Fault-Tolerant Control of an Overactuated 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, overactuated platforms can continue flying with appropriate fault-tolerant control (FTC). This article presents a robust FTC controller for an overactuated 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: 1) a low-level adjustment strategy to prevent propeller-level thrust saturation; 2) a compensation loop for mitigating introduced disturbances; and 3) 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*—Fault-tolerant control (FTC), input redundancy, nullspace allocation, optimization, overactuated unmanned aerial vehicles (UAVs), propeller failure.**

I. Introduction

OVERACTUATED multirotor unmanned aerial vehicles (UAVs) have been proposed in the last decade to overcome the underactuation issue of traditional colinear 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], [3], [4] employs multiple propeller-motor pairs in various or varying directions to achieve full or overactuation. The second group of works [5], [6], [7], [8], [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 **propeller failure** compared to conventional quadcopters: without sufficient fault-tolerant control (FTC) algorithms, overactuated 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.

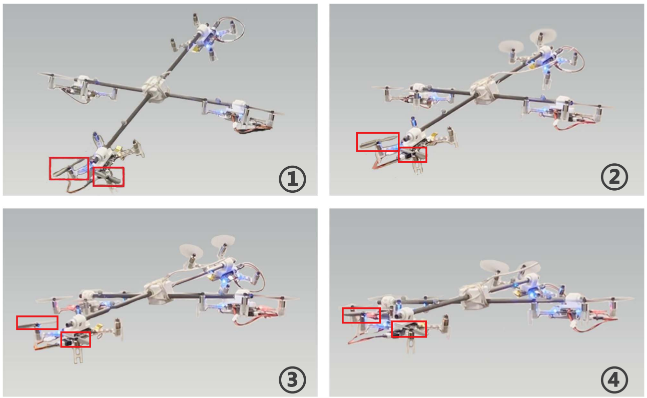


Fig. 1. With two failed propellers on one quadcopter module, our proposed FTC algorithm can keep the platform from crashing and remain stable trajectory tracking. (Failed propellers are labeled by red boxes).

In this article, we first implement the previously proposed nullspace-based control allocation framework [10] to address the **thrust force saturation** issue in overactuated UAV platforms. We validate this approach on our customized overactuated platform [8], where four miniquadcopters are connected to the mainframe via 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

Manuscript received 20 February 2023; revised 24 April 2023; accepted 13 June 2023. Recommended by Technical Editor Phanindra Tallapragada and Senior Editor Kostas J. Kyriakopoulos. *(Yao Su and Pengkang Yu contributed equally to this work.) (Corresponding authors: Yao Su; Lecheng Ruan.)*

The authors are with the Mechanical and Aerospace Engineering Department, University of California, Los Angeles, CA 90095 USA (e-mail: yaosu@ucla.edu; paulyu1994@ucla.edu; gerber211@ucla.edu; ruanlecheng@ucla.edu; ttsao@ucla.edu).

The video of experiments is available at https://www.youtube.com/watch?v=Y-8l1EBH5gU

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.2023.3288032.

Digital Object Identifier 10.1109/TMECH.2023.3288032

scenarios in which one or more propellers on a single quadcopter (denoted as **Bad QC**) fail while the other three quadcopters (denoted as **Good QCs**) continue functioning properly (see Fig. 1). This FTC controller employs a hierarchical structure with three main components: 1) a low-level controller to adjust the propeller-level thrust force distribution on the Bad QC, 2) a high-level controller to reallocate quadcopter-level thrust force distribution among the Good and Bad QCs, and 3) 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 overactuated 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 overactuated 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 overactuated UAV platforms.

The rest of the article is organized as follows. Related work is summarized in Section II. The dynamics models of the overactuated UAV platform are reviewed in Section III. Section IV presents the control architecture and the nullspace-based framework for control allocation. Section V describes the FTC framework to handle propeller failure. Sections VI and VII present the simulation and experiment results. Finally, Section VIII concludes this article.

II. Related Work

A. Control Allocation

Control allocation in *overactuated 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. introduced 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 sacrificed input redundancies. Furthermore, iterative approach [13] and separation method [14] were proposed for improved efficiency. However, none of these methods [2], [12], [13], [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 article, we further implement this framework to address the issue of thrust force saturation and propeller failure on a different overactuated UAV platform.

B. Fault-Tolerant Control

UAV 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], [18], [19], [20], [21], [22]. Multirotor UAVs with more than six controllable inputs [4], [23], [24], [25] or tilt-rotor quadcopters [26], [27], [28], [29] 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 [30]. Related FTC controllers were presented in [31] based on modifications of control allocation and in [32] based on center-of-mass shifting. A new type of overactuated 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 article, 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], [26], [27], [33], [34], 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 and Property

The overactuated multirotor platform discussed in this article adopts commercial quadcopters with passive hinges, serving as two-degree-of-freedom (DoF) tiltable thrust generators [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 $\boldsymbol{\xi} = [x, y, z]^\mathsf{T}$, the attitude is defined in the roll-pitch-yaw convention as $\boldsymbol{\eta} = [\phi, \theta, \psi]^\mathsf{T}$, and the angular velocity is defined as $\boldsymbol{\nu} = [p, q, r]^\mathsf{T}$.

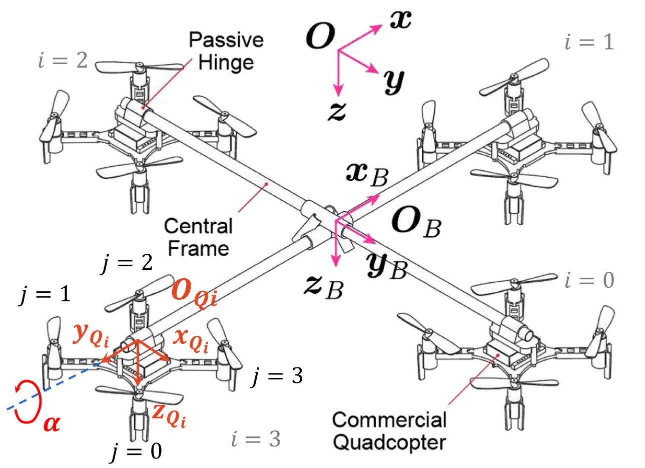


Fig. 2. Shown is the prototype used in this article; four commercial quadcopters are passively hinged to the central frame of the platform.

