

Fault-tolerant Control of an Over-actuated UAV Platform Built on Quadcopters and Passive Hinges

Yao Su, *Member, IEEE*, Pengkang Yu, Matthew J. Gerber, Lecheng Ruan
and Tsu-Chin Tsao, *Senior Member, IEEE*

Abstract—Propeller failure is a major cause of multirotor Unmanned Aerial Vehicles (UAVs) crashes. While conventional multirotor systems struggle to address this issue due to underactuation, over-actuated platforms can continue flying with appropriate fault-tolerant control (FTC). This paper presents a robust FTC controller for an over-actuated UAV platform composed of quadcopters mounted on passive joints, offering input redundancy at both the high-level vehicle control and the low-level quadcopter control of vectored thrusts. To maximize the benefits of input redundancy during propeller failure, the proposed FTC controller features a hierarchical control architecture with three key components: (i) a low-level adjustment strategy to prevent propeller-level thrust saturation; (ii) a compensation loop for mitigating introduced disturbances; (iii) a nullspace-based control allocation framework to avoid quadcopter-level thrust saturation. Through reallocating actuator inputs in both the low-level and high-level control loops, the low-level quadcopter control can be maintained with up to two failed propellers, ensuring that the whole platform remains stable and avoids crashing. The proposed controller's superior performance is thoroughly examined through simulations and real-world experiments.

Index Terms—Over-actuated UAV, propeller failure, fault-tolerant control (FTC), nullspace allocation, input redundancy, optimization

I. INTRODUCTION

OVER-ACTUATED multirotor UAVs have been proposed in the last decade to overcome the underactuation issue of traditional co-linear multirotor UAVs [1], leveraging vectored thrusts to improve dynamic properties. There are mainly two categories of realizations in this field. The first group of works [2–4] employs multiple propeller-motor pairs in various or varying directions to achieve full or over-actuation. The second group of works [5–9] utilizes standard quadcopters mounted on passive joints as actuation modules, simplifying the design and prototyping process while reducing internal disturbance levels [9]. With additional actuators onboard, these platforms should demonstrate increased robustness against

Manuscript received: 2 August 2022; revised 20 February 2023 and 24 April 2023; accepted: 12 May, 2023. Date of publication xxxx, 2023; date of current version xxxx, 2023. Recommended by Technical Editor xxx and Senior Editor Kostas Kyriakopoulos. (Yao Su and Pengkang Yu contributed equally to this work.) (Corresponding authors: Yao Su and Lecheng Ruan.)

Yao Su, Pengkang Yu, Matthew J. Gerber, Lecheng Ruan, and Tsu-Chin Tsao are with Mechanical and Aerospace Engineering Department, University of California, Los Angeles (UCLA), Los Angeles, CA 90095 USA (e-mail: yaosu@ucla.edu; paulyu1994@ucla.edu; gerber211@ucla.edu; ruanlecheng@ucla.edu; ttsao@ucla.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at <https://doi.org/10.1109/TMECH.2023>.

Digital Object Identifier

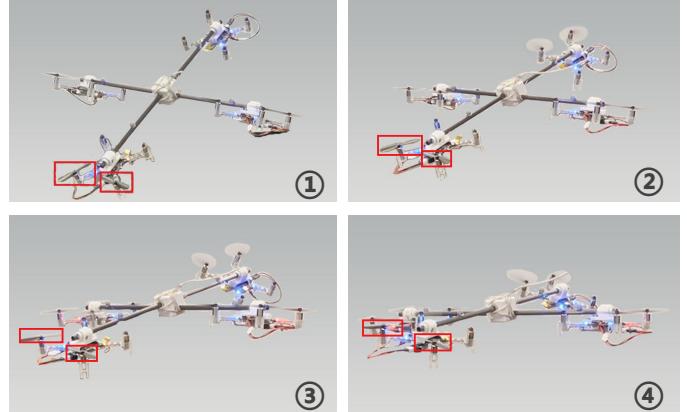


Fig. 1: With two failed propellers on one quadcopter module, our proposed FTC algorithm can prevent the platform from crashing and maintain the capability of trajectory tracking. (Failed propellers are labeled by red boxes).

propeller failure compared to conventional quadcopters: without sufficient FTC algorithms, over-actuated multirotor systems are more likely to suffer from propeller failure than quadcopters, resulting in crashes. Therefore, developing FTC algorithms to reduce crash likelihood in the event of propeller failure is of great interest.

In this paper, we implement the previously proposed nullspace-based control allocation framework [10] to address the **thrust force saturation** issue in over-actuated UAV platforms. We validate this approach on our customized over-actuated platform [8], where four mini-quadcopters are connected to the mainframe via 1 Degree-of-freedom (DoF) passive hinges, serving as tiltable thrust generators. The platform features input redundancy in both high-level wrench control and low-level tiltable thrusts control [11]. Following this, we propose a FTC algorithm specifically for scenarios in which one or more propellers on a single quadcopter (denoted as **Bad QC**) fail while the other quadcopters (denoted as **Good QCs**) remain functional (see Fig. 1). This FTC controller employs a hierarchical structure with three main components: (i) a low-level controller to adjust the propeller-level thrust force distribution on the Bad QC, (ii) a high-level controller to reallocate quadcopter-level thrust force distribution among the Good and Bad QCs, and (iii) a compensation loop for disturbance attenuation.

In the low-level controller, due to the reduced thrust and torque capacity, the thrust distribution among the propellers of the Bad QC is adjusted to maintain control of the tilting angle

and the thrust. However, this low-level adjustment introduces an interaction torque between the central frame and the Bad QC, acting as a disturbance. To attenuate this undesirable torque, the redundant inputs of the three Good QCs are utilized to formulate a compensation loop [11]. Finally, in the high-level controller, the nullspace-based allocation framework is implemented to optimize the thrust distribution of all four quadcopters. This control framework is validated through both dynamic simulations and real-world experiments.

Our contributions are highlighted as follows:

- (1) We analyze the thrust force saturation issue in over-actuated UAV platforms, and implement the nullspace-based control allocation framework to address it.
- (2) We develop a FTC algorithm to fully utilize the redundancy of over-actuated UAV when some of the propellers on a single quadcopter module fail. We analyze and compare two different low-level control methods, design a compensation loop for disturbance attenuation, and incorporate a nullspace-based control allocation to find the optimal allocation solution under this scenario.
- (3) We present simulation and experimental validations to demonstrate the effectiveness of our proposed control algorithm in handling the thrust force saturation and propeller failure in over-actuated UAV platforms.

The remainder of the paper is organized as follows. Related work is summarized in Sec. II. The dynamics models of the over-actuated UAV platform are reviewed in Sec. III. Sec. IV presents the control architecture and the nullspace-based framework for control allocation. Sec. V describes the FTC framework to handle propeller failure. Sec. VI and Sec. VII present the simulation and experiment results. Finally, we conclude the paper with a discussion in Sec. VIII and Sec. IX.

II. RELATED WORK

A. Control Allocation

Control allocation in over-actuated UAV platforms, which computes individual actuator commands from the desired total wrench, is a constrained nonlinear optimization problem that is generally difficult to solve with high efficiency. Ryll *et al.* first utilized dynamic output linearization for control allocation at a higher differential level, requiring accurate acceleration measurements or estimation [12]. Kamel *et al.* introduce a Force Decomposition (FD)-based method, transforming the nonlinear allocation problem into a linear one by defining intermediate variables [2]. This method improved computational speed by directly choosing the least-square solution but sacrificing input redundancies. Furthermore, iterative approach [13] and separation method [14] were proposed for improved efficiency. However, none of these methods [2, 12–14] considered input constraints, leading to instability when the input constraints were triggered [15].

The Quadratic Programming (QP)-based framework [16] relied on discretization and linearization to incorporate both inequality and equality constraints. Nonetheless, it only generated approximate solutions, introducing additional disturbance to the control system. In our previous work [10], we developed

a **nullspace-based** allocation framework to combine the benefits of FD-based and QP-based frameworks and provide exact allocation solutions that satisfied the defined input constraints in real-time. Specifically, we demonstrated its capability by addressing the kinematic-singularity problem of a twist-and-tilt rotor platform [6]. In this paper, we further implement this framework to address the issue of thrust force saturation and propeller failure on a different over-actuated UAV platform.

B. Fault-tolerant Control

UAV **fault-tolerant control (FTC)** strategies can be categorized based on their configurations. For quadcopters, due to underactuation, propeller failure typically requires sacrificing yaw motion control to maintain full translational control [17–21]. Multirotor UAVs with more than six controllable inputs [4, 22–24], or tilt-rotor quadcopters [25–28] exhibit greater robustness against propeller loss due to **input redundancy**. The Y-shaped hexarotor platform with tilted rotors exhibited enhanced rotor-failure robustness compared to the standard star-shaped hexarotor platform [29]. Related FTC controllers were presented in [30] based on modifications of control allocation and in [31] based on Center-of-Mass shifting. A new type of over-actuated UAV platforms integrates quadcopters and passive joints to achieve full actuation [5, 6, 8]. Inherently, these platforms possess more propellers than standard tilt-rotor platforms, increasing the likelihood of propeller failure. In this paper, we present an FTC for this type of UAV platform that sufficiently utilizes the redundancy of the entire platform at both high-level and low-level control, enhancing platform robustness against propeller failure. As a result, various UAV platforms [6, 13, 25, 26, 32, 33], which may have different thrust generation capabilities among propellers or thrust-generation modules under propeller failure, can achieve improved rotor-failure robustness.

