2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Robust and Energy-Efficient Control for Multi-task Aerial Manipulation with Automatic Arm-switching

Ying Wu[1], Zida Zhou[1], Mingxin Wei[2] and Hui Cheng*[1]

***Abstract*— Aerial manipulation has received increasing research interest with wide applications of drones. To perform specific tasks, robotic arms with various mechanical structures will be mounted on the drone. It results in sudden disturbances to the aerial manipulator when switching the robotic arm or interacting with the environment. Hence, it is challenging to design a generic and robust control strategy adapted to various robotic arms when achieving multi-task aerial manipulation. In this paper, we present a learning-based control algorithm that allows online trajectory optimization and tracking to accomplish various aerial interaction tasks without manual adjustment. The proposed energy-saved trajectory planning approach integrates coupled dynamics model with a single rigid body to generate the energy-efficient trajectory for the aerial manipulator. Addressing the challenges of precise control when performing aerial manipulation tasks, this paper presents a controller based on deep neural networks that classifies and learns accurate forces and moments caused by different robotic arms and interactions. Moreover, the forces arising from robotic arm motions are delicately used as part of the drone's power to save energy. Extensive real-world experiments demonstrate that the proposed method can adapt to various robotic arms and interactions when performing multi-task aerial manipulation.**

I. INTRODUCTION

Nowadays, there is much research on aerial robots in civilian applications [1]. While aerial robots continue to be a highly active field of study, most of these studies focus on contact-free flight such as object detection [2], precision agriculture [3], and disaster monitoring [4]. This is due to that physical contact of the aerial manipulation system causes additional perturbations to drones. Unmanned aerial vehicles (UAVs), especially multi-rotors, are inherently flexible, easy to control, and highly versatile. Besides the various applications like surveillance [5] and navigation [6], drones show great potential in aerial manipulation tasks. Aerial manipulation brings the ability that interact with other robots to UAVs, leading to more potential for multi-robot cooperative tasks. Moreover, the physical interaction of the manipulation system provides UAVs the capability to exchange manipulation actuators and charge batteries in ground robots. This technique is efficient from the energy point of view and task diversity, which presents widespread applications for aerial manipulation.

Aerial manipulation involves many challenges for UAVs: 1) A UAV carrying a load and a manipulation actuator leads to modeling uncertainties on the whole system. In practice, due to the complexity of aerial manipulation systems, it is difficult to obtain a precise model. Without a precise model, the planning and control of a dynamically stabilized interaction will be quite difficult; 2) The disturbances caused by the robotic arm's dynamic behavior should be compensated to keep the aerial manipulation system stable. The energy loss caused by compensatory of robotic arm disturbances reduces endurance time. To address these challenges, we present a generalized, robust, and energy-efficient control strategy for multi-task aerial manipulation with various robotic arms.

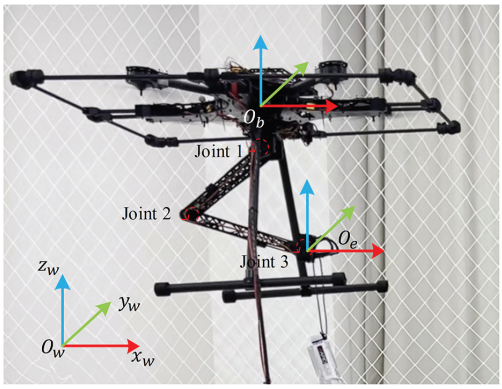


Fig. 1. The quadrotor grips a 570 ml bottle of water through a three-degree-of-freedom robotic arm. The aerial manipulation system performs different manipulation tasks (grabbing, affixing a hook, et al.) with robotic arms of various configurations.

To perform an interaction with environment, e.g., aerial grasping [7], [8], opening a drawer/door [9], [10], sliding along irregular surfaces [11], trajectory optimization is critical for an aerial manipulation system. Compared to a single-UAV system, aerial manipulation is more concerned about the end-effector trajectory of the robotic arm and less concerned about the quadrotor trajectory. [12], [13] convert the UAV state into the end-effector state considering the coupled dynamics model through the differential flatness property of system, which relies on an accurate model. This methods designed for the coupled dynamics model have a strong reliance on model precision that could not replace manipulation actuators with different configurations. [14], [15] decouple the mission trajectory planning of the aerial manipulation system into two parts: UAV and robotic arm, meaning that the flight trajectory of UAVs is not related to the flight trajectory of the robotic arms. This approach can deal with different tasks, but the manual adjustments of parameters are indispensable when changing robotic arms. One trajectory optimization method for complicated models

*This work was supported by the Key-Area Research and Development Program of Guangzhou (202206030003)

[1]Ying Wu, Zida Zhou, and Hui Cheng are with School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China. Corresponding author: Hui Cheng chengh9@mail.sysu.edu.cn

[2]Mingxin Wei is with School of Artificial Intelligence, Sun Yat-Sen University, Guangzhou 511400, China.

is simplifying a complicated model as a single rigid body (SRB) for the optimal reduced-order trajectory planning that has proven to be practical and effective in Quadrupedal robot [16]. However, this optimization method through SRB model has never been studied in the aerial manipulation field for trajectory optimization.

