A Fast and Efficient Attitude Control Algorithm of a Tilt-Rotor Aerial Platform Using Inputs Redundancies

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

***Abstract*—Overactuated multirotor unmanned aerial vehicles (UAVs) usually consist of multiple tiltable thrust actuators. The controllers are mostly designed by regarding the thrust forces and actuator tilting angles as inputs of outer-loop position and attitude controllers, while formulating an inner-loop controller for each actuator to track the thrust and angle as required by the outer-loop. This hierarchical control strategy separates the complicated combined dynamics into two relatively simple systems, and thus simplifies the control design. However, the interaction between the two systems is neglected and therefore the control performance will be degraded when the inner-loop dynamics are not sufficiently fast. This letter investigates the capability of a new overactuated multirotor UAV configuration, where regular quadcopters are passively hinged onto the frame as tiltable thrust actuators. Apart from the thrust force and tilting angle, each actuator has additional auxiliary torque inputs, which exhibit fast responses as they are not subject to the inner-loop actuator tilting angle dynamics. In this letter, an add-on attitude compensation control is designed exploiting the auxiliary inputs to reduce the tracking and disturbance-rejection errors from the nominal control loop. The effectiveness is demonstrated in simulation and verified by experiment.**

***Index Terms*—Overactuated multirotor, attitude control, model mismatch, add-on compensation, input redundancy, disturbance rejection.**

I. Introduction

OVERACTUATED multirotor unmanned aerial vehicles (UAVs) are suitable for exploration and interaction applications that are challenging for traditional multirotors (*e.g.* quadcopters, hexacopters), because of their advantages in decoupling the position and attitude control compared with collinear multirotor UAVs [1]. Different configurations of overactuated UAV have been proposed in recent years, such as tilt-rotor platforms [2]–[4], fixed tilting angle platforms [5], [6], modular platforms [7], [8], and passive joints platforms [9]–[11].

To generate the thrusts and torques in arbitrary directions, the mechanical complexity of these platforms is increased, making it challenging to accurately model their dynamics for controller design. The standard approach for controller design for overactuated UAV is based on model simplification, where the dynamics of the whole platform are separated into two parts: the main body which is subjected to the thrust force vectors generated by individual actuators, and low-level actuator dynamics. The dynamics separation lends itself to the hierarchical control structure [2]. In the outer-loop, desired inputs of the main body from position and attitude controller are represented by virtual force and torque commands (*e.g.* 6 Degree-of-freedom (DoF) wrench) based on feedback linearization of the nonlinear rigid body dynamics. Then the desired force and angle of every actuator are calculated by an allocation mapper that accounts for redundant actuations to meet the desired wrench [3], [12]–[15]. The desired force and angle for every actuator are tracked in the inner loop by the low-level actuator control loops. However, because the inner-loop dynamics are neglected in the outer-loop control design, the hierarchical controller introduces model mismatch into the system, which will cause additional disturbances. This uncertainty can only be compensated by adding integrators in the outer-loop trajectory-tracking controller, which reduces the transient performance and stability of the system and are not able to react to fast-changing modeling errors [16].

To solve the model mismatch problem and improve the disturbance rejection capabilities, several approaches have been proposed. An analytical method was introduced in [17] to compare the disturbance-rejection capability of different overactuated UAV platforms. From this perspective, a novel design was presented in [18] which demonstrated improved disturbance-rejection capabilities with optimal design parameters. A learning-based method was implemented in [16], [19] to compensate for the unmodeled dynamics, and an active disturbance rejection controller (ADRC) approach was used in [20], [21]. However, all of these works can only compensate for the unmodeled dynamics at the virtual wrench level, which is generated by the inner-loop control system (normally a second-order PID loop) and the response speed is generally slow.

Overactuated UAVs based on regular quadcopters and passive joints have been proposed in [10], [22] for their mechanical

Manuscript received September 9, 2021; accepted December 6, 2021. Date of publication December 28, 2021; date of current version January 5, 2022. This letter was recommended for publication by Associate Editor Santhakumar Mohan and Editor Pauline Pounds upon evaluation of the reviewers' comments. *(Yao Su, Lecheng Ruan, and Pengkang Yu contributed equally to this work.) (Corresponding author: Yao Su.)*

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

Chen-Huan Pi is with the Mechanical Engineering Department, National Yang Ming Chiao Tung University, HsinChu 300093, Taiwan (e-mail: john40532.me00@g2.nctu.edu.tw).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2021.3138806, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3138806

simplicity and inherent circumvention of disturbance torques from propeller drag and tilting reaction torques. Compared with other overactuated aerial platforms [2]–[6], this platform has eight independent auxiliary torque inputs, which can be set as zero for simplicity [23], or they can be utilized to improve control performance [24]. Furthermore, these torques are generated by the differences of the rotor speeds and as such have faster response than the torques that would normally be created via the inner-loop control of the actuator tilting angles and rotor speeds.

Exploiting the unique feature of this aerial platform, this letter extends the min-max allocation method in [23] and introduces the fast auxiliary torque inputs to compensate for disturbances and unmodeled dynamics. A Quadratic Programming (QP) problem is formulated to find the optimal auxiliary torque inputs at each time step. Three cases will be presented to demonstrate the effectiveness of this add-on compensation loop in disturbance-rejection and trajectory tracking.

The remainder of this letter is organized as follows. Section II reviews the dynamics of the platform proposed in [10]. Section III describes the nominal controller. Section IV shows the add-on compensation loop formulated by the auxiliary torque inputs. The simulation and experiment setup are shown in Section V. Both simulation and experiment results are presented in Section VI. The conclusion is addressed in Section VII.

