boldsymbol{\tau}_g$  are functions of the manipulator’s joint angles  $\mathbf{q}_M$  defined by kinematic relationships. And

$$\mathbf{u} = \begin{bmatrix} \sum_{i=1}^4 {}^B_i\mathbf{R} T_i \hat{\mathbf{z}} \\ \sum_{i=1}^4 (\mathbf{d}_i \times {}^B_i\mathbf{R} T_i \hat{\mathbf{z}}) \end{bmatrix} = \begin{bmatrix} \mathbf{J}_v(\alpha, \beta) \\ \mathbf{J}_\omega(\alpha, \beta) \end{bmatrix} T, \quad (2)$$

where  $T_i$ ,  $\alpha_i$ , and  $\beta_i$  denote the magnitude of thrust, tilting, and twisting angles of the  $i$ th thrust generator, respectively.  $\mathbf{d}_i$  is the distance vector from  $\mathcal{F}_B$ ’s center to each  $\mathcal{F}_i$ .

### C. Manipulator Dynamics

The dynamics of the manipulator are modeled following Luo *et al.* [25], formally as:

$$\mathbf{M}_M(\mathbf{q}_M)\ddot{\mathbf{q}}_M + \mathbf{C}_M(\mathbf{q}_M, \dot{\mathbf{q}}_M) + \mathbf{G}_M(\mathbf{q}_M) = \boldsymbol{\tau}_M + \mathbf{J}_{ext}\mathbf{F}_{ext}, \quad (3)$$

where  $\mathbf{M}_M \in \mathbb{R}^{4 \times 4}$  is the manipulator’s inertia matrix,  $\mathbf{C}_M \in \mathbb{R}^{4 \times 1}$  the vector of the Coriolis and centrifugal terms,  $\mathbf{G}_M \in \mathbb{R}^{4 \times 1}$  the gravitational force vector,  $\boldsymbol{\tau}_M$  the torque

command of each joint actuator,  $\mathbf{F}_{ext}$  external forces, and  $\mathbf{J}_{ext}$  the related Jacobian matrix.

## IV. SEQUENTIAL AERIAL MANIPULATION PLANNING

In this section, we start by describing three essential steps to construct VKCs [17, 18, 26] for our UAM platform. Next, we formulate the sequential manipulation planning problem on VKCs and solve it through trajectory optimization for aerial manipulation tasks.

### A. Modeling UAMs with VKCs

**Kinematic inversion** reverses the kinematic model of an articulated object by converting the attachable link into the new root in the inverted kinematic model. Of note, in addition to reversing the parent-child relationship for every two adjacent frames between the base link and the attachable link of the object, the spatial transformation of each joint must also be updated appropriately since a joint typically constrains the child link’s motion w.r.t. child link’s frame.

**Virtual joint** defines the spatial transformation between two body frames and the joint type that constrains the relative motion between them. The manipulator and the manipulated object form a single serial kinematic chain by inserting a virtual joint between the manipulator’s end-effector and an attachable link in the object model. If a manipulated object is articulate, its kinematic model has to be inverted for the constructed kinematic chain to remain serial.

**Virtual base** reflects the motion constraints imposed on the floating base. In our UAM platform, the floating base is an over-actuated UAV that can achieve free motion in space. Specifically, starting from the ground, we add three perpendicular prismatic joints for linear motion, followed by three revolute joints at the center of the UAV body frame for angular motions. These six joints together form a virtual chain that mimics the possible motions of the floating base.

A VKC for aerial manipulation planning is constructed by augmenting a virtual base to the UAM’s kinematic model; see Fig. 2. During the manipulation, the end-effector connects to the inverted object model via a virtual joint. From this VKC perspective, performing an aerial manipulation task is treated as altering the VKC’s state, equivalent to solving a motion planning problem on VKCs.

### B. Motion Planning on VKCs

The state vector  $\mathbf{x} \in \mathcal{X}_{\text{free}}$  describes the state of a VKC, where  $\mathcal{X}_{\text{free}} \in \mathbb{R}^n$  is the collision-free configuration space. The motion planning problem on VKCs is equivalent to finding a  $T$ -step path  $\mathbf{x}_{1:T} \in \mathcal{X}_{\text{free}}$ , which can be formulated and solved by trajectory optimization. Following Jiao *et al.* [17, 18], the objective function of the trajectory optimization is:

$$\min_{\mathbf{x}_{1:T}} \sum_{t=1}^{T-1} \|\mathbf{W}_v \delta \mathbf{x}_t\|_2^2 + \sum_{t=2}^{T-1} \|\mathbf{W}_a \delta \dot{\mathbf{x}}_t\|_2^2, \quad (4)$$