A. Platform Dynamics Model

The dynamics model of this platform can be simplified as

$$\begin{bmatrix} {}^W\ddot{\boldsymbol{\xi}} \\ {}^B\dot{\boldsymbol{\nu}} \end{bmatrix} = \begin{bmatrix} \frac{1}{m}\,{}^W_B\boldsymbol{R} & 0 \\ 0 & {}^B\boldsymbol{I}^{-1} \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} {}^W\boldsymbol{G} \\ \boldsymbol{0} \end{bmatrix} \tag{1}$$

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

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

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

$$\begin{bmatrix} T_i \\ M_i^x \\ M_i^y \\ M_i^z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -b & -b & b & b \\ b & -b & -b & b \\ c_\tau & -c_\tau & c_\tau & -c_\tau \end{bmatrix} \begin{bmatrix} t_{i0} \\ t_{i1} \\ t_{i2} \\ t_{i3} \end{bmatrix} \tag{3}$$

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]

$$\ddot{\alpha}_i = \frac{1}{I_i^y} M_i^y - \sin\left(\frac{\pi}{2} i\right)\dot{p} - \cos\left(\frac{\pi}{2} i\right)\dot{q} \tag{4}$$

where I_i^y is the inertia in $\boldsymbol{y}_{Q_i}$ direction.

C. Platform Property

As introduced in [9] and [35], this type of overactuated UAV platform has unlimited joint angle ranges and greatly reduces the mechanical complexity compared to existing tiltable-rotor configurations [2], [12], [27], [36], [37], where each rotor's tilting is actuated by a servo motor. Moreover, our proposed configuration eliminates propeller drags, gyroscopic momenta, and tilting reaction torques—factors treated as disturbances or unmodeled dynamics in other tiltable-rotor platforms—due to paired propellers rotating in opposite directions and the zero-torque transmission in passive hinges. Therefore, this type of platform has the best trajectory tracking performance among these configurations.

Another notable feature of this platform is the presence of fast auxiliary inputs M_i^x and M_i^z in the low-level control of $\mathcal{Q}_i$ (3). In the event of propeller failure, different from other overactuated UAV configurations requiring abandonment of the propeller-motor pair and related servo motors, the low-level input redundancy of our platform offers partial thrust generation capability, preventing the platform crashes with more weights onboard. In addition, auxiliary inputs can improve control performance (e.g., as done in [11] and [38]) because the dynamics of each motor are sufficiently fast and considered as feed-through dynamics.

Furthermore, our proposed configuration could potentially become a modular reconfigurable quadcopter system by equipping each quadcopter with a docking frame. Various similar designs are proposed to transform quadcopter swarms into connected overactuated flying structures through docking process [39], [40], [41], [42]. The popularity of constructing flying structures with quadcopters highlights the significance of our platform and associated control challenges.

IV. Nominal Control

A. Hierarchical Architecture and Tracking Control

The overall controller features a hierarchical structure, consisting of 1) 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; 2) 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:

$$\boldsymbol{u}^d = \begin{bmatrix} \boldsymbol{J}_\xi \\ \boldsymbol{J}_\nu \end{bmatrix} \boldsymbol{T} = \begin{bmatrix} m\,{}^W_B\boldsymbol{R}^\intercal & \boldsymbol{0} \\ \boldsymbol{0} & {}^B\boldsymbol{I} \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{u}_\xi \\ \boldsymbol{u}_\nu \end{bmatrix} - \begin{bmatrix} {}^W\boldsymbol{G} \\ \boldsymbol{0} \end{bmatrix} \right) \tag{5}$$

where the superscript d indicates the desired values, $\boldsymbol{u}_\xi$ and $\boldsymbol{u}_\nu$ are virtual inputs for position control and attitude control, respectively. Combining (5) with (1), the platform dynamics is equivalent to a double integrator and can be written in a state-space form as

$$\dot{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{B}}\widetilde{\boldsymbol{u}} \tag{6}$$

where

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} 0 & 0 & \boldsymbol{I}_3 & 0 \\ 0 & 0 & 0 & \boldsymbol{I}_3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{\mathcal{B}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \boldsymbol{I}_3 & 0 \\ 0 & \boldsymbol{I}_3 \end{bmatrix}$$

$$\boldsymbol{\mathcal{X}} = \begin{bmatrix} \boldsymbol{\xi}^{\mathsf{T}} & \boldsymbol{\eta}^{\mathsf{T}} & \dot{\boldsymbol{\xi}}^{\mathsf{T}} & \boldsymbol{\nu}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}, \quad \widetilde{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u}_\xi^{\mathsf{T}} & \boldsymbol{u}_\nu^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}. \tag{7}$$

We design a linear quadratic integral (LQI) control scheme [43], [44], [45] to close the control loop with the augmented system states as

$$\widetilde{\boldsymbol{\mathcal{X}}} = \begin{bmatrix} \boldsymbol{e}_\xi^{\mathsf{T}} & \boldsymbol{e}_\eta^{\mathsf{T}} & \dot{\boldsymbol{e}}_\xi^{\mathsf{T}} & \dot{\boldsymbol{e}}_\eta^{\mathsf{T}} & \int \boldsymbol{e}_\xi dt^{\mathsf{T}} & \int \boldsymbol{e}_\eta dt^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}. \tag{8}$$

The cost function is

$$\widetilde{\boldsymbol{J}}(\widetilde{\boldsymbol{\mathcal{X}}}, \widetilde{\boldsymbol{u}}) = \int_0^\infty (\widetilde{\boldsymbol{\mathcal{X}}}^{\mathsf{T}} \widetilde{\boldsymbol{\mathcal{Q}}} \widetilde{\boldsymbol{\mathcal{X}}} + \widetilde{\boldsymbol{u}}^{\mathsf{T}} \widetilde{\boldsymbol{\mathcal{R}}} \widetilde{\boldsymbol{u}}) dt \tag{9}$$

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

$$\widetilde{\boldsymbol{u}} = -\boldsymbol{K}\widetilde{\boldsymbol{\mathcal{X}}} \tag{10}$$

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

B. Nominal Allocation Framework

1) FD-Based Allocation:

Given $\boldsymbol{u}^d$, we aim to determine the desired tilting angle $\boldsymbol{\alpha}$ and thrust $\boldsymbol{T}$ for the four quadcopters through the nonlinear mapping (5)—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]

$$\boldsymbol{F} = [F_{s0} \quad F_{c0} \quad \dots \quad F_{s3} \quad F_{c3}]^{\mathsf{T}} \tag{11}$$

where

$$F_{si} = \sin\alpha_i T_i, \; F_{ci} = \cos\alpha_i T_i. \tag{12}$$

With these new variables, (5) can be rewritten as

$$\boldsymbol{u}^d = \begin{bmatrix} \boldsymbol{J}_\xi \\ \boldsymbol{J}_\nu \end{bmatrix} \boldsymbol{T} = \boldsymbol{W}\boldsymbol{F} \tag{13}$$

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

$$\boldsymbol{F} = \boldsymbol{W}^\dagger \boldsymbol{u}^d + \boldsymbol{N}_W \boldsymbol{Z} \tag{14}$$

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

$$T_i = \sqrt{F_{si}^2 + F_{ci}^2}, \; \alpha_i = \text{atan2}(F_{si}, \, F_{ci}). \tag{15}$$

2) Thrust Force Saturation Issue:

However, this FD-based allocation framework (referred 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] and [9], where the nominal FD-based allocation framework proved inadequate for utilizing full thrust capability of the platform, and an analytical solution was provided for a 1-D 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 (will be introduced next) determines the optimal tilting angle and thrust for each quadcopter, subject to a predefined cost function and input constraints (see Section 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

$$\min_{\Delta\boldsymbol{X}, \boldsymbol{s}} \quad \boldsymbol{J} = \Delta\boldsymbol{X}^{\mathsf{T}} \boldsymbol{P} \Delta\boldsymbol{X} + \boldsymbol{s}^{\mathsf{T}} \boldsymbol{Q} \boldsymbol{s} \tag{16}$$

$$\text{s.t.} \quad \boldsymbol{W}\left(\boldsymbol{s} + \boldsymbol{F}(\boldsymbol{\alpha}_o, \boldsymbol{T}_o) + \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{X}}\right|_{\boldsymbol{X}=\boldsymbol{X}_o} \Delta\boldsymbol{X}\right) = \boldsymbol{u}^d \tag{17}$$

$$\boldsymbol{X}_{\min} - \boldsymbol{X}_o \le \Delta\boldsymbol{X} \le \boldsymbol{X}_{\max} - \boldsymbol{X}_o \tag{18}$$

$$\Delta\boldsymbol{X}_{\min} \le \Delta\boldsymbol{X} \le \Delta\boldsymbol{X}_{\max} \tag{19}$$

where (16) is the object function, with $\boldsymbol{X}$ defined as

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{\alpha}^{\mathsf{T}} & \boldsymbol{T}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \tag{20}$$

$\boldsymbol{P}$ and $\boldsymbol{Q}$ are weighting matrices. Equation (17) uses first-order linearization to approximate the nonlinear equality constraint (14), 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 $\boldsymbol{s}$ serves as a slack variable. Equations (18) and (19) 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

$$\boldsymbol{X} = \boldsymbol{X}_o + \Delta\boldsymbol{X}. \tag{21}$$

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

$$\boldsymbol{F}^* = (\boldsymbol{I}_{3n} - \boldsymbol{N}_W \boldsymbol{N}_W^\dagger) \boldsymbol{W}^\dagger \boldsymbol{u}^d + \boldsymbol{N}_W \boldsymbol{N}_W^\dagger \boldsymbol{F}(\boldsymbol{X}). \tag{22}$$

Finally, $\boldsymbol{\alpha}^*$ and $\boldsymbol{T}^*$ can be determined from $\boldsymbol{F}^*$ using (15). The nullspace-based allocation framework takes into account input constraints while still providing an exact solution for (13). 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 Section V.

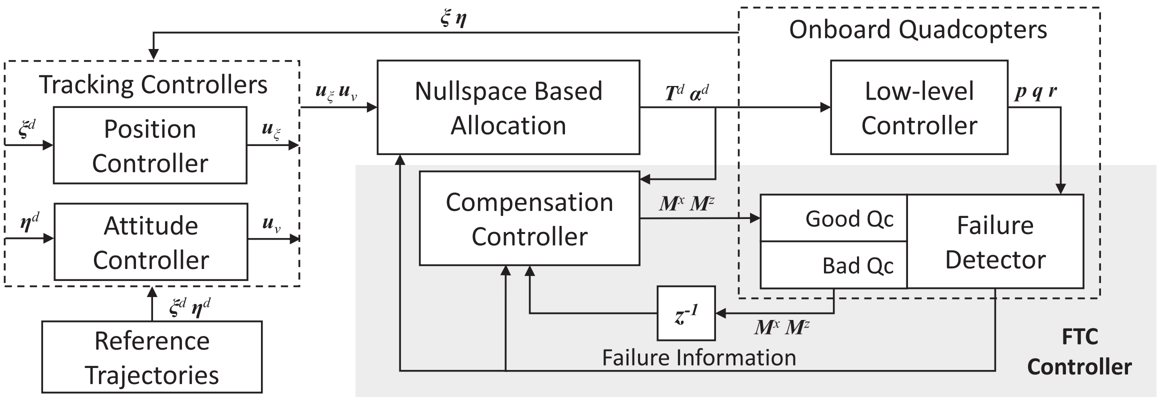


Fig. 3. Fault-Tolerant Controller Architecture. Each quadcopter is equipped with onboard failure detection and motor-control modules. When propeller failure is detected, the failure detection module communicates with the onboard controller to modify the low-level control strategy according to identified failure combinations. In addition, 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.

D. Low-Level Control

The thrust T_i can be directly controllable in the low-level controller for each $\mathcal{Q}_i$, whereas the tilting angle α_i is controlled by M_i^y through the motor's second-order rotational dynamics (4). Therefore, a double-loop PID controller is applied to track the tilting angle [1], [8].

Considering these relationships and neglecting the fast motor dynamics that drive the propeller speeds, the propeller thrusts can be calculated from (3)

$$\begin{bmatrix} t_{i0} \\ t_{i1} \\ t_{i2} \\ t_{i3} \end{bmatrix} = sat\left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ -b & -b & b & b \\ b & -b & -b & b \\ c_\mathcal{T} & -c_\mathcal{T} & c_\mathcal{T} & -c_\mathcal{T} \end{bmatrix}^{-1} \begin{bmatrix} T_i \\ M_i^x \\ M_i^y \\ M_i^z \end{bmatrix} \right) \tag{23}$$

where $sat(\cdot)$ 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, $\mathcal{Q}_3$ 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], [46]. Therefore, we assume that the propeller failure combinations has been accurately detected in this article.