III. PLATFORM DYNAMICS

The over-actuated UAV discussed in this paper is built upon regular quadcopters mounted on passive hinges and is a representative configuration for over-actuated UAVs with auxiliary inputs [8, 9]. As shown in Fig. 2, we define the world frame, body frame, and quadcopter frames as \mathcal{F}_W , \mathcal{F}_B , and \mathcal{F}_{Q_i} , respectively. The position of the central frame is defined as $\xi = [x, y, z]^T$, the attitude is defined in the roll-pitch-yaw convention as $\eta = [\phi, \theta, \psi]^T$, and the angular velocity is defined as $\nu = [p, q, r]^T$.

A. Platform Dynamics Model

The dynamics model of this platform can be simplified as,

$$\begin{bmatrix} {}^W\xi \\ {}^B\nu \end{bmatrix} = \begin{bmatrix} \frac{1}{m} {}^W_B\mathbf{R} & 0 \\ 0 & {}^B\mathbf{I}^{-1} \end{bmatrix} \mathbf{u} + \begin{bmatrix} {}^W\mathbf{G} \\ \mathbf{0} \end{bmatrix}, \quad (1)$$

where m is the total mass of the platform, \mathbf{G} is the gravitational acceleration, ${}^W_B\mathbf{R}$ is the rotation matrix from \mathcal{F}_B to \mathcal{F}_W , \mathbf{I} is the inertia matrix of the platform. And

$$\mathbf{u} = \begin{bmatrix} \sum_{i=1}^4 {}^B_Q_i \mathbf{R} T_i \hat{\mathbf{z}} \\ \sum_{i=1}^4 (\mathbf{d}_i \times {}^B_Q_i \mathbf{R} T_i \hat{\mathbf{z}}) \end{bmatrix} = \begin{bmatrix} \mathbf{J}_\xi(\boldsymbol{\alpha}) \\ \mathbf{J}_\nu(\boldsymbol{\alpha}) \end{bmatrix} \mathbf{T}, \quad (2)$$

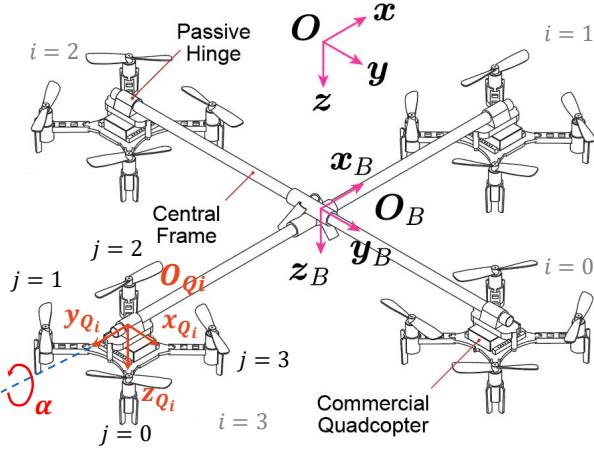


Fig. 2: The platform presented in this paper: four commercial quadcopters are passively hinged to the central frame with 1 DoF connection.

where \mathbf{d}_i is the distance vector from \mathcal{F}_B 's center to \mathcal{F}_{Q_i} , and

$$\begin{aligned} \mathbf{J}_\xi &= \begin{bmatrix} -\sin \alpha_0 & 0 & \sin \alpha_2 & 0 \\ 0 & \sin \alpha_1 & 0 & -\sin \alpha_3 \\ \cos \alpha_0 & \cos \alpha_1 & \cos \alpha_2 & \cos \alpha_3 \end{bmatrix}, \\ \mathbf{J}_\nu &= l \begin{bmatrix} -\cos \alpha_0 & 0 & \cos \alpha_2 & 0 \\ 0 & \cos \alpha_1 & 0 & -\cos \alpha_3 \\ \sin \alpha_0 & \sin \alpha_1 & \sin \alpha_2 & \sin \alpha_3 \end{bmatrix}, \\ \boldsymbol{\alpha} &= [\alpha_0 \quad \alpha_1 \quad \alpha_2 \quad \alpha_3]^T, \quad \mathbf{T} = [T_0 \quad T_1 \quad T_2 \quad T_3]^T, \end{aligned} \quad (3)$$

l is the distance from the geometric center of each quadcopter to the geometric center of the central frame, α_i is the tilting angle of quadcopter i (denoted as \mathcal{Q}_i), and T_i is the magnitude of the thrust generated by \mathcal{Q}_i .

B. Actuator Dynamics

For each \mathcal{Q}_i , the four rotating propellers collectively generate an independent force and torque output according to:

$$\begin{bmatrix} T_i \\ M_i^x \\ M_i^y \\ M_i^z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -b & -b & b & b \\ b & -b & -b & b \\ c_\tau & -c_\tau & c_\tau & -c_\tau \end{bmatrix} \begin{bmatrix} t_{i0} \\ t_{i1} \\ t_{i2} \\ t_{i3} \end{bmatrix}, \quad (4)$$

where M_i^x , M_i^y , and M_i^z are the torque outputs in \mathcal{F}_{Q_i} ; b is a constant defined as $b = a/\sqrt{2}$ with a the arm length of the quadcopter; c_τ is a constant defined as $c_\tau = K_\tau/K_T$ with K_τ the propeller drag constant and K_T the propeller thrust constant; and t_{ij} is the thrust force generated by propeller j (denoted as \mathcal{P}_j) of \mathcal{Q}_i , defined by: $t_{ij} = K_T \omega_{ij}^2$, where ω_{ij} is the rotational speed of \mathcal{P}_j on \mathcal{Q}_i . The torque outputs M_i^y are related to the hinge angles α_i through the tilting dynamics [8]:

$$\ddot{\alpha}_i = \frac{1}{I_i^y} M_i^y - \sin\left(\frac{\pi}{2}i\right)\dot{p} - \cos\left(\frac{\pi}{2}i\right)\dot{q}, \quad (5)$$

where I_i^y is the inertia in the y_{Q_i} direction.

IV. NOMINAL CONTROL

A. Hierarchical Architecture & Tracking Control

The overall controller features a hierarchical structure, consisting of (i) a high-level controller that provides the

desired wrench commands for the platform to track a reference trajectory, and maps these commands to the inputs of each thrust generator through control allocation; (ii) a low-level controller for each quadcopter to achieve rapid response in tracking the desired joint angle and thrust. The stability of this controller has been demonstrated in our previous work, please refer to [9] for more details.

In high-level control, feedback linearization is implemented, and the six DoF wrench command is designed as follows:

$$\mathbf{u}^d = \begin{bmatrix} \mathbf{J}_\xi \\ \mathbf{J}_\nu \end{bmatrix} \mathbf{T} = \begin{bmatrix} m_B^W \mathbf{R}^\top & \mathbf{0} \\ \mathbf{0} & B \mathbf{I} \end{bmatrix} \left(\begin{bmatrix} \mathbf{u}_\xi \\ \mathbf{u}_\nu \end{bmatrix} - \begin{bmatrix} W \mathbf{G} \\ \mathbf{0} \end{bmatrix} \right), \quad (6)$$

where the superscript d indicates the desired values, \mathbf{u}_ξ and \mathbf{u}_ν are virtual inputs for position control and attitude control, respectively. Combining Eq. (6) with Eq. (1), the platform dynamics is equivalent to a double integrator and can be written in a state-space form as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\tilde{\mathbf{u}}, \quad (7)$$

where

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0 & 0 & \mathbf{I}_3 & 0 \\ 0 & 0 & 0 & \mathbf{I}_3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \mathbf{I}_3 & 0 \\ 0 & \mathbf{I}_3 \end{bmatrix}, \\ \mathbf{x} &= [\xi^\top \quad \eta^\top \quad \dot{\xi}^\top \quad \dot{\eta}^\top]^\top, \quad \tilde{\mathbf{u}} = [\mathbf{u}_\xi^\top \quad \mathbf{u}_\nu^\top]^\top. \end{aligned} \quad (8)$$

We design a LQI control scheme [34, 35] to close the control loop with the augmented system states as

$$\tilde{\mathbf{x}} = [e_\xi^\top \quad e_\eta^\top \quad \dot{e}_\xi^\top \quad \dot{e}_\eta^\top \quad \int e_\xi dt^\top \quad \int e_\eta dt^\top]^\top, \quad (9)$$

with

$$\begin{aligned} e_\xi &= \xi^r - \xi, \\ \dot{e}_\xi &= \dot{\xi}^r - \dot{\xi}, \\ e_\eta &= \frac{1}{2} [\mathbf{R}(\eta)^T \mathbf{R}(\eta^r) - \mathbf{R}(\eta^r)^T \mathbf{R}(\eta)]_\vee, \\ \dot{e}_\eta &= \mathbf{R}(\eta)^T \mathbf{R}(\eta^r) \nu^r - \nu, \end{aligned} \quad (10)$$

where the superscript r indicates the reference value, $\mathbf{R}(\cdot)$ is the transformation from Euler angles to a standard rotation matrix, and $[\cdot]_\vee$ is the mapping from SO(3) to \mathbb{R}^3 .