where we penalize the overall traveled distance and overall smoothness of the trajectory  $\mathbf{x}_{1:T}$ .  $\mathbf{W}_v$  and  $\mathbf{W}_a$  are two diagonal weighting matrices for each DoF,  $\delta \mathbf{x}_t$  and  $\delta \dot{\mathbf{x}}_t$  are finite forward difference and second-order finite central difference

of  $\mathbf{x}_t$ , respectively. An equality constraint is imposed on the constructed VKC, which specifies the physical constraints of the object or the environment:

$$h_{\text{chain}}(\mathbf{x}_t) = 0, \quad \forall t = 1, 2, \dots, T. \quad (5)$$

Failing to account for this type of constraint (*e.g.*, the kinematic constraint of the manipulator or the object) may damage the UAM or the object being manipulated, resulting in failed executions.

The goal of a sequential aerial manipulation task is formulated as an inequality constraint:

$$\|f_{\text{task}}(\mathbf{x}_T) - \mathcal{G}_{\text{goal}}\|_2^2 \leq \xi_{\text{goal}}, \quad (6)$$

which bounds the final state  $\mathbf{x}_T$  of the VKC and the task goal  $\mathcal{G}_{\text{goal}} \in \mathcal{G}$  with a small tolerance  $\xi_{\text{goal}}$ . The function  $f_{\text{task}} : \mathbb{R}^n \rightarrow \mathbb{R}^k$  maps  $\mathbf{x}_T$  from the configuration space  $\mathcal{X}$  to the task-dependent goal space  $\mathcal{G} \in \mathbb{R}^k$ . For example, in an object-picking task,  $f_{\text{task}}$  represents the forward kinematics of the VKC, and  $\mathcal{G}_{\text{goal}}$  is the end-effector pose prior to grasping [27].

Additional safety constraints are further imposed on the motion planning problem:

$$\mathbf{x}_{\min} \leq \mathbf{x}_t \leq \mathbf{x}_{\max}, \quad \forall t = 1, 2, \dots, T \quad (7)$$

$$\|\delta \mathbf{x}_t\|_{\infty} \leq \dot{\mathbf{x}}_{\max}, \quad \|\delta \dot{\mathbf{x}}_t\|_{\infty} \leq \ddot{\mathbf{x}}_{\max}, \quad \forall t = 2, 3, \dots, T-1 \quad (8)$$

$$\sum_{i=1}^{N_{\text{link}}} \sum_{j=1}^{N_{\text{obj}}} |\text{dist}_{\text{safe}} - f_{\text{dist}}(L_i, O_j)|^+ \leq \xi_{\text{dist}}, \quad (9)$$

$$\sum_{i=1}^{N_{\text{link}}} \sum_{j=1}^{N_{\text{link}}} |\text{dist}_{\text{safe}} - f_{\text{dist}}(L_i, L_j)|^+ \leq \xi_{\text{dist}}, \quad (10)$$

where  $|\cdot|^+$  is defined as  $|x|^+ = \max(x, 0)$ . Equations (7) and (8) are inequality constraints that define the joint capability and implicitly constrain the workspace of a UAM. Equation (9) and Equation (10) penalize collisions with obstacles and self-collisions, respectively.  $\text{dist}_{\text{safe}}$  is a predefined safety distance, and  $f_{\text{dist}}$  is a function that calculates the signed distance between a pair of objects.

## V. CONTROL

Using a hierarchical control architecture, we devise the UAM's overall controller with two subsystems, shown in Fig. 3. The high-level controller calculates the desired commands for trajectory tracking remotely and sends them wirelessly to the low-level controller that runs on the platform with high frequency.

### A. Flying Vehicle Control

**High-level control:** The feedback-linearization method is applied to Equation (1) to transfer the nonlinear system dynamics to a linear double integrator [28–30]:

$$\mathbf{u}^d = \begin{bmatrix} m^W_B R^T & \mathbf{0} \\ \mathbf{0} & {}^B J(\mathbf{q}_M) \end{bmatrix} \left( \begin{bmatrix} \mathbf{u}_v \\ \mathbf{u}_{\omega} \end{bmatrix} - \begin{bmatrix} g \hat{\mathbf{z}} \\ {}^B \tau_g(\mathbf{q}_M) - {}^B \boldsymbol{\omega} \times {}^B J(\mathbf{q}_M) {}^B \boldsymbol{\omega} \end{bmatrix} \right), \quad (11)$$

where the superscript  $d$  indicates the desired values,  $\mathbf{u}_v$  and  $\mathbf{u}_{\omega}$  are two virtual inputs that can be designed with

