Fault-tolerant Control of Overactuated Multirotor UAV Platform under Propeller Failure

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

***Abstract*—Propeller failure is one major reason for the falling and crashing of multirotor Unmanned Aerial Vehicles (UAVs). While conventional multirotors can barely handle this issue due to underactuation, over-actuated platforms can still pursue the flight with proper fault-tolerant control (FTC). This paper investigates such a controller for one such over-actuated multirotor aerial platform composing quadcopters mounted on passive joints with input redundancy in both the high-level vehicle control and the low-level quadcopter control of vectored thrusts. To fully utilize the input redundancies of the whole platform under propeller failure, our proposed FTC controller has a hierarchical control architecture with three main components: (i) a low-level adjustment strategy to avoid propeller-level thrust saturation; (ii) a compensation loop to attenuate introduced disturbance; (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 at most two failed propellers and the whole platform can be stabilized without crashing. The proposed controller is extensively studied in both simulation and real-world experiments to demonstrate its superior performance.**

***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] through vectoring of thrusts and provide better dynamics property. There are mainly two categories of realizations in this field. The first group of works [2–4] adopt multiple propeller-motor pairs in various or varying directions to achieve full or over actuation. The second group of works [5–9] make use of regular quadcopter as mounted on passive hinges as actuation modules to simplify the design and prototyping process and reduce the internal disturbance level [9]. With more actuators onboard, these platforms must exhibit better robustness towards **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, it is of interest to develop FTC algorithms to reduce the likelihood of crashes in the event of propeller failure.

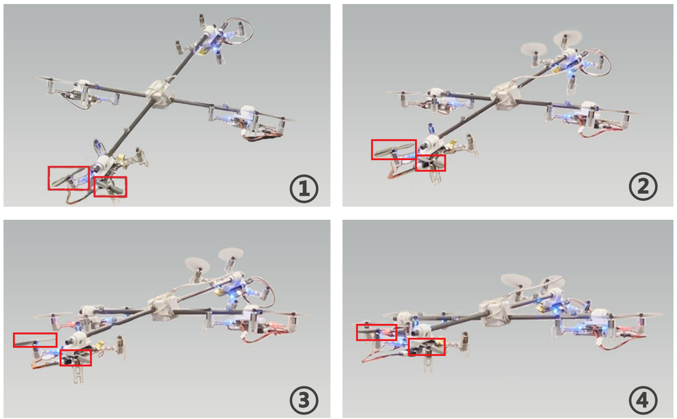


Fig. 1: **With two failed propellers on one quadcopter module, our proposed FTC algorithm can keep the platform from crashing and remain stable trajectory tracking.** (Failed propellers are labeled by red boxes).

In this paper, we first implement the previously proposed nullspace-based control allocation framework [10] to address the **thrust force saturation** issue of over-actuated UAV platforms. We verify this approach on our customized over-actuated platform [8] where four mini-quadcopters are connected to the mainframe by passive hinges as tiltable thrust generators. And it has input redundancy in both the high-level wrench control and low-level tiltable thrusts control [11]. Following this, we propose a FTC algorithm specifically for the scenario where one or more propellers on a single quadcopter (denoted as **Bad QC**) are lost, but the other three quadcopters (denoted as **Good QCs**) are still functioning properly (see Fig. 1). This FTC controller exhibits a hierarchical structure with three main components: (i) a low-level controller to adjust 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 Bad QC's propellers 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 as disturbance. To attenuate this undesirable torque, the redundant inputs of the three Good QCs are utilized to formulate

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

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).

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

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 verified in both dynamic simulations and real-world experiments.

Our contributions are highlighted in the followings:

(1) We analyze the thrust force saturation issue of over-actuated UAV platforms, and implement the nullspace-based control allocation framework to address it.

(2) We propose a FTC algorithm to fully utilize the redundancy of over-actuated UAV when some of the propellers on one quadcopter module are failed. Two different low-level control methods are analyzed and compared while a compensation loop is designed for disturbance attenuation, and nullspace-based control allocation is included to find the optimal allocation solution under this scenario.

(3) We provide simulation and experimental validations to demonstrate the effectiveness of our proposed control algorithm in handling the thrust force saturation and propeller failure of 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, the paper is concluded in Sec. VIII.

II. Related Work

A. Control Allocation

The **control allocation** of **over-actuated UAV** platforms, which computes the command for each actuator from the desired total wrench, is a constrained nonlinear optimization problem and is generally difficult to solve with high efficiency. Ryll *et al*. first utilized the dynamic output linearization to do control allocation at a higher differential level which required accurate acceleration measurements or estimation [12]. An Force Decomposition (FD)-based method was introduced by Kamel *et al*., which transformed the nonlinear allocation problem into a linear one by defining intermediate variables [2]. This method improved the computational speed by directly choosing the least-square solution, but at the expense of losing input redundancies. Furthermore, iterative approach [13] and separation method [14] were proposed for improved efficiency. However, none of these methods [2, 12–14] could include input constraints, which leads to instability when the input constraints are triggered [15].

The Quadratic Programming (QP)-based framework [16] relied on discretization and linearization to incorporate both inequality and equality constraints. But it only generated approximate solutions and introduced additional disturbance to the control system. In our previous work [10], a **nullspace-based** allocation framework was created 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 ability by solving 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 divided into several groups according to their configurations. For quadcopters, due to the underactuation, when propeller failure happens, normally the yaw motion is sacrificed to maintain full control of translation [17–22]. Much research has been conducted using this approach, such as reinforcement learning [23], fuzzy gain-scheduled PID [24], and LQR [25]. Multirotor UAV platforms with more than six controllable inputs [4, 26–28], or tilt-rotor quadcopters [29–32] are more robust against propeller loss due to **input redundancy**. The Y-shaped hexarotor platform with tilted rotors was proved to have better rotor-failure robustness compared with the standard star-shaped hexarotor platform [33]. Related FTC controllers were presented in [34] based on modifications of control allocation and in [35] based on Center-of-Mass (CoM) shifting.