From a control perspective, aerial physical interactions also cause uncertain perturbations in UAVs. There exists much literature on robot control under such perturbations including mechanical improvement of robotic arm [17], [18], the use of force/torque sensor [11], [19], [20] and disturbance estimator [21], [22]. The smart mechanical design of the joint between the UAV and the arm can zero the reaction forces caused by the motion of robotic arm[17], [18]. The model-based controller presented in [24] and [25] strongly relies on the precise measurements of the force sensor to react to various environmental properties. Both these techniques further improve the mechanical complexity and the total weight of the whole system. The disturbance estimator can handle different perturbations without the need for additional mechanical design [21], [22]. However, most disturbance estimators rely on state feedback and the tracking accuracy would be affected by the hysteresis of feedback. Indeed, the perturbations caused by robotic arms obey a certain distribution law related to the state of the UAV and its arm. And the learning-based approach could be used to learn a distribution law of external disturbance [23], [26]. Examples combining the neural network and adaptive controller can be found in UAVs [26][27], manipulation [23], and Quadrupedal [16]. Particularly, [26] presents a learning-based approach (Neural-Fly) that can adapt to perturbations of different wind speeds.

Given the challenges of multi-task aerial manipulation, this paper presents a novel control strategy to perform different aerial manipulation tasks with automatic arm-switching in the air. Firstly, by simplifying the modeling of an aerial manipulation system, the disturbances from the robotic arm are exploited to generate minimum energy trajectory. In addition, under the perturbation caused by robotic arm movements, a deep neural network is employed to learn different disturbances. This learning method only uses 10 minutes flight data of each robotic arm for offline training. In this paper, a complete multi-task aerial manipulation is performed including peg placement, hanging a picture, and exchanging robotic arms in the air. The main contributions of the paper are summarized as follows:

- An aerial manipulation system is simplified a model containing a spring and center of mass. The simplified model can efficiently plan optimized reference trajectories for different robotic arms with various masses and spatial configurations. In addition, the perturbations arising from robotic arm motions are delicately used as part of the drone's power to save energy.
- The learning-based control law can efficiently online learn and adapt to the disturbances caused by various robotic arms as well as the environmental interactions. Extensive real-world experiments show that, with the proposed method, the aerial manipulation system can automatically replace various robotic arms to perform different environmental interaction tasks without manual parameter adjustment.

II. Modeling

The aerial manipulation system contains a quadrotor and a robotic arm, and the robotic arm physically interacts with the environment. It is assumed that the dynamics of quadrotor, the mechanical structure of robotic arm, and the environment are known. This paper adopts a decoupling modeling approach for quadrotor and robotic arms. Hence this section introduces the dynamics of quadrotor and the generic model of robotic arm.

A. Quadrotor

The aerial manipulation model contains a quadrotor and a multi-axis robot arm that attaches to the quadrotor. As shown in Fig. 1, O_w represents the world coordinate system, O_b represents the body coordinate system fixed at the quadrotor center of mass, and O_e represents the end effector coordinate system fixed at the robotic arm end-effector. Since quadrotor is an underactuated system, its dynamics model can be written as:

$$m\ddot{\boldsymbol{p}}_Q = \boldsymbol{R}\boldsymbol{F} + \boldsymbol{F}_{arm} + \boldsymbol{G} \tag{1}$$

$$\boldsymbol{\tau} = \boldsymbol{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \boldsymbol{J}\boldsymbol{\Omega} - \boldsymbol{\tau}_{arm} \tag{2}$$

$$\dot{\boldsymbol{p}}_Q = \dot{\boldsymbol{p}}_Q \tag{3}$$

$$\dot{\boldsymbol{R}} = \boldsymbol{R}\hat{\boldsymbol{\Omega}}. \tag{4}$$

Herein, $\boldsymbol{p}_Q$ denotes global position of quadrotor, and $\boldsymbol{\Omega}$ represents body angular velocity of quadrotor. The attitude rotation matrix of quadrotor is given by $\boldsymbol{R}$. $\boldsymbol{F}$ and $\boldsymbol{\tau}$ are the total thrust and torque generated by the propellers. $\boldsymbol{G}$ represents the gravitational force, and $\boldsymbol{J}$ is the inertia matrix of the drone's body. Additionally, $\boldsymbol{F}_{arm}$ and $\boldsymbol{\tau}_{arm}$ refer to the force and torque generated by the robotic arm. Our approach is implemented in the position control and calculates a desired force $\boldsymbol{u}_d$ that decomposed to a desired force and desired attitude matrix for DJI N3 controller. Therefore, its dynamics model can be simplified for position control in the following format:

$$\boldsymbol{M}(\boldsymbol{p}_Q)\ddot{\boldsymbol{p}}_Q + \boldsymbol{C}(\boldsymbol{p}_Q, \dot{\boldsymbol{p}}_Q)\dot{\boldsymbol{p}}_Q + \boldsymbol{G} = \boldsymbol{u_d} + \boldsymbol{F}_{arm} \tag{5}$$

where $\boldsymbol{M} = m\boldsymbol{I}$ and $\boldsymbol{C} = \boldsymbol{0}$.