The cost function is

$$\tilde{\mathbf{J}}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) = \int_0^\infty (\tilde{\mathbf{x}}^\top \tilde{\mathbf{Q}} \tilde{\mathbf{x}} + \tilde{\mathbf{u}}^\top \tilde{\mathbf{R}} \tilde{\mathbf{u}}) dt, \quad (11)$$

where $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{R}}$ are designed matrices that determine the closed-loop dynamics, and the optimal input $\tilde{\mathbf{u}}$ is given by

$$\tilde{\mathbf{u}} = -\mathbf{K} \tilde{\mathbf{x}}, \quad (12)$$

where \mathbf{K} is the solution to the algebraic Riccati equation of the augmented system.

B. Nominal Allocation Framework

1) *Force Decomposition-based Allocation*: Given \mathbf{u}^d , we aim to determine the desired tilting angle $\boldsymbol{\alpha}$ and thrust \mathbf{T} for the four quadcopters through the nonlinear mapping Eq. (6)—this process is known as the “allocation problem”. One heuristic solution employs FD to transform this nonlinear

mapping problem into a linear one by defining intermediate variables [2, 36]:

$$\mathbf{F} = \begin{bmatrix} F_{s0} & F_{c0} & \dots & F_{s3} & F_{c3} \end{bmatrix}^T, \quad (13)$$

where

$$F_{si} = \sin \alpha_i T_i, \quad F_{ci} = \cos \alpha_i T_i. \quad (14)$$

With these new variables, Eq. (6) can be rewritten as

$$\mathbf{u}^d = \begin{bmatrix} \mathbf{J}_\xi \\ \mathbf{J}_\nu \end{bmatrix} \mathbf{T} = \mathbf{W} \mathbf{F}, \quad (15)$$

where $\mathbf{W} \in \mathbb{R}^{6 \times 8}$ is a constant allocation matrix with full row rank. The general solution of \mathbf{F} is expressed as

$$\mathbf{F} = \mathbf{W}^\dagger \mathbf{u}^d + \mathbf{N}_W \mathbf{Z}, \quad (16)$$

where $\mathbf{N}_W \in \mathbb{R}^{8 \times 2}$ is the nullspace of \mathbf{W} and $\mathbf{Z} \in \mathbb{R}^{2 \times 1}$ is an arbitrary vector. The least-squares solution can be acquired to minimize $\|F\|^2 = \|T\|^2$ by setting $\mathbf{Z} = 0$. The real inputs T_i and α_i for the low-level controller can then be recovered as

$$T_i = \sqrt{F_{si}^2 + F_{ci}^2}, \quad \alpha_i = \text{atan2}(F_{si}, F_{ci}). \quad (17)$$

2) *Thrust Force Saturation Issue*: However, this FD-based allocation framework (referred to as the nominal allocation framework) does not account for input constraints. Specifically, it could generate a desired thrust exceeding motor saturation, leading to platform instability. This issue, known as thrust force saturation, was investigated previously in [7, 9], where the nominal FD-based allocation framework proved inadequate for utilizing the full thrust capability of the platform, and an analytical solution was provided for a one-dimensional rotation scenario by formulating it as a min-max optimization problem. In this study, we generalize this problem for a standard trajectory-tracking scenario. At each timestep, the nullspace-based allocation framework (which will be introduced next) determines the optimal tilting angle and thrust for each quadcopter, subject to a predefined cost function and input constraints (see Sec. VII-B).

C. Nullspace-based Allocation Framework

In our previous work, we proposed a nullspace-based allocation framework [10] which has the advantages of both the FD-based and QP-based allocation frameworks while avoiding their known issues. In this framework, a QP problem is first formulated at each time step as:

$$\min_{\Delta \mathbf{X}, \mathbf{s}} \mathbf{J} = \Delta \mathbf{X}^T \mathbf{P} \Delta \mathbf{X} + \mathbf{s}^T \mathbf{Q} \mathbf{s} \quad (18)$$

$$\text{s.t. } \mathbf{W} \left(\mathbf{s} + \mathbf{F}(\boldsymbol{\alpha}_o, \mathbf{T}_o) + \frac{\partial \mathbf{F}}{\partial \mathbf{X}} \Big|_{\mathbf{X}=\mathbf{X}_o} \Delta \mathbf{X} \right) = \mathbf{u}^d \quad (19)$$

$$\mathbf{X}_{\min} - \mathbf{X}_o \leq \Delta \mathbf{X} \leq \mathbf{X}_{\max} - \mathbf{X}_o \quad (20)$$

$$\Delta \mathbf{X}_{\min} \leq \Delta \mathbf{X} \leq \Delta \mathbf{X}_{\max} \quad (21)$$

where Eq. (18) is the object function, with \mathbf{X} defined as

$$\mathbf{X} = [\boldsymbol{\alpha}^T \quad \mathbf{T}^T]^T, \quad (22)$$

\mathbf{P} and \mathbf{Q} are weighting matrices. Eq. (19) uses first-order linearization to approximate the nonlinear equality constraint

Eq. (16), where $[\cdot]_o$ is the value of a variable at previous timestep, $\Delta[\cdot]$ is the difference with respect to the previous timestep of a variable, and \mathbf{s} serves as a slack variable. Eqs. (20) and (21) are two inequality constraints to limit the value of a variable or its rate of change.

The desired inputs for the current step can be approximated as,

$$\mathbf{X} = \mathbf{X}_o + \Delta \mathbf{X}. \quad (23)$$

Then, we can eliminate the approximation errors with the nullspace projection method,

$$\mathbf{F}^* = (\mathbf{I}_{3n} - \mathbf{N}_W \mathbf{N}_W^\dagger) \mathbf{W}^\dagger \mathbf{u}^d + \mathbf{N}_W \mathbf{N}_W^\dagger \mathbf{F}(\mathbf{X}). \quad (24)$$

Finally, $\boldsymbol{\alpha}^*$ and \mathbf{T}^* can be determined from \mathbf{F}^* using Eq. (17). The nullspace-based allocation framework takes into account input constraints while still providing an exact solution for Eq. (15). This makes it more broadly widely than existing methods. For a detailed explanation of the implementation, the reader can refer to [10]. The FTC utilizing this constrained allocation framework will be presented in Sec. V.

D. Low-level Control

The thrust T_i can be directly controllable in the low-level controller for each Q_i , whereas the tilting angle α_i is controlled by M_i^y through the motor's second-order rotational dynamics Eq. (5). Therefore, a double-loop PID controller is applied to track the tilting angle [1, 8]:

$$M_i^y = k_{D\alpha} \dot{e}_\alpha + k_{P\alpha} e_\alpha + k_{I\alpha} \int e_\alpha dt, \quad (25)$$

where $k_{[\cdot]\alpha}$ are constant gains and the error term is defined as $e_\alpha = \alpha_i^d - \alpha_i$.

Considering these relationships and neglecting the fast motor dynamics that drive the propeller speeds, the propeller thrusts can be calculated from Eq. (4):

$$\begin{bmatrix} t_{i0} \\ t_{i1} \\ t_{i2} \\ t_{i3} \end{bmatrix} = \text{sat} \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ -b & -b & b & b \\ b & -b & -b & b \\ c_\tau & -c_\tau & c_\tau & -c_\tau \end{bmatrix}^{-1} \begin{bmatrix} T_i \\ M_i^x \\ M_i^y \\ M_i^z \end{bmatrix} \right). \quad (26)$$

where $\text{sat}(\cdot)$ is the saturation function. In the event of a failed propeller, the speed and thrust become nearly zero, subsequently altering the quadcopter's maximum thrust and moments.

V. FAULT-TOLERANT CONTROLLER

This section examines the FTC control of the platform when some propellers are completely failed. Specifically, the focus lies on scenarios when one or two propellers on a single quadcopter have failed, while the other three quadcopters continue to function properly. Here, as an example, Q_3 is assumed as the Bad QC. Of note, UAV fault detection has been extensively studied with various well-developed methods available in the literature [4, 18, 37]. Therefore, we assume that the propeller failure combinations have been accurately detected in this paper.