As a new type of over-actuated UAV platforms that use quadcopters and passive joints to achieve full actuation [5, 6, 8], they inherently have more propellers than standard tilt-rotor platforms and thus the likelihood of propeller failure is increased. In this paper, we present an FTC for this type of UAV platform to sufficiently use the redundancy of the entire platform in both high-level and low-level control, thus improving the robustness of the platform against propeller failure. As a result, various UAV platforms [6, 13, 29, 30, 36, 37] that may have different thrust generation capabilities among propellers/thrust-generation modules under propeller failure can have better rotor-failure robustness.

III. Platform Dynamics & Property

The over-actuated multirotor platform discussed in this paper 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 $\boldsymbol{\nu} = [p, q, r]^\mathsf{T}$.

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

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

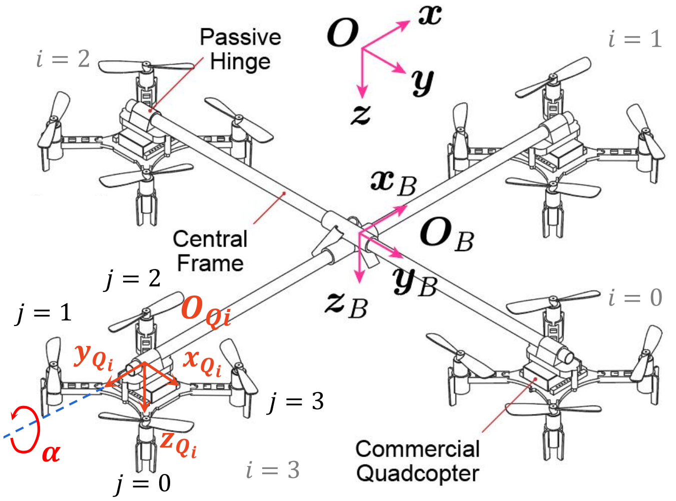


Fig. 2: Shown is the prototype used in this paper; four commercial quadcopters are passively hinged to the central frame of the platform.

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 the $\boldsymbol{y}_{Q_i}$ direction.

C. Platform Property

The proposed

As introduced in [9], this type of over-actuated UAV platform has unlimited joint angle ranges and greatly reduces the mechanical complexity, compared to existing tiltable-rotor configurations [2, 12, 30, 38, 39] where the tilting of each rotor is actuated by a servo motor. Besides, our proposed configuration can eliminate the propeller drags, gyroscopic momentums and tilting reaction torques, which had been treated as disturbances or unmodeled dynamics in other tiltable-rotor platforms, because of the paired propellers rotating in opposite directions and the zero-torque transmission in the passive hinges. Therefore, this type of platform has the best trajectory tracking performance among these configurations.

Another feature of this platform is the presence of fast auxiliary inputs M_i^x and M_i^z in the low-level control of $\mathcal{Q}_i$ (Eq. (3)). When propeller failure happens, different from other over-actuated UAV configurations that the propeller-motor pair and related servo motors have to be given up, the low-level input redundancy in our platform, offers partial thrust generation capability which could prevent the platform from crashing with more weights onboard. Besides, the auxiliary inputs can be controlled to improve control performance (e.g., as done in [11, 40]) because the dynamics of each motor are sufficiently fast and can be regarded as feed-through dynamics.

Furthermore, our proposed configuration has the potential to become a modular reconfigurable quadcopter system by equipping each quadcopter with a docking frame. Various similar designs are proposed to transform a quadcopter swarm to a connected over-actuated flying structure through the docking process [41–45]. The popularity of building flying structures with quadcopters also points out the significance of our platform and related control problems.

IV. Nominal Control

A. Hierarchical Architecture & Tracking Control

The overall controller has a hierarchical structure: (i) a high-level controller that provides the desired wrench commands for the platform to track a reference trajectory, and maps the wrench commands to the inputs of each thrust generator through control allocation (ii) a low-level controller on each quadcopter to track the desired joint angle and thrust with fast response. The stability of this controller has been proved 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:

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

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 Eq. (5) with Eq. (1), the platform dynamics is equivalent to a double integrator, and can be written in state-space form as

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

where

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} 0 & 0 & \boldsymbol{I}_3 & 0 \\ 0 & 0 & 0 & \boldsymbol{I}_3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{\mathcal{B}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \boldsymbol{I}_3 & 0 \\ 0 & \boldsymbol{I}_3 \end{bmatrix}, \tag{7}$$
$$\boldsymbol{\mathcal{X}} = \begin{bmatrix} \boldsymbol{\xi}^\top & \boldsymbol{\eta}^\top & \dot{\boldsymbol{\xi}}^\top & \boldsymbol{\nu}^\top \end{bmatrix}^\top, \quad \tilde{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u}_\xi^\top & \boldsymbol{u}_\nu^\top \end{bmatrix}^\top.$$

We design a LQI control scheme [46–48] to close the control loop with the augmented system states as