B. Manipulator

The structure of the robotic arms in this paper is illustrated in Fig. 1 and Fig. 3. To model the n-axis robot arms with different structures, we first employ a kinematic representation method based on $SE(3)$. The transformation matrix from joint $i-1$ to joint i is as follows:

$$\boldsymbol{T}_{i-1}^{i} = \begin{bmatrix} \boldsymbol{R}_{i-1}^{i} & \boldsymbol{p}_{i-1}^{i} \\ \boldsymbol{0_{1\times 3}} & 1 \end{bmatrix} \tag{6}$$

where $\boldsymbol{R} = \exp(\hat{\boldsymbol{\Omega}}_i\theta_i) = \boldsymbol{I} + \hat{\boldsymbol{\Omega}}_i sin\theta_i + \hat{\boldsymbol{\Omega}}_i^2(1 - cos\theta_i)$ denotes the rotation matrix from joint $i-1$ to joint i, θ_i

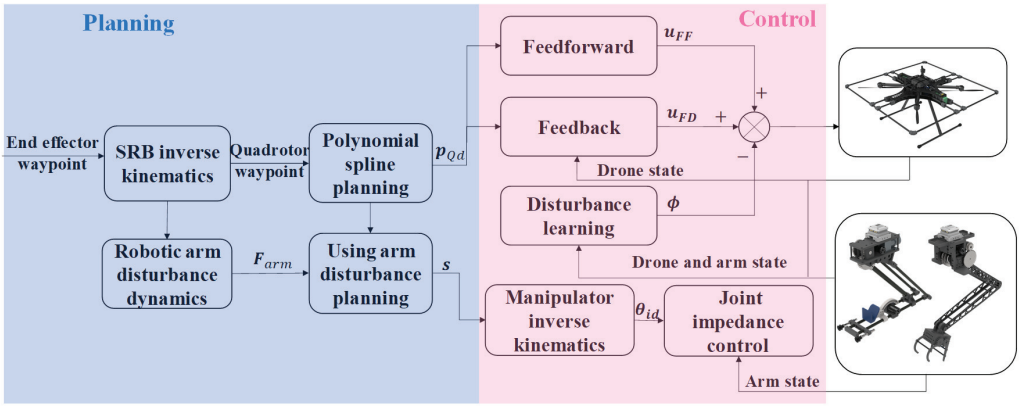


Fig. 2. The control framework for aerial manipulation with different arms. The blue area corresponds to the planning part for generating energy-effective trajectories of the quadrotor and the robotic arms. The pink area corresponds to the control part for online learning and adapt to perturbations caused by various robotic arms and the environmental interactions.

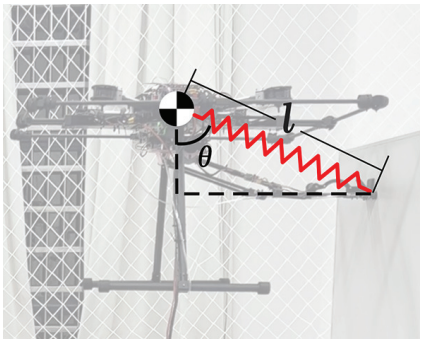


Fig. 3. The single rigid body of the manipulation system. The quadrotor is simplified as a mass point (sphere) while the arm is simplified as a spring (red broken line). The l and θ show the relative position of the mass point and the spring.

means the angle of i th joint, Ω_i means the rotation axis of i th joint and $\boldsymbol{p}_{i-1}^{i}$ represents the relative displacement from joint $i-1$ to joint i. The kinematic model of the robotic arm is:

$$\boldsymbol{T}_0^i(\theta_1, \theta_2, \cdots \theta_i) = \boldsymbol{T}_0^1 \boldsymbol{T}_1^2 \cdots \boldsymbol{T}_{t-1}^t \tag{7}$$

where $\boldsymbol{T}_0^i$ represents the transformation matrix from O_b to O_e, and $\boldsymbol{p}_0^i$ represents the position of the end effector in body coordinate system. For a given desired trajectory of end effector, the angle of each joint could be calculated through inverse kinematics of (7).

III. PLANNING AND CONTROL FRAMEWORK

The goal of our method is to enable a quadrotor to execute different manipulation tasks with various robotic arms while autonomously planning energy-effective trajectories and replacing robotic arms. As shown in Fig. 2, our optimization and control strategy is as follows:

1) The whole system of aerial manipulation is simplified as a center-of-mass spring model for trajectory optimization with different robotic arms, which reduces the computation complexity of the planning algorithm. The trajectory planning algorithm of coupled system uses the perturbation by the robotic arm as part of UAV flight power to achieve energy optimization and trajectory smoothing.

2) The control strategy consists of an adaptive control law of quadrotor, an impedance controller of a robotic arm and a neural policy to learn interactions between the quadrotor and arm. This method combines data-driven and model-based approach to keep stability of the system with various robotic arms motion through short-duration offline training.

A. SRB model

Since quadrotors are underactuated, the robotic arm motion first affects the quadrotor attitude then changes the position. Firstly, the robotic arm motion would affect the mass distribution of manipulation system. And the center mass of the whole system shifts towards the direction of the arm reaching motions. That means the $z-$axis of O_b would offset in the arm reaching direction and quadrotor offset to that direction. Utilizing the centroid displacement caused by the robotic arm in an unmanned manipulation system, the disturbance of robotic arm could be exploited as a "joystick" to move the quadrotor attitude to the desired direction. Therefore, as shown in Fig. 3, we present a reduced-order model that the robotic arm is modeled as a spring (robotic arm) pulling the quadrotor toward its stretch direction. The kinematic model of a reduced-order model between the end effector position $\boldsymbol{p}_m$ and $\boldsymbol{p}_Q$ is:

$$\boldsymbol{p}_Q = \boldsymbol{p}_m - l \begin{bmatrix} \cos\psi \sin\theta \\ \sin\psi \sin\theta \\ \cos\theta \end{bmatrix} \tag{8}$$

where ψ is the yaw angle of the quadrotor. And the disturbance force model from spring acted on quadrotor is written as:

$$\boldsymbol{F}_{arm} = (k \cdot \triangle l + b \cdot \dot{l}) \begin{bmatrix} \cos\psi \sin\theta \\ \sin\psi \sin\theta \\ \cos\theta \end{bmatrix} \tag{9}$$

where k and b represents the elastic coefficient and damping coefficient of spring.