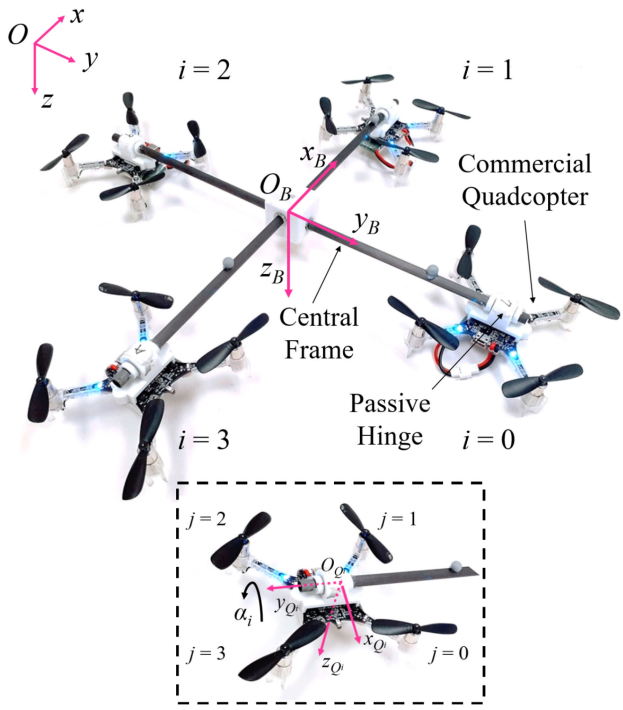


Fig. 1. **The prototype and coordination systems of the overactuated UAV platform.** Four commercial quadcopters are passively hinged on the central frame with equal distance to the center of the main frame as tiltable actuators.

II. Platform

A. Configuration

The controller designed in this letter is based on the platform proposed in [10], [22], where four regular quadcopters are mounted on the UAV central frame via passive hinges (see Fig. 1). Three coordinate systems are defined in this platform: the world frame under North-East-Down (NED) convention $\mathcal{F}_W : \{O; \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}$, the body frame attached to the platform geometric center $\mathcal{F}_B : \{O_B; \boldsymbol{x}_B, \boldsymbol{y}_B, \boldsymbol{z}_B\}$, and the quadcopter frames on each quadcopter i as $\mathcal{F}_{Q_i} : \{O_{Q_i}; \boldsymbol{x}_{Q_i}, \boldsymbol{y}_{Q_i}, \boldsymbol{z}_{Q_i}\}$. The position of the central frame center is defined as $\boldsymbol{\xi} = [x, y, z]^\mathsf{T}$, the attitude in the roll-pitch-yaw convention as $\boldsymbol{\eta} = [\phi, \theta, \psi]^\mathsf{T}$ and the platform angular velocity in $\mathcal{F}_B$ as $\boldsymbol{\nu} = [p, q, r]^\mathsf{T}$.

B. Actuator

In this platform, each quadcopter on the passive hinge is regarded as a tiltable thrust actuator. For each quadcopter, the four spinning propellers collectively generate four independent inputs

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

where T_i is the total thrust force provided by the quadcopter along $\boldsymbol{z}_{Q_i}$, M_i^x, M_i^y and M_i^z refer to the external torques in $\mathcal{F}_{Q_i}$, t_{ij} is the thrust force generated by propeller j in quadcopter i, and it is defined by

$$t_{ij} = K_T \omega_{ij}^2, \tag{2}$$

where K_T is the propeller thrust constant and ω_{ij} is the spinning speed of propeller j in quadcopter i. $b = \frac{a}{\sqrt{2}}$, $c_\tau = \frac{K_\tau}{K_T}$, a is the scalar distance of each propeller to the quadcopter center, and K_τ is the propeller drag constant.

Furthermore, the actuator tilting angle α_i can be determined through the dynamics

$$\ddot{\alpha}_i = \frac{1}{I_i^y} M_i^y - s\left(\frac{\pi}{2} i\right) \dot{p} - c\left(\frac{\pi}{2} i\right) \dot{q}, \tag{3}$$

where I_i^y is the inertia in $\boldsymbol{y}_{Q_i}$ direction and $s[\cdot]$ and $c[\cdot]$ denote $\sin[\cdot]$ and $\cos[\cdot]$, respectively.

In addition, M_i^x and M_i^z for each quadcopter can be directly exerted on the platform central frame. M_i^z is generated by propeller drag torques and usually set as zero because the magnitude is small. M_i^x is an independent auxiliary input of the actuator, and can be controlled directly because the dynamics of each motor are usually sufficiently fast and regarded as feedthrough dynamics.

C. Platform Dynamics

The platform is fully-actuated with inputs T_i, α_i and M_i^x. The translational movements can be described as

$$\ddot{\boldsymbol{\xi}} = \frac{1}{m} {}_B^W \boldsymbol{R}^\mathsf{T} \boldsymbol{J}_\xi \boldsymbol{T} + \boldsymbol{G}, \tag{4}$$

where m refers to the total mass of the platform, $\boldsymbol{G}$ is the gravitational acceleration in $\mathcal{F}_W$, ${}_B^W \boldsymbol{R}$ is the rotation matrix from