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

The cost function is

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

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

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

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

B. Nominal Allocation Framework

1) Force Decomposition-based Allocation: Given $\boldsymbol{u}^d$, we try to solve for the desired tilting angle $\boldsymbol{\alpha}$ and thrust $\boldsymbol{T}$ of the four quadcopters through the nonlinear mapping Eq. (5)—this is known as the "allocation problem". One heuristic solution uses FD to transform this nonlinear mapping problem to a linear one by defining intermediate variables [2]:

$$\boldsymbol{F} = \begin{bmatrix} F_{s0} & F_{c0} & \dots & F_{s3} & F_{c3} \end{bmatrix}^\top, \tag{11}$$

where

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

With these new variables, Eq. (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 take input constraints into consideration. Specifically, it could generate a desired thrust greater than motor saturation, leading to the instability of the platform. This is known as the thrust force saturation issue, which was investigated previously in [9], where the nominal FD-based allocation framework was shown to be insufficient to use 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. Here, we generalize this problem in a standard trajectory-tracking scenario. At each timestep, the nullspace-based allocation framework (will be introduced next) can find 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:

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

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

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

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

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

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

$\boldsymbol{P}$ and $\boldsymbol{Q}$ are weighting matrices. Eq. (17) uses first-order linearization to approximate the nonlinear equality constraint Eq. (14) where $[\cdot]_o$ is the value of a variable at the previous time step, $\Delta[\cdot]$ is the difference with respect to the previous time step of a variable, and $\boldsymbol{s}$ is a slack variable. Eqs. (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 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 solved from $\boldsymbol{F}^*$ with Eq. (15). This nullspace-based allocation framework can include input constraints while still providing an exact solution for Eq. (13). Therefore, it is more broadly applicable than the existing methods and the reader can refer to [10] for details on its implementation. We will present the FTC in Sec. V utilizing this constrained allocation framework.

D. Low-level Control

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

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

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

where $sat(\cdot)$ is the saturation function. A failed propeller means the speed and thrust is near zero and thus the quadcopter's maximum thrust and moments are changed.

V. Fault-tolerant Controller

In this section, we investigate the FTC control of this platform when some propellers are completely failed. Specifically, we focus on the scenario when one or two propellers

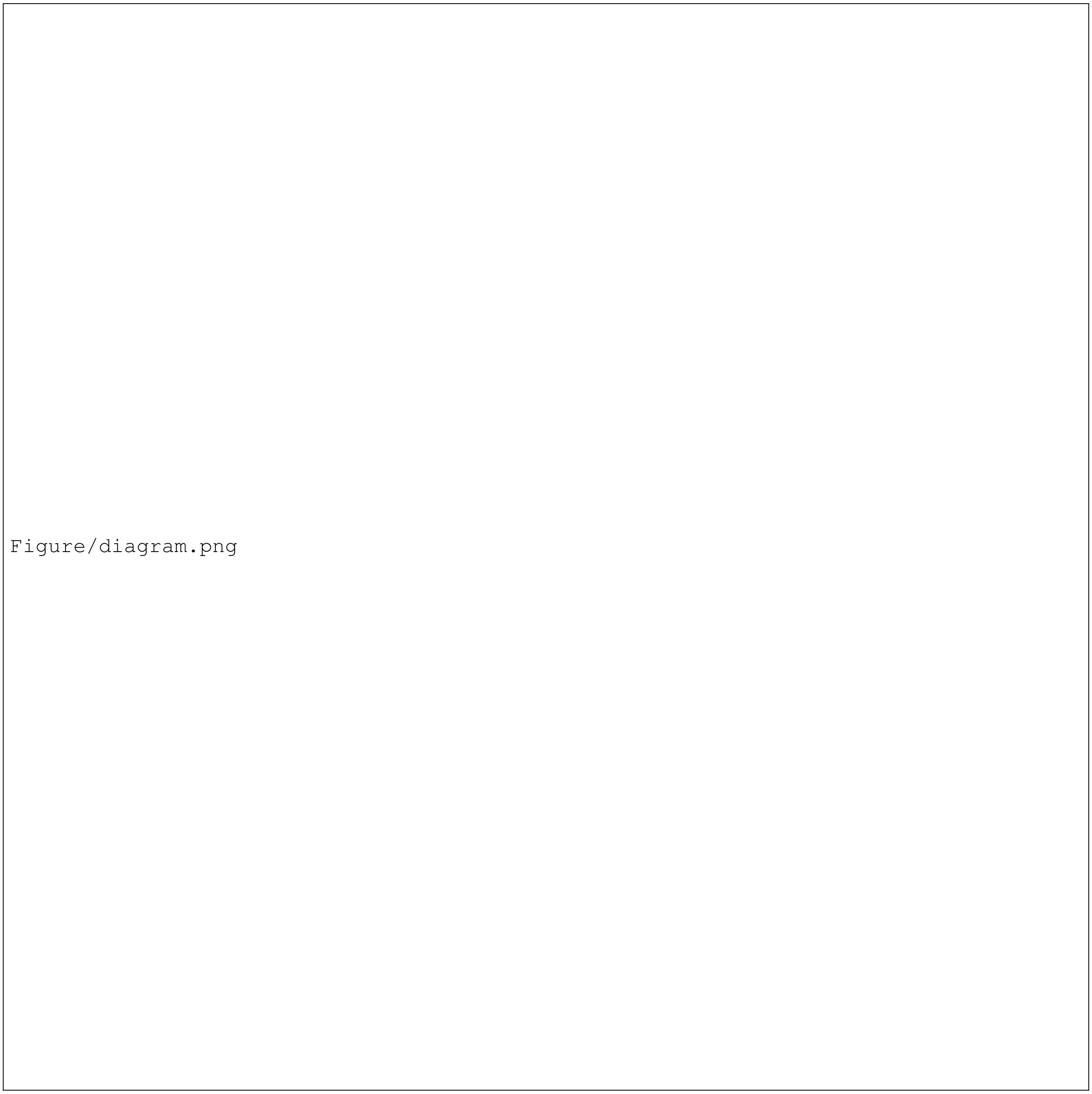


Fig. 3: **Fault-Tolerant Controller Architecture.** Each quadcopter has its own onboard failure detection and motor-control modules. When propeller failure is detected, the failure-detection module will communicate with its onboard controller to change the low-level control strategy according to the failure combinations, and sent the failure information to the high-level controller, and the nullspace-based control allocation will adjust the thrust distribution among the four individual quadcopters.

on a single quadcopter have failed, while the other three quadcopters remain functional. 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 [4, 18, 49], therefore we assume the propeller failure combination is well detected in this paper.