B. Trajectory planning of coupled system

The trajectory planning of coupled system can generate the trajectories of quadrotor and robotic arm. The planner in this paper can exploit the force and torque generated by robotic

arm as power for quadrotor flight. First, for a given waypoints of end effector, the desired waypoints of quadrotor are calculated from (8). Then a desired trajectory of quadrotor $\boldsymbol{p}_{Qd}(t), (0 < t < t_e)$ is generated by polynomial spline with the constraint that quadrotor has zero velocity and acceleration at the start and terminal position.

To use the perturbation force by robotic arm, the quadrotor acceleration and the perturbation force from arm acted on quadrotor could not be along opposite direction. The direction of perturbation force $\boldsymbol{F}_{arm}$ can be calculated by assuming $\triangle l = l_e - l_0$. Thus, if $(\boldsymbol{F}_{arm})^T \ddot{\boldsymbol{p}_Q} \geq 0$, the desired trajectory of robotic arm $\boldsymbol{s} = [l_d(t), \theta_d(t)]$ is generated by polynomial spline in $(0 < t < \frac{t_e}{2})$ as:

$$\begin{cases} \boldsymbol{s}(t) = \boldsymbol{a} + \boldsymbol{d}t^3 + \boldsymbol{e}t^4 + \boldsymbol{f}t^5, (0 < t < \dfrac{t_e}{2}) \\ \boldsymbol{s}(t) = \boldsymbol{s}_e, (\dfrac{t_e}{2} < t < t_e). \end{cases} \tag{10}$$

If $(\boldsymbol{F}_{arm})^T \ddot{\boldsymbol{p}_Q} \leq 0$, the desired trajectory of robotic arm $\boldsymbol{s} = [l_d(t), \theta_d(t)]$ is generated by polynomial spline in $(0 < t < \frac{t_e}{2})$ as:

$$\begin{cases} \boldsymbol{s}(t) = \boldsymbol{s}_0, (0 < t < \dfrac{t_e}{2}) \\ \boldsymbol{s}(t) = \boldsymbol{a} + \boldsymbol{d}t^3 + \boldsymbol{e}t^4 + \boldsymbol{f}t^5, (\dfrac{t_e}{2} < t < t_e) \end{cases} \tag{11}$$

where $\boldsymbol{a}$, $\boldsymbol{d}$, $\boldsymbol{e}$ and $\boldsymbol{f}$ is the parameters of polynomial spline. The subscrs 0 and e means the start and end state.

C. Adaptive control

We begin with defining some notations. The error of impedance control for each robotic arm joint is defined as:

$$e_{\theta_i} = \theta_i - \theta_{id} \tag{12}$$

where θ_{id} is desired angle of joint i calculated by $\boldsymbol{s}$ in (10) and (11). The impedance control law of the arm joint i is:

$$u_{\theta_i} = m_d \ddot{e}_{\theta_i} + b_d \dot{e}_{\theta_i} + k_d e_{\theta_i} \tag{13}$$

where m_d, b_d and k_d are the parameters of the impedance controller. The track error of quadrotor is:

$$\boldsymbol{e_p} = \boldsymbol{p}_Q - \boldsymbol{p}_{Qd}. \tag{14}$$

The control law of this quadrotor $\boldsymbol{u}_{AD}$ is:

$$\boldsymbol{u}_{AD} = \boldsymbol{u}_{FF} + \boldsymbol{u}_{FD} - \boldsymbol{\phi}(\boldsymbol{p}_Q, \dot{\boldsymbol{p}}_Q, \boldsymbol{p}_m, \dot{\boldsymbol{p}}_m) \tag{15}$$

$$\boldsymbol{u}_{FF} = \boldsymbol{M}(\boldsymbol{p}_Q)\ddot{\boldsymbol{p}_{Qd}} + \boldsymbol{C}(\boldsymbol{p}_Q, \dot{\boldsymbol{p}}_Q)\dot{\boldsymbol{p}}_Q + \boldsymbol{G} \tag{16}$$

$$\boldsymbol{u}_{FD} = -\boldsymbol{K}_P \boldsymbol{e}_p - \boldsymbol{K}_D \dot{\boldsymbol{e}}_p - \boldsymbol{K}_I \int \boldsymbol{e}_p dt \tag{17}$$

where the $\boldsymbol{u}_{FF}$ is nonlinear feedforward terms, $\boldsymbol{u}_{FD}$ is PID feedback terms and $\boldsymbol{\phi}$ is the learning-based feedforward term. And $\boldsymbol{K}_P$, $\boldsymbol{K}_D$ and $\boldsymbol{K}_I$ are positive gains.