A. Fault-Tolerant Controller Architecture

As shown in Fig. 3, the FTC controller comprises three primary components:

1) 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.
2) 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 faster response within the saturation constraints.
3) 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 $\mathcal{Q}_3$ are assumed functional, while propeller $\mathcal{P}_0$ is considered to have failed. As a result, the four outputs in (3) cannot be independently controlled [18]. To account for this, (23) is adjusted to calculate the low-level commands as follows, where the control of M_3^z is lost (its magnitude is relatively small)

$$\begin{bmatrix} t_{31} \\ t_{32} \\ t_{33} \end{bmatrix} = sat\left(\begin{bmatrix} \frac{1}{2} & -\frac{1}{2b} & 0 \\ 0 & \frac{1}{2b} & -\frac{1}{2b} \\ \frac{1}{2} & 0 & \frac{1}{2b} \end{bmatrix} \begin{bmatrix} T_3 \\ M_3^x \\ M_3^y \end{bmatrix} \right). \tag{24}$$

Given that the mapping matrix in (24) 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 toque M_3^x.

Based on the aforementioned analysis, (24) is not employed in the low-level control. Instead, it is reformulated as follows:

$$\begin{bmatrix} t_{31} \\ t_{32} \\ t_{33} \end{bmatrix} = sat\left(\begin{bmatrix} \frac{1}{4} & -\frac{1}{4b} \\ \frac{1}{4} & -\frac{1}{4b} \\ \frac{1}{2} & \frac{1}{2b} \end{bmatrix}\begin{bmatrix} T_3 \\ M_3^y \end{bmatrix}\right). \tag{25}$$

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 (24) and (25) in low-level control will be shown in Section VI.

b) High-Level Control: From (25), it can be shown that T_3 will not be evenly distributed among the three remaining propellers, with $\mathcal{P}_3$ 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 (18) must be changed from $t_{\max} \cdot [4,4,4,4]^\mathsf{T}$ to $t_{\max} \cdot [4,4,4,2]^\mathsf{T}$. Note that the existing allocation strategies [2], [12], [13], [14] cannot accommodate input constraints, necessitating the use of the nullspace-based constrainted 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

$${}^B\dot{\boldsymbol{\nu}} = {}^B\boldsymbol{I}^{-1}\left(\boldsymbol{J}_v\boldsymbol{T} + \boldsymbol{J}_M^x\boldsymbol{M}^x + \boldsymbol{J}_M^z\boldsymbol{M}^z\right) \tag{26}$$

where

$$\boldsymbol{J}_M^x = \begin{bmatrix} -\cos\alpha_0 & 0 & \cos\alpha_2 & 0 \\ 0 & \cos\alpha_1 & 0 & -\cos\alpha_3 \\ \sin\alpha_0 & \sin\alpha_1 & \sin\alpha_2 & \sin\alpha_3 \end{bmatrix}$$

$$\boldsymbol{M}^x = \begin{bmatrix} M_0^x & M_1^x & M_2^x & M_3^x \end{bmatrix}^\mathsf{T}$$

$$\boldsymbol{J}_M^z = \begin{bmatrix} \sin\alpha_0 & 0 & -\sin\alpha_2 & 0 \\ 0 & -\sin\alpha_1 & 0 & \sin\alpha_3 \\ \cos\alpha_0 & \cos\alpha_1 & \cos\alpha_2 & \cos\alpha_3 \end{bmatrix}$$

$$\boldsymbol{M}^z = \begin{bmatrix} M_0^z & M_1^z & M_2^z & M_3^z \end{bmatrix}^\mathsf{T}. \tag{27}$$

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

$$\boldsymbol{J}_M^x\boldsymbol{M}^x + \boldsymbol{J}_M^z\boldsymbol{M}^z + \boldsymbol{k} = 0 \tag{28}$$

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		×

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

$$J(\boldsymbol{y},\boldsymbol{k}) = \boldsymbol{y}^\mathsf{T}\boldsymbol{A}\boldsymbol{y} + \boldsymbol{k}^\mathsf{T}\boldsymbol{B}\boldsymbol{k} \tag{29}$$

where $\boldsymbol{y}$ is defined as

$$\boldsymbol{y} = [M_0^x \quad M_1^x \quad M_2^x \quad M_0^z \quad M_1^z \quad M_2^z]^\mathsf{T} \tag{30}$$

and $\boldsymbol{A}$ and $\boldsymbol{B}$ are constant, positive semidefinite gain matrices. Saturation is included as the following inequality constraints:

$$0 \leq \begin{bmatrix} 1 & 1 & 1 & 1 \\ -b & -b & b & b \\ b & -b & -b & b \\ c_\tau & -c_\tau & c_\tau & -c_\tau \end{bmatrix}^{-1}\begin{bmatrix} T_i \\ M_i^x \\ M_i^y \\ M_i^z \end{bmatrix} \leq t_{\max} \quad \forall i = 0,\,1,\,2. \tag{31}$$

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 overactuated UAV platforms [47], [48], 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 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 $\mathcal{P}_2$ (see Table I).

As an example, we consider the case where both $\mathcal{P}_0$ and $\mathcal{P}_1$ are failed. The low-level controller is designed as

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

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 Section V-B1, the high-level maximum thrust constraint is modified, and the compensation loop is implemented for disturbance attenuation.

TABLE II
PHYSICAL AND SOFTWARE PROPERTIES IN SIMULATION

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

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 Table 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 article, the quadcopter opposing to 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 nonopposing 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 (see Section 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 Table 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 (24) and (25) were compared in this simulation. The nominal controller with FD-based allocation was utilized at the high level to track the reference position and attitude trajectory. In both tests, $\mathcal{P}_0$ on $\mathcal{Q}_3$ began to fail at 2 s (see Fig. 4).

As illustrated in Fig. 4(a), with the first strategy (24), the desired thrust t_{32} could become a nonpositive 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 9 s, leading to instability in the position and attitude control of the whole platform.

In contrast, with the second strategy (25), both position and attitude control remained stable throughout the trajectory, as shown in Fig. 4(b). In the low-level control of $\mathcal{Q}_3$, all propeller thrusts stayed within the saturation range, thus ensuring proper tilting angle control.