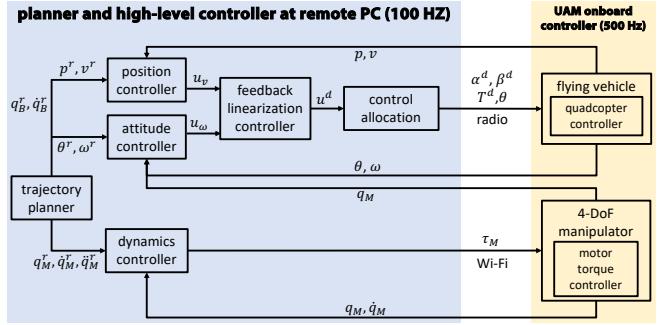


Fig. 3: **Hierarchical control architecture of the UAM platform.** The high-level controller of the flying vehicle (i) calculates desired wrench command  $\mathbf{u}^d$  for trajectory tracking and (ii) allocates it to desired thrusts and joint angles of thrust generators through control allocation. Each quadcopter has its own onboard controller to regulate the joint angles and thrust to desired values.

translational and rotational errors to track the reference position and attitude trajectory:

$$\begin{aligned} \mathbf{u}_v &= \dot{\mathbf{v}}^r + K_{v1} \mathbf{e}_v + K_{v2} \mathbf{e}_p + K_{v3} \int \mathbf{e}_p dt, \\ \mathbf{u}_{\omega} &= \dot{\boldsymbol{\omega}}^r + K_{\omega1} \mathbf{e}_{\omega} + K_{\omega2} \mathbf{e}_{\theta} + K_{\omega3} \int \mathbf{e}_{\theta} dt, \end{aligned} \quad (12)$$

where  $K_{pi}$  and  $K_{wi}$  are constant gain matrices, and the superscript  $r$  indicates the reference value from the VKC-based motion planning; see Sec. IV. The error terms are defined following Su *et al.* [31]:

$$\begin{aligned} \mathbf{e}_p &= \mathbf{p}^r - \mathbf{p}, \quad \mathbf{e}_v = \mathbf{v}^r - \mathbf{v}, \\ \mathbf{e}_{\theta} &= \frac{1}{2} [\mathbf{R}(\boldsymbol{\theta})^T \mathbf{R}(\boldsymbol{\theta}^r) - \mathbf{R}(\boldsymbol{\theta}^r)^T \mathbf{R}(\boldsymbol{\theta})]_{\vee}, \\ \mathbf{e}_{\omega} &= \mathbf{R}(\boldsymbol{\theta})^T \mathbf{R}(\boldsymbol{\theta}^r) \boldsymbol{\omega}^r - \boldsymbol{\omega}, \end{aligned} \quad (13)$$

where  $\mathbf{R}(\cdot)$  is the transformation from Euler angles to a standard rotation matrix, and  $[\cdot]_{\vee}$  is the mapping from  $\text{SO}(3)$  to  $\mathbb{R}^3$ . Combining Equations (11) to (13) with Equation (1), we have the error dynamics as:

$$\begin{aligned} \dot{\mathbf{e}}_v + K_{v1} \mathbf{e}_v + K_{v2} \mathbf{e}_p + K_{v3} \int \mathbf{e}_p dt &= 0, \\ \dot{\mathbf{e}}_{\omega} + K_{\omega1} \mathbf{e}_{\omega} + K_{\omega2} \mathbf{e}_{\theta} + K_{\omega3} \int \mathbf{e}_{\theta} dt &= 0, \end{aligned} \quad (14)$$

which is an asymptotically stable system.

**Control allocation and low-level control:** The control allocation solves for desired command  $\alpha_i^d$ ,  $\beta_i^d$ , and  $T_i^d$  for each 3-DoF thrust generator from total wrench command of whole flying vehicle  $\mathbf{u}^d$ . Among various approaches [21–23, 28, 32], we implement the downwash-aware control allocation method [22] to avoid the large disturbance caused by downwash flows that counteract other thrust generators while maintaining high thrust efficiency, critical for a smooth aerial manipulation, especially when interacting with an object.

In low-level control, two separated PID controllers are designed to allow each quadcopter to track the desired tilting and twisting angles,  $\alpha_i^d$  and  $\beta_i^d$ , with tilting torque commands. This is combined later with the thrust force command  $T_i^d$  to determine each actuator's angular velocity. Finally, it is converted to a PWM command to drive the actuators [23].