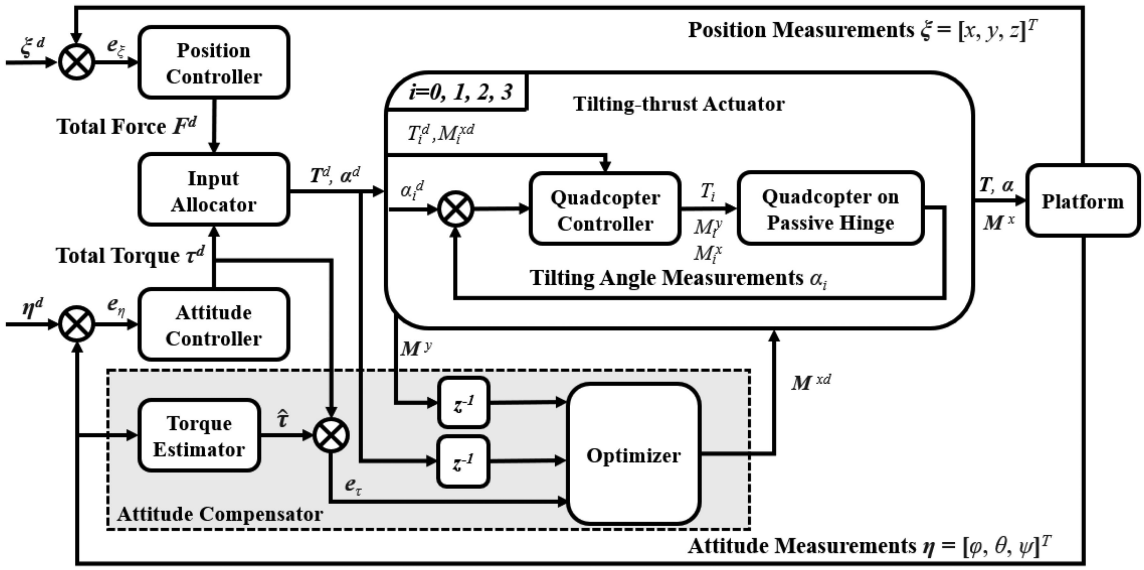


Fig. 2. **Hierarchical control structure for the overactuated UAV platform.** High-level position and attitude tracking controller outputs $\boldsymbol{u}^d$ to the allocation. This whole-body input is then allocated as the desired thrusts $\boldsymbol{T}^d$ and tilting angles $\boldsymbol{\alpha}^d$ for each quadcopter. Each quadcopter will regulate its tilting angle and thrust with onboard low-level controllers. Add-on attitude compensators are utilized to compensate for modeling errors with auxiliary inputs and thus improve attitude control performance.

$\mathcal{F}_B$ to $\mathcal{F}_W$ and

$$\boldsymbol{J}_\xi = \begin{bmatrix} s\alpha_0 & 0 & -s\alpha_2 & 0 \\ 0 & -s\alpha_1 & 0 & s\alpha_3 \\ c\alpha_0 & c\alpha_1 & c\alpha_2 & c\alpha_3 \end{bmatrix},$$

$$\boldsymbol{T} = \begin{bmatrix} T_0 & T_1 & T_2 & T_3 \end{bmatrix}^\mathsf{T}. \tag{5}$$

The rotational dynamics are

$$\dot{\boldsymbol{\nu}} = \boldsymbol{I}^{-1}(-\boldsymbol{\nu} \times (\boldsymbol{I}\boldsymbol{\nu}) + \boldsymbol{\tau}), \tag{6}$$

where $\boldsymbol{I}$ is the inertial matrix, and $\boldsymbol{\tau} \in \mathbb{R}^3$ is the total external torque exerted on the platform as

$$\boldsymbol{\tau} = \boldsymbol{\tau}_T + \boldsymbol{\tau}_M. \tag{7}$$

Here $\boldsymbol{\tau}_T$ is generated by actuator thrust forces

$$\boldsymbol{\tau}_T = \boldsymbol{J}_\nu \boldsymbol{T}, \tag{8}$$

where

$$\boldsymbol{J}_\nu = l \begin{bmatrix} -c\alpha_0 & 0 & c\alpha_2 & 0 \\ 0 & c\alpha_1 & 0 & -c\alpha_3 \\ s\alpha_0 & s\alpha_1 & s\alpha_2 & s\alpha_3 \end{bmatrix}, \tag{9}$$

and l refers to the identical distance from $\mathcal{F}_W$ to $\mathcal{F}_{Q_i}$.

$\boldsymbol{\tau}_M$ is a result of the actuator auxiliary inputs M_i^x as

$$\boldsymbol{\tau}_M = \boldsymbol{J}_M \boldsymbol{M}^x, \tag{10}$$

with

$$\boldsymbol{J}_M = \begin{bmatrix} -c\alpha_0 & 0 & c\alpha_2 & 0 \\ 0 & c\alpha_1 & 0 & -c\alpha_3 \\ s\alpha_0 & s\alpha_1 & s\alpha_2 & s\alpha_3 \end{bmatrix}, \tag{11}$$

$$\boldsymbol{M}^x = \begin{bmatrix} M_0^x & M_1^x & M_2^x & M_3^x \end{bmatrix}^\mathsf{T}. \tag{12}$$

III. Nominal Controller

A. Hierarchical Architecture

The controller of multirotor aerial platforms with tiltable thrust actuators usually follows a hierarchical architecture, as shown in the unshaded region of Fig. 2. This controller design we refer to in this letter as the "nominal controller". The controller uses four thrust forces and four tilting angles as system inputs, by setting

$$\boldsymbol{M}^x = \boldsymbol{0}, \tag{13}$$

and rewriting the dynamics as

$$\begin{bmatrix} \ddot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\nu}} \end{bmatrix} = \begin{bmatrix} \frac{1}{m}{}_B^W\boldsymbol{R} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{J}_\xi \\ \boldsymbol{J}_\nu \end{bmatrix} \boldsymbol{T} + \begin{bmatrix} \boldsymbol{G} \\ \boldsymbol{0} \end{bmatrix}. \tag{14}$$

Define two virtual inputs $\boldsymbol{u}_\xi$ and $\boldsymbol{u}_\nu$ for position and attitude, respectively. The outer-loop dynamics for six DoF can be expressed as

$$\begin{bmatrix} \ddot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\nu}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_\xi \\ \boldsymbol{u}_\nu \end{bmatrix}, \tag{15}$$

with the feedback linearized inputs

$$\boldsymbol{u}^d = \begin{bmatrix} \boldsymbol{J}_\xi \\ \boldsymbol{J}_\nu \end{bmatrix} \boldsymbol{T} = \begin{bmatrix} \boldsymbol{F}^d \\ \boldsymbol{\tau}^d \end{bmatrix} = \begin{bmatrix} m_B^W \boldsymbol{R}^\mathsf{T} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{u}_\xi \\ \boldsymbol{u}_\nu \end{bmatrix} - \begin{bmatrix} \boldsymbol{G} \\ \boldsymbol{0} \end{bmatrix} \right). \tag{16}$$

The outer-loop (15) can be closed by any stabilizing controllers. Here, we apply a LQR controller, similar to [23] to take into consideration communication delay and improve system robustness.

