2023 International Conference on Unmanned Aircraft Systems (ICUAS)
June 6-9, 2023 | Lazarski University, Warsaw, Poland

Online Trajectory Generation for Aerial Manipulator Subject to Multi-tasks and Inequality Constraints

Rui Chen[1], Qianyuan Liu[1], Zeshuai Chen[1], Kexin Guo[2,*], Xiang Yu[1], and Lei Guo[1]

2023 International Conference on Unmanned Aircraft Systems (ICUAS) | 979-8-3503-1037-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICUAS57906.2023.10156288

***Abstract*— This article tackles the problem of generating coordinated trajectory for unmanned aerial manipulator (UAM) system. The kinematic redundancy nature of this class of system makes it challenging to design constraints-satisfied trajectories of both the aerial vehicle and the robotic arm simultaneously that can accomplish a series of tasks with varying levels of priority. This paper presents a redundancy utilized trajectory generation method based on hierarchical quadratic programming (HQP). The method is computationally inexpensive to execute online, allowing the UAM to dynamically adjust its configuration within inequality constraints (e.g. velocity bounds) to execute multi-tasks such as end-effector tracking, joint limits avoidance, and center of gravity (CoG) alignment. An experiment case study, where UAM is assigned to track and grasp a moving target, has been reported to illustrate the effectiveness of our approach.**

I. INTRODUCTION

The increasing need for active tasks such as aerial grasping, equipment installation, infrastructure maintenance, and the demand for aerial robots with active operation capability have sparked the transformation to unmanned aerial manipulator systems (UAMs) [1]–[3]. In [4], an avian-inspired UAM system that integrates a 1-DoF robotic arm with a special gripper is designed to accomplish flying grasping task. In [5], a novel UAM is constructed for aerial visual servoing with a moving battery used to counterbalance the statics of a 4-DoF robotic arm. The UAM system designed in [6] is equipped with two 4-DoF robotic arms cooperating to turn valves on infrastructure. A parallel dexterous 3-DoF robotic arm is attached under the aerial vehicle for tree cavity inspection [7]. However, attachment of the multi-link robotic arm endows UAM system with kinematic redundancy. The utilization of this nature to coordinate motion of the aerial platform and the robotic arm is a challenging issue, especially when the system is faced with multi-tasks in hierarchy and subject to state constraints.

This work was supported by National Key Research and Development Program of China (Grant Number 2020YFA0711200), National Natural Science Foundation of China (Grant Numbers 61833013, 61973012, 62273023, and 62227813), Defense Industrial Technology Development Program (Grant Number JCKY2020601C016), Program for Changjiang Scholars and Innovative Research Team (Grant Number IRT 16R03), Key Research and Development Program of Zhejiang (Grant Number 2021C03158), Science and Technology Key Innovative Project of Hangzhou (Grant Number 20182014B06), Natural Science Foundation of Zhejiang Province (Grant Number LD21F030001).

[1]R. Chen, Q. Liu, Z. Chen, X. Yu, and L. Guo are with the School of Automation Science and Electrical Engineering, Beihang University, 100191, Beijing, China. E-mail: rchen_buaa@163.com, {liuqianyuan, zschen, xiangyu_buaa, lguo} @buaa.edu.cn.

[2]K. Guo is with the School of Aeronautic Science and Engineering, Beihang University, 100191, Beijing, China. E-mail: kxguo@buaa.edu.cn.

* Corresponding author.

To utilize redundancy and coordinate the motion for UAM, weighted damped least-squares method is used in [8] to generate reference trajectories for multiple tasks with different priorities including end-effector tracking, obstacle avoidance, and manipulability maximization. The weighting matrices assign greater or lesser significance to the associated tasks relative to the others. Redundancy utilization problem is resolved via the null space-based (NSB) paradigm in [9]–[14]. In these cases, feasible trajectories are generated for UAM to deal with end-effector tracking as well as other secondary tasks, including obstacle avoidance [9], joint limits [10], visual servoing [11], CoG alignment [12], and dual arm cooperation [13]. In [14], this paradigm is further extended to multi-UAMs system coordination control. In the aforementioned NSB paradigm, the contributions of lower-priority tasks to the reference trajectories are projected onto the null space of the higher-priority ones, and trajectories are derived through a series of algebraic operations, including differentiation, pseudo-inversions, and null-space projections. The generation of trajectories can also be formulated to an optimization problem and thereby resolved by a solver. In [15], an optimization-based method employing quadratic programming (QP) is utilized to compute joint acceleration references satisfying both equality and inequality constraints. Nonlinear model predictive control (NMPC) is proposed in [16] to accomplish multi-tasks while maintaining UAM configuration within imposed equality and inequality bounds.

The current literature on UAM systems has yet to consider redundancy utilized trajectory generation under strict task-priority and inequality constraints. In the method [8], strict priority among tasks is lost because of the weighting strategy, and inequality constraints are not considered, either. NSB paradigm handles multiple tasks in a strict order of hierarchy, which ensures the primary task is prioritized and accomplished at first when different tasks conflict with each other, but with limited consideration of inequality constraints. Neglecting inequality constraints may result in generation of infeasible trajectories. The optimization-based method [15], [16] is able to solve the problem with the consideration of inequality constraints, but multi-tasks are not strictly prioritized due to the exploitation of a similar weighting strategy as in [8].