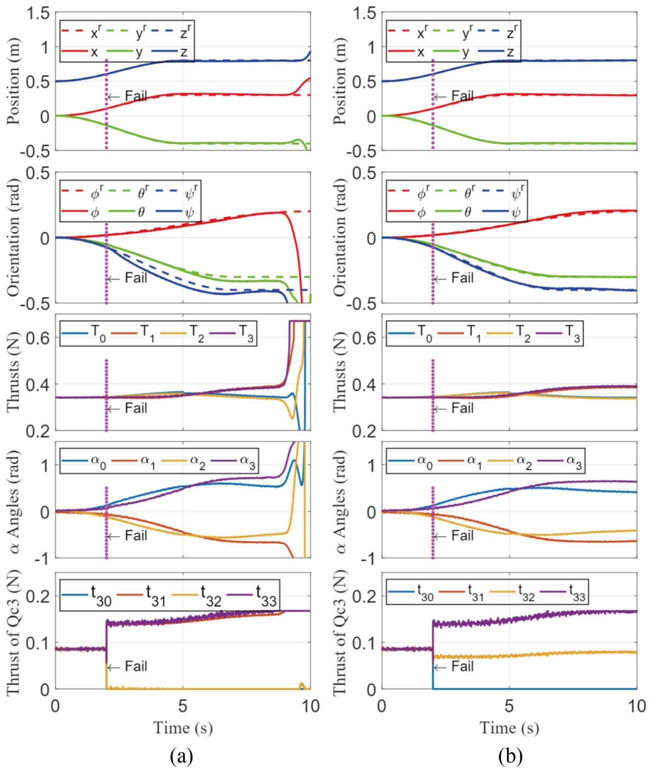


Fig. 4. Simulation: Trajectory tracking performance of two low-level adjustment strategies under propeller failure. (a) corresponds to (24), (b) corresponds to (25), (a) first strategy, (b) second strategy.

VII. Experiments

A. Experiment Setup

The Crazyflie 2.1 quadcopter served as the basis for each individual unit within the platform. To achieve greater thrust force, the battery and motors on each quadcopter were upgraded, resulting in a maximum thrust force of 0.67 $N(4 \times t_{\max})$ 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 × 36 × 6 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.

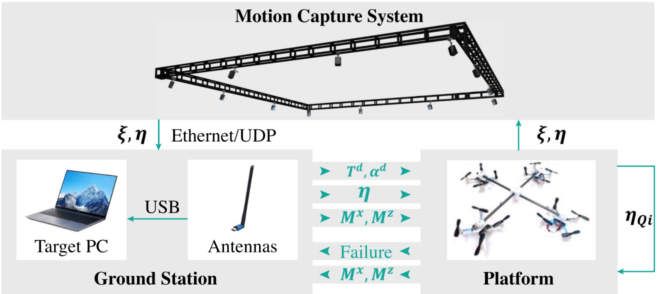
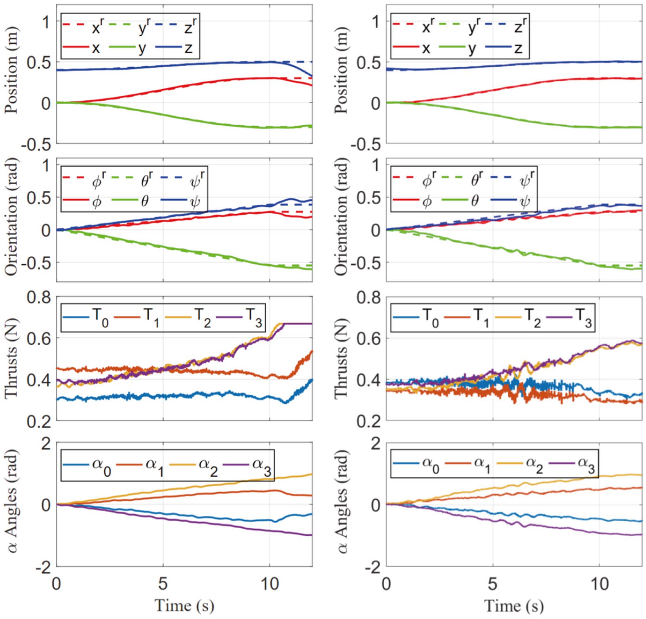


Fig. 5. Experimental setup. A host PC runs at 100 Hz the high-level controller of the whole platform with measurements from the OptiTrack motion-capture system. Control signals are sent to each unit-copter that runs the low-level controller at 500 Hz.



(a) (b)

Fig. 6. Experiment: Using different allocation frameworks to handle thrust force saturation. (a) FD-based framework. (b) Nullspace-based framework.

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 (see 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 2 s. As shown in Fig. 6(a), 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 10.7 s, 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 [see Fig. 6(b)]. As analyzed in Section 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.

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:

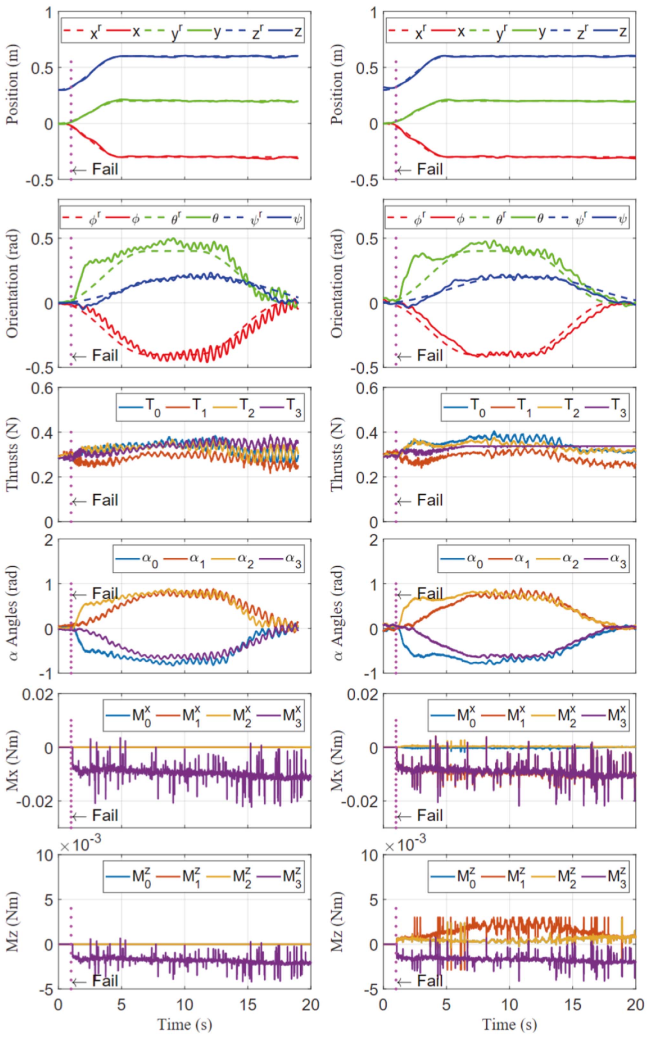
a) Unsaturated Trajectory: In this experiment, an unsaturated trajectory was employed, ensuring that $T_3 \leq 2\,t_{\max}$ at each timestep. As analyzed in Section V-B1, the N+L controller maintained the stability of the platform when the trajectory did not trigger low-level saturation of $\mathcal{Q}_3$. 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 1 s. 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 [see Fig. 7(a)], 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 [see Fig. 7(b)] 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 $T_3 > 2\,t_{\max}$ at some timesteps [see Fig. 8(a)] 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 1 s. The tracking performance of the two controllers is plotted in Fig. 8.