B. Quadcopter Control

After the required total force and torques are determined by (16), they are allocated to real system inputs T_i and α_i. Define

$$\boldsymbol{F} = \begin{bmatrix} F_{s0} & F_{c0} & \dots & F_{s3} & F_{c3} \end{bmatrix}^{\mathsf{T}}, \tag{17}$$

where

$$F_{si} = s\alpha_i T_i,\ F_{ci} = c\alpha_i T_i. \tag{18}$$

Then

$$\boldsymbol{u}^d = \begin{bmatrix} \boldsymbol{J}_\xi \\ \boldsymbol{J}_\nu \end{bmatrix} \boldsymbol{T} = \boldsymbol{W}\boldsymbol{F}, \tag{19}$$

where $\boldsymbol{W} \in \mathbb{R}^{6\times 8}$ is a constant allocation matrix with full row rank. Therefore, the original inputs can be resumed with a least-square solution [3]

$$\boldsymbol{F} = \boldsymbol{W}^{\dagger} \begin{bmatrix} \boldsymbol{J}_\xi \\ \boldsymbol{J}_\nu \end{bmatrix} \boldsymbol{T}, \tag{20}$$

and

$$\begin{aligned} T_i &= \sqrt{F_{si}^2 + F_{ci}^2}, \\ \alpha_i &= \text{atan2}(F_{si}, F_{ci}). \end{aligned} \tag{21}$$

For each quadcopter i, the thrust force T_i can be directly controlled, but the tilting angle α_i must be controlled by M_i^y under second-order dynamics (3). A double-loop PID controller is applied for tilt-angle tracking, as stated in [10]. The propeller thrusts can be reversely calculated as

$$\begin{bmatrix} t_{i0} \\ t_{i1} \\ t_{i2} \\ t_{i3} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -b & -b & b & b \\ b & -b & -b & b \\ c_\tau & -c_\tau & c_\tau & -c_\tau \end{bmatrix}^{-1} \begin{bmatrix} T_i \\ M_i^x \\ M_i^y \\ M_i^z \end{bmatrix}. \tag{22}$$

These thrusts are converted to PWM signals to drive the motors.

IV. Add-on Attitude Compensator With Auxiliary Inputs

A. Attitude Model Mismatch

As analyzed in [16], [19], there are modeling errors and unknown disturbance in reality which will influence the control performance. On this platform, modeling error has three main components: (1) There is a distance between the quadcopter center of mass (CoM) and the tilting axis. When the quadcopter is tilted at a non-zero angle, the inertia matrix of the entire platform changes as well. (2) The four quadcopters are not perfectly installed at the same height (quadcopters 1 and 3 are above quadcopters 0 and 2 with 5 mm vertical distance). (3) The low-level dynamics of regulating the tilting angle with double loop PID control are neglected in the outer-loop controller.

With model mismatch considered, the attitude dynamics becomes

$$\boldsymbol{\tau}^{real} = \boldsymbol{\tau}^d + e_{\boldsymbol{\tau}}, \tag{23}$$

where $\boldsymbol{\tau}^{real}$ is the real achieved tilting torque, the estimation of $\boldsymbol{\tau}^{real}$ is $\hat{\boldsymbol{\tau}}$, which can be acquired by

$$\hat{\boldsymbol{\tau}} = \boldsymbol{I}\frac{d\boldsymbol{\nu}}{dt}, \tag{24}$$

$\boldsymbol{\tau}^d$ is the desired tilting torque calculated by the attitude controller, e_τ is the additive modeling error, and it can be estimated by

$$e_{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}} - \boldsymbol{\tau}^d. \tag{25}$$

B. Compensation Method

As mentioned in (10), the unique dynamics of the proposed system allow for independent auxiliary inputs $\boldsymbol{M}^x$, which can be utilized to improve the control specifications. Here a separate loop is formulated, as an add-on compensator of the nominal controller, to compensate for the unmodeled dynamics, thus improving tracking accuracy and attenuating unknown disturbances. Note that the add-on loop can only compensate for the attitude controller.

The dynamics with respect to the auxiliary inputs $\boldsymbol{M}^x$ can be simplified as

$$\dot{\boldsymbol{\nu}} = \boldsymbol{I}^{-1}\boldsymbol{J}_M\boldsymbol{M}^x. \tag{26}$$

A QP problem is formulated at each time step to find the optimal auxiliary inputs $\boldsymbol{M}^x$ for modeling error compensation. Combining (25), and (26), the equality constraint is designed as

$$-e_{\boldsymbol{\tau}} = \boldsymbol{J}_M\boldsymbol{M}^x + \boldsymbol{s}, \tag{27}$$

where $\boldsymbol{s}$ is a slack variable. And the object function is

$$\boldsymbol{J}(\boldsymbol{M}^x, \boldsymbol{s}) = \boldsymbol{M}^{x\mathsf{T}}\boldsymbol{P}\boldsymbol{M}^x + \boldsymbol{s}^{\mathsf{T}}\boldsymbol{Q}\boldsymbol{s}, \tag{28}$$

where $\boldsymbol{P}$ and $\boldsymbol{Q}$ are the weighting matrices.

Saturation constraints are also included as,

$$0 \le \begin{bmatrix} 1 & 1 & 1 & 1 \\ -b & -b & b & b \\ b & -b & -b & b \\ c_\tau & -c_\tau & c_\tau & -c_\tau \end{bmatrix}^{-1} \begin{bmatrix} T_i \\ M_i^x \\ M_i^y \\ 0 \end{bmatrix} \le t_{max} \cdot \boldsymbol{1} \tag{29}$$

for $i = 0, 1, 2, 3$.

In this problem, T_i and α_i are known from the nominal controller in the previous time step, M_i^y is sent from the quadcopter onboard controller as feedback, and t_{max} refers to the maximum thrust force that can be generated by each propeller of the quadcopter. The control architecture related to this part is plotted in the gray region of Fig. 2.

C. Discussion

An additional method to utilize these four auxiliary inputs is to perform input allocation on all 12 inputs together, as

$$\boldsymbol{u}^d = \begin{bmatrix} \boldsymbol{J}_\xi \\ \boldsymbol{J}_\nu \end{bmatrix} \boldsymbol{T} + \boldsymbol{J}_M\boldsymbol{M}^x \tag{30}$$

and attempt to solve for $\boldsymbol{\alpha}$, $\boldsymbol{T}$, and $\boldsymbol{M}^x$ from the virtual input $\boldsymbol{u}^d$. However, in this case the change of variables strategy (17) and (18) is not applicable to transform the nonlinear allocation problem to a linear one because $\boldsymbol{M}^x$ is also coupled with $\boldsymbol{\alpha}$