In general, the aforementioned methods [8]–[16] are incapable of generating trajectories within inequality bounds in a hierarchical order for multi-tasks. To address this gap, this paper employs the HQP framework, which is a prioritized task-regulation framework based on a sequence of QPs [17]. The HQP framework can guarantee the strict priority

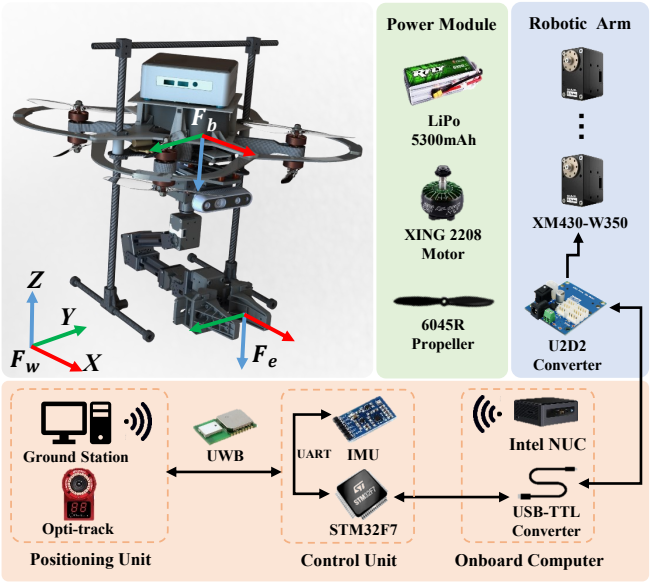


Fig. 1. Coordinates and hardware architecture of UAM system.

order of multi-tasks while explicitly accounting for inequality constraints imposed at any task level. It has been effectively applied to humanoid robot systems [18], mobile manipulator systems [19], and multi-robot systems [20]. In particular, this paper contributes to exploiting the redundancy of the entire system through the HQP framework to perform internal motions that address challenges (e.g. joint limits avoidance, CoG alignment) and satisfy inequality constraints (e.g. velocity bounds) while the end-effector is performing the primary task (e.g. end-effector tracking). To the authors' knowledge, this is the first time HQP framework has been experimentally applied and validated in UAM system to handle system redundancy and inequality constraints simultaneously.

The remainder of this paper is organized as follows: Section II describes the kinematics of UAM system. Section III analyzes the multiple tasks executed by UAM. In Section IV, the trajectory generation strategy based on HQP framework is formulated. Experiment results are introduced and analyzed in Section V. Lastly, Section VI concludes the paper.

II. UAM Model

A. System Coordinates

Consider a UAM system comprising a coaxial quadrotor and a multi-link robotic arm (shown in Fig. 1). Three coordinates are defined to describe the system kinematics: the world inertial frame $\mathcal{F}_w$, the quadrotor body frame $\mathcal{F}_b$ attached to the centroid of the quadrotor, and the end-effector frame $\mathcal{F}_e$.

B. Kinematics

According to [13], position and orientation of the end-effector are given as

$$\begin{cases} \boldsymbol{P}_e = \boldsymbol{P}_b + \boldsymbol{R}_b^w \boldsymbol{P}_{eb}^b \\ \boldsymbol{R}_e^w = \boldsymbol{R}_b^w \boldsymbol{R}_e^b \end{cases}, \tag{1}$$

where $\boldsymbol{P}_e$ and $\boldsymbol{R}_e^w$ are the relative position and orientation of the frame $\mathcal{F}_e$ w.r.t the frame $\mathcal{F}_w$ expressed in $\mathcal{F}_w$, $\boldsymbol{R}_b^w$ is the relative orientation of the frame $\mathcal{F}_b$ w.r.t the frame $\mathcal{F}_w$, $\boldsymbol{P}_b$ is the relative position of the frame $\mathcal{F}_b$ w.r.t the frame $\mathcal{F}_w$ expressed in $\mathcal{F}_w$, $\boldsymbol{R}_e^b$ is the relative orientation of the frame $\mathcal{F}_e$ w.r.t the frame $\mathcal{F}_b$, $\boldsymbol{P}_{eb}^b$ is the relative position of the frame $\mathcal{F}_e$ w.r.t the frame $\mathcal{F}_b$ expressed in $\mathcal{F}_b$.

By differentiating Eq.(1), linear velocity $\dot{\boldsymbol{P}}_e$ and angular velocity $\dot{\boldsymbol{\Omega}}_e$ of the end-effector are described as

$$\begin{bmatrix} \dot{\boldsymbol{P}}_e \\ \dot{\boldsymbol{\Omega}}_e \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_3 & -S(\boldsymbol{R}_b^w \boldsymbol{P}_{eb}^b) \\ \boldsymbol{O}_3 & \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{P}}_b \\ \dot{\boldsymbol{\Omega}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{R}_b^w & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{R}_b^w \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{P}}_{eb}^b \\ \dot{\boldsymbol{\omega}}_{eb}^b \end{bmatrix}, \tag{2}$$

where $\dot{\boldsymbol{P}}_b$ and $\dot{\boldsymbol{\Omega}}_b$ are, respectively, linear and angular velocity of the quadrotor body frame $\mathcal{F}_b$, $\boldsymbol{I}_3$ and $\boldsymbol{O}_3$ are, respectively, (3×3) identity and null matrix, $S(\cdot)$ represents the skew symmetric operator. The relative velocities of $\mathcal{F}_e$ w.r.t. $\mathcal{F}_b$ expressed in the frame $\mathcal{F}_b$, $\dot{\boldsymbol{P}}_{eb}^b$ and $\dot{\boldsymbol{\omega}}_{eb}^b$ can be obtained by the differential kinematics of the robotic arm directly

$$\begin{bmatrix} \dot{\boldsymbol{P}}_{eb}^b \\ \dot{\boldsymbol{\omega}}_{eb}^b \end{bmatrix} = \boldsymbol{J}_{eb}^b \dot{\boldsymbol{q}}, \tag{3}$$

with $\boldsymbol{J}_{eb}^b \in \mathbb{R}^{6\times n}$ the Jacobian matrix of the robotic arm expressed in $\mathcal{F}_b$ and $\boldsymbol{q} = [q_1, q_2, ..., q_n]^{\mathrm{T}} \in \mathbb{R}^n$ the joint positions vector. n is the number of DoF of the robotic arm.