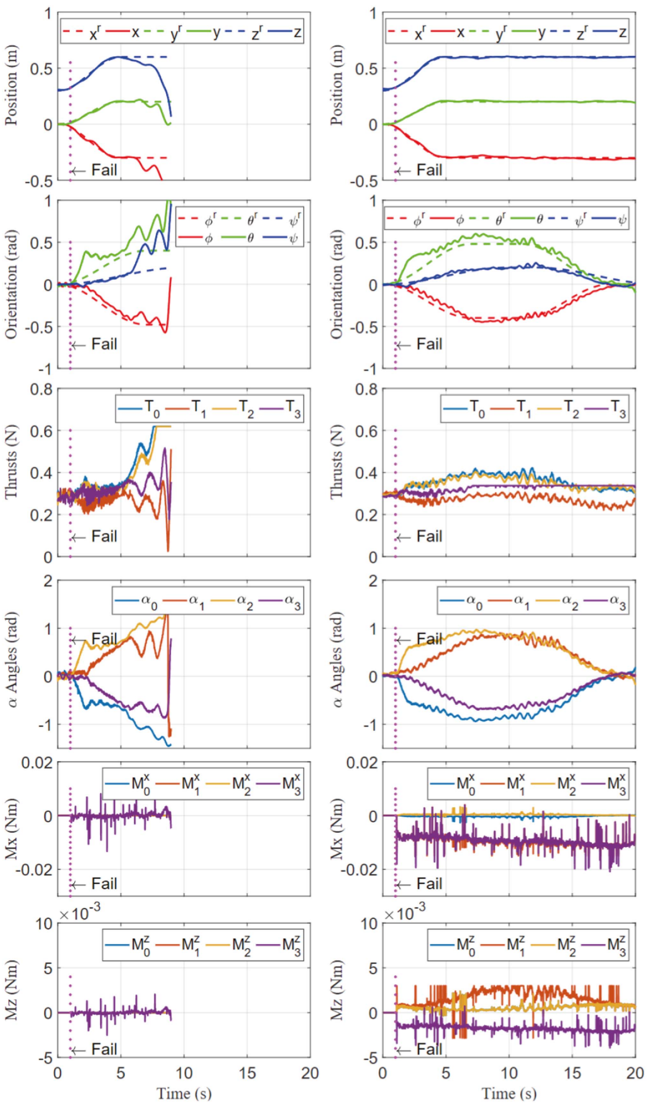
As shown in Fig. 8(a), when the low-level saturation constraint was triggered. the N+L controller caused the UAV platform to become unstable at approximately 7 s. Two explanations are given here. First, the nominal allocation strategy did not consider the thrust-saturation constraint, leading to output desired thrust T_3 beyond the capability of $\mathcal{Q}_3$. Second, the disturbance torques M_x^3 and M_z^3 generated by the low-level control were transferred to the central frame without compensation.

For the FTC controller [see Fig. 8(b)], 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) (b)

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



(a) (b)

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.) (a) One Fail (S): N+L. (b) One Fail (S): FTC

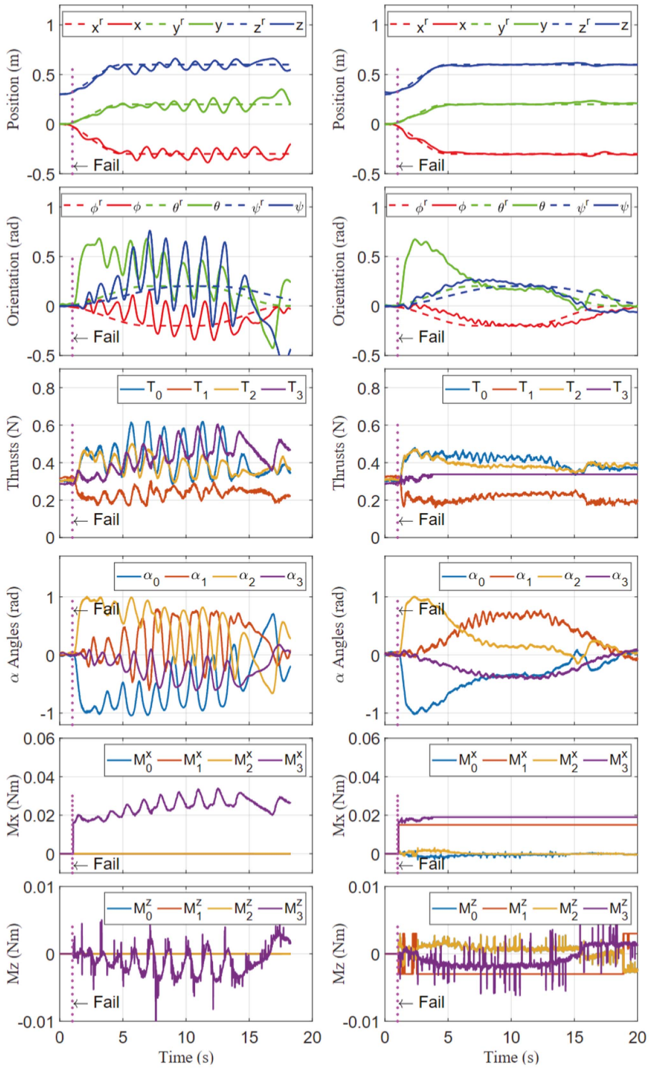
2) Two Failed Propellers: In this experiment, $\mathcal{P}_0$ and $\mathcal{P}_1$ of $\mathcal{Q}_3$ began to fail at 1 s while tracking the reference trajectory. The tracking performance of the N+L controller and FTC controller are both plotted in Fig. 9. In this scenario, the disturbance torques M_3^x 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. 9(a), 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 [see Fig. 9(b)]. Due to the larger interacting disturbance torques produced by low-level adjustment in this case, the overall performance for both controllers was worse than that observed in Case 2. Some video clips of this experiment are shown in Fig. 1.