### B. Manipulator Control

With Equation (3), we design the manipulator's controller,

$$\tau_M = \mathbf{M}_M(\mathbf{q}_M)\ddot{\mathbf{q}}_M^d + \mathbf{C}_M(\mathbf{q}_M, \dot{\mathbf{q}}_M) + \mathbf{G}_M(\mathbf{q}_M), \quad (15)$$

where

$$\begin{aligned} \ddot{\mathbf{q}}_M^d &= \ddot{\mathbf{q}}_M^r + K_{M1}\mathbf{e}_M + K_{M2}\dot{\mathbf{e}}_M + K_{M3} \int \mathbf{e}_M dt, \\ \mathbf{e}_M &= \mathbf{q}_M^r - \mathbf{q}_M, \end{aligned} \quad (16)$$

where  $K_{Mi}$  are constant gain matrices.

## VI. SIMULATION

### A. Setup

Before conducting experiments, we develop a co-simulation platform in MATLAB Simulink and ROS/Gazebo. The ROS simulator and low-level controller are modified from an open-source quadcopter simulator CrazyS [33] using our customized hardware design and configuration with parameters listed in Tab. I.

We use *JointEffortController* in *ros\_control* to set desired torques to each joint of the manipulator. The trajectory planner is implemented in MATLAB, and the reference trajectory is fed into the high-level controller built in Simulink, where all the characteristics of the physical system are included, such as control frequencies and measurement noise. The controller communicates with ROS through *ROS Toolbox* by subscribing and publishing topics.

### B. Results

Fig. 4 and 5 summarize the simulation results of accomplishing two sequential aerial manipulation tasks—installing a light bulb and relocating an object into a closed cabinet—using the proposed manipulation planning framework.

**Task 1:** As shown in Fig. 4a, the light bulb installation task was divided into four steps in our VKC-based motion planning framework: ① approach the light bulb, ② pick it up with the manipulator, ③ flip the platform to transport the light bulb to the bottom of the target position, and ④ move up to install it. A 10-DoF VKC for the UAM platform is built—six for the flying vehicle and four for the manipulator. A feasible and collision-free reference trajectory is acquired within the physical constraints. As shown in Fig. 4b, with the hierarchical controller introduced in Sec. V, the proposed UAM platform can accurately track the reference trajectory to accomplish the task.

**Task 2:** Relocating an object into the cabinet requires a six-step action sequence of the UAM: ① approach the door and grasp the handle with the manipulator, ② open the door, ③ ungrasp the handle and move to pick up the object, ④ put the object into the cabinet, ⑤ approach and grasp to the handle again, and ⑥ close the door. Some keyframes are shown in Fig. 5, and the planned reference trajectory by the VKC-based motion planner and the tracking performance are shown in Fig. 5b.

These simulation results indicate that the VKC-based motion planning framework and the proposed UAM platform effectively achieve sequential aerial manipulation.