A. Fault-tolerant Controller Architecture

As shown in Fig. 3, the FTC controller has three main parts: (i) The low-level control on each quadcopter has two onboard functions. One performs propeller failure detection (assumed to be sufficiently fast), and the other performs quadcopter attitude and thrust control. (ii) The compensation loop is an addition to the high-level position-attitude controller and is implemented with the auxiliary inputs to improve the trajectory tracking performance and attenuate disturbance with faster response within the saturation constraints. (iii) The nullspace-based control allocation framework is utilized 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 will change its strategy according to the failure combination to maintain the control of thrust forces and tilting angle. In the meantime, the high-level control allocation can adjust thrust force distribution among the four individual quadcopters based on their different thrust generation capabilities to prevent saturation. The compensation loop will utilize the auxiliary torque inputs of Good QCs to compensate for the disturbance torques caused 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 are assumed to be functioning on $\mathcal{Q}_3$, while the propeller $\mathcal{P}_0$ is assumed to have failed. As a result, four outputs in Eq. (3) cannot be independently controlled [18]. Therefore, Eq. (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),

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

Since the mapping matrix in Eq. (24) is in full-rank, T_3, M_3^x, and M_3^y can still be controlled independently with the three remaining thrusts without considering the saturation of t_{3j}. However, in reality $t_{32} \leqslant 0$ could occur from $t_{32} = \frac{1}{2b}(M_3^x - M_3^y)$ which is obviously unreasonable. Therefore, the control of the tilting angle with M_3^y must take precedence over the toque M_3^x.

Based on the above analysis, Eq. (24) is not adopted in the low-level control. Instead, it is changed to

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

This means that the control of both M_3^x and M_3^z are lost. Although this strategy will transfer a torque disturbance to the central frame ($M_3^x \neq 0$ and $M_3^z \neq 0$), the control of α_3 can be guaranteed. The magnitude of the torque disturbances are proportional to T_3 and can be decreased by adjusting the thrust distribution across all quadcopters using the nullspace-allocation framework [40]. In addition, this torque disturbance will be compensated by the the add-on compensation loop by the other quadcopters, which will be introduced later. The comparison of using Eq. (24) and Eq. (25) in low-level control will be shown in Sec. VI.

b) High-level Control: From Eq. (25), it can be shown that T_3 will not be equally distributed on the three remaining propellers and that $\mathcal{P}_3$ must contribute half of the total required thrust. Therefore, the maximum value T_3 has to be changed from $4t_{\max}$ to $2t_{\max}$, where $t_{\max}$ is the maximum thrust of a single propeller. In other words, half of the thrust-generation capability will be lost even though only one propeller is failed on a quadcopter. Therefore, the maximum thrust vector $T_{\max}$ in Eq. (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, 12–14] cannot take input constraints, thereby necessitating the nullspace-based constrainted allocation framework.

c) Fast Compensation Loop: We use the auxiliary inputs M_i^x and M_i^z of the three Good QCs to formulate a compensation loop to eliminate the disturbance caused by the low-level control of $\mathcal{Q}_3$ [11]. The platform's complete rotational dynamics derived in [9], and with the neglect of ${}^B\boldsymbol{\nu} \times ({}^B\boldsymbol{I}\,{}^B\boldsymbol{\nu})$, is

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

where

$$\begin{aligned} \boldsymbol{J}_M^x &= \begin{bmatrix} -\cos\alpha_0 & 0 & \cos\alpha_2 & 0 \\ 0 & \cos\alpha_1 & 0 & -\cos\alpha_3 \\ \sin\alpha_0 & \sin\alpha_1 & \sin\alpha_2 & \sin\alpha_3 \end{bmatrix}, \\ \boldsymbol{M}^x &= \begin{bmatrix} M_0^x & M_1^x & M_2^x & M_3^x \end{bmatrix}^\mathsf{T}, \\ \boldsymbol{J}_M^z &= \begin{bmatrix} \sin\alpha_0 & 0 & -\sin\alpha_2 & 0 \\ 0 & -\sin\alpha_1 & 0 & \sin\alpha_3 \\ \cos\alpha_0 & \cos\alpha_1 & \cos\alpha_2 & \cos\alpha_3 \end{bmatrix}, \\ \boldsymbol{M}^z &= \begin{bmatrix} M_0^z & M_1^z & M_2^z & M_3^z \end{bmatrix}^\mathsf{T}. \end{aligned} \tag{27}$$

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

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

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

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

where $\boldsymbol{y}$ is defined as

$$\boldsymbol{y} = \begin{bmatrix} M_0^x & M_1^x & M_2^x & M_0^z & M_1^z & M_2^z \end{bmatrix}^\mathsf{T}, \tag{30}$$

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

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

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 $\mathcal{Q}_{0-2}$.