TABLE I
PHYSICAL AND SOFTWARE PROPERTIES IN SIMULATION

Parameter	Value
m_0	$0.036\ kg$
m_i	$0.031\ kg$
I_0	$\text{diag}([3\ 3\ 4.5])\ kg \cdot cm^2$
I_i	$\text{diag}([0.16\ 0.16\ 0.29])\ kg \cdot cm^2$
l	$0.14\ m$
a	$0.032\ m$
d	$0.0144\ m$
t_{max}	$0.147\ N$
Communication delay	$0.02\ sec$
Host PC controller rate	$100\ Hz$
Onboard controller rate	$500\ Hz$

TABLE II
ACCUMULATED RMS ERRORS FOR THREE CASES

	x(mm)	y(mm)	z(mm)	roll(rad)	pitch(rad)	yaw(rad)
Case1:N (Exp)	-	-	-	-	0.173	-
Case1:Mx(Exp)	-	-	-	-	0.110	-
Case2:N (Exp)	10.8	14.4	12.8	0.012	0.045	0.011
Case2:Mx(Exp)	11.4	9.9	9.1	0.013	0.018	0.011
Case3:N (Sim)	7.4	5.0	19.0	0.013	0.066	0.026
Case3:Mx(Sim)	3.7	2.9	14.7	0.007	0.050	0.010
Case3:N (Exp)	10.3	5.1	20.5	0.017	0.110	0.011
Case3:Mx(Exp)	9.4	4.0	11.2	0.011	0.087	0.011

through the multiplication with $\boldsymbol{J}_M$. In addition, this method is still based on the simplified dynamics model to design $\boldsymbol{u}^d$. As introduced in Section I, the unmodeled dynamics are not considered in the nominal controller framework and they can only be compensated by the integral operation of the tracking controller. Therefore, even if additional inputs are included or the allocation is changed to another allocation method [13]–[15], [25], as long as the goal is to solve for the control inputs from (30), the control performance will be difficult to improve.

Based on these reasons, we decided to use the auxiliary inputs to formulate a compensation loop. Although it has a one-step delay, it will not affect the control allocation part of the main controller and the unmodeled dynamics can be directly compensated. Thus this compensation loop can work together with any existing main controller that may have different allocation strategies.

V. SIMULATION AND EXPERIMENT SETUP

A. Simulation Setup

We built a simulation in Maltab Simulink/Simscape environment to test the control performance prior to performing the experiments. This simulator includes all the characteristics of the real hardware system, such as physical parameters obtained from system identification, control frequencies, measurement noise, communication delays and noise, dynamics of propeller motors, and thrust force saturation. Table I summarizes the physical and software properties included in the simulation.

In this simulation, the dynamics of the entire platform are multi-body dynamics calculated by Simscape. Therefore, the inertia matrix of the entire platform will change when the quadcopters are tilted at different angles, although in the controller design process we assume that whole platform has a constant inertia matrix. In Table I, m_0 and I_0 refer to the mass and inertia matrix of main frame while m_i and I_i refer to the mass and inertia matrix of each quadcopter with passive hinge. It also includes the vertical distance between the quadcopters 0,2 and quadcopters 1,3. There is also a distance between quadcopter CoM and rotation axis (d refers to this distance), which has a pendulum effect to influence tilting angle regulation of each quadcopter in low-level control.

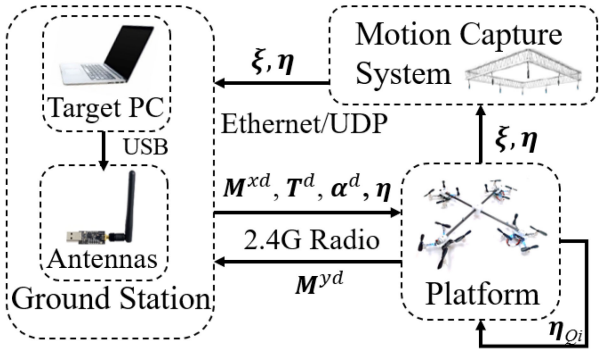


Fig. 3. **Experimental communication setup.** A ground PC runs the high-level controller at 100 Hz with measurements from an OptiTrack motion-capture system. Control signals are sent to each quadcopter, each running a low-level controller at 500 Hz.

B. Experiment Setup

The platform prototype is shown in Fig. 1. The central frame consists of two perpendicular carbon fiber tubes rigidly connected at the geometric center. The quadcopters are connected with the central frame by light-weight, 3D-printed hinges, which have no rotation-angle limitations. We use Crazyflie 2.1 as the quadcopter module. The weight of Crazyflie 2.1 is 31 g (including passive hinge) and the maximum total payload is 60 g (maximum thrust of 0.59 N). The total mass of the entire platform is 160 g.

In the experiment, we use an Optitrack motion capture system to measure the position and attitude of the central frame. The main controller runs on a ground PC, which communicates with the motion capture system through Ethernet. The main controller calculates the desired thrust $\boldsymbol{T}^d$, hinge angle $\boldsymbol{\alpha}^d$, and auxiliary torque $\boldsymbol{M}^{xd}$ for each quadcopter and receives the desired tilting torque $\boldsymbol{M}^{yd}$ as feedback. The communication between the ground PC and each quadcopter is achieved by Crazy Radio PA antennas. Each quadcopter is embedded with an onboard IMU module, and it can estimate the hinge rotation angle knowing the attitude of central frame $\boldsymbol{\eta}$. Then, the onboard controller regulates the hinge angle and thrust to the desired values. The measurement rate of the Optitrack, the ground PC controller rate, and the data communication rate with each quadcopter are all set to 100 Hz. The onboard controller of the quadcopter is set to 500 Hz to ensure fast low-level response. The software architecture is shown in Fig. 3.