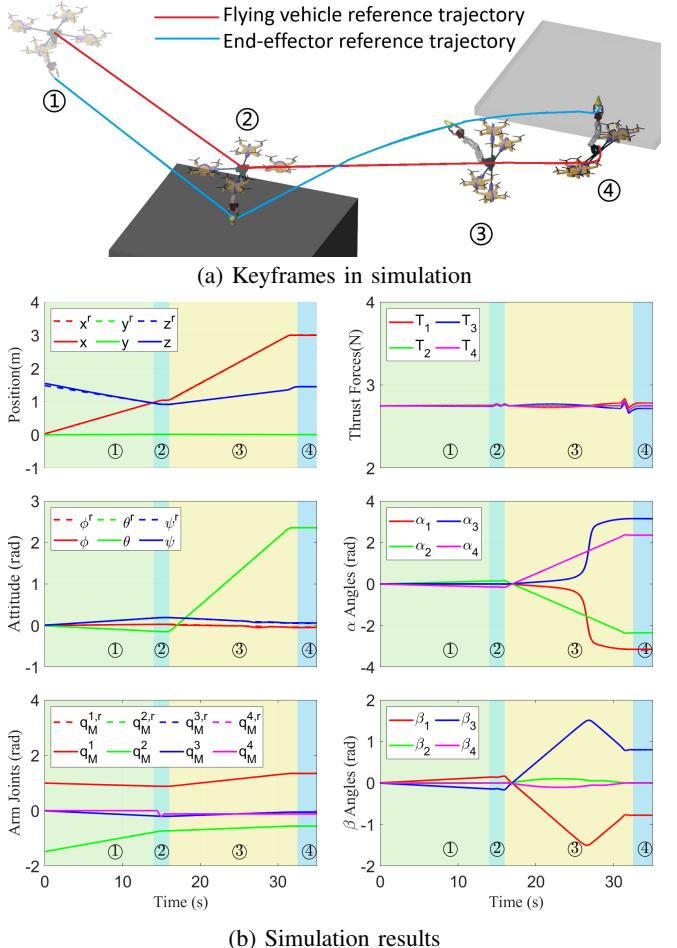


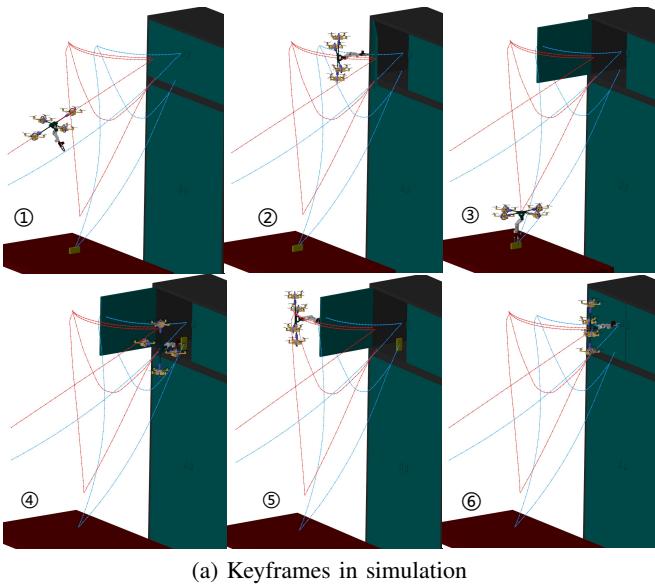
Fig. 4: **Simulation Task 1: Install a light bulb.** The action sequence of the UAM can be divided into four steps: ① approach, ② pick-up, ③ rotate and translate, and ④ feed in.

## VII. EXPERIMENT

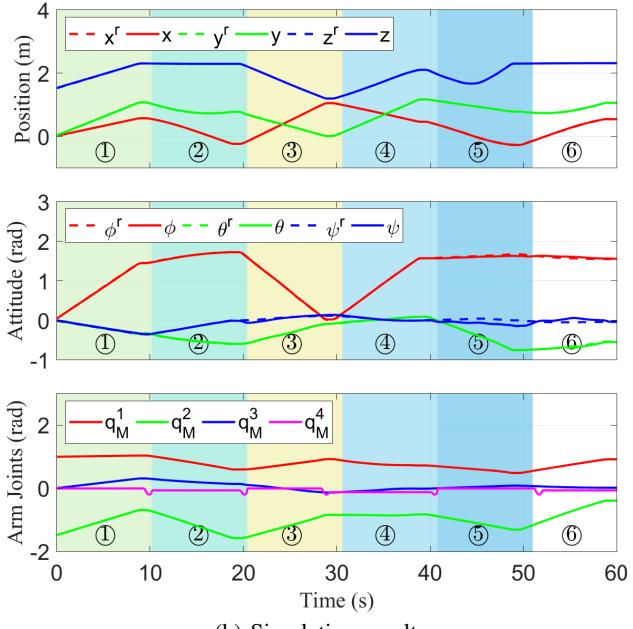
### A. Setup

To further demonstrate the sequential aerial manipulation capability, we conduct experiments by implementing a table arrangement task in the physical world. Specifically, we use the Vicon motion capture system (MoCap) to measure the position and attitude of the UAM platform. The trajectory planner and main controller of the UAM systems runs on a remote PC (AMD Ryzen9 5950X CPU, 64 GB RAM), which communicates with the MoCap through Ethernet and solves the controller commands with high efficiency. The flying vehicle's primary controller is modified from the Crazyflie python library; it calculates the desired thrust  $\mathbf{T}^d$ , tilting angles  $\alpha^d$ , and twisting angles  $\beta^d$  for all quadcopter modules of the thrust generators and sends them through Crazy Radio PA antennas (2.4G Hz). Each quadcopter is embedded with an onboard IMU module. Its firmware is modified to estimate the rotation angle given the attitude of central frame  $\theta$ , regulates the tilting and twisting angles to desired values with two PID loops, and provides the required thrust with 500 Hz for a fast low-level response.