2) Two Propellers are Failed:

a) Low-level Control: In this case, we assume two propellers on $\mathcal{Q}_3$ have failed. To maintain control of T_3 and M_3^y, there must be requirements on the propeller failure combination to ensure M_3^y remains controllable. There are two cases that cannot be handled by this framework when propeller failure occurs on 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 $\mathcal{P}_1$ are failed. The low-level controller is designed as

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

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	0	✓
	1	✓
	2	✓
	3	✓
Two	0, 1	✓
	0, 2	✓
	0, 3	×
	1, 2	×
	1, 3	✓
	2, 3	✓
Three or Four		×

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

Obviously, the case of three or four propeller failures means the quadcopter loses total control. In the cases of quadcopter-level failure (marked with × in Tab. I), the platform would still be 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 our four quadcopter configuration investigated in this paper, the quadcopter opposing to the failed one would also have to be disabled and the platform control would rely on the remaining pair of the quadcopters. Specifically, the high-level control's thrust saturation limit for each quadcopter has to be changed accordingly, where the thrust limits of the lost pair of quadcopters are set to zero. Finally, when two non-opposing quadcopters lost control due to the aforementioned cases of propeller failures, the platform would fail. Thus, we have completed the treatments for every possible propeller failures.

VI. Simulation

A. Simulation Setup

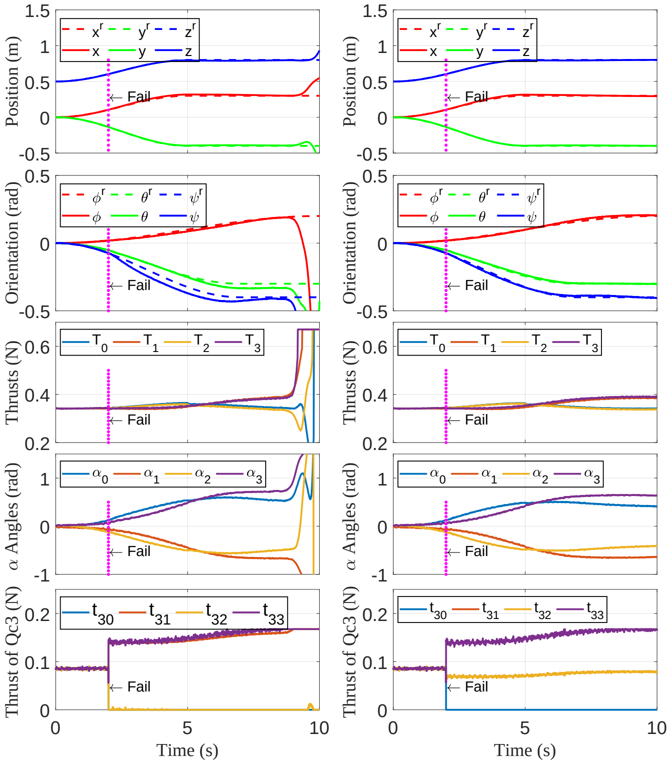
To compare the two low-level control adjustment strategies (Sec. V-B1), a dynamic simulation was built in Matlab/Simulink. The Simscape Multibody module was used to simulate the dynamics of the complete platform. All the known characteristics of the hardware were included in the simulated model, including sampling frequencies, measurement noise, communication delay, motor dynamics, and so on, the full list is shown in Table II, where m_0 and I_0 refer to the mass and inertia matrix of 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. (24) and Eq. (25)) were compared in this simulation. The nominal controller with FD-based allocation was utilized at high-level to track the reference position and attitude trajectory. For both tests, $\mathcal{P}_0$ on $\mathcal{Q}_3$ started to fail at $2s$ (Fig. 4).

TABLE II: Physical and Software Properties in Simulation

Parameter	Value	Unit
m_0	0.036	kg
m_i	0.027	kg
I_0	diag ([3 3 4.5])	kg·cm^2
I_i	diag ([0.16 0.16 0.29])	kg·cm^2
l	0.14	m
t_{max}	0.167	N
Communication delay	0.02	s



(a) The first strategy.

(b) The second strategy

Fig. 4: **Simulation: Trajectory tracking performance of two low-level adjustment strategies under propeller failure.** (a) corresponds to Eq. (24) and (b) corresponds to Eq. (25)

As we can see in Fig. 4a, with the first strategy (Eq. (24)) the desired thrust t_{32} could be non-positive value, which was then set to zero due to propeller limitation. This deteriorated the regulation of the tilting angle. In result, the high-level thrust finally saturated at approximately $9s$, and then the position and attitude control of whole platform became unstable.

For the second strategy (Eq. (25)), both position and attitude control remained stable along the whole trajectory as shown in Fig. 4b. And in the low-level control of $\mathcal{Q}_3$, all the propeller thrusts remained in the saturation range, so the tilting angle control was guaranteed.

VII. Experiments

A. Experiment Setup

The Crazyflie 2.1 was used for each individual quadcopter of the platform. To provide 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

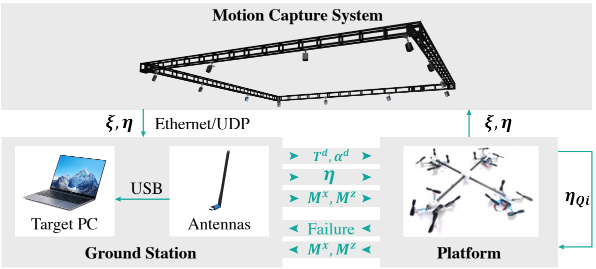


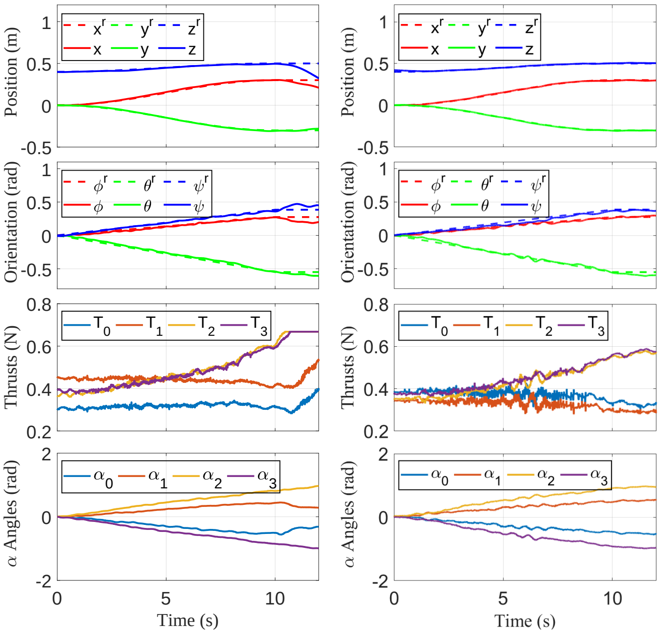
Fig. 5: **Experimental setup.** A host PC runs at 100 Hz the high-level controller of the whole platform with measurements from OptiTrack motion-capture system. Control signals are sent to each unit-copter that runs low-level controller at 500 Hz.