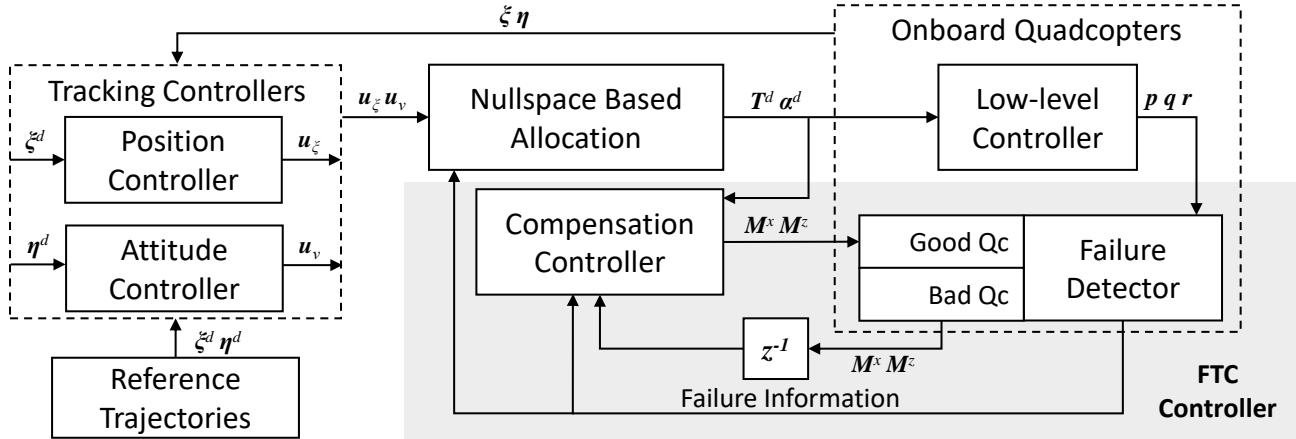


Fig. 3: Fault-Tolerant Controller Architecture. Each quadcopter is equipped with onboard failure detection and motor control modules. When propeller failure is detected, the failure detection module communicates with the onboard controller to modify the low-level control strategy according to identified failure combinations. Additionally, it transmits failure information to the high-level controller, which leverages the nullspace-based control allocation to adjust thrust distribution among the four individual quadcopters.

A. Fault-tolerant Controller Architecture

As shown in Fig. 3, the FTC controller comprises three primary components: (i) The low-level control on each quadcopter encompasses two onboard functions—one for propeller failure detection (assumed to be sufficiently fast), and another for quadcopter attitude and thrust control. (ii) The compensation loop serves as an addition to the high-level position-attitude controller and employs auxiliary inputs to improve trajectory tracking performance and attenuate disturbances with a faster response within the saturation constraints. (iii) The nullspace-based control allocation framework is engaged to incorporate input constraints and adjust thrust force distribution among the four individual quadcopters.

When a propeller failure is detected, the low-level control module adjusts its strategy according to the failure combination to maintain control over thrust forces and tilting angle. Simultaneously, the high-level control allocation modifies thrust force distribution among the four individual quadcopters, considering their different thrust generation capabilities to prevent saturation. The compensation loop utilizes the auxiliary torque inputs of Good QCs to compensate for the disturbance torques generated by the low-level control of the Bad QC.

B. Propeller Failure Handling

1) One Propeller is Failed:

a) Low-level Control: In this case, three propellers on Q_3 are assumed functional, while propeller P_0 is considered to have failed. As a result, the four outputs in Eq. (4) cannot be independently controlled [18]. To account for this, Eq. (26) is adjusted to calculate the low-level commands as follows, where the control of M_3^z is lost (its magnitude is relatively small),

$$\begin{bmatrix} t_{31} \\ t_{32} \\ t_{33} \end{bmatrix} = \text{sat} \left(\begin{bmatrix} \frac{1}{2} & -\frac{1}{2b} & 0 \\ 0 & \frac{1}{2b} & -\frac{1}{2b} \\ \frac{1}{2} & 0 & \frac{1}{2b} \end{bmatrix} \begin{bmatrix} T_3 \\ M_3^x \\ M_3^y \end{bmatrix} \right). \quad (27)$$

Given that the mapping matrix in Eq. (27) is in full-rank, T_3 , M_3^x , and M_3^y can be controlled independently using the three remaining thrusts, without considering the saturation of t_{3j} . However, in practice, the occurrence of $t_{32} \leq 0$ as a result of $t_{32} = \frac{1}{2b}(M_3^x - M_3^y)$ is evidently unreasonable. Therefore, the control of the tilting angle with M_3^y must take precedence over the torque M_3^x .

Based on the aforementioned analysis, Eq. (27) is not employed in the low-level control. Instead, it is reformulated as follows:

$$\begin{bmatrix} t_{31} \\ t_{32} \\ t_{33} \end{bmatrix} = \text{sat} \left(\begin{bmatrix} \frac{1}{4} & -\frac{1}{4b} \\ \frac{1}{4} & -\frac{1}{4b} \\ \frac{1}{2} & \frac{1}{2b} \end{bmatrix} \begin{bmatrix} T_3 \\ M_3^y \\ M_3^x \end{bmatrix} \right). \quad (28)$$

This implies the loss of control over both M_3^x and M_3^z . Although this strategy introduces a torque disturbance to the central frame ($M_3^x \neq 0$ and $M_3^z \neq 0$), it ensures control of α_3 . The magnitude of the torque disturbances, which are proportional to T_3 , can be reduced by adjusting thrust distribution across all quadcopters using the nullspace-allocation framework [38]. In addition, this torque disturbance will be compensated by the add-on compensation loop involving the other quadcopters, which will be introduced later. The comparison between using Eq. (27) and Eq. (28) in low-level control will be shown in Sec. VI.

b) High-level Control: From Eq. (28), it can be shown that T_3 will not be evenly distributed among the three remaining propellers, with P_3 contributing half of the total required thrust. Therefore, the maximum value of T_3 must be changed from $4t_{\max}$ to $2t_{\max}$, where t_{\max} denotes the maximum thrust of a single propeller. In essence, half of the thrust-generation capability is lost despite only one propeller failing on a quadcopter. Therefore, the maximum thrust vector T_{\max} in Eq. (20), must be changed from $t_{\max} \cdot [4, 4, 4, 4]^T$ to $t_{\max} \cdot [4, 4, 4, 2]^T$. Note that the existing allocation strategies [2, 12–14] cannot accommodate input constraints, necessitating the use of the

nullspace-based constrained allocation framework.

c) Fast Compensation Loop: We utilize the auxiliary inputs M_i^x and M_i^z from the three Good QCs to formulate a compensation loop, mitigating disturbances induced by the low-level control of \mathcal{Q}_3 [11]. The platform's complete rotational dynamics derived in [9], and neglecting ${}^B\boldsymbol{\nu} \times ({}^B\boldsymbol{I} {}^B\boldsymbol{\nu})$, can be expressed as:

$${}^B\dot{\boldsymbol{\nu}} = {}^B\boldsymbol{I}^{-1} (\mathbf{J}_v \boldsymbol{T} + \mathbf{J}_M^x \mathbf{M}^x + \mathbf{J}_M^z \mathbf{M}^z), \quad (29)$$

where

$$\begin{aligned} \mathbf{J}_M^x &= \begin{bmatrix} -\cos \alpha_0 & 0 & \cos \alpha_2 & 0 \\ 0 & \cos \alpha_1 & 0 & -\cos \alpha_3 \\ \sin \alpha_0 & \sin \alpha_1 & \sin \alpha_2 & \sin \alpha_3 \end{bmatrix}, \\ \mathbf{M}^x &= [M_0^x \ M_1^x \ M_2^x \ M_3^x]^\top, \\ \mathbf{J}_M^z &= \begin{bmatrix} \sin \alpha_0 & 0 & -\sin \alpha_2 & 0 \\ 0 & -\sin \alpha_1 & 0 & \sin \alpha_3 \\ \cos \alpha_0 & \cos \alpha_1 & \cos \alpha_2 & \cos \alpha_3 \end{bmatrix}, \\ \mathbf{M}^z &= [M_0^z \ M_1^z \ M_2^z \ M_3^z]^\top. \end{aligned} \quad (30)$$

When a propeller fails, a QP problem is formulated to solve for the optimal auxiliary inputs of \mathcal{Q}_{0-2} for disturbance compensation. The equality constraint is designed as

$$\mathbf{J}_M^x \mathbf{M}^x + \mathbf{J}_M^z \mathbf{M}^z + \mathbf{k} = 0, \quad (31)$$

where \mathbf{k} is a slack variable. The object function is

$$\mathbf{J}(\mathbf{y}, \mathbf{k}) = \mathbf{y}^\top \mathbf{A} \mathbf{y} + \mathbf{k}^\top \mathbf{B} \mathbf{k}, \quad (32)$$

where \mathbf{y} is defined as

$$\mathbf{y} = [M_0^x \ M_1^x \ M_2^x \ M_0^z \ M_1^z \ M_2^z]^\top, \quad (33)$$

and \mathbf{A} and \mathbf{B} are constant, positive semi-definite gain matrices. Saturation is included as the following inequality constraints:

$$0 \leq \begin{bmatrix} 1 & 1 & 1 & 1 \\ -b & -b & b & b \\ b & -b & -b & b \\ c_\tau & -c_\tau & c_\tau & -c_\tau \end{bmatrix}^{-1} \begin{bmatrix} T_i \\ M_i^x \\ M_i^y \\ M_i^z \end{bmatrix} \leq t_{\max}, \forall i = 0, 1, 2. \quad (34)$$

Note that for this QP problem, M_3^x and M_3^z from \mathcal{Q}_3 are used as feedback along with T_i and M_i^y from \mathcal{Q}_{0-2} .