VI. SIMULATION AND EXPERIMENT RESULTS

In this section, we present the performance of the add-on compensation loop in three cases: (1) A preliminary test with pitch reference trajectory tracking with the other five DoFs

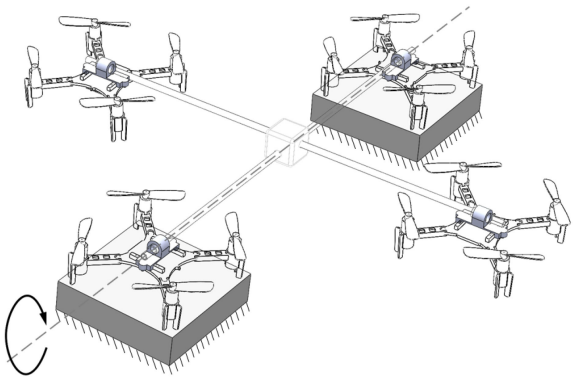
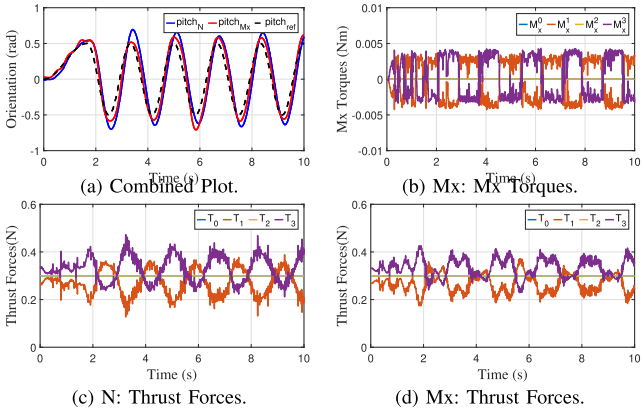
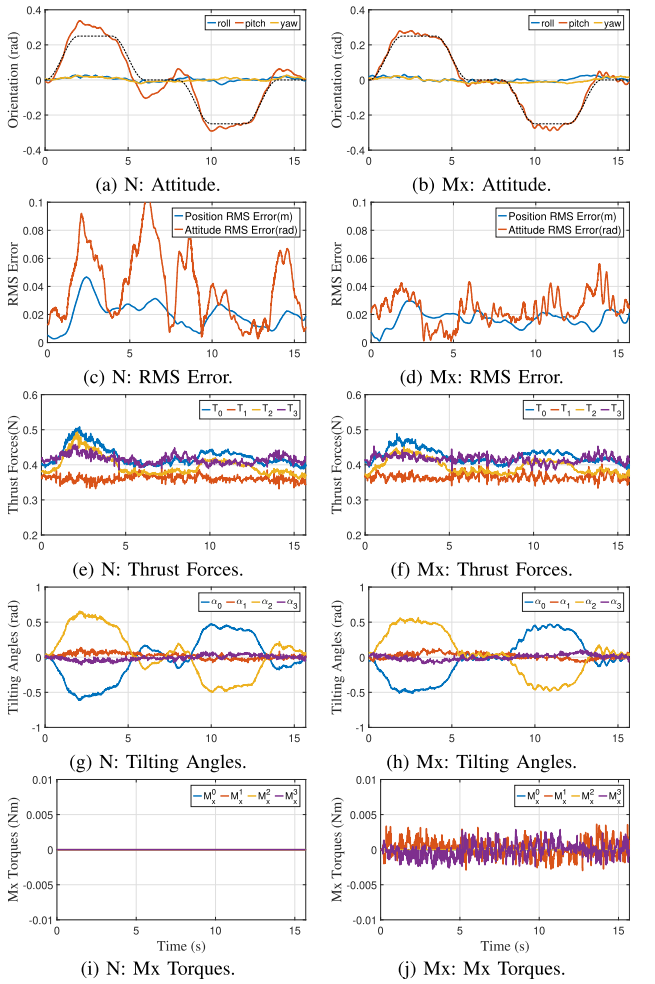


Fig. 4. **Preliminary test setup.** Five DoFs of the platform are constrained by fixing two diagonal quadcopters. The remaining one rotational DoF is realized by the passive hinges.



(a) Combined Plot. (b) Mx: Mx Torques. (c) N: Thrust Forces. (d) Mx: Thrust Forces.

Fig. 5. Case 1: Pitch trajectory tracking performance with and without the add-on attitude compensation loop while the other five DoFs are fixed. (N for nominal control, Mx for with add-on attitude compensation loop. Same notations are applied for the rest of this letter.)



(a) N: Attitude. (b) Mx: Attitude. (c) N: RMS Error. (d) Mx: RMS Error. (e) N: Thrust Forces. (f) Mx: Thrust Forces. (g) N: Tilting Angles. (h) Mx: Tilting Angles. (i) N: Mx Torques. (j) Mx: Mx Torques.

Fig. 6. Case 2: Trajectory tracking performance with and without add-on attitude compensation loop.

fixed. (2) Attitude reference trajectory tracking while hovering. (3) Disturbance rejection while hovering. For each case, the nominal controller was also tested for comparison. Simulation and experiment results are provided. Each scenario is tested multiple times, and the results are repeatable. The related errors are shown in Table II.