In general, a standard PID controller combined with feedforward terms (baseline controller mentioned below) can keep dynamically stable with a nonlinear dynamics model. However, this baseline controller compensates the unknown perturbations only through PID terms leading to slow reaction to unmodeled perturbation force. Thus, we improve this controller by a learned representation $\boldsymbol{\psi}$ of unmodeled perturbation of different robotic arms. Assume $\boldsymbol{\psi}$ can infinitely approximate $\boldsymbol{F}_{arm}$, the dynamics model of quadrotor in (5) could be written as:

$$\ddot{\boldsymbol{e}}_p = -\boldsymbol{K}_P \boldsymbol{e}_p - \boldsymbol{K}_D \dot{\boldsymbol{e}}_p - \boldsymbol{K}_I \int \boldsymbol{e}_p dt. \tag{18}$$

The system is exponentially stable with proper gains $\boldsymbol{K}_P$, $\boldsymbol{K}_D$ and $\boldsymbol{K}_I$. Substituting (16) and (17) into (5), (5) is calculated as:

$$m\ddot{\boldsymbol{p}}_Q + G - (\boldsymbol{u}_{FF} + \boldsymbol{u}_{FD}) = -m\ddot{\boldsymbol{e}}_p + \boldsymbol{u}_{FD} + \boldsymbol{F}_{arm}. \tag{19}$$

Hence, $m\ddot{\boldsymbol{p}}_Q + G - (\boldsymbol{u}_{FF} + \boldsymbol{u}_{FD})$ exponentially approaches $\boldsymbol{F}_{arm}$.

Algorithm 1: Train process of $\boldsymbol{\phi}$

param: $\alpha \geq 0$, $0 < \mu \leq 1$
Data: $\mathbf{D} = \{D_{a_1}, \cdots, D_{a_k}\}$
Initial : Neural network $\boldsymbol{\phi}$ and $\boldsymbol{h}$
repeat
  Random sample $\mathbf{N} = \{N_1, \cdots, N_k\}$ from $\mathbf{D}$;
  Train $\boldsymbol{\phi}$ use stochastic gradient descent and spectral normalization with loss:

$$\sum_{k=1}^{K}\sum_{i=1}^{N_k}(\|\boldsymbol{y}_k^{(i)} - \boldsymbol{\phi}(\boldsymbol{x}_k^{(i)})\| - \alpha \cdot loss(\boldsymbol{h}(\boldsymbol{\phi}(\boldsymbol{x}_k^{(i)})), k));$$

  if $rand() \leq \mu$ **then**
    Train $\boldsymbol{h}$ use stochastic gradient descent and spectral normalization with loss:

$$\sum_{k=1}^{K}\sum_{i=1}^{N_k}(\alpha \cdot loss(\boldsymbol{h}(\boldsymbol{\phi}(\boldsymbol{x}_k^{(i)})), k));$$

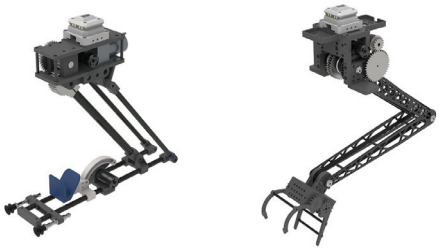
  end
until *convergence*;

D. Learning to close the gap

This subsection is to introduce the detail of learning a representation $\boldsymbol{\phi}$ to approximate $\boldsymbol{F}_{arm}$. First, to generate training data of $\boldsymbol{\phi}$, the quadrotor and robotic arm both track some random trajectories with the baseline controller. The input of training data $\boldsymbol{x}_k^{(i)}$ is robotic arm index, the attitude of quadrotor, relative position between quadrotor and end effector, and total thrust of quadrotor. The label of training data $\boldsymbol{y}_k^{(i)}$ is $m\ddot{\boldsymbol{p}}_Q + G - (\boldsymbol{u}_{FF} + \boldsymbol{u}_{FD})$. And D_{a_k} is the training data of the kth robotic arm. Given the training data, the goal of learning is to learn a representation to approximate $\boldsymbol{F}_{arm}$ for any robotic arm. Hence, the representation $\boldsymbol{\phi}$ solves the optimization problem as:

$$\min_{\boldsymbol{\phi}} \sum_{k=1}^{K}\sum_{i=1}^{N_k} \|\boldsymbol{y}_k^{(i)} - \boldsymbol{\phi}(\boldsymbol{x}_k^{(i)})\| \tag{20}$$

where N_k means the amounts of data D_{a_k}. In this paper, we use deep neural network (DNN) to represent $\boldsymbol{\phi}$ that is shared by each robotic arm. As long as $\boldsymbol{\phi}$ has adequate



(a) arm for pegging (b) arm with a gripper

Fig. 4. The arms using to place signboard on the wall.

(a) Reaching action (b) Contraction actions

Fig. 5. Schematic of the reaching and contraction movement

neurons, the $\boldsymbol{F}_{arm}$ is approximated by $\boldsymbol{\phi}$. However, for same input $\boldsymbol{x}_k^{(i)}$, $\boldsymbol{F}_{arm}$ vary from various arms. By (20), the distribution of $\boldsymbol{x}_k^{(i)}$ is memorized by $\boldsymbol{\phi}$, while $\boldsymbol{\phi}$ could not reflect the distributions of different robotic arm. To solve this problem, inspired by [26], we designed a modified optimization problem as:

$$\max_{\boldsymbol{h}} \min_{\boldsymbol{\phi}} \sum_{k=1}^{K} \sum_{i=1}^{N_k} (\|\boldsymbol{y}_k^{(i)} - \boldsymbol{\phi}(\boldsymbol{x}_k^{(i)})\| - \alpha \cdot loss(\boldsymbol{h}(\boldsymbol{\phi}(\boldsymbol{x}_k^{(i)})), k)) \tag{21}$$

where $\boldsymbol{h}$ is a DNN as learned regularization to avoid overfitting, $loss(\cdot)$ is a cross entropy loss, and $\alpha \geq 0$ is a hyper-parameter of DNN $\boldsymbol{h}$. Intuitively, $\boldsymbol{h}$ is the discriminator to predict the model index out of K robotic arms (training by the outer max), while $\boldsymbol{\phi}$ is to follow $\boldsymbol{F}_{arm}$ as closely as possible (training by the inner min). The details of training process of the DNN $\boldsymbol{\phi}$ and $\boldsymbol{h}$ are shown in the Algorithm 1.