Compared to other disturbance attenuation approaches for over-actuated UAV platforms [39, 40], which primarily compensate for unmodeled dynamics by adjusting the virtual wrench command via a second-order PID loop, this auxiliary-input-based compensation loop can be regarded as feed-through dynamics. Consequently, it enables a faster response to torque disturbances introduced by the low-level adjustment of Bad QC.

2) Two Propellers are Failed:

a) Low-level Control: In this case, it is assumed that two propellers on \mathcal{Q}_3 have failed. In order to maintain control over T_3 and M_3^y , specific requirements must be imposed on the propeller failure combinations to ensure M_3^y remains controllable. This framework is unable to address two situations that may occur when propeller failure affects both \mathcal{P}_0 and \mathcal{P}_3 or both \mathcal{P}_1 and \mathcal{P}_2 (Tab. I).

As an example, we consider the case where both \mathcal{P}_0 and

TABLE I: Controllability of M_3^y under Different Propeller Failure Combinations (M_3^y uncontrollable cases lead to quadcopter-level failure.)

Group	Failure Combination	M_3^y Controllable?
One		✓
Two	0, 1	✓
	0, 2	✓
	0, 3	✗
	1, 2	✗
	1, 3	✓
	2, 3	✓
Three or Four		✗

\mathcal{P}_1 are failed. The low-level controller is designed as

$$\begin{bmatrix} t_{32} \\ t_{33} \end{bmatrix} = \text{sat} \left(\begin{bmatrix} 1 & 1 \\ -b & b \end{bmatrix}^{-1} \begin{bmatrix} T_3 \\ M_3^y \end{bmatrix} \right). \quad (35)$$

In this case, M_3^x and M_3^z will be transferred to the central frame as disturbance torques with larger magnitudes than those in case of a single propeller failure. As in Sec. V-B1, the high-level maximum thrust constraint is modified, and the compensation loop is implemented for disturbance attenuation.

Obviously, in the event of three or four propeller failures, the quadcopter effectively loses all control. Nonetheless, in cases of quadcopter-level failure (marked with ✗ in Tab. I), the platform remains controllable if the other Good QCs retain the controllability for the DoFs to be controlled, which must include at least the gravity direction [18]. For the four-quadcopter configuration investigated in this paper, the quadcopter opposing the failed one would also need to be disabled, and platform control would rely on the remaining pair of quadcopters. Specifically, the high-level control's thrust saturation limit for each quadcopter must be adjusted accordingly, with the thrust limits of the lost pair of quadcopters set to zero. Finally, when two non-opposing quadcopters lost control due to the aforementioned propeller failures cases, the platform would fail. Thus, all possible propeller failure scenarios have been addressed.

VI. SIMULATION

A. Simulation Setup

To compare the two low-level control adjustment strategies (Sec. V-B1), a dynamic simulation was constructed using Matlab/Simulink. The Simscape Multibody module was employed to simulate the platform's complete dynamics. All known hardware characteristics were included in the simulated model, such as sampling frequencies, measurement noise, communication delay, motor dynamics, and more, with the full list provided in Tab. II. Here, m_0 and I_0 refer to the mass and inertia matrix of the mainframe, while m_i and I_i refer to the mass and inertia matrix of each quadcopter with the passive hinge.

B. Simulation Results

Two low-level adjustment strategies (Eq. (27) and Eq. (28)) were compared in this simulation. The nominal controller with FD-based allocation was utilized at the high-level to track the

TABLE II: Physical and Software Properties in Simulation

Parameter	Value
m_0	0.036 kg
m_i	0.027 kg
I_0	$\text{diag}([3 \ 3 \ 4.5]) / \text{kg}\cdot\text{cm}^2$
I_i	$\text{diag}([0.16 \ 0.16 \ 0.29]) / \text{kg}\cdot\text{cm}^2$
l	0.14 m
t_{\max}	0.167 N
communication delay	0.02 s

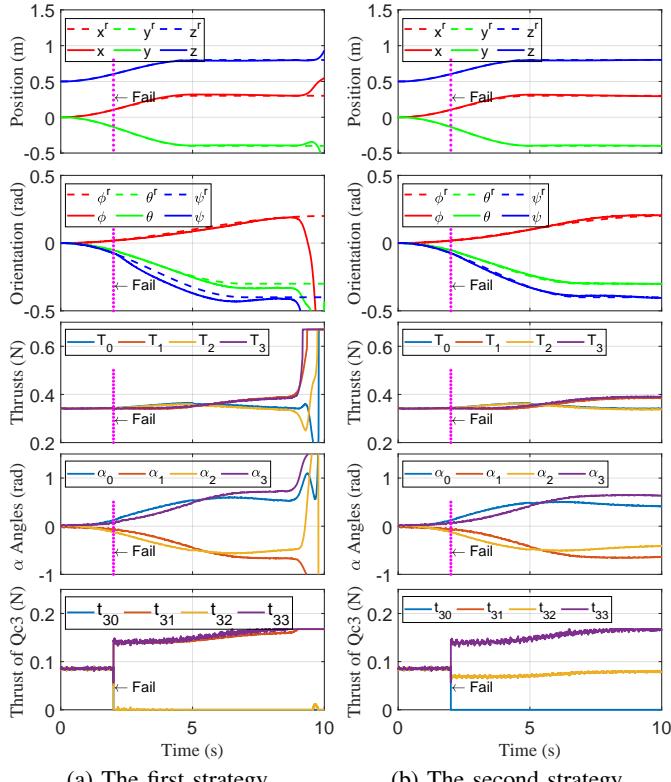


Fig. 4: Simulation: Trajectory tracking performance of two low-level adjustment strategies under propeller failure. (a) corresponds to Eq. (27) and (b) corresponds to Eq. (28)

reference position and attitude trajectory. In both tests, \mathcal{P}_0 on \mathcal{Q}_3 began to fail at 2s (Fig. 4).

As illustrated in Fig. 4a, with the first strategy (Eq. (27)), the desired thrust t_{32} could become a non-positive value, which was subsequently set to zero due to propeller limitation. This deteriorated the regulation of the tilting angle. As a result, the high-level thrust finally saturated at approximately 9s, leading to instability in the position and attitude control of the whole platform. In contrast, with the second strategy (Eq. (28)), both position and attitude control remained stable throughout the trajectory, as shown in Fig. 4b. In the low-level control of \mathcal{Q}_3 , all propeller thrusts stayed within the saturation range, thus ensuring proper tilting angle control.

VII. EXPERIMENTS

A. Experiment Setup

The Crazyflie 2.1 quadcopter served as the basis for each individual unit within the platform. To achieve greater

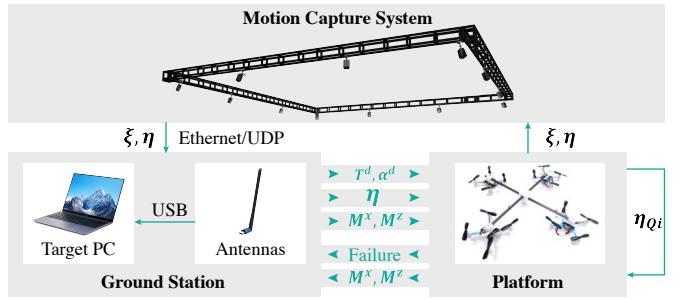


Fig. 5: Experimental setup. A host PC runs the high-level controller at 100 Hz with platform position and attitude feedback from the OptiTrack motion-capture system. The onboard low-level controller runs at 500 Hz on each quadcopter.

thrust force, the battery and motors on each quadcopter were upgraded, resulting in a maximum thrust force of $0.67 \text{ N}(4 \times t_{\max})$ and a mass (including the passive hinge) of 27 g . The entire platform has a total mass of 144 g , with overall dimensions of $36 \times 36 \times 6 \text{ cm}$. A light-weighted tether was attached from the ceiling of the indoor environment to the center of the platform to protect the hardware in the case of failure. This tether remained loose and exerted negligible force and torque on the platform throughout all the experiments.

An OptiTrack motion-capture system served as the external sensor for measuring the platform's position and attitude. The main controller operated on a ground-based computer and communicated through Ethernet with the motion-capture system to calculate T_i , α_i , M_i^x , M_i^z for each quadcopter. The radio-communication antenna transmitted these values, along with the attitude of the central frame, to the individual quadcopter. Every quadcopter was equipped with its own microprocessor, IMU, and modules for failure detection and onboard control. The system presumed the propeller failure combinations to be identifiable by the efficient failure detection module; it utilized the control module to adjust the low-level control strategy based on the propeller-failure combination and to regulate T_i and α_i . The experimental setup is shown in Fig. 5.