(including the passive hinge) of 27 g. The total mass of the entire platform was 144 g with overall dimensions of $36 \times 36 \times 6$ 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 was used as an external sensor to measure the position and attitude of the platform. The main controller was run on a ground-based computer, which communicated via Ethernet with the motion-capture system to calculate T_i, α_i, M_i^x, M_i^z for each quadcopter. These values and the attitude of the central frame were sent to each individual quadcopter via a radio-communication antenna. Each individual quadcopter was equipped with its own microprocessor, IMU, and modules for failure detection and onboard control. The combinations of propeller failure was assumed to be known with sufficiently fast failure detection module; the control module was used to adjust the low-level control strategy based on the propeller-failure combination in addition to regulating 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 handle the thrust force saturation issue. Specifically, the desired thrust commands T_2 and T_3 continued to increase and saturated at approximately $10.7s$, resulting in an unstable system. Meanwhile, T_0 and T_1 remained below 0.4 N, implying that the platform had the potential to generate sufficient thrust to remain airborne. For the nullspace-based allocation framework, the platform successfully reached the desired attitude and remained stable (Fig. 6b). As analyzed in Sec. IV-B2, this experiment further demonstrates that the FD-based framework cannot fully utilize the capability of the platform, while the nullspace-based framework shows improved performance through the inclusion of the constraints.



(a) FD-based framework. (b) Nullspace-based framework.

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

C. Propeller Failure

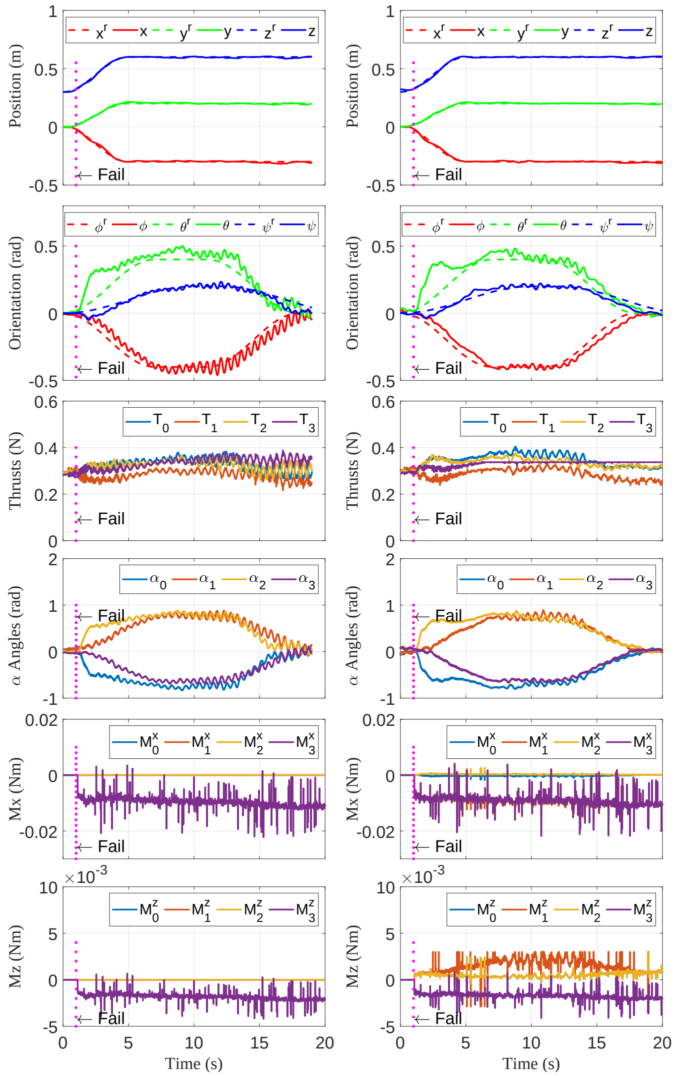
Three cases are designed to verify the effectiveness of the proposed FTC controller when one or two propellers have failed during trajectory tracking. For each case, the nominal allocation strategy with low-level adjustment (denoted as N+L controller) is compared with the FTC controller. The failure is created by setting the speed of related propellers to zero.

1) One Failed Propeller:

a) Unsaturated Trajectory: In this experiment, an unsaturated trajectory was used which satisfied $T_3 \leqslant 2\,t_{\text{max}}$ at each time step. As analyzed in Sec. V-B1, when the trajectory did not trigger low-level saturation of $\mathcal{Q}_3$, the N+L controller could maintain the stability of the platform. In such a scenario, 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.