Substitute Eq.(3) into Eq.(2) and denote

$$\boldsymbol{J}_q = \begin{bmatrix} \boldsymbol{I}_3 & -S(\boldsymbol{R}_b^w \boldsymbol{P}_{eb}^b) \\ \boldsymbol{O}_3 & \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{P}}_b \\ \dot{\boldsymbol{\Omega}}_b \end{bmatrix}, \ \boldsymbol{J}_a = \begin{bmatrix} \boldsymbol{R}_b^w & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{R}_b^w \end{bmatrix} \boldsymbol{J}_{eb}^b,$$

with the Jacobian matrices $\boldsymbol{J}_q$, $\boldsymbol{J}_a$ representing the contribution to the end-effector velocity from the quadrotor and the robotic arm, respectively. Therefore, Eq.(2) can be rewritten in the following compact form

$$\dot{\boldsymbol{v}}_e = \boldsymbol{J}_q \dot{\boldsymbol{v}}_b + \boldsymbol{J}_a \dot{\boldsymbol{q}}, \tag{4}$$

where $\dot{\boldsymbol{v}}_e = [\dot{\boldsymbol{P}}_e^{\mathrm{T}}, \dot{\boldsymbol{\Omega}}_e^{\mathrm{T}}]^{\mathrm{T}}$, $\dot{\boldsymbol{v}}_b = [\dot{\boldsymbol{P}}_b^{\mathrm{T}}, \dot{\boldsymbol{\Omega}}_b^{\mathrm{T}}]^{\mathrm{T}}$ are the generalized velocity of the end-effector and the quadrotor. By adopting the roll-pitch-yaw Euler angles $\boldsymbol{\Theta}_b$ and $\boldsymbol{\Theta}_e$ for the attitude of the quadrotor and the end-effector, and substituting the angular velocity with the time derivative of the Euler angles, we can rewrite (4) in the form of

$$\dot{\boldsymbol{x}}_e = \boldsymbol{T}_e^{-1}[\boldsymbol{J}_q \boldsymbol{T}_b \dot{\boldsymbol{x}}_b + \boldsymbol{J}_a \dot{\boldsymbol{q}}] = \boldsymbol{T}_e^{-1} \boldsymbol{J}_q \boldsymbol{T}_b \dot{\boldsymbol{x}}_b + \boldsymbol{T}_e^{-1} \boldsymbol{J}_a \dot{\boldsymbol{q}}, \tag{5}$$

where $\dot{\boldsymbol{x}}_e = [\dot{\boldsymbol{P}}_e^{\mathrm{T}}, \dot{\boldsymbol{\Theta}}_e^{\mathrm{T}}]^{\mathrm{T}}$, $\dot{\boldsymbol{x}}_b = [\dot{\boldsymbol{P}}_b^{\mathrm{T}}, \dot{\boldsymbol{\Theta}}_b^{\mathrm{T}}]^{\mathrm{T}}$ and

$$\boldsymbol{T}_e = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \overline{\boldsymbol{T}}_e \end{bmatrix}, \boldsymbol{T}_b = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \overline{\boldsymbol{T}}_b \end{bmatrix},$$

with $\overline{\boldsymbol{T}}_e$, $\overline{\boldsymbol{T}}_b$ denoting the mapping matrices relating the corresponding angular velocity with Euler angle time derivative.

Due to the under-actuated property of the quadrotor, position $\boldsymbol{P}_b$ and yaw angle ψ_b are controlled variables, while roll and pitch angles ϕ_b, ϑ_b are intermediate control inputs. Therefore, it is reasonable to define the controlled and uncontrolled motion vector of UAM as

$$\boldsymbol{\xi}_c = \begin{bmatrix} \boldsymbol{P}_b \\ \psi_b \\ \boldsymbol{q} \end{bmatrix} \in \mathbb{R}^{4+n}, \ \boldsymbol{\xi}_{uc} = \begin{bmatrix} \phi_b \\ \vartheta_b \end{bmatrix} \in \mathbb{R}^2,$$

Consequently, Eq.(5) can be further rewritten as

$$\dot{\boldsymbol{x}}_e = \boldsymbol{J}_c\dot{\boldsymbol{\xi}}_c + \boldsymbol{J}_{uc}\dot{\boldsymbol{\xi}}_{uc}, \tag{6}$$

via selecting the columns referred to the corresponding controlled and uncontrolled variables, where $\boldsymbol{J}_c$ and $\boldsymbol{J}_{uc}$ are the Jacobian matrices referred to $\dot{\boldsymbol{\xi}}_c$ and $\dot{\boldsymbol{\xi}}_{uc}$ obtained from $\boldsymbol{T}_e^{-1}\boldsymbol{J}_q\boldsymbol{T}_b$ and $\boldsymbol{T}_e^{-1}\boldsymbol{J}_a$.

III. UAM Tasks Analysis And Formulation

In order to achieve coordinated motion of UAM, we consider several tasks for UAM to accomplish. This section elaborates these tasks and formulates their cost functions.

A. End-effector Tracking

The interaction task of UAM is executed by the end-effector. Thus, it is important to track the desired end-effector trajectory. With the given desired end-effector velocity $\dot{\boldsymbol{x}}_{e,d}$, the end-effector tracking task can be formulated by finding the desired $\dot{\boldsymbol{\xi}}_c^d$ that minimizes the following cost function

$$f_1 = \min_{\dot{\boldsymbol{\xi}}_c} ||\boldsymbol{J}_c\dot{\boldsymbol{\xi}}_c - (\dot{\boldsymbol{x}}_{e,d} - \boldsymbol{J}_{uc}\dot{\boldsymbol{\xi}}_{uc})||^2, \tag{7}$$

where $\dot{\boldsymbol{x}}_{e,d}$ is computed by

$$\dot{\boldsymbol{x}}_{e,d} = \boldsymbol{K}_1(\boldsymbol{x}_t - \boldsymbol{x}_e) + \dot{\boldsymbol{x}}_t, \tag{8}$$

with $\boldsymbol{x}_t$, $\dot{\boldsymbol{x}}_t$ desired end-effector trajectory and its time derivative, respectively, and $\boldsymbol{K}_1$ the gain matrix.

B. Joint Limits Avoidance

To ensure flight safety and prevent risks such as mechanical limits of joints and self-collision of the robotic arm, the motion range of each joint should be determined.