B. Thrust Force Saturation

In this experiment (Fig. 6), both FD-based and nullspace-based allocation frameworks were implemented on the UAV platform to track a six DoF reference trajectory and remain at the final attitude for 2s. As shown in Fig. 6a, the FD-based allocation framework could not address the thrust force saturation issue. Specifically, the desired thrust commands T_2 and T_3 continued to increase and saturated at approximately 10.7s, leading to an unstable system. Meanwhile, T_0 and T_1 remained below 0.4 N, implying that the platform could generate sufficient thrust to remain airborne. For the nullspace-based allocation framework, the platform successfully reached the desired attitude and maintained stability (Fig. 6b). As analyzed in Sec. IV-B2, this experiment further demonstrates that the FD-based framework does not fully exploit the platform's capabilities, while the nullspace-based framework enhances performance by incorporating the constraints.

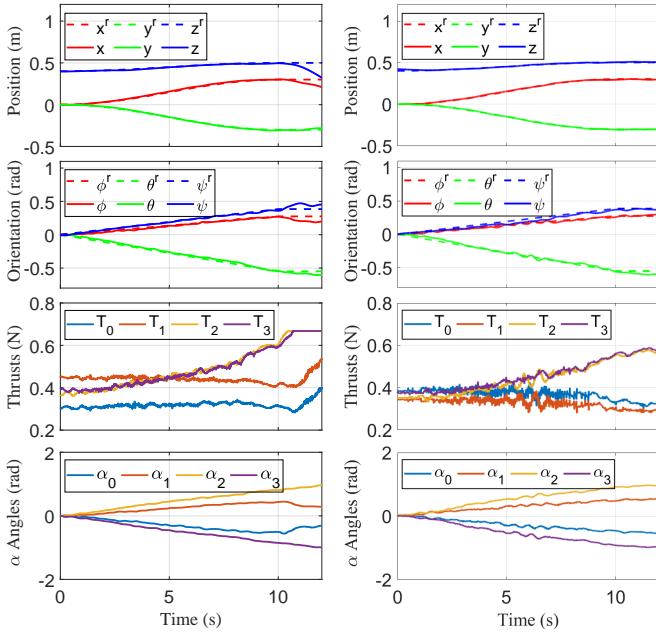


Fig. 6: Experiment: Using different allocation frameworks to handle thrust force saturation.

C. Propeller Failure

Three cases were designed to verify the effectiveness of the proposed FTC controller in scenarios where one or two propellers failed during trajectory tracking. For each case, the nominal allocation strategy with low-level adjustment (denoted as N+L controller) is compared with the FTC controller. Propeller failure was simulated by setting the speed of the corresponding propellers to zero.

1) One Failed Propeller:

a) Unsaturated Trajectory: In this experiment, an unsaturated trajectory was employed, ensuring that $T_3 \leq 2t_{\max}$ at each timestep. As analyzed in Sec. V-B1, the N+L controller maintained the stability of the platform when the trajectory did not trigger low-level saturation of \mathcal{Q}_3 . In this situation, the main difference between the N+L controller and the FTC controller was the fast compensation loop for disturbance rejection. The \mathcal{P}_0 on \mathcal{Q}_3 began to fail at 1s. The tracking performance of the two controllers is plotted in Fig. 7.

As shown in Fig. 7, both controllers maintained the stability of the UAV platform. However, for the N+L controller (Fig. 7a), disturbance torques M_3^x and M_3^z introduced by low-level adjustment could only be compensated by the integral action in the trajectory tracking controller, leading to the oscillation of the platform. In contrast, the FTC controller (Fig. 7b) incorporated a compensation loop to attenuate disturbance, resulting in improved tracking performance and reduced oscillation for the entire platform.

b) Saturated Trajectory: In this test, a more challenging reference trajectory that requires $T_3 > 2t_{\max}$ at some timesteps (see Fig. 8a) was designed, activating the low-level saturation constraint of \mathcal{Q}_3 . Likewise, \mathcal{P}_0 of \mathcal{Q}_3 was set to fail at 1s. The tracking performance of the two controllers is plotted in Fig. 8.

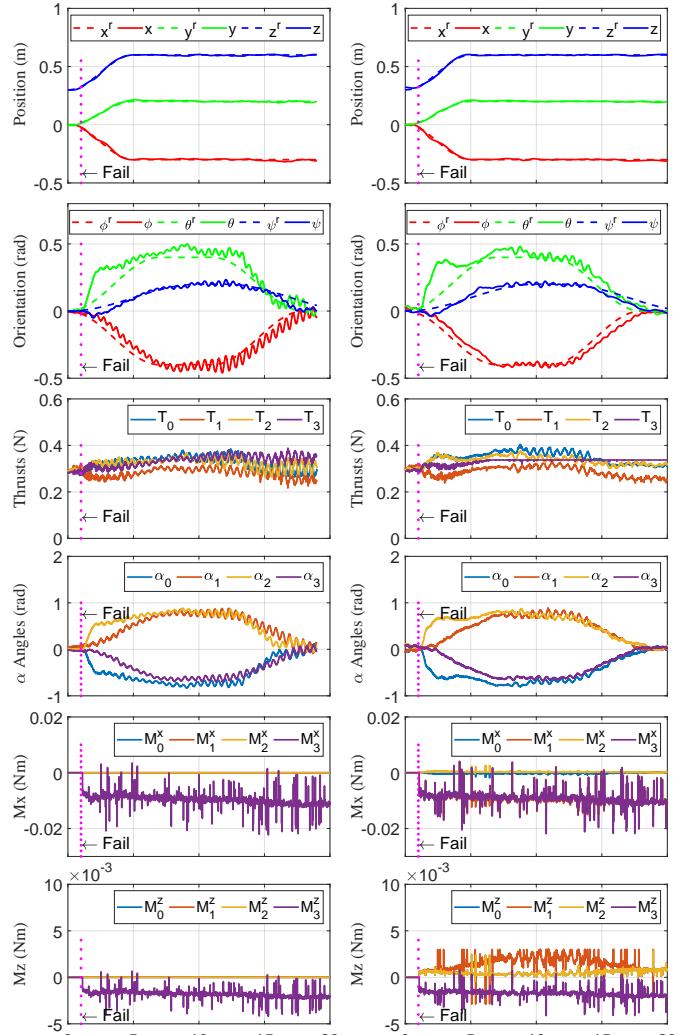
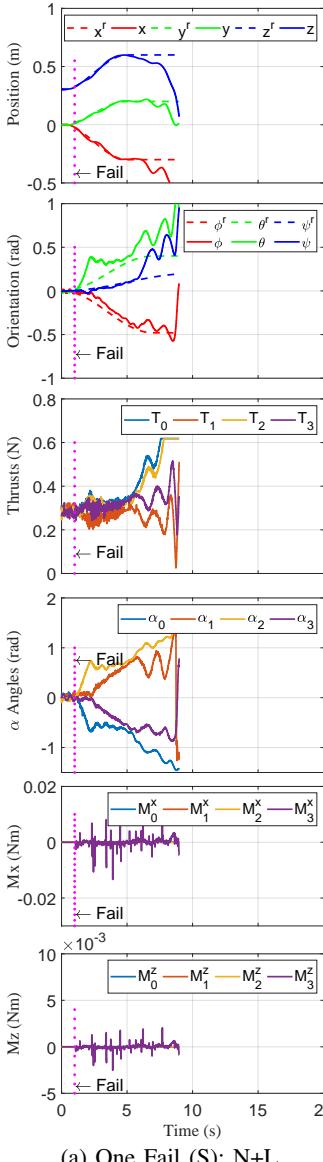


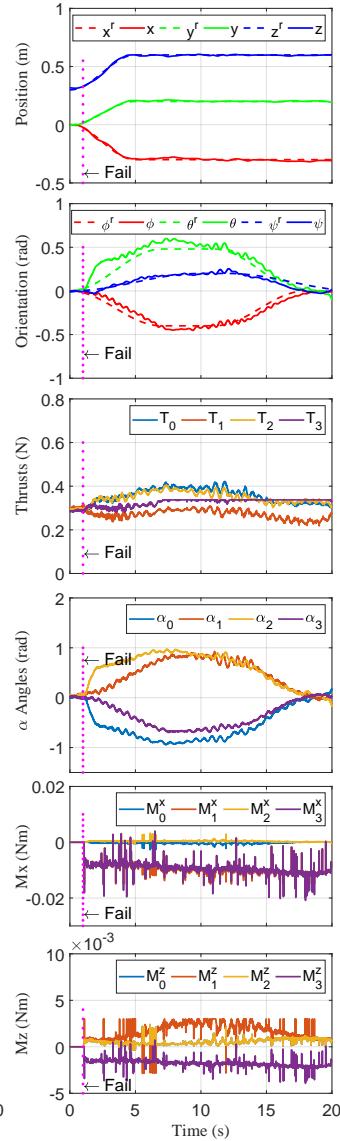
Fig. 7: Case 1: Trajectory tracking performance when one propeller is failed. (U stands for the unsaturated trajectory that satisfies $T_3 \leq 2t_{\max}$ all the time, N+L stands for the nominal controller + low-level adjustment, and FTC stands for the FTC framework we proposed. The same notations are applied for the rest of this paper.)

As shown in Fig. 8a, when the low-level saturation constraint was triggered, the N+L controller caused the UAV platform to become unstable at approximately 7s. Two explanations are given here. First, the nominal allocation strategy did not consider the thrust-saturation constraint, leading to output desired thrust T_3 beyond the capability of \mathcal{Q}_3 . Second, the disturbance torques M_3^x and M_3^z generated by the low-level control were transferred to the central frame without compensation.