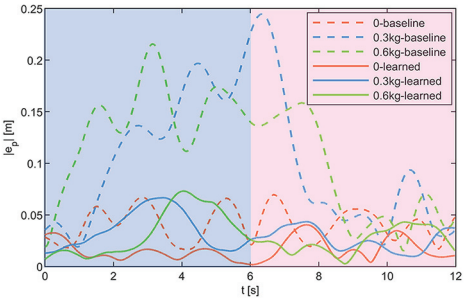
IV. EXPERIMENTS

In this section our control strategy is evaluated based on two selected manipulation tasks: 1) The robotic arm carries a heavy object ($0.6kg$) to execute different actions behind quadrotor; 2) The aerial manipulation system executes a complete task that contains peg placement, switching robotic arm, and hanging a picture.

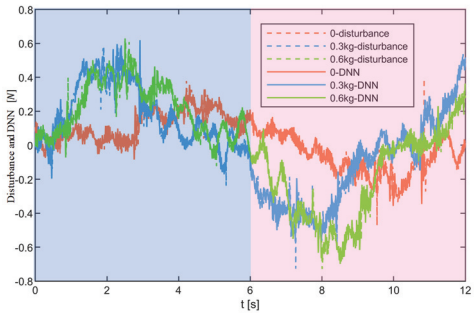
A. Experimental setup

The quadrotor and robotic arms used in the experiments are shown in Fig. 1 and Fig. 4. The experiments are performed indoors equipped with an Optirack motion capture system. The quadrotor with mass $m = 6\ kg$ is assembled with a NUC i7 computer and a DJI N3 controller, while each robotic arm with stretch length $0.6\ m$ is assembled with a STM32 chip for independent motion control. The quadrotor communicates with robotic arm through a serial port protocol.

To perform the task of placing a signboard on the wall, we have designed two robotic arms with different structures.



(a) The position error of quadrotor



(b) The disturbance $\boldsymbol{y}(1)$ and DNN output $\boldsymbol{\phi}(1)$

Fig. 6. The state of quadrotor when robotic arm executes reaching and contraction actions. The blue area in a) and b) means arm executes reaching action and the pink area in a) and b) means arm executes contraction action.

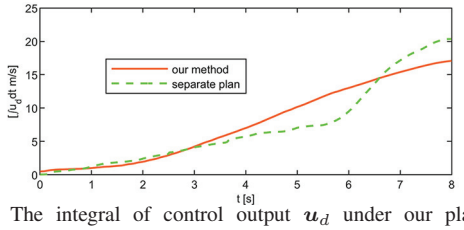


Fig. 7. The integral of control output $\boldsymbol{u}_d$ under our planning and control framework and planning separately strategy. The solid line means our planning and control framework and the dotted line means planning separately strategy.

As shown in Fig. 4, since the peg placement requires apply some pressure to the wall, the arm of peg placement (Fig. 4a) is equipped with a slider–crank mechanism and two vacuum chucks. This pegging arm has only a single degree of freedom for straighten that the vacuum chuck can be attached to the wall, and then slider–crank mechanism exert force on the hook to adhere to the wall. The arm of hanging a signboard (Fig. 4b) is a three-degree-of-freedom arm with a gripper on the end, which is more flexible to grab objects from the surrounding environment.

The quadrotor collects the training data for 10 minutes with each robotic arm. And the hyper-parameter of learning network is $\alpha = 0.01$ and $\mu = 0.5$. The parameters of baseline controller are defined as:

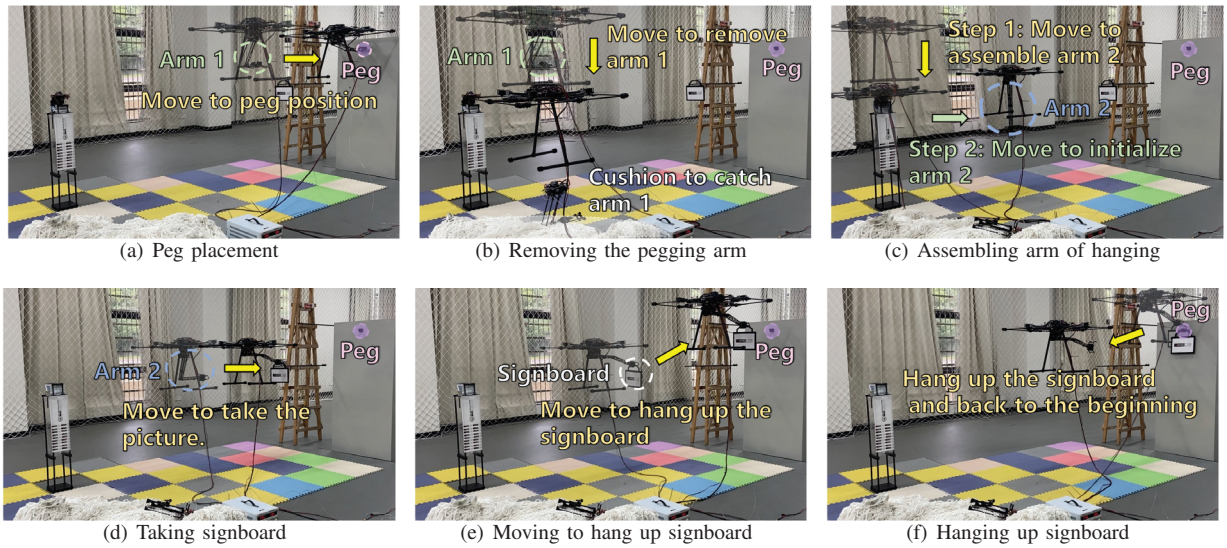
$$\boldsymbol{K}_P = diag([3.0, 3.0, 3.0]) \tag{22}$$