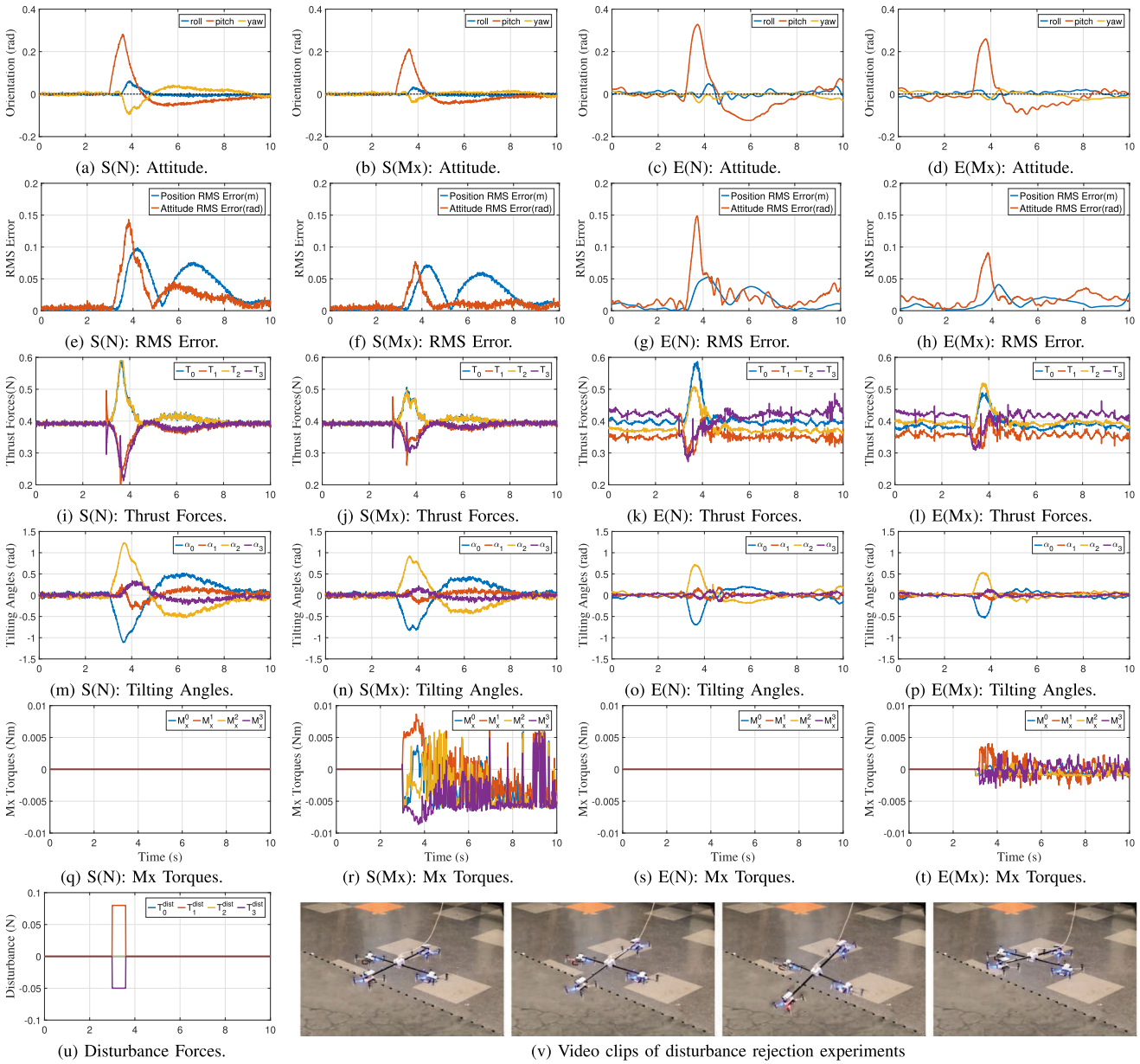
A. Preliminary Test

Before any experiment is conducted in air, a preliminary experiment is designed to prove that the add-on attitude compensation loop can improve attitude tracking performance. In this experiment, quadcopter 0 and quadcopter 2 are fixed, and the platform can only rotate around its pitch axis by the passive hinges that pass through the geometric center, which is shown in Fig. 4. The pitch angle of the platform is controlled to track a sinusoidal trajectory. In this test, only quadcopter 1 and quadcopter 3 are used and their desired tilting angles are zero. Because this case does not require any tiling angle actuation (only zero angle regulation), it should be more challenging to improve the control performance for the add-on compensator than the case that requires tiling angle actuation.

The experimental results are plotted in Fig. 5. From Fig. 5(a) it is obvious that the implementation of add-on attitude compensation loop can improve the trajectory tracking performance. The test with the add-on attitude compensation loop has smaller delay and overshot. The RMS error of the pitch angle is improved from $0.173\,rd$ to $0.110\,rd$ (a 36% improvement). As shown in Figs. 5(c) and (d) with the compensation loop, the magnitude of the thrust force can be smaller, which is helpful in avoiding the saturation of the thrust force. This preliminary test verifies the effectiveness of the add-on attitude compensation loop in a challenging scenario, and gives us enough confidence to conduct the following aerial experiments.

B. Trajectory Tracking

In this case, we compare the attitude reference trajectory tracking performance for the nominal controller with and without the add-on attitude compensation loop, and experiment results are shown in Fig. 6. In Fig. 6(a), it is obvious that the nominal controller can not track the pitch reference trajectory well due to the unmodeled dynamics. It has an obvious overshot and cannot quickly stabilize to a steady state value. After combining with the add-on attitude compensation loop, the trajectory tracking performance improves significantly. As shown in Fig. 6(b), it has a smaller overshot and shorter response time. The RMS error along the entire trajectory is improved from $0.04\,rd$ to $0.02\,rd$ (50% improvement) (Figs. 6(c) and (d)).



(a) S(N): Attitude. (b) S(Mx): Attitude. (c) E(N): Attitude. (d) E(Mx): Attitude.

(e) S(N): RMS Error. (f) S(Mx): RMS Error. (g) E(N): RMS Error. (h) E(Mx): RMS Error.

(i) S(N): Thrust Forces. (j) S(Mx): Thrust Forces. (k) E(N): Thrust Forces. (l) E(Mx): Thrust Forces.

(m) S(N): Tilting Angles. (n) S(Mx): Tilting Angles. (o) E(N): Tilting Angles. (p) E(Mx): Tilting Angles.

(q) S(N): Mx Torques. (r) S(Mx): Mx Torques. (s) E(N): Mx Torques. (t) E(Mx): Mx Torques.

(u) Disturbance Forces. (v) Video clips of disturbance rejection experiments

Fig. 7. Case 3: Disturbance rejection performance with and without the add-on attitude compensation loop. (S for simulation, E for experiment).

C. *Disturbance Rejection*

In this case, we compare the disturbance rejection performance of the nominal controller with and without the add-on attitude compensation loop while hovering. Both the simulation and the experimental results are shown in Fig. 7. The disturbance used in this experiment is manually created by adding a disturbance thrust-force signal (Fig. 7(u)) onto the desired thrust force commands $\boldsymbol{T}^d$. This disturbance signal begins at 3 s and lasts for 0.6 s.

In Figs. 7(c) and (d) the nominal controller requires larger movement to attenuate this disturbance comparing with the one with the add-on attitude compensation loop. Their position and attitude RMS errors are plotted in Figs. 7(g) and (h) which shows that with the add-on attitude compensation loop, the attitude RMS error is improved from 0.11 rd to 0.087 rd (20.6% improvement). Similar to the result of Section VI-A, the magnitude of the desired thrust forces are smaller with the add-on attitude compensation loop, as shown on Figs. 7(k) and (l). In Fig. 7(l), the thrust force command T_0 is close to the maximum value (0.59 N), meaning its margin for stabilization is small. If the disturbance is larger, this platform is likely to become unstable. But in Fig. 7(k), we notice that the maximum thrust force is only 0.52 N, implying that with the compensation loop, the platform can maintain its stability even under larger disturbance torques. We can find similar results in Figs. 7(o) and (p), where the required tilting angles can be smaller when the add-on attitude compensation loop is implemented. Some video snapshots for this experiment are shown in Fig. 7(v).