For the FTC controller (Fig. 8b), the high-level control facilitated thrust distribution adjustment through the nullspace-based allocation strategy, enabling different saturation values for different quadcopters. In addition, the disturbance torques M_3^x and M_3^z can be compensated with the auxiliary inputs. Consequently, position and attitude control remained stable along the entire trajectory.



(a) One Fail (S): N+L

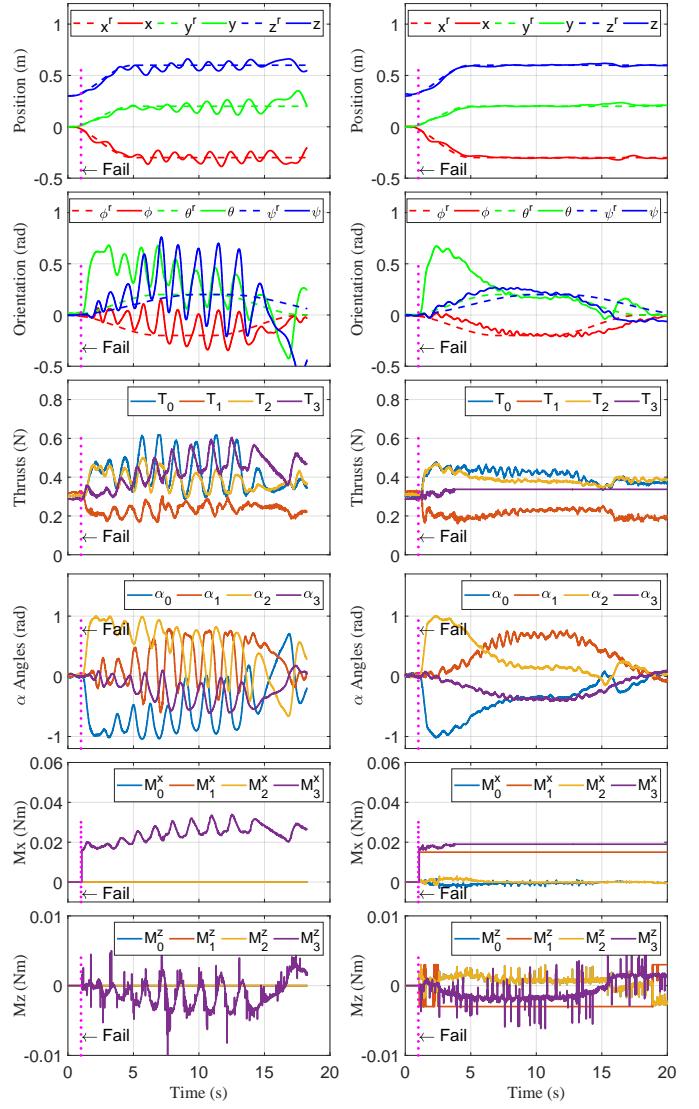


(b) One Fail (S): FTC

Fig. 8: Case 2: Trajectory tracking performance when one propeller is failed. (S stands for the saturated trajectory that $T_3 > 2 t_{\max}$ at some time steps. Reference attitude trajectory with $\phi_{\max} = 0.5 \text{ rad}$, $\theta_{\max} = 0.2 \text{ rad}$, $\psi_{\max} = 0.4 \text{ rad}$.

2) Two Failed Propellers: In this experiment, \mathcal{P}_0 and \mathcal{P}_1 of \mathcal{Q}_3 began to fail at 1s while tracking the reference trajectory. The tracking performance of the N+L controller and FTC controller are both plotted in Fig. 9. In this scenario, the disturbance torques M_3^x and M_3^z introduced by the low-level controller were larger than in Case 2, making platform stabilization more challenging despite the reference trajectory possessing a relatively smaller tilting attitude compared to Case 2.

As shown in Fig. 9a, the entire platform became unstable with the N+L controller. For the FTC controller, the platform successfully tracked the reference position and attitude reference trajectories (Fig. 9b). Due to the larger interacting disturbance torques produced by low-level adjustment in this case, the overall performance for both controllers was worse



(a) Two Fails: N+L

Fig. 9: Case 3: Trajectory tracking performance when two propellers failed. (Reference attitude trajectory with $\phi_{\max} = 0.2 \text{ rad}$, $\theta_{\max} = 0.2 \text{ rad}$, $\psi_{\max} = 0.2 \text{ rad}$.)

than that observed in Case 2. Some video clips of this experiment are shown in Fig. 1.

VIII. DISCUSSION

This paper proposed an FTC to handle certain propeller failure situations for over-actuated UAV platforms built on quadcopters and passive hinges, which is an increasingly popular category of over-actuated UAV configurations in the recent years [5] for its simplicity in design and prototyping, in addition to the natural attenuation to internal disturbance from propeller drag, momentum and reaction torque, as introduced in [6–9, 38].

One notable feature of the presented configuration is the existence of high-bandwidth auxiliary inputs M_i^x and M_i^z in the low-level control of \mathcal{Q}_i (Eq. (4)), as analyzed in [9]. These inputs can either be used to improve the tracking performance as in [11], compensate for the air dynamics as

in [41], or improve system robustness under certain failure, as demonstrated in this paper. Particularly, in the event of propeller failure, the thrust generator in this type of configuration partially reserves its actuation functionality due to the compensation from auxiliary inputs, so that crashes can be prevented. However, we also witnessed that when the low-level pose regulation controller on the passive joint fails, as shown in Tab. I, the uncontrolled relative motion between the quadcopter and mainframe will introduce a huge disturbance to the platform, which shall be further investigated by future research.

The characteristics of the presented configuration also make it befitting as the foundation module of reconfigurable aerial platforms, so that the applications can be extended in multiple directions including flight formation [42] or aerial manipulation [43]. The popularity of constructing flying structures with quadcopters reiterates the significance of our platform and the associated control challenges.

IX. CONCLUSION

In this paper, we addressed the fault-tolerant control (FTC) method for over-actuated UAV platforms. Our approach adopts the widely-used hierarchical control architecture and the nullspace-based constrained control allocation we have recently developed for omnidirectional flight.

The logic of FTC within the hierarchical control is presented in the following manner: The low-level control maintains the quadcopter's control of the orientation and thrust whenever possible. The high-level control sets up the maximum thrust available to each quadcopter unit and solves desired commands using the nullspace-based constrained allocation framework. Uncontrolled disturbances generated by the Bad QC are compensated by the Good QCs whenever within the saturation limit. Both simulation and experimental results, in the cases of one or two propeller failures, have demonstrated FTC's stability and improved trajectory-tracking performance compared to the nominal control method.

Our FTC analysis and methods encompass all possible combinations of propeller failures and can be applied to other platform configurations with similar low-level propeller actuators and maximum thrust limit setups in the high-level control. For aerial platforms with fixed rotor angles, the high-level control method can be readily applied with the maximum thrust of failed propellers set to zero. Finally, in situations where the platform is inevitably failing, graceful crashing is desirable and warrants future investigation.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Hangxin Liu, Dr. Zeyu Zhang, Dr. Wenzhong Yan, and Dr. Ankur Mehta at UCLA for the figure polishing and the technical assistance with the motion-capture system.