$$\boldsymbol{K}_D = diag([1.5, 1.5, 1.5]) \tag{23}$$

$$\boldsymbol{K}_I = diag([0.2, 0.2, 0.2]). \tag{24}$$

B. The robotic arm execute dissimilar actions

This experiment considers three conditions: 1) The robotic arm carries different payload weighing (0 kg, 0.3 kg and 0.6



(a) Peg placement (b) Removing the pegging arm (c) Assembling arm of hanging

(d) Taking signboard (e) Moving to hang up signboard (f) Hanging up signboard

Fig. 8. Task diagram for hanging signboard with automatic arm changing. In (a), the quadrotor with the pegging arm moves to place peg in a cabinet. After peg placement, the quadrotor automatically removes the pegging arm (b) and assembles an arm to hang a signboard (c). Then the quadrotor takes the signboard hanging from a ladder in (d) and moves to the front of the cabinet to hang a signboard (e). Finally, the quadrotor puts the signboard on the peg (f).

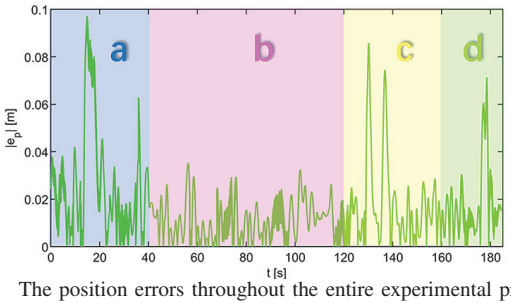


Fig. 9. The position errors throughout the entire experimental process: a) peg placement, b) robotic arm switching, c) signboard acquisition, and d) signboard hanging

kg) to execute dissimilar actions under baseline controller; 2) The robotic arm carries different payload weighing (0 kg, 0.3 kg and 0.6 kg) to execute dissimilar actions under our learning-based controller;

The robotic arm first executes the reaching action and then executes contraction actions. Fig. 5 shows the contraction and reaching actions of the robotic arm. As shown in Fig. 6, when robotic arm makes reaching movement, the disturbance reaches the maximal range, and our learning-based control algorithm outperforms baseline controller in all conditions. Even carrying a load weighing up 0.6 kg, the error caused by the continuous movement of arm can also be controlled under the accuracy of 0.10 m by our algorithm, while the steady-state error of baseline controller is 0.40 m. Fig. 6 also shows that our learning-based controller can better match the dynamics of arm disturbance in different mass arm.

C. The aerial manipulation system executes different tasks

Two aerial manipulation experiments are conducted in two conditions. One is that aerial manipulation system executes reaching movement under our planning and control framework, and the other is that aerial manipulation system executes reaching movement under planning separately strategy and our learning-based controller. Then, this whole experiment is performed in the following order: 1) peg placement 2) switching robotic arm and 3) hanging a picture.

Our method calculates and executes the movement trajectory of the robotic arm during flying. For planning separately strategy, the quadrotor first flies to the target position and then executes the reaching action of robotic arm. As shown in Fig. 7, it is obvious that the integral of control output $\boldsymbol{u}_d$ of our framework is less than that of planning separately strategy, which shows energy efficiency of our methods. Fig. 8 and Fig. 9 show the whole manipulation steps of our methods containing peg placement, dropping the pegging arm, assembling arm of hanging, taking signboard and hanging signboard. Meanwhile, this method can precisely track desired trajectories with error less than 0.10 m throughout the experiments. It has been demonstrated that our learning-based controller automatically learns perturbations of different robotics arm motions, and no manual interventions were necessary.

V. CONCLUSIONS

This paper presents a universal and robust control framework that can generate an energy-effective trajectory and stably track the desired trajectory for different aerial manipulation tasks. The proposed approach delicately utilizes the perturbations caused by the robotic arms to partly control the quadrotor's rotation. With the proposed control strategy, manual parameter adjustment is not needed when the drone replaces various robotic arms. Extensive real-world experiments demonstrate the energy optimization as well as perturbation resistance of the proposed controller, which enables automatically performing multi-task aerial manipulation. Future work will consider more complicated aerial manipulation tasks and study multi-robot collaborative aerial manipulation tasks.

REFERENCES

[1] S. Chen, D. F. Laefer, and E. E. Mangina, "State of technology review of civilian uavs," Recent Patents on Engineering, vol. 10, pp. 160–174, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:56431741

[2] N. Tijtgat, W. Van Ranst, B. Volckaert, T. Goedemé, and F. De Turck, "Embedded real-time object detection for a uav warning system," in 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), 2017, pp. 2110–2118.

[3] D. C. Tsouros, S. Bibi, and P. G. Sarigiannidis, "A Review on UAV-Based Applications for Precision Agriculture," Information, vol. 10, no. 11, p. 349, Nov. 2019, doi: 10.3390/info10110349.

[4] C. Yuan, Z. Liu, and Y. Zhang, "Uav-based forest fire detection and tracking using image processing techniques," in 2015 International Conference on Unmanned Aircraft Systems (ICUAS), 2015, pp. 639–643.