D. Discussion

Utilizing the hierarchical control architecture, the high-level control of our proposed configuration is the same as other overactuated UAV configurations, therefore thrust-force saturation issue we solved in this work is a general problem that could meet by all overactuated UAV platforms.

The specialty of our configuration lies in the input redundancy in the low-level control loop, which offers partial thrust-generation capability when propeller failure happens. If the low-level redundancy ensures passive joint control, the partial thrust-generation utilized by our proposed FTC could provide better performance than giving up the failed quadcopter module. If the control of the passive joint is not maintained by the low-level control (failed cases in Table I), the uncontrollable relative motion between the quadcopter and mainframe will



(a) (b)

Fig. 9. Case 3: Trajectory tracking performance when two propellers are failed. (a) Two Fails: N+L. (b) Two Fails: FTC.

introduce a huge disturbance to the platform which could be future work for this research.

VIII. Conclusion

In this article, we have addressed the FTC method for over-actuated UAV platforms. Our approach is applicable to the widely used hierarchical control architecture and utilizes the nullspace-based constrained control allocation we have recently developed.

To summarize the FTC within the hierarchical control, 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. Furthermore, uncontrolled disturbances generated by the Bad QC are compensated by the Good QCs whenever within the saturation range of them. Both simulation and experimental results, in 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 [49], [50], [51]. 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.

Acknowledgment

The authors would like to thank Dr. Hangxin Liu, Dr. Zeyu Zhang, Dr. Wenzhong Yan, and Dr. Ankur Mehta for the figure improvement 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 Trans. Mechatron.*, vol. 25, no. 4, pp. 1773–1782, Aug. 2020.

[2] M. Kamel et al., "The Voliro omniorientational hexacopter: An agile and maneuverable tiltable-rotor aerial vehicle," *IEEE Robot. Autom. Mag.*, vol. 25, no. 4, pp. 34–44, Dec. 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 *Proc. Int. Conf. Robot. Autom.*, 2015, pp. 4006–4013.

[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 octorotor," in *Proc. Int. Conf. Robot. Autom.*, 2015, pp. 5266–5271.

[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 Trans. Robot.*, vol. 34, no. 2, pp. 353–369, Apr. 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 Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6828–6835, Oct. 2021.

[7] L. Ruan, "Independent position and attitude control on multirotor aerial platforms," Ph.D. thesis, Univ. California, Los Angeles, CA, USA, 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 *Proc. IEEE Conf. Control Technol. Appl.*, 2021, pp. 329–334.

[9] L. Ruan, C.-H. Pi, Y. Su, P. Yu, S. Cheng, and T.-C. Tsao, "Control and experiments of a novel tiltable-rotor aerial platform comprising quadcopters and passive hinges," *Mechatronics*, vol. 89, 2023, Art. no. 102927.

[10] Y. Su, P. Yu, M. Gerber, L. Ruan, and T.-C. Tsao, "Nullspace-based control allocation of overactuated UAV platforms," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 8094–8101, Oct. 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 Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1214–1221, Apr. 2022.

[12] M. Ryll, H. H. Bülthoff, and P. R. Giordano, "A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 2, pp. 540–556, Mar. 2015.

[13] M. Zhao, K. Okada, and M. Inaba, "Enhanced modeling and control for multilinked aerial robot with two DoF force vectoring apparatus," *IEEE Robot. Autom. Lett.*, vol. 6, no. 1, pp. 135–142, Jan. 2021.

[14] M. Santos, L. Honório, A. Moreira, M. Silva, and V. Vidal, "Fast real-time control allocation applied to over-actuated quadrotor tilt-rotor," *J. Intell. Robot. Syst.*, 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 Trans. Control Syst. Technol.*, vol. 12, no. 1, pp. 211–216, Jan. 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 Trans. Mechatron.*, vol. 26, no. 5, pp. 2548–2557, Oct. 2021.

[18] S. J. Lee, I. Jang, and H. J. Kim, “Fail-safe flight of a fully-actuated quadrotor in a single motor failure,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6403–6410, Oct. 2020.

[19] W. Chung and H. Son, “Fault-tolerant control of multirotor UAVs by control variable elimination,” *IEEE/ASME Trans. Mechatron.*, vol. 25, no. 5, pp. 2513–2522, Oct. 2020.

[20] W. Zhang, M. W. Mueller, and R. D’Andrea, “Design, modeling and control of a flying vehicle with a single moving part that can be positioned anywhere in space,” *Mechatronics*, vol. 61, pp. 117–130, 2019.

[21] M. W. Mueller and R. D’Andrea, “Relaxed hover solutions for multicopters: Application to algorithmic redundancy and novel vehicles,” *Int. J. Robot. Res.*, vol. 35, no. 8, pp. 873–889, 2016.

[22] X. Shao, G. Sun, W. Yao, J. Liu, and L. Wu, “Adaptive sliding mode control for quadrotor UAVs with input saturation,” *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 3, pp. 1498–1509, Jun. 2022.

[23] G. Michieletto, M. Ryll, and A. Franchi, “Control of statically hoverable multi-rotor aerial vehicles and application to rotor-failure robustness for hexarotors,” in *Proc. Int. Conf. Robot. Automat.*, 2017, pp. 2747–2752.

[24] G.-X. Du, Q. Quan, B. Yang, and K.-Y. Cai, “Controllability analysis for multirotor helicopter rotor degradation and failure,” *J. Guidance, Control, Dyn.*, vol. 38, no. 5, pp. 978–985, 2015.

[25] A. Marks, J. F. Whidborne, and I. Yamamoto, “Control allocation for fault tolerant control of a VTOL octorotor,” in *Proc. UKACC Int. Conf. Control*, 2012, pp. 357–362.