REFERENCES

- [1] L. Qian, S. Graham, and H. H.-T. Liu, "Guidance and control law design for a slung payload in autonomous landing: A drone delivery case study," *IEEE/ASME Transactions on Mechatronics (TMECH)*, vol. 25, no. 4, pp. 1773–1782, 2020.
- [2] M. Kamel, S. Verling, O. Elkhatib, C. Sprecher, P. Wulkop, Z. Taylor, R. Siegwart, and I. Gilitschenski, "The voliro omniorientational hexacopter: An agile and maneuverable tilttable-rotor aerial vehicle," *IEEE Robotics and Automation Magazine (RA-M)*, vol. 25, no. 4, pp. 34–44, 2018.
- [3] S. Rajappa, M. Ryll, H. H. Bülthoff, and A. Franchi, "Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2015.
- [4] M. Saied, B. Lussier, I. Fantoni, C. Francis, H. Shraim, and G. Sanahuja, "Fault diagnosis and fault-tolerant control strategy for rotor failure in an octocopter," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2015.
- [5] H.-N. Nguyen, S. Park, J. Park, and D. Lee, "A novel robotic platform for aerial manipulation using quadrotors as rotating thrust generators," *IEEE Transactions on Robotics (T-RO)*, vol. 34, no. 2, pp. 353–369, 2018.
- [6] 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," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 4, pp. 6828–6835, 2021.
- [7] L. Ruan, *Independent position and attitude control on multirotor aerial platforms*. PhD thesis, University of California, Los Angeles, 2020.
- [8] 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.
- [9] L. Ruan, C.-H. Pi, Y. Su, P. Yu, S. Cheng, and T.-C. Tsao, "Control and experiments of a novel tilttable-rotor aerial platform comprising quadcopters and passive hinges," *Mechatronics*, vol. 89, p. 102927, 2023.
- [10] Y. Su, P. Yu, M. Gerber, L. Ruan, and T.-C. Tsao, "Nullspace-based control allocation of overactuated uav platforms," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 4, pp. 8094–8101, 2021.
- [11] 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," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, pp. 1214–1221, 2021.
- [12] 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.
- [13] M. Zhao, K. Okada, and M. Inaba, "Enhanced modeling and control for multilinked aerial robot with two dof force vectoring apparatus," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 1, pp. 135–142, 2020.
- [14] M. Santos, L. Honório, A. Moreira, M. Silva, and V. Vidal, "Fast real-time control allocation applied to over-actuated quadrotor tilt-rotor," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 3, pp. 1–20, 2021.
- [15] T. A. Johansen and T. I. Fossen, "Control allocation—a survey," *Automatica*, vol. 49, no. 5, pp. 1087–1103, 2013.
- [16] T. A. Johansen, T. I. Fossen, and S. P. Berge, "Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 1, pp. 211–216, 2004.
- [17] D.-T. Nguyen, D. Saussie, and L. Saydy, "Design and experimental validation of robust self-scheduled fault-tolerant control laws for a multicopter uav," *IEEE/ASME Transactions on Mechatronics (TMECH)*, vol. 26, no. 5, pp. 2548–2557, 2020.
- [18] S. J. Lee, I. Jang, and H. J. Kim, "Fail-safe flight of a fully-actuated quadrotor in a single motor failure," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 4, pp. 6403–6410, 2020.
- [19] W. Chung and H. Son, "Fault-tolerant control of multirotor uavs by control variable elimination," *IEEE/ASME Transactions on Mechatronics (TMECH)*, vol. 25, no. 5, pp. 2513–2522, 2020.
- [20] M. W. Mueller and R. D'Andrea, "Relaxed hover solutions for multicopters: Application to algorithmic redundancy and novel vehicles," *International Journal of Robotics Research (IJRR)*, vol. 35, no. 8, pp. 873–889, 2016.
- [21] X. Shao, G. Sun, W. Yao, J. Liu, and L. Wu, "Adaptive sliding mode control for quadrotor uavs with input saturation," *IEEE/ASME Transactions on Mechatronics (TMECH)*, vol. 27, no. 3, pp. 1498–1509, 2021.
- [22] G. Michieletto, M. Ryll, and A. Franchi, "Control of statically hoverable multi-rotor aerial vehicles and application to rotor-failure robustness for hexarotors," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2017.

- [23] G.-X. Du, Q. Quan, B. Yang, and K.-Y. Cai, "Controllability analysis for multirotor helicopter rotor degradation and failure," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 5, pp. 978–985, 2015.
- [24] A. Marks, J. F. Whidborne, and I. Yamamoto, "Control allocation for fault tolerant control of a vtol octorotor," in *Proceedings of UKACC International Conference on Control*, 2012.
- [25] M. J. Gerber and T.-C. Tsao, "Twisting and tilting rotors for high-efficiency, thrust-vectorized quadrotors," *Journal of Mechanisms and Robotics*, vol. 10, no. 6, p. 061013, 2018.
- [26] B. Li, L. Ma, D. Huang, and Y. Sun, "A flexibly assembled and maneuverable reconfigurable modular multirotor aerial vehicle," *IEEE/ASME Transactions on Mechatronics (TMECH)*, vol. 27, no. 3, pp. 1704–1714, 2021.
- [27] C. Ding and L. Lu, "A tilting-rotor unmanned aerial vehicle for enhanced aerial locomotion and manipulation capabilities: Design, control, and applications," *IEEE/ASME Transactions on Mechatronics (TMECH)*, vol. 26, no. 4, pp. 2237–2248, 2020.
- [28] S. Park, J. Lee, J. Ahn, M. Kim, J. Her, G.-H. Yang, and D. Lee, "Odar: Aerial manipulation platform enabling omnidirectional wrench generation," *IEEE/ASME Transactions on Mechatronics (TMECH)*, vol. 23, no. 4, pp. 1907–1918, 2018.
- [29] G. Michieletto, M. Ryll, and A. Franchi, "Fundamental actuation properties of multirotors: Force–moment decoupling and fail-safe robustness," *IEEE Transactions on Robotics (T-RO)*, vol. 34, no. 3, pp. 702–715, 2018.
- [30] D.-T. Nguyen, D. Saussie, and L. Saydy, "Fault-tolerant control of a hexacopter uav based on self-scheduled control allocation," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018.
- [31] C. Pose and J. Giribet, "Multirotor fault tolerance based on center-of-mass shifting in case of rotor failure," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021.
- [32] M. A. da Silva Ferreira, M. F. T. Begazo, G. C. Lopes, A. F. de Oliveira, E. L. Colombini, and A. da Silva Simões, "Drone reconfigurable architecture (dra): A multipurpose modular architecture for unmanned aerial vehicles (uavs)," *Journal of Intelligent & Robotic Systems*, vol. 99, pp. 517–534, 2020.
- [33] A. F. Şenkul and E. Altuğ, "System design of a novel tilt-roll rotor quadrotor uav," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 575–599, 2016.
- [34] P. Yu, Y. Su, L. Ruan, and T.-C. Tsao, "Compensating aerodynamics of over-actuated multi-rotor aerial platform with data-driven iterative learning control," *IEEE Robotics and Automation Letters (RA-L) (submitted)*, 2023.
- [35] O. Kose and T. Oktay, "Simultaneous quadrotor autopilot system and collective morphing system design," *Aircraft Engineering and Aerospace Technology*, vol. 92, no. 7, pp. 1093–1100, 2020.
- [36] P. Yu, *An Over-Actuated Multi-Rotor Aerial Platform and Iterative Learning Control Applications*. PhD thesis, University of California, Los Angeles, 2022.
- [37] X. Zhang, Z. Zhao, Z. Wang, and X. Wang, "Fault detection and identification method for quadcopter based on airframe vibration signals," *Sensors*, vol. 21, no. 2, p. 581, 2021.
- [38] 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.
- [39] R. Yang, L. Zheng, J. Pan, and H. Cheng, "Learning-based predictive path following control for nonlinear systems under uncertain disturbances," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, pp. 2854–2861, 2021.
- [40] Z. Wang, Z. Gong, Y. Chen, M. Sun, and J. Xu, "Practical control implementation of tri-tiltrotor flying wing unmanned aerial vehicles based upon active disturbance rejection control," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 234, no. 4, pp. 943–960, 2020.
- [41] Y. Su, C. Chu, M. Wang, Y. Liu, Y. Zhu, and H. Liu, "Downwash-aware control allocation for over-actuated uav platforms," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2022.
- [42] Y. Su, Z. Jiao, Z. Zhang, C. Chu, J. Li, H. Li, M. Wang, and H. Liu, "Flight structure optimization of modular reconfigurable uavs," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS) (submitted)*, 2023.
- [43] Y. Su, J. Li, Z. Jiao, M. Wang, C. Chu, H. Li, Y. Zhu, and H. Liu, "Sequential manipulation planning for over-actuated uams," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS) (submitted)*, 2023.



Yao Su (Member, IEEE) received the B.S. degree from the School of Mechatronic Engineering, Harbin Institute of Technology in 2016, and the M.S. and Ph.D. degrees from the Department of Mechanical and Aerospace Engineering, University of California, Los Angeles in 2017 and 2021. He is now a research scientist at National Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI). His research interests include robotics, control, optimization, trajectory planning, and mechatronics.



Pengkang Yu received the B.Eng. degree in Mechanical Engineering from Hong Kong University of Science and Technology, Hong Kong, in 2016. He received the M.S. degree in 2017 and the Ph.D. degree in 2022 in Mechanical Engineering from the University of California, Los Angeles. His research interests include control, optimization, planning, robotics and mechatronics.



Matthew J. Gerber received the B.Eng. degree in Mechanical Engineering from Hong Kong University of Science and Technology, Hong Kong, in 2016. He received the M.S. degree in 2017 and the Ph.D. degree in 2022 in Mechanical Engineering from the University of California, Los Angeles. His research interests include control, optimization, planning, robotics and mechatronics.



Lecheng Ruan received the B.S. honor degree from the School of Mechatronic Engineering, Harbin Institute of Technology in 2015, and the Ph.D. degree from the Department of Mechanical and Aerospace Engineering, University of California, Los Angeles in 2020. He is now affiliated with the National Key Laboratory of General Artificial Intelligence and Peking University. His research interests include control and optimization, mechatronics, robotics, perception and signal processing.



Tsu-Chin Tsao (Senior Member, IEEE; Fellow, ASME) received the B.S. degree in engineering from National Taiwan University, Taipei, Taiwan, in 1981, and the M.S. and Ph.D. degrees in mechanical engineering from the University of California Berkeley, Berkeley, CA, USA, in 1984 and 1988, respectively. He is currently a Professor with the Mechanical and Aerospace Engineering Department, University of California, Los Angeles, Los Angeles CA, USA. His research interests include precision motion control, mechatronics, and robotics.