The manipulator controller is modified from the Robotics Dynamixel SDK, which runs on the RPi Zero. It receives



(a) Keyframes in simulation



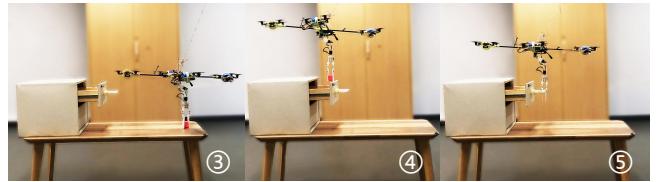
(b) Simulation results

**Fig. 5: Simulation Task 2: Relocate an object into a cabinet.** The action sequence of the UAM are divided into six steps: ① approach to the door, ② open the door, ③ pick up the object, ④ put the object into the cabinet, ⑤ approach to the door, and ⑥ close the door.

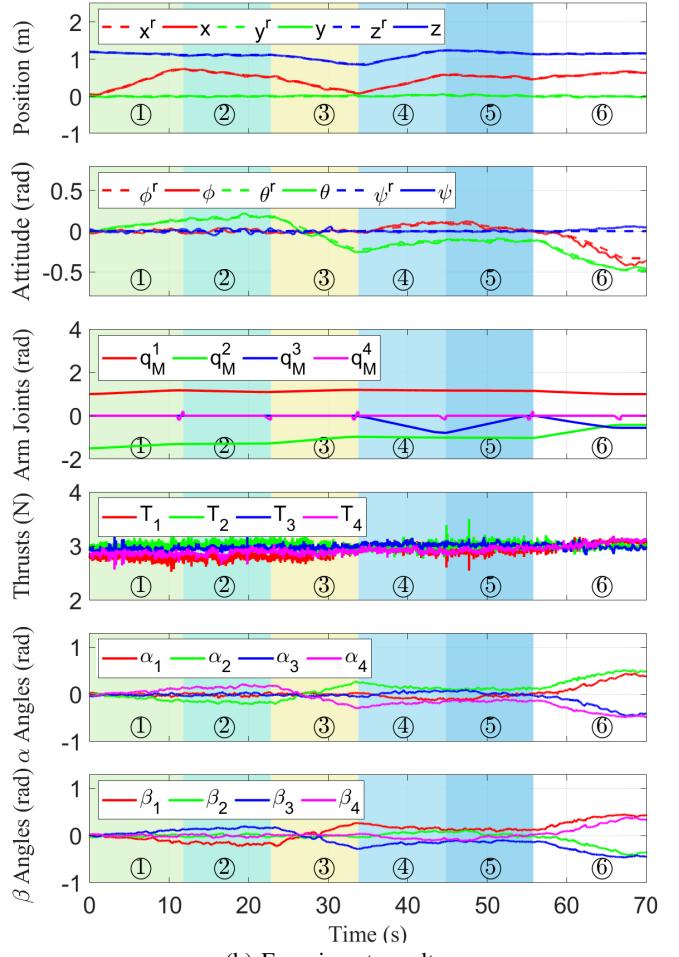
commands from the remote PC via WiFi and sends commands to the motors through Dynamixel U2D2 converter. Each Dynamixel XC330-M228-T motor has its own controller, and the control mode is set as current control. The measurement rate of the motion capture system, the remote PC controller, and the data communication with the UAM platform is all set to 100 Hz.

## B. Results

Fig. 6 summarizes the experimental results of relocating an object into the drawer, which was divided into six steps: ① approach to the drawer and grasp the handle with the manipulator, ② open the drawer, ③ ungrasp with the handle



(a) Keyframes in experiment



(b) Experiment results

**Fig. 6: Experiment Task: Relocate an object into the drawer.** The action sequence of the UAM is divided into six steps: ① approach to the drawer, ② open the drawer, ③ approach the toy and pick it up, ④ drop off the toy to the drawer, ⑤ approach to the drawer handle, and ⑥ close the drawer.

and move to pick up the toy, ④ move to the opened drawer and drop off the toy inside, ⑤ approach and grasp the handle again, and ⑥ close the drawer. The 10-DoF reference trajectory provided by our VKC-based motion planner were accurately tracked by the UAM with the hierarchical controller, and the desired commands for each omnidirectional thrust generator are plotted in Fig. 6b. Fig. 1 and 6a show some experiment keyframes.