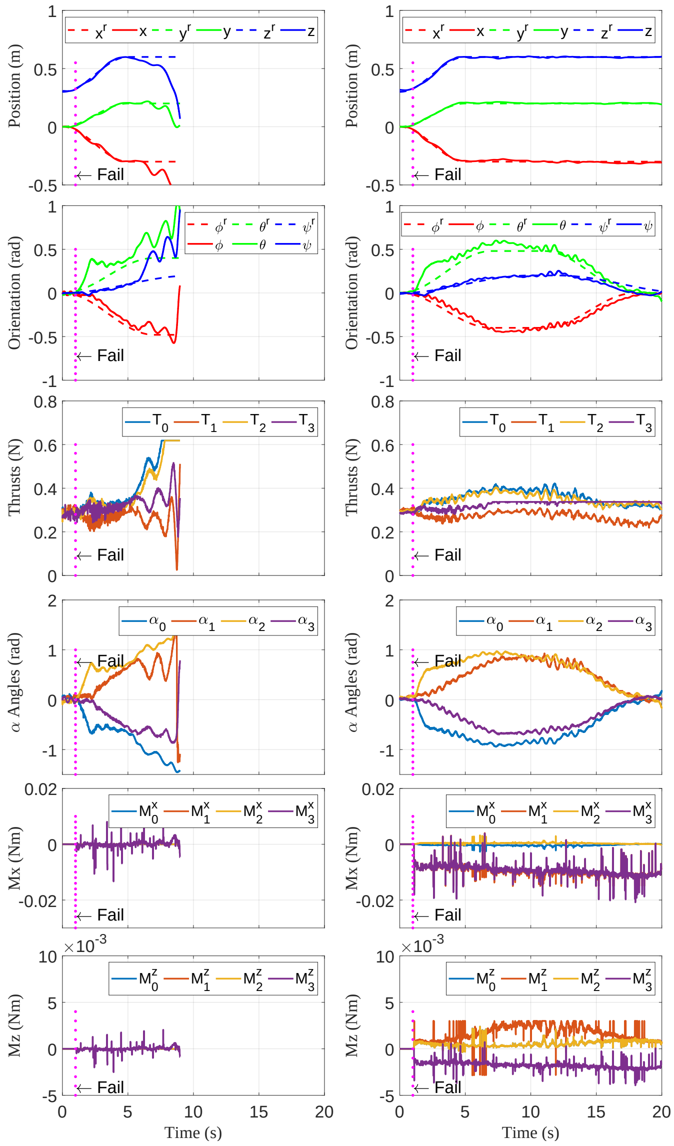
As shown in Fig. 7, both controllers could maintain the stability of the UAV platform. However, for the N+L controller (Fig. 7a), the disturbance torques M_3^x and M_3^z introduced by low-level adjustment could only be compensated by the integral action in the trajectory tracking controller, which resulted in the oscillation of the platform. For the FTC controller (Fig. 7b), since it included the compensation loop to attenuate disturbance, the tracking performance of whole platform was improved with minimal oscillation.

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



(a) One Fail (U): N+L (b) One Fail (U): FTC

Fig. 7: **Case 1: Trajectory tracking performance when one propeller is failed.** (U stands for the unsaturated trajectory that satisfies $T_3 \leqslant 2\,t_{\max}$ all the time, N+L stands for the nominal controller + low-level adjustment, FTC stands for the FTC framework we proposed. Reference attitude trajectory with $\phi_{\max} = 0.4\ rad$, $\theta_{\max} = 0.2\ rad$, $\psi_{\max} = 0.4\ rad$. Same notations are applied for the rest of this paper.)



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

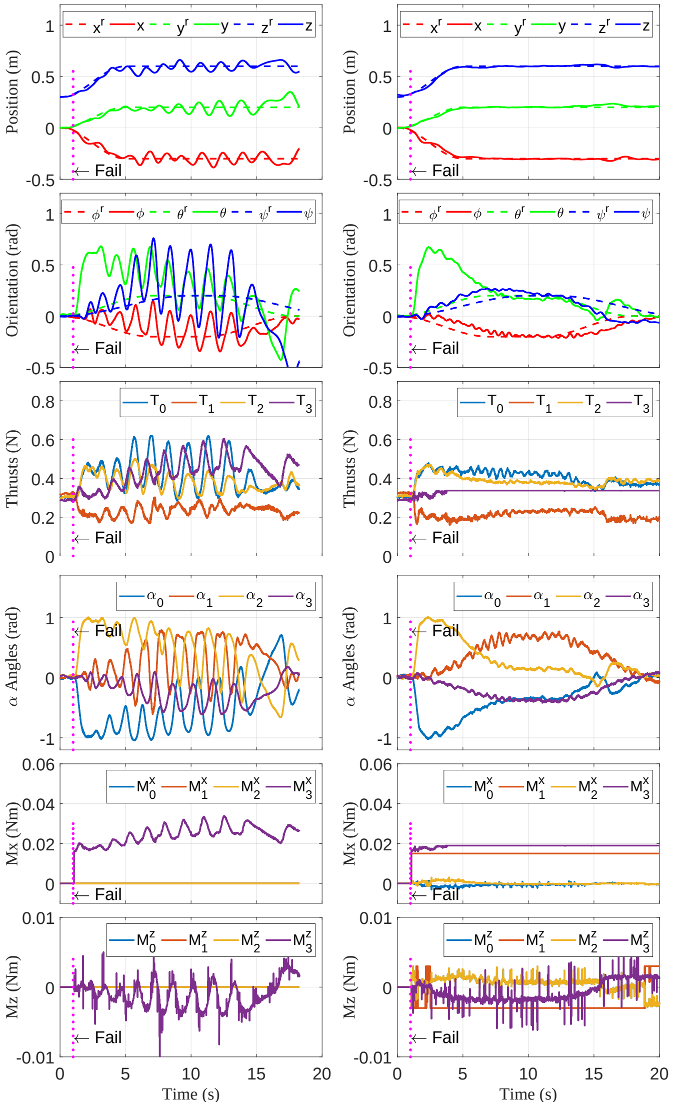
Fig. 8: **Case 2: Trajectory tracking performance when one propeller is failed.** (S stands for the saturated trajectory that $T_3 > 2\,t_{\max}$ at some time steps. Reference attitude trajectory with $\phi_{\max} = 0.5\ rad$, $\theta_{\max} = 0.2\ rad$, $\psi_{\max} = 0.4\ rad$.)