Introduce the following task function similar to that is proposed in [10].

$$\sigma_L = \sum_{i=1}^{n} l_i(q_i), \tag{9}$$

where

$$l_i(q_i) = \begin{cases} \frac{(q_i - \underline{q}_i)^2}{n}, & q_i \le \underline{q}_i \\ 0, & \underline{q}_i < q_i < \overline{q}_i \\ \frac{(\overline{q}_i - q_i)^2}{n}, & \overline{q}_i \le q_i \end{cases}$$

and $\underline{q}_i$, $\overline{q}_i$represent the lower and upper limit of the i^{th} joint, respectively. The differential relation between σ_L and $\boldsymbol{\xi}_c$ is

$$\dot{\sigma}_L = \boldsymbol{J}_L\dot{\boldsymbol{\xi}}_c, \tag{10}$$

where $\boldsymbol{J}_L = \begin{bmatrix}\boldsymbol{0}_{1\times 4} & \boldsymbol{J}_l\end{bmatrix} \in \mathbb{R}^{1\times(4+n)}$ is the task Jacobian and $\boldsymbol{J}_l = \begin{bmatrix}\frac{\partial l_1}{\partial q_1} & \frac{\partial l_2}{\partial q_2} & \dots & \frac{\partial l_n}{\partial q_n}\end{bmatrix} \in \mathbb{R}^{1\times n}$.

When the joint positions are within their limitations, the task should be deactivated and $\sigma_L = 0$, allowing the system redundancy to be utilized for other tasks. Therefore, define the cost function

$$f_2 = \min_{\dot{\boldsymbol{\xi}}_c} ||\boldsymbol{J}_L\dot{\boldsymbol{\xi}}_c - \dot{\sigma}_{L,d}||^2, \tag{11}$$

where $\dot{\sigma}_{L,d}$ is the desired value of $\dot{\sigma}_L$ which is obtained by

$$\dot{\sigma}_{L,d} = \boldsymbol{K}_2(0 - \sigma_L) = -\boldsymbol{K}_2\sigma_L, \tag{12}$$

with $\boldsymbol{K}_2$ the gain matrix.

C. CoG Alignment

When CoG of the robotic arm is not aligned with the gravity vector of the quadrotor, undesirable torque will significantly affect the motion of the flying base [21]. In order to avoid this disturbance, it is beneficial to construct a task to ensure the alignment.

The CoG vector of the robotic arm in the frame $\mathcal{F}_b$, ${}^b\boldsymbol{p}_G \in \mathbb{R}^3$ can be expressed as

$${}^b\boldsymbol{p}_G = \frac{\sum_{i=1}^{n} m_i {}^b\boldsymbol{p}_{Gi}}{\sum_{i=1}^{n} m_i}, \tag{13}$$

where m_i is the mass of the i^{th} link, ${}^b\boldsymbol{p}_{Gi} \in \mathbb{R}^3$ is the CoG vector of the i^{th} link expressed in the frame $\mathcal{F}_b$. Denote ${}^w\boldsymbol{p}_{Gxy} \in \mathbb{R}^2$ as the displacement of the CoG in the horizontal plane of the frame $\mathcal{F}_w$

$${}^w\boldsymbol{p}_{Gxy} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}\boldsymbol{R}_b^{wb}\boldsymbol{p}_G, \tag{14}$$

and introduce the square distance of the robotic arm CoG w.r.t. the z axis of the frame $\mathcal{F}_w$, which can be written as

$$\sigma_G = {}^w\boldsymbol{p}_{Gxy}^T{}^w\boldsymbol{p}_{Gxy}. \tag{15}$$

The differential relation between σ_G and $\boldsymbol{\xi}_c$ is

$$\dot{\sigma}_G = \boldsymbol{J}_G\dot{\boldsymbol{\xi}}_c, \tag{16}$$

with $\boldsymbol{J}_G$ representing the corresponding task Jacobian.

To achieve the CoG alignment, introduce the cost function

$$f_3 = \min_{\dot{\boldsymbol{\xi}}_c} ||\boldsymbol{J}_G\dot{\boldsymbol{\xi}}_c - \dot{\sigma}_{G,d}||^2, \tag{17}$$

where $\dot{\sigma}_{G,d}$ is the desired value of $\dot{\sigma}_G$ to be computed as

$$\dot{\sigma}_{G,d} = \boldsymbol{K}_3(0 - \sigma_G) = -\boldsymbol{K}_3\sigma_G, \tag{18}$$

with $\boldsymbol{K}_3$ the gain matrix.

IV. Hierarchical Quadratic Programming For Trajectory Generation

HQP is a sequence of QPs in which each QP aims to minimize a cost function subject to equality and inequality constraints as well as solutions of other QPs with higher priority. For the first hierarchical task, the QP model is formulated as

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} ||\boldsymbol{A}_1\boldsymbol{\alpha} - \boldsymbol{b}_1||^2, \\ &s.t.\ \ \boldsymbol{C}_1\boldsymbol{\alpha} \le \boldsymbol{d}_1,\ \ \boldsymbol{E}_1\boldsymbol{\alpha} = \boldsymbol{f}_1. \end{aligned} \tag{19}$$

The solution of the first QP $\boldsymbol{\alpha}_1^*$ is then used to constrain the following QPs successively. At the k^{th} level of priority, the QP model can be formulated as

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} ||\boldsymbol{A}_k\boldsymbol{\alpha} - \boldsymbol{b}_k||^2, \\ &s.t.\ \ \boldsymbol{C}_1\boldsymbol{\alpha} \le \boldsymbol{d}_1,\ \dots,\ \boldsymbol{C}_k\boldsymbol{\alpha} \le \boldsymbol{d}_k, \\ &\boldsymbol{E}_1\boldsymbol{\alpha} = \boldsymbol{f}_1,\ \dots,\ \boldsymbol{E}_k\boldsymbol{\alpha} = \boldsymbol{f}_k, \\ &\boldsymbol{A}_1\boldsymbol{\alpha} = \boldsymbol{A}_1\boldsymbol{\alpha}_1^*,\ \dots,\ \boldsymbol{A}_{k-1}\boldsymbol{\alpha} = \boldsymbol{A}_{k-1}\boldsymbol{\alpha}_{k-1}^*, \end{aligned} \tag{20}$$

where matrix $\boldsymbol{A}_i$ and vector $\boldsymbol{b}_i$ describe the cost function of the i^{th} QP, matrix $\boldsymbol{C}_i$ and vector $\boldsymbol{d}_i$ describe the inequality

constraints, while matrix $\boldsymbol{E}_i$ and vector $\boldsymbol{d}_i$ describe the equality constraints, $\boldsymbol{\alpha}_i^*$ is the solution of the QP at the i^{th} level. Notably, each QP inherits the equality constraints from the preceding hierarchy of level. The solution of the last QP serves as the ultimate solution of the whole HQP framework.