[5] W. J. Yun, S. Park, J. Kim, M. Shin, S. Jung, D. A. Mohaisen, and J.-H. Kim, "Cooperative multiagent deep reinforcement learning for reliable surveillance via autonomous multi-uav control," IEEE Transactions on Industrial Informatics, vol. 18, no. 10, pp. 7086–7096, 2022.

[6] Y. Lu, Z. Xue, G.-S. Xia, and L. Zhang, "A survey on vision-based uav navigation," Geo-spatial Information Science, vol. 21, pp. 21 – 32, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:81985472

[7] G. Zhang, Y. He, B. Dai, F. Gu, L. Yang, J. Han, and G. Liu, "Aerial grasping of an object in the strong wind: Robust control of an aerial manipulator," Applied Sciences, vol. 9, no. 11, p. 2230, 2019.

[8] Z. Ouyang, R. Mei, Z. Liu, M. Wei, Z. Zhou, and H. Cheng, "Control of an aerial manipulator using a quadrotor with a replaceable robotic arm," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 153–159.

[9] S. Kim, H. Seo, and H. J. Kim, "Operating an unknown drawer using an aerial manipulator," in 2015 IEEE international conference on robotics and automation (ICRA). IEEE, 2015, pp. 5503–5508.

[10] M. Brunner, G. Rizzi, M. Studiger, R. Siegwart, and M. Tognon, "A planning-and-control framework for aerial manipulation of articulated objects," IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 10689–10696, 2022.

[11] W. Zhang, L. Ott, M. Tognon, and R. Siegwart, "Learning variable impedance control for aerial sliding on uneven heterogeneous surfaces by proprioceptive and tactile sensing," IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 11275–11282, 2022.

[12] H. Seo, S. Kim, and H. J. Kim, "Locally optimal trajectory planning for aerial manipulation in constrained environments," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 1719–1724.

[13] M. Tognon, B. Yüksel, G. Buondonno, and A. Franchi, "Dynamic decentralized control for protocentric aerial manipulators," in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 6375–6380.

[14] R. Ragel, I. Maza, F. Caballero, and A. Ollero, "Comparison of motion planning techniques for a multi-rotor uas equipped with a multi-joint manipulator arm," in 2015 Workshop on research, education and development of unmanned aerial systems (RED-UAS). IEEE, 2015, pp. 133–141.

[15] J. Fishman and L. Carlone, "Control and trajectory optimization for soft aerial manipulation," in 2021 IEEE Aerospace Conference (50100). IEEE, 2021, pp. 1–17.

[16] A. Pandala, R. T. Fawcett, U. Rosolia, A. D. Ames, and K. A. Hamed, "Robust predictive control for quadrupedal locomotion: Learning to close the gap between reduced-and full-order models," IEEE Robotics and Automation Letters, vol. 7, no. 3, pp. 6622–6629, 2022.

[17] K. Suryavanshi, S. Hamaza, V. van der Wijk, and J. Herder, "Adapt: A 3 degrees of freedom reconfigurable force balanced parallel manipulator for aerial applications," in 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023, pp. 11936–11942.

[18] Y. S. Sarkisov, A. Coelho, M. G. Santos, M. J. Kim, D. Tsetserukou, C. Ott, and K. Kondak, "Hierarchical whole-body control of the cable-suspended aerial manipulator endowed with winch-based actuation," in 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023, pp. 5366–5372.

[19] E. Cataldi, G. Muscio, M. A. Trujillo, Y. Rodr´ıguez, F. Pierri, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Impedance control of an aerial-manipulator: Preliminary results," in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016, pp. 3848–3853.

[20] G. Nava, Q. Sable, M. Tognon, D. Pucci, and A. Franchi, "Direct force feedback control and online multi-task optimization for aerial manipulators," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 331–338, 2019.

[21] M. Brunner, L. Giacomini, R. Siegwart, and M. Tognon, "Energy tank-based policies for robust aerial physical interaction with moving objects," in 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022, pp. 2054–2060.

[22] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, D. Bicego, and A. Franchi, "6d interaction control with aerial robots: The flying end-effector paradigm," The International Journal of Robotics Research, vol. 38, no. 9, pp. 1045–1062, 2019.

[23] L. Roveda, J. Maskani, P. Franceschi, A. Abdi, F. Braghin, L. Molinari Tosatti, and N. Pedrocchi, "Model-based reinforcement learning variable impedance control for human-robot collaboration," Journal of Intelligent & Robotic Systems, vol. 100, no. 2, pp. 417–433, 2020.

[24] S. A. Emami and A. Banazadeh, "Simultaneous trajectory tracking and aerial manipulation using a multi-stage model predictive control," Aerospace Science and Technology, vol. 112, p. 106573, 2021.

[25] D. Tzoumanikas, F. Graule, Q. Yan, D. Shah, M. Popovic, and S. Leutenegger, "Aerial manipulation using hybrid force and position nmpc applied to aerial writing," arXiv preprint arXiv:2006.02116, 2020.

[26] M. O'Connell, G. Shi, X. Shi, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural-fly enables rapid learning for agile flight in strong winds," Science Robotics, vol. 7, no. 66, p. eabm6597, 2022.

[27] M. Wei, L. Zheng, H. Li, and H. Cheng, "Adaptive neural network-based model path-following contouring control for quadrotor under diversely uncertain disturbances," IEEE Robotics and Automation Letters, pp. 1–8, 2024.