As shown in Fig. 8a, with the N+L controller, the UAV platform became unstable at approximately $7s$ when the low-level saturation constraint was triggered. Two explanations are given here. First, the nominal allocation strategy could not account for the thrust-saturation constraint and therefore output the 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 passed to the central frame without compensation.

For the FTC controller (Fig. 8b), the thrust distribution can be adjusted in the high-level control with the nullspace-based allocation strategy by setting 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. Therefore, the position and attitude control remained stable along the entire trajectory.

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 Case 2, so it would be more difficult to stabilize the platform although the reference trajectory had a relative smaller tilting attitude compared with 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). Because in this case the interacting disturbance torques created by the low-level adjustment were larger, the overall performance of both controllers was worse



(a) Two Fails: N+L

(b) Two Fails: FTC

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

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

D. Discussion

This paper proposed a fault-tolerant control strategy, which can be used on a group of over-actuated UAV platforms built on quadcopters and passive hinges.

Utilizing the hierarchical control architecture, the high-level control of our proposed configuration is the same as other over-actuated UAV configurations, therefore thrust-force saturation issue we solved in this work is a general problem that could meet by all over-actuated 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 Tab. I), the uncontrollable relative motion between the quadcopter and mainframe will introduce a huge disturbance to the platform which could be a future work for this research.

VIII. Conclusion

In this paper, we have addressed the fault-tolerant control method for over-actuated UAV platforms, where our method applies to the popular hierarchical control architecture and utilizes the nullspace-based constrained control allocation we have recently developed.

To summarize the FTC in the hierarchical control, the low-level control retains 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 by using the nullspace-based constrained allocation framework. Furthermore, the uncontrolled disturbances generated by the Bad QC are compensated by the Good QCs whenever within the saturation range of them. The simulation and experimental results for the case of one or two propeller failures respectively have demonstrated FTC's stability and superior performance in trajectory-tracking performance compared to the nominal control method.

Our FTC analysis and methods cover all possible combinations of propeller failures upon the detection of the propeller failures and they can also be applied to other platform configurations with similar low-level propeller actuators and similar setup of the maximum thrust limits in the high-level control [50–52]. In the case of aerial platforms with fixed rotor angles, the high-level control method can readily be applied with the maximum thrust of the failed propellers set to zero. Finally, for the cases where the platform faces failing inevitably, graceful crashing is desirable and is of interest for future work.

Acknowledgments

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 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 tiltable-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 octorotor," 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 tiltable-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] 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," *International Journal of Robotics Research (IJRR)*, 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 Transactions on Mechatronics (TMECH)*, vol. 27, no. 3, pp. 1498–1509, 2021.

[23] R. Arasanipalai, A. Agrawal, and D. Ghose, "Mid-flight propeller failure detection and control of propeller-deficient quadcopter using reinforcement learning," *arXiv preprint arXiv:2002.11564*, 2020.

[24] M. H. Amoozgar, A. Chamseddine, and Y. Zhang, "Fault-tolerant fuzzy gain-scheduled pid for a quadrotor helicopter testbed in the presence of actuator faults," *IFAC Proceedings Volumes*, vol. 45, no. 3, pp. 282–287, 2012.

[25] G. P. S. Rible, N. A. A. Arriola, and M. C. Ramos, "Fail-safe controller architectures for quadcopter with motor failures," in *International Conference on Control, Automation and Robotics (ICCAR)*, 2020.

[26] 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.

[27] 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.

[28] 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.

[29] M. J. Gerber and T.-C. Tsao, "Twisting and tilting rotors for high-efficiency, thrust-vectored quadrotors," *Journal of Mechanisms and Robotics*, vol. 10, no. 6, p. 061013, 2018.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] T. Anzai, M. Zhao, X. Chen, F. Shi, K. Kawasaki, K. Okada, and M. Inaba, "Multilinked multirotor with internal communication system for multiple objects transportation based on form optimization method," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[39] 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 Robotics and Automation Letters (RA-L)*, 2022.

[40] 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.

[41] G. Li, B. Gabrich, D. Saldana, J. Das, V. Kumar, and M. Yim, "Modquad-vi: A vision-based self-assembling modular quadrotor," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2019.

[42] D. Saldana, P. M. Gupta, and V. Kumar, "Design and control of aerial modules for inflight self-disassembly," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 4, pp. 3410–3417, 2019.

[43] B. Gabrich, G. Li, and M. Yim, "Modquad-dof: A novel yaw actuation for modular quadrotors," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2020.

[44] J. Xu, D. S. D'Antonio, and D. Saldaña, "H-modquad: Modular multi-rotors with 4, 5, and 6 controllable dof," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, IEEE, 2021.

[45] J. Xu, D. S. D'Antonio, and D. Saldaña, "Modular multi-rotors: From quadrotors to fully-actuated aerial vehicles," *arXiv preprint arXiv:2202.00788*, 2022.

[46] K. Oguz and T. Oktay, "Hexarotor longitudinal flight control with deep neural network, pid algorithm and morphing," *Avrupa Bilim ve Teknoloji Dergisi*, no. 27, pp. 115–124, 2021.

[47] O. Köse and T. OKTAY, "Non simultaneous morphing system desing for quadrotors," *Avrupa Bilim ve Teknoloji Dergisi*, no. 16, pp. 577–588, 2019.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.