In order to complete three tasks proposed in Section III, an HQP framework incorporating three QP models is constructed to solve for the controlled motion input vector in a strict order of priority. In this framework, the primary task is end-effector tracking, with joint limits avoidance and CoG alignment as secondary and tertiary tasks, respectively.

A. QP for End-effector Tracking

In this article we assume that UAM is not meant for acrobatic maneuvers. Therefore, the uncontrolled motion variables ϕ_b and ϑ_b are assumed to be 0. Expanding the right hand side of cost function (7) obtains

$$f_1 = \min_{\dot{\boldsymbol{\xi}}_c} \dot{\boldsymbol{\xi}}_c^T \boldsymbol{J}_c^T \boldsymbol{J}_c \dot{\boldsymbol{\xi}}_c - 2\dot{\boldsymbol{x}}_{e,d}^T \boldsymbol{J}_c. \tag{21}$$

Rewrite Eq.(21) in the following form

$$\min_{\dot{\boldsymbol{\xi}}_c} \frac{1}{2} \dot{\boldsymbol{\xi}}_c^T \boldsymbol{H}_1 \dot{\boldsymbol{\xi}}_c + \boldsymbol{g}_1 \dot{\boldsymbol{\xi}}_c, \tag{22}$$

where $\boldsymbol{H}_1 = 2\boldsymbol{J}_c^T \boldsymbol{J}_c$ and $\boldsymbol{g}_1 = -2\dot{\boldsymbol{x}}_{e,d}^T \boldsymbol{J}_c$. In the first QP model, only inequality constraint is imposed

$$s.t. \ \ \underline{\dot{\boldsymbol{\xi}}}_c \leq \dot{\boldsymbol{\xi}}_c \leq \overline{\dot{\boldsymbol{\xi}}}_c, \tag{23}$$

where $\underline{\dot{\boldsymbol{\xi}}}_c$ and $\overline{\dot{\boldsymbol{\xi}}}_c$ are the lower and upper bounds of $\dot{\boldsymbol{\xi}}_c$.

B. QP for Joint Limits Avoidance

Similarly, expanding the right hand side of cost function (11) renders

$$\min_{\dot{\boldsymbol{\xi}}_c} \frac{1}{2} \dot{\boldsymbol{\xi}}_c^T \boldsymbol{H}_2 \dot{\boldsymbol{\xi}}_c + \boldsymbol{g}_2 \dot{\boldsymbol{\xi}}_c, \tag{24}$$

where $\boldsymbol{H}_2 = 2\boldsymbol{J}_L^T \boldsymbol{J}_L$ and $\boldsymbol{g}_3 = -2\dot{\sigma}_{L,d}^T \boldsymbol{J}_L$. The constraints of this QP model are

$$s.t. \ \ \begin{cases} \underline{\dot{\boldsymbol{\xi}}}_c \leq \dot{\boldsymbol{\xi}}_c \leq \overline{\dot{\boldsymbol{\xi}}}_c \\ \boldsymbol{J}_c \dot{\boldsymbol{\xi}}_c = \boldsymbol{J}_c \dot{\boldsymbol{\xi}}_{c1}^* \end{cases}, \tag{25}$$

with an additional equality constraint inheriting from the first QP model, where $\dot{\boldsymbol{\xi}}_{c1}^*$ is the solution of (22) constrained by (23).

C. QP for CoG Alignment

Expand the right hand side of cost function (17) and it can be written as

$$\min_{\dot{\boldsymbol{\xi}}_c} \frac{1}{2} \dot{\boldsymbol{\xi}}_c^T \boldsymbol{H}_3 \dot{\boldsymbol{\xi}}_c + \boldsymbol{g}_3 \dot{\boldsymbol{\xi}}_c, \tag{26}$$

where $\boldsymbol{H}_3 = 2\boldsymbol{J}_G^T \boldsymbol{J}_G$ and $\boldsymbol{g}_3 = -2\dot{\boldsymbol{\sigma}}_{G,d}^T \boldsymbol{J}_G$. The constraints of this last QP model are

$$s.t. \ \ \begin{cases} \underline{\dot{\boldsymbol{\xi}}}_c \leq \dot{\boldsymbol{\xi}}_c \leq \overline{\dot{\boldsymbol{\xi}}}_c \\ \boldsymbol{J}_c \dot{\boldsymbol{\xi}}_c = \boldsymbol{J}_c \dot{\boldsymbol{\xi}}_{c1}^* \\ \boldsymbol{J}_L \dot{\boldsymbol{\xi}}_c = \boldsymbol{J}_L \dot{\boldsymbol{\xi}}_{c2}^* \end{cases}, \tag{27}$$

where $\dot{\boldsymbol{\xi}}_{c2}^*$ is the solution of (24) constrained by (25). Denote $\dot{\boldsymbol{\xi}}_{c3}^*$ as the solution of the last QP model, and it is also the ultimate solution of the HQP framework.