[26] M. J. Gerber and T.-C. Tsao, “Twisting and tilting rotors for high-efficiency, thrust-vectored quadrotors,” *J. Mechanisms Robot.*, vol. 10, no. 6, 2018, Art. no. 061013.

[27] B. Li, L. Ma, D. Huang, and Y. Sun, “A flexibly assembled and maneuverable reconfigurable modular multirotor aerial vehicle,” *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 3, pp. 1704–1714, Jun. 2022.

[28] C. Ding and L. Lu, “A tilting-rotor unmanned aerial vehicle for enhanced aerial locomotion and manipulation capabilities: Design, control, and applications,” *IEEE/ASME Trans. Mechatron.*, vol. 26, no. 4, pp. 2237–2248, Aug. 2021.

[29] S. Park et al., “ODAR: Aerial manipulation platform enabling omnidirectional wrench generation,” *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 4, pp. 1907–1918, Aug. 2018.

[30] G. Michieletto, M. Ryll, and A. Franchi, “Fundamental actuation properties of multirotors: Force–moment decoupling and fail–safe robustness,” *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 702–715, Jun. 2018.

[31] D.-T. Nguyen, D. Saussie, and L. Saydy, “Fault-tolerant control of a hexacopter UAV based on self-scheduled control allocation,” in *Int. Conf. Unmanned Aircr. Syst.*, 2018, pp. 385–393.

[32] C. Pose and J. Giribet, “Multirotor fault tolerance based on center-of-mass shifting in case of rotor failure,” in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2021, pp. 38–46.

[33] M. A. da Silva Ferreira et al., “Drone reconfigurable architecture (DRA): A multipurpose modular architecture for unmanned aerial vehicles (UAVs),” *J. Intell. Robotic Syst.*, vol. 99, pp. 517–534, 2020.

[34] A. F. Şenkul and E. Altuğ, “System design of a novel tilt-roll rotor quadrotor UAV,” *J. Intell. Robot. Syst.*, vol. 84, no. 1, pp. 575–599, 2016.

[35] P. Yu, “An over-actuated multi-rotor aerial platform and iterative learning control applications,” Ph.D. thesis, Univ. California, Los Angeles, CA, USA, 2022.

[36] T. Anzai et al., “Multilinked multirotor with internal communication system for multiple objects transportation based on form optimization method,” in *Proc. Int. Conf. Intell. Robots Syst.*, 2017, pp. 5977–5984.

[37] H. Li, X. Zheng, H. He, and L. Liao, “Design and longitudinal dynamics decoupling control of a tilt-rotor aerial vehicle with high maneuverability and efficiency,” *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1191–1198, Mar. 2023.

[38] Y. Su, “Compensation and control allocation with input saturation limits and rotor faults for multi-rotor copters with redundant actuations,” Ph.D. thesis, Univ. California, Los Angeles, CA, USA, 2021.

[39] G. Li, B. Gabrich, D. Saldana, J. Das, V. Kumar, and M. Yim, “Modquad-VI: A vision-based self-assembling modular quadrotor,” in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 346–352.

[40] D. Saldana, P. M. Gupta, and V. Kumar, “Design and control of aerial modules for inflight self-disassembly,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3410–3417, Oct. 2019.

[41] B. Gabrich, G. Li, and M. Yim, “Modquad-DoF: A novel yaw actuation for modular quadrotors,” in *Proc. Int. Conf. Robot. Autom.*, 2020, pp. 8267–8273.

[42] J. Xu, D. S. D’Antonio, and D. Saldaña, “H-modquad: Modular multi-rotors with 4, 5, and 6 controllable DoF,” in *Proc. Int. Conf. Robot. Autom.*, 2021, pp. 190–196.

[43] K. Oguz and T. Oktay, “Hexarotor longitudinal flight control with deep neural network, PID algorithm and morphing,” *Avrupa Bilim ve Teknoloji Dergisi*, vol. 27, pp. 115–124, 2021.

[44] 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 Robot. Autom. Lett.*, submitted for publication.

[45] O. Kose and T. Oktay, “Simultaneous quadrotor autopilot system and collective morphing system design,” *Aircr. Eng. Aerosp. Technol.*, vol. 92, no. 7, pp. 1093–1100, 2020.

[46] 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, 2021, Art. no. 581.

[47] R. Yang, L. Zheng, J. Pan, and H. Cheng, “Learning-based predictive path following control for nonlinear systems under uncertain disturbances,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2854–2861, Apr. 2021.

[48] 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,” *Proc. Inst. Mech. Eng. Part G: J. Aerosp. Eng.*, vol. 234, no. 4, pp. 943–960, 2020.

[49] Y. Su, C. Chu, M. Wang, Y. Liu, Y. Zhu, and H. Liu, “Downwash-aware control allocation for over-actuated UAV platforms,” in *Proc. Int. Conf. Intell. Robots Syst.*, 2022, pp. 10478–10485.

[50] Y. Su et al., “Sequential manipulation planning for over-actuated unmanned aerial manipulators,” in *Proc. Int. Conf. Intell. Robots Syst. (IROS)*, 2023.

[51] Y. Su et al., “Flight structure optimization of modular reconfigurable UAVs,” in *Proc. Int. Conf. Intell. Robots Syst.*, 2023.

Yao Su (Member, IEEE) received the B.S. degree in mechanical design manufacturing and automation from the Harbin Institute of Technology, Harbin, China, in 2016, and the M.S. and Ph.D. degrees in mechanical engineering from the University of California, Los Angeles, CA, USA, in 2017 and 2021, respectively.

He is now a Research Scientist with National Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI), Beijing, China. His research interests include robotics, control, optimization, trajectory planning, and mechatronics.

Pengkang Yu received the B.Eng. degree in mechanical engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2016. He received the M.S. and Ph.D. degrees in mechanical engineering from the University of California, Los Angeles, CA, USA, in 2017 and 2022, respectively.

His research interests include control, optimization, planning, robotics, and mechatronics.

Matthew J. Gerber received the B.Eng. degree in mechanical engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2016. He received the M.S. and Ph.D. degrees in mechanical engineering from the University of California, Los Angeles, in 2017 and 2022, respectivley.

His research interests include control, optimization, planning, robotics, and mechatronics.

Lecheng Ruan received the B.S. (hons.) degree in mechatronic engineering from the Harbin Institute of Technology, Harbin, China, in 2015, and the Ph.D. degree in mechanical engineering from the University of California, Los Angeles, CA, USA, in 2020.

He is now 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) 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.