## VIII. CONCLUSION

In this paper, we presented a solution to the sequential aerial manipulation problem of UAMs, an unexplored topic until now. Unlike previous work in UAM that solves the

motion planning and control problems of one-step manipulation tasks, accomplishing sequential aerial manipulation requires (i) a highly efficient UAM platform, (ii) a specialized motion planner that can well-coordinate motions of the flying vehicle, the manipulator, and the manipulated object under varied settings, and (iii) an effective control scheme to track the desired trajectory. To jointly tackle these challenges, we designed a novel UAM platform based on an overactuated UAV that can achieve omnidirectional flight with high thrust efficiency. To produce a long sequence of motions that coordinates well with each other, we extended the idea of VKC used for ground mobile manipulators and developed a VKC-based aerial manipulation planning framework for UAMs. Together with a hierarchical control scheme, we validated our solution in both simulation and experiment. The results demonstrated that our approach endowed a new capability of sequential aerial manipulation for UAMs and could open up new venues in the field of aerial manipulation.

**Acknowledgement:** The authors would like to thank Zeyu Zhang, Zhen Chen, Yangyang Wu, Zihang Zhao, and Hao Liang at BIGAI and Qing Lei at PKU for their help on Vicon, figures, and hardware design. This work is supported in part by the National Key R&D Program of China (2021ZD0150200) and the Beijing Nova Program.