TABLE I
D-H PARAMETERS OF THE 4-DOF ROBOTIC ARM

Link	$\theta_i(^\circ)$	$\alpha_i(^\circ)$	$a_i(m)$	$d_i(m)$
1	q_1	90	0	0.077
2	q_2	0	0.130	0
3	q_3	0	0.135	0
4	q_4	0	0.126	0

TABLE II
HQP PARAMETERS AND CONSTRAINTS

Section	**Parameters**
Gain Matrix	$\boldsymbol{K}_1 = diag(1.1, 1.1, 1.1, 0.0, 0.5, 2.0)$ $\boldsymbol{K}_2 = 2.0$ $\boldsymbol{K}_3 = 2.0$
Inequality Constraint	$\underline{\dot{\boldsymbol{\xi}}}_c$ = [-0.5,-0.5,-0.2,-0.4, 0.0,-0.45,-0.3,-0.15] $\overline{\dot{\boldsymbol{\xi}}}_c = [0.5, 0.5, 0.2, 0.4, 0.0, 0.45, 0.3, 0.15]$
Joint Limits	$\underline{\boldsymbol{q}}$ = [0.0,-1.25,-1.0,-0.5] $\overline{\boldsymbol{q}} = [0.0, 1.25, 1.0, 0.5]$

V. EXPERIMENT

A. System Setup

The UAM system used in the experiment (shown in Fig. 1) consists of a coaxial quadrotor and a 4-DoF robotic arm. The system integrates an onboard computer and a control unit running planning and control programs, respectively. The position of the UAM is obtained via a motion capture system. Table I lists the Denavit-Hartenberg (D-H) parameters of the robotic arm. System software architecture (shown in Fig. 2) includes four major software modules which are all programmed in C/C++ within the ROS environment (on Ubuntu 20.04). HQP Planner resolves the HQP framework in Section IV online via qpOASES toolkit [22]. State Manager is a supervisor that collects and monitors system states, and activates HQP Planner by ROS Service. Arm Control receives referenced joint velocities from UAM State Manager and transmits the references to actuators after numerical integration. Likewise, the UAV Control transmits the referenced trajectory to the quadrotor after numerical integration and differentiation of the original data. The structure of low level controller of the quadrotor is the same as in our previous work [21]. Moreover, a disturbance observer (DO) [23] is employed to estimate and compensate disturbance torque and

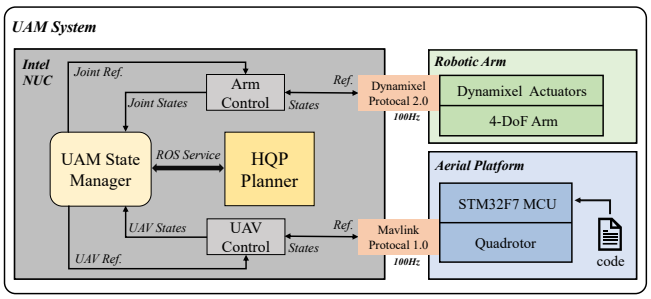


Fig. 2. Software architecture of UAM system.

Fig. 3. Snapshots of the aerial grasping experiment. Flight video is available at https://youtu.be/2JXBj3hAiuo.

model uncertainty. Details about the system setup and control algorithm can be referred to [21].

B. Experiment Scenarios and Results

To validate the proposed method, indoor real-world experiments are carried out, in which UAM is assigned to grasp an object on a moving platform. During the grasping process, the proposed algorithm generates constraints-satisfied trajectories for both the quadrotor and the arm joints, such that the end-effector is able to track the moving target and UAM attempts to adjust its configuration to avoid joint limits and minimize CoG shift. The assigned task consists of three phases:

- *Steady Phase*: End-effector hovers at a setpoint with the gripper open. Move the quadrotor and arm joints within inequality constraints and joint limits to minimize CoG shift in order to reduce power consumption and enhance trajectory tracking precision in the subsequent phase.
- *Approaching Phase*: During this phase, the end-effector begins to track the moving target while UAM trying to maintain its previous configuration.
- *Grasping Phase*: The gripper close to grasp the target when the final position of the end-effector reach a good accuracy within the error of 2 cm.

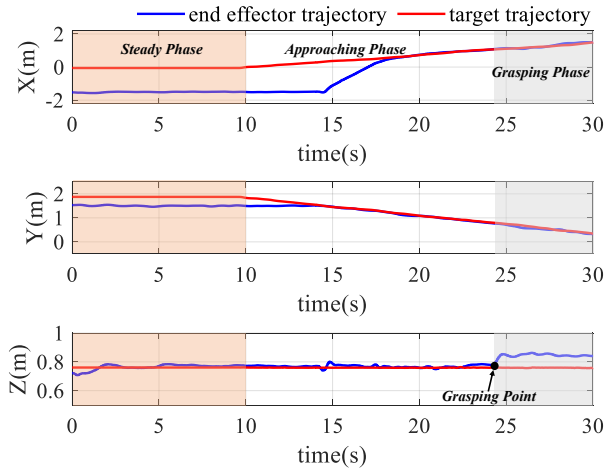


Fig. 4. Comparison between the trajectories of the end-effector and the moving target.

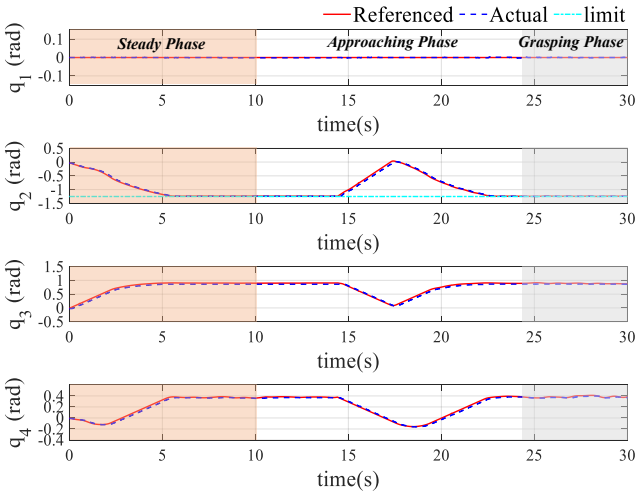


Fig. 5. Reference and actual joint angles of the robotic arm.

Figs. 3-8 show the aerial grasping experimental results, obtained by selecting the parameters and constraints listed in Table II.