In simulation, we can also find similar results, namely that when the compensation loop is implemented, the attitude RMS error is improved from 0.07 rd to 0.05 rd (28% improvement) under the same disturbance (Figs. 7(e) and (f)), and the

magnitude of desired thrust force (Figs. 7(i) and (j)) and tilting angles (Figs. 7(m) and (n)) is smaller which will be helpful to prevent saturation.

D. Discussion

In these three experiments we have proven that with the add-on attitude compensation loop, the attitude control performance of our customized overactuated UAV platform with quadcopters and passive hinges can be dramatically improved. The reason for this improvement is twofold. First, when the add-on attitude compensation loop is implemented, each quadcopter has one additional control input in the control system, which can be used to compensate for modeling mismatch. In other words, the number of total inputs is increased from 8 to 12 by adding this compensation loop. Second, inside the add-on attitude compensation loop, the unmodeled dynamics are estimated as torque commands, which are directly compensated by the auxiliary torque inputs via control of the rotor speeds. In contrast, the nominal controller does not take the unmodeled dynamics into consideration and it can only be compensated by the integrator of a tracking controller. Furthermore, this compensation must pass through inner-loop dynamics, where the actuator angle responses are slower than the rotor speed responses with one extra relative degree of the plant dynamics. Therefore, with the add-on attitude compensation loop, the control system can achieve faster response speeds compared with the nominal controller.

VII. Conclusion

We have proposed a fast and efficient attitude control strategy for the tilt-rotor aerial platform with passive hinges and quadcopters. Compared with other tiltable actuator aerial platforms, the auxiliary torque inputs of each quadcopter are unique on this platform and they are utilized to formulate an add-on compensation loop to dynamically compensate the estimated unmodeled dynamics. This approach is shown to exhibit superior performance compared with the nominal controller where the compensation of the unmodeled dynamics purely relies on the slower response integral action of an attitude controller and the inner-loop tilting-angle control loop. The simulation and real-world experimental results from the three cases have clearly demonstrated the effectiveness of the proposed method.

Acknowledgment

The authors would like to thank Dr. Hangxin Liu, Mr. Wenzhong Yan and Dr. Ankur Mehta for the access and technical assistance in the motion capture system.

References

[1] M. Furci, C. Nainer, L. Zaccarian, and A. Franchi, "Input allocation for the propeller-based overactuated platform rospo," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2720–2727, Nov. 2020.

[2] 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.

[3] M. Kamel *et al.*, "The voliro omniorientational hexacopter: An agile and maneuverable tiltable-rotor aerial vehicle," *IEEE Robot. Automat. Mag.*, vol. 25, no. 4, pp. 34–44, Dec. 2018.

[4] 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.

[5] M. Ryll *et al.*, "6D physical interaction with a fully actuated aerial robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 5190–5195.

[6] R. Rashad, F. Califano, and S. Stramigioli, "Port-Hamiltonian passivity-based control on SE (3) of a fully actuated UAV for aerial physical interaction near-hovering," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 4378–4385, Oct. 2019.

[7] M. Zhao, T. Anzai, F. Shi, X. Chen, K. Okada, and M. Inaba, "Design, modeling, and control of an aerial robot dragon: A dual-rotor-embedded multilink robot with the ability of multi-degree-of-freedom aerial transformation," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 1176–1183, Apr. 2018.

[8] J. Xu, D. S. D'Antonio, and D. Saldaña, "H-modquad: Modular multi-rotors with 4, 5, and 6 controllable DOF," 2021, *arXiv:2106.04048*.

[9] 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.

[10] 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.

[11] 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. Automat. Lett.*, vol. 6, no. 4, pp. 6828–6835, Oct. 2021.

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

[13] Y. Su, P. Yu, M. Gerber, L. Ruan, and T.-C. Tsao, "Nullspace-based control allocation of overactuated UAV platforms," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8094–8101, Oct. 2021.

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

[15] 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. Robotic Syst.*, vol. 102, no. 3, pp. 1–20, 2021.

[16] W. Zhang, M. Brunner, L. Ott, M. Kamel, R. Siegwart, and J. Nieto, "Learning dynamics for improving control of overactuated flying systems," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5283–5290, Oct. 2020.

[17] Z. J. Chen, J. X. Bannwarth, K. A. Stol, and P. J. Richards, "Analysis of a multirotor UAV with tilted-rotors for the purposes of disturbance rejection," in *Proc. IEEE Int. Conf. Unmanned Aircraft Syst.*, 2018, pp. 864–873.

[18] Z. Chen, K. Stol, and P. Richards, "Preliminary design of multirotor UAVs with tilted-rotors for improved disturbance rejection capability," *Aerosp. Sci. Technol.*, vol. 92, pp. 635–643, 2019.

[19] 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.

[20] 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. Engineers, Part G: J. Aerosp. Eng.*, vol. 234, no. 4, pp. 943–960, 2020.

[21] Z. Wang, J. Li, and D. Duan, "Manipulation strategy of tilt quad rotor based on active disturbance rejection control," *Proc. Inst. Mech. Eng., Part G: J. Aerosp. Eng.*, vol. 234, no. 3, pp. 573–584, 2020.

[22] L. Ruan, "Independent position and attitude control on multirotor aerial platforms," Ph.D. dissertation, Univ. California, Los Angeles, 2020.

[23] L. Ruan, C. Pi, Y. Su, P. Yu, S. Cheng, and T. Tsao, "Control and experiments of a novel tiltable-rotor aerial platform comprising quadcopters and passive hinges," *Mechatronics*(submitted), Elsevier, 2021.

[24] Y. Su, "Compensation and control allocation with input saturation limits and rotor faults for multi-rotor copters with redundant actuations," Ph.D. dissertation, Univ. California, Los Angeles, 2021.

[25] M. Santos, L. Honório, A. Moreira, P. Garcia, M. Silva, and V. Vidal, "Analysis of a fast control allocation approach for nonlinear over-actuated systems," *ISA Trans.*, 2021.