## REFERENCES

- [1] M. Fumagalli, R. Naldi, A. Macchelli, F. Forte, A. Q. Keemink, S. Stramigioli, R. Carloni, and L. Marconi, “Developing an aerial manipulator prototype: Physical interaction with the environment,” *IEEE Robotics and Automation Magazine (RA-M)*, vol. 21, no. 3, pp. 41–50, 2014.
- [2] A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi, “Past, present, and future of aerial robotic manipulators,” *Transactions on Robotics (T-RO)*, 2021.
- [3] M. Ryll and R. K. Katzschmann, “Smors: A soft multirotor uav for multimodal locomotion and robust interaction,” in *International Conference on Robotics and Automation (ICRA)*, 2022.
- [4] G. Jiang, R. Voyles, K. Sebesta, and H. Greiner, “Estimation and optimization of fully-actuated multirotor platform with nonparallel actuation mechanism,” in *International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [5] S. Park, J. Lee, J. Ahn, M. Kim, J. Her, G.-H. Yang, and D. Lee, “Odar: Aerial manipulation platform enabling omnidirectional wrench generation,” *Transactions on Mechatronics (TMECH)*, vol. 23, no. 4, pp. 1907–1918, 2018.
- [6] S. Yi, K. Watanabe, and I. Nagai, “Modeling and control of a fully-actuated quadrotor manipulator with tiltable rotors,” in *Recent Advances in Intelligent Computational Systems (RAICS)*, 2020.
- [7] M. Zhao, K. Okada, and M. Inaba, “Versatile articulated aerial robot dragon: Aerial manipulation and grasping by vectorable thrust control,” *International Journal of Robotics Research (IJRR)*, p. 02783649221112446, 2022.
- [8] M. Tognon, E. Cataldi, H. A. T. Chavez, G. Antonelli, J. Cortés, and A. Franchi, “Control-aware motion planning for task-constrained aerial manipulation,” *Robotics and Automation Letters (RA-L)*, vol. 3, no. 3, pp. 2478–2484, 2018.
- [9] K. Bodie, M. Tognon, and R. Siegwart, “Dynamic end effector tracking with an omnidirectional parallel aerial manipulator,” *Robotics and Automation Letters (RA-L)*, vol. 6, no. 4, pp. 8165–8172, 2021.
- [10] M. Zhao, K. Nagato, K. Okada, M. Inaba, and M. Nakao, “Forceful valve manipulation with arbitrary direction by articulated aerial robot equipped with thrust vectoring apparatus,” *Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, pp. 4893–4900, 2022.
- [11] M. Brunner, G. Rizzi, M. Studiger, R. Siegwart, and M. Tognon, “A planning-and-control framework for aerial manipulation of articulated objects,” *Robotics and Automation Letters (RA-L)*, 2022.
- [12] N. Sugito, M. Zhao, T. Anzai, T. Nishio, K. Okada, and M. Inaba, “Aerial manipulation using contact with the environment by thrust vectorable multilinked aerial robot,” in *International Conference on Robotics and Automation (ICRA)*, 2022.
- [13] P. Yu, *An Over-Actuated Multi-Rotor Aerial Platform and Iterative Learning Control Applications*. PhD thesis, University of California, Los Angeles, 2022.
- [14] D. Berenson, J. Kuffner, and H. Choset, “An optimization approach to planning for mobile manipulation,” in *International Conference on Robotics and Automation (ICRA)*, 2008.
- [15] S. Chitta, B. Cohen, and M. Likhachev, “Planning for autonomous door opening with a mobile manipulator,” in *International Conference on Robotics and Automation (ICRA)*, 2010.
- [16] K. Gochev, A. Safanova, and M. Likhachev, “Planning with adaptive dimensionality for mobile manipulation,” in *International Conference on Robotics and Automation (ICRA)*, 2012.
- [17] Z. Jiao, Z. Zhang, X. Jiang, D. Han, S.-C. Zhu, Y. Zhu, and H. Liu, “Consolidating kinematic models to promote coordinated mobile manipulations,” in *International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [18] Z. Jiao, Z. Zhang, W. Wang, D. Han, S.-C. Zhu, Y. Zhu, and H. Liu, “Efficient task planning for mobile manipulation: a virtual kinematic chain perspective,” in *International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [19] J. Haviland, N. Sünderhauf, and P. Corke, “A holistic approach to reactive mobile manipulation,” *Robotics and Automation Letters (RA-L)*, 2022.
- [20] A. Röfer, G. Bartels, W. Burgard, A. Valada, and M. Beetz, “Kiniverse: A symbolic articulation model framework for model-agnostic mobile manipulation,” *Robotics and Automation Letters (RA-L)*, 2022.
- [21] Y. Su, P. Yu, M. Gerber, L. Ruan, and T.-C. Tsao, “Nullspace-based control allocation of overactuated uav platforms,” *Robotics and Automation Letters (RA-L)*, vol. 6, no. 4, pp. 8094–8101, 2021.
- [22] Y. Su, C. Chu, M. Wang, J. Li, Y. Liu, Y. Zhu, and H. Liu, “Downwash-aware control allocation for over-actuated uav platforms,” in *International Conference on Intelligent Robots and Systems (IROS)*, 2022.
- [23] P. Yu, Y. Su, M. J. Gerber, L. Ruan, and T.-C. Tsao, “An over-actuated multi-rotor aerial vehicle with unconstrained attitude angles and high thrust efficiencies,” *Robotics and Automation Letters (RA-L)*, vol. 6, no. 4, pp. 6828–6835, 2021.
- [24] C. Pi, L. Ruan, P. Yu, Y. Su, S. Cheng, and T. Tsao, “A simple six degree-of-freedom aerial vehicle built on quadcopters,” in *Proceedings of IEEE Conference on Control Technology Applications (CCTA)*, 2021.
- [25] J. Luo, Z. Gong, Y. Su, L. Ruan, Y. Zhao, H. H. Asada, and C. Fu, “Modeling and balance control of supernumerary robotic limb for overhead tasks,” *Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, pp. 4125–4132, 2021.
- [26] Z. Jiao, Y. Niu, Z. Zhang, S.-C. Zhu, Y. Zhu, and H. Liu, “Planning sequential tasks on contact graph,” in *International Conference on Intelligent Robots and Systems (IROS)*, 2022.
- [27] Z. Jiao, *A Virtual Kinematic Chain Perspective for Robot Task and Motion Planning*. PhD thesis, University of California, Los Angeles, 2022.
- [28] L. Ruan, C.-H. Pi, Y. Su, P. Yu, S. Cheng, and T.-C. Tsao, “Control and experiments of a novel tiltable-rotor aerial platform comprising quadcopters and passive hinges,” *Mechatronics*, vol. 89, p. 102927, 2023.
- [29] Y. Su, L. Ruan, P. Yu, C.-H. Pi, M. J. Gerber, and T.-C. Tsao, “A fast and efficient attitude control algorithm of a tilt-rotor aerial platform using inputs redundancies,” *Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, pp. 1214–1221, 2021.
- [30] Y. Su, P. Yu, M. Gerber, L. Ruan, and T. Tsao, “Fault-tolerant control of overactuated multirotor uav platform under propeller failure,” *Transactions on Mechatronics (TMECH) (submitted)*, 2022.
- [31] Y. Su, *Compensation and Control Allocation with Input Saturation Limits and Rotor Faults for Multi-Rotor Copters with Redundant Actuations*. PhD thesis, University of California, Los Angeles, 2021.
- [32] M. Ryll, H. H. Bülthoff, and P. R. Giordano, “A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 540–556, 2014.
- [33] G. Silano and L. Iannelli, “Crazys: A software-in-the-loop simulation platform for the crazyflie 2.0 nano-quadcopter,” in *Robot Operating System (ROS)*, pp. 81–115, Springer, 2020.