Fig.3 gives some experiment snapshots. In detail, the first snapshot depicts the initial configuration of UAM with quadrotor hovering and all the robotic arm joints at $0rad$. In the second snapshot, UAM configuration reaches the first phase where CoG shift is minimized and joint limits are satisfied. The last two snapshots illustrate the approaching phase and the grasping phase, respectively.

The end-effector and moving target trajectories in Fig. 4 prove that our main task is completed with good tracking performance. The target starts to move at $10s$ and subsequently the end-effector is driven by the quadrotor and the robotic arm to follow the target. The grasping phase is reached at around $24s$ when the gripper closes and UAM ascends to disengage the moving platform.

Figs. 5-6 compare the quadrotor and robotic arm measured trajectories to the HQP reference trajectories. As shown, all joint angles are within upper and lower bounds, and joint q_2 is saturated at $1.25rad$, indicating that our secondary task (i.e. joint limits) is activated on the basis of end-effector tracking.

Section I states that the HQP framework can resolve the redundancy utilization with inequality constraints at any task

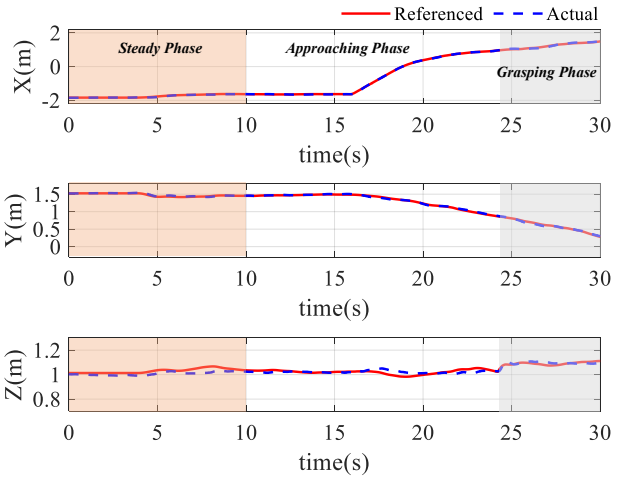
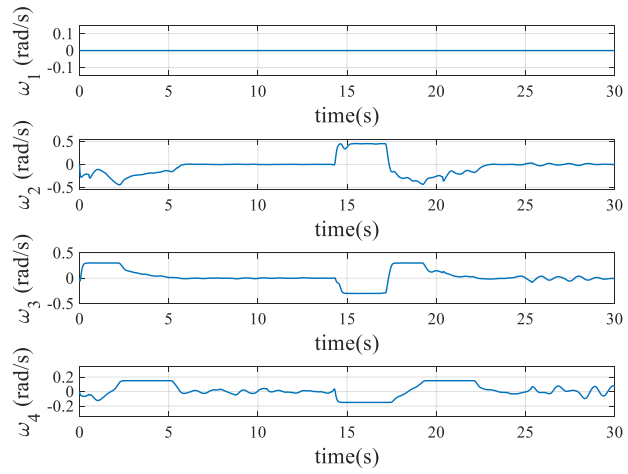
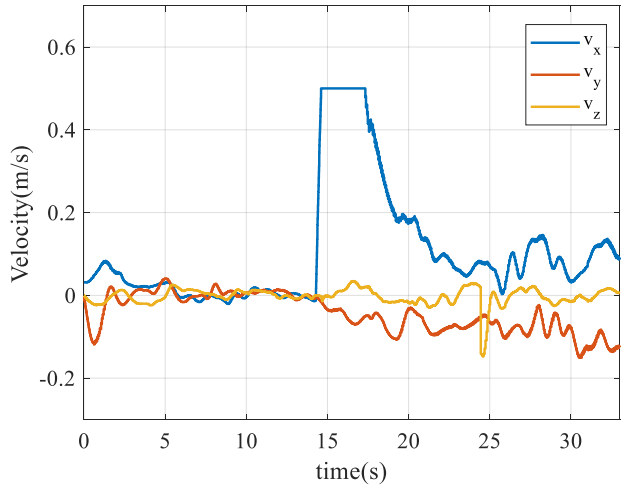


Fig. 6. Reference and actual trajectories of the quadrotor.

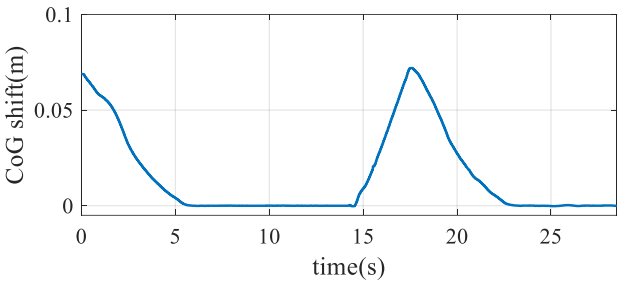


(a) Reference angular speed of the joints

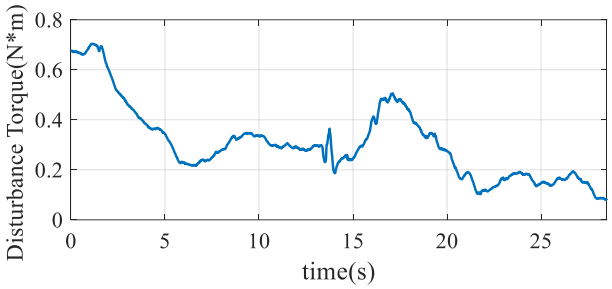


(b) Reference velocity of the quadrotor

Fig. 7. Reference quadrotor velocity and joints angular speed generated by HQP. All the joint references and the quadrotor reference in X direction have reached their boundary of constraints.



(a) CoG shift of the UAM



(b) Disturbance torque estimated by DO

Fig. 8. CoG shift of the system and disturbance torque estimated by DO.

level. As illustrated in Fig. 7, the reference quadrotor velocity and joints angular speed is clearly generated under inequality constraints given in Table II, which avoids high speed and enhances flight stability.

CoG alignment is assigned to be the tertiary task. The dynamic shift of the CoG is presented in Fig. 8(a). Notably, the shift has been minimized to nearly zero during the steady phase, but increases as UAM approaches the target. This is because the primary task requires relatively larger velocity of the end-effector when UAM is far from the target, but the maximum reference velocity of the quadrotor is limited to $0.5m/s$. Therefore, more speed has to be allocated to the manipulator joints and the robotic arm stretch forward to contribute to the motion of the end-effector. In this condition, the primary and tertiary tasks contradict each other. HQP framework can deal with this contradiction in a task priority-based manner. By comparing Figs. 4, 7(b), and 8(a), the reference quadrotor velocity is no longer saturated and the CoG shift starts to decrease when the end-effector is close to the target (at about $17.5s$).

Fig. 8(b) shows the estimated results of the disturbance torque provide by DO. Observably, the two curves in Fig. 8 share the same trend. However, the estimated disturbance torque at the second peak of the curve of the CoG shift is less than its initial value ($0.50Nm$ vs. $0.67Nm$). This occurs due to the DO used in this paper only guarantees the exponential convergence to constant interference, and the estimation error occurs when the robotic arm is moving.

VI. Conclusion

In this work, a coordinated trajectory generation method for UAM system by optimization-based approach is pro-

posed and demonstrated. To generate feasible trajectories for both the quadrotor and the robotic arm, different tasks are formulated in a strict order of priority with either equality or inequality constraints, which are solved using the hierarchical quadratic programming framework. Validations of the proposed approach are shown in experiment using an actual aerial robot.

REFERENCES

[1] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 2018.

[2] X. Ding, P. Guo, K. Xu, and Y. Yu, "A review of aerial manipulation of small-scale rotorcraft unmanned robotic systems," *Chinese Journal of Aeronautics*, vol. 32, no. 1, pp. 200–214, 2019.

[3] A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi, "Past, present, and future of aerial robotic manipulators," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 626–645, 2022.

[4] C. E. Doyle, J. J. Bird, T. A. Isom, J. C. Kallman, D. F. Bareiss, D. J. Dunlop, R. J. King, J. J. Abbott, and M. A. Minor, "An avian-inspired passive mechanism for quadrotor perching," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 2, pp. 506–517, 2013.

[5] F. Ruggiero, M. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Perz, V. Lippiello, A. Ollero, and B. Siciliano, "A multilayer control for multirotor UAVs equipped with a servo robot arm," in *Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, 2015, pp. 4014–4020.

[6] C. Korpela, M. Orsag, and P. Oh, "Towards valve turning using a dual-arm aerial manipulator," in *Proceedings of 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, USA, 2014, pp. 3411–3416.

[7] K. Steich, M. Kamel, P. Beardsley, M. K. Obrist, R. Siegwart, and T. Lachat, "Tree cavity inspection using aerial robots," in *Proceedings of 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea (South), 2016, pp. 4856–4862.

[8] N. Imanberdiyev, J. Monica, and E. Kayacan, "A multi-task velocity-based redundancy resolution strategy for unmanned aerial manipulators," in *Proceedings of 2019 18th European Control Conference (ECC)*, Naples, Italy, 2019, pp. 1130–1135.

[9] M. Ryll, D. Bicego, and A. Franchi, "A truly redundant aerial manipulator exploiting a multi-directional thrust base," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 138–143, 2018.

[10] K. Baizid, G. Giglio, F. Pierri, M. A. Trujillo, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Experiments on behavioral coordinated control of an unmanned aerial vehicle manipulator system," in *Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, 2015, pp. 4680–4685.

[11] V. Lippiello, J. Cacace, A. Santamaria-Navarro, J. Andrade-Cetto, M. . Trujillo, Y. R. Rodrguez Esteves, and A. Viguria, "Hybrid visual servoing with hierarchical task composition for aerial manipulation," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 259–266, 2016.

[12] A. Santamaria-Navarro, V. Lippiello, and J. Andrade-Cetto, "Task priority control for aerial manipulation," in *Proceedings of 2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*, Hokkaido, Japan, 2014, pp. 1–6.

[13] E. Cataldi, F. Real, A. Suarez, P. Di Lillo, F. Pierri, G. Antonelli, F. Caccavale, G. Heredia, and A. Ollero, "Set-based inverse kinematics control of an anthropomorphic dual arm aerial manipulator," in *Proceedings of 2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, 2019, pp. 2960–2966.

[14] G. Muscio, F. Pierri, M. A. Trujillo, E. Cataldi, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Coordinated control of aerial robotic manipulators: Theory and experiments," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1406–1413, 2018.

[15] R. Rossi, A. Santamaria-Navarro, J. Andrade-Cetto, and P. Rocco, "Trajectory generation for unmanned aerial manipulators through quadratic programming," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 389–396, 2017.

[16] N. Imanberdiyev and E. Kayacan, "Redundancy resolution based trajectory generation for dual-arm aerial manipulators via online model predictive control," in *Proceedings of IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, 2020, pp. 674–681.

[17] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.

[18] A. Escande, N. Mansard, and P. B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[19] S. Kim, K. Jang, S. Park, Y. Lee, S. Y. Lee, and J. Park, "Whole-body control of non-holonomic mobile manipulator based on hierarchical quadratic programming and continuous task transition," in *Proceedings of 2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, Toyonaka, Japan, 2019, pp. 414–419.

[20] D. Koung, O. Kermorgant, I. Fantoni, and L. Belouaer, "Cooperative multi-robot object transportation system based on hierarchical quadratic programming," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6466–6472, 2021.

[21] W. Zhang, Q. Liu, M. Wang, J. Jia, S. Lyu, K. Guo, X. Yu, and L. Guo, "Design of an aerial manipulator system applied to capture missions," in *Proceedings of 2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, Athens, Greece, 2021, pp. 1063–1069.

[22] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[23] K. Guo, J. Jia, X. Yu, L. Guo, and L. Xie, "Multiple observers based anti-disturbance control for a quadrotor UAV against payload and wind disturbances," *Control Engineering Practice*, vol. 102, p. 104560, 2020.