


# Versatile articulated aerial robot DRAGON: Aerial manipulation and grasping by vectorable thrust control

The International Journal of  
Robotics Research  
2022, Vol. 0(0) 1–35  
© The Author(s) 2022  
Article reuse guidelines:  
[sagepub.com/journals-permissions](https://sagepub.com/journals-permissions)  
DOI: 10.1177/02783649221112446  
[journals.sagepub.com/home/ijr](https://journals.sagepub.com/home/ijr)  


Moju Zhao<sup>1</sup> , Kei Okada<sup>2</sup> and Masayuki Inaba<sup>2</sup>

## Abstract

*Various state-of-the-art works have achieved aerial manipulation and grasping by attaching additional manipulator to aerial robots. However, such a coupled platform has limitations with respect to the interaction force and mobility. In this paper, we present the successful implementation of aerial manipulation and grasping by a novel articulated aerial robot called DRAGON, in which a vectorable rotor unit is embedded in each link. The key to performing stable manipulation and grasping in the air is the usage of rotor vectoring apparatus having two degrees-of-freedom. First, a comprehensive flight control methodology for aerial transformation using the vectorable thrust force is developed with the consideration of the dynamics of vectoring actuators. This proposed control method can suppress the oscillation due to the dynamics of vectoring actuators and also allow the integration with external and internal wrenches for object manipulation and grasping. Second, an online thrust-level planning method for bimanual object grasping using the two ends of this articulated model is presented. The proposed grasping style is unique in that the vectorable thrust force is used as the internal wrench instead of the joint torque. Finally, we show the experimental results of evaluation on the proposed control and planning methods for object manipulation and grasping.*

## Keywords

Articulated aerial robot, design and control, manipulation and grasping

## 1. Introduction

During the last decade, aerial robots, particularly multirotors, have undergone considerable development (Floreato and Wood 2015; Kumar and Michael 2012). The precise maneuvering of multirotors has been achieved in various autonomous applications, such as disaster response (Michael et al., 2012), inspection (Ozaslan et al., 2017), surveillance (Doitsidis et al., 2012), and cinematography (Bonatti et al., 2020), whereas the agility enables fast passage through narrow gaps via dynamic motion in well-known environments (Falanga et al., 2017; Loianno et al., 2017; Riviere et al., 2018). Moreover, state-of-the-art studies increasingly focus on aerial manipulation (Bonyan Khamseh et al., 2018) as a means of replacing human labor at high elevations. Generally, an additional actuator, such as a gripper (Mellinger et al., 2011) or an arm manipulator (Heredia et al., 2014), is attached to the bottom of the multirotor body to enable grasping or physical interaction. However, the main challenge of such a coupled system is that the multirotor part is required to generate sufficient force and moment (wrench) to balance the translational and rotational loads that arise from the physical interaction. For instance, the reaction force on the end-effector can induce a large rotational load on the multirotor

and cause saturation in the moment control. Therefore, the number and size of the propellers must be increased to the handle reaction force. However, such a bulky platform is inefficient and also limits the mobility in confined environments.

Another platform design that can mitigate the above issues is a modular structure in which the entire body can serve as an arm manipulator (Yang et al., 2018) or gripper (Gabrich et al., 2018). A common characteristic of these aerial modular structures is that each module unit is an independent multirotor composed of more than four propellers. However, if all the propellers are unidirectional, the physical interaction that involves all degrees-of-freedom (DoF) is still impossible. Therefore, to achieve the full-pose

<sup>1</sup>Department of Mechanical Engineering, The Graduate School of Engineering, The University of Tokyo, Tokyo, Japan

<sup>2</sup>Department of Mechano-Infomatics, The Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan

## Corresponding author:

Moju Zhao, Department of Mechanical Engineering, The Graduate School of Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan.

Email: [zhou@jsk.t.u-tokyo.ac.jp](mailto:zhou@jsk.t.u-tokyo.ac.jp)

control for each module unit, a two-DoF rotor vectoring apparatus was proposed in our previous work (Zhao et al., 2018a), and an articulated aerial robot named DRAGON (Figure 1(a)) has been developed. This vectoring apparatus enables omnidirectional operation for the individual rotors (i.e., individual thrust force), and thus maximizes the efficiency of the thrust force for counteracting the gravity and external force. This enables the superior performance of DRAGON in terms of physical interaction.

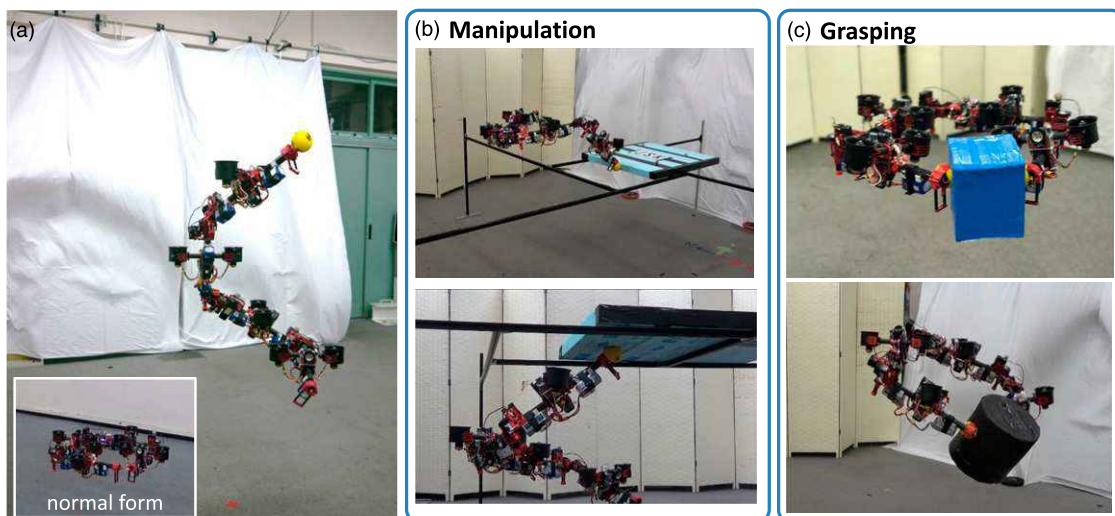
In this study, we develop a comprehensive flight control methodology for the vectorable thrust force with the aim of considering the influence of vectoring actuators' dynamics on flight stability, and further investigate the versatility of this articulated aerial structure in object manipulation and grasping as shown in Figure 1(b) and (c), which demonstrate the novel solution of blurring the line between floating- and fixed-base manipulators.

### 1.1. Related work

To achieve aerial manipulation or grasping, the design of the aerial robot is critical. Generally, a model with unidirectional propellers (e.g., quadrotor) is applied, and an actuator attached either to the bottom (Lippiello et al., 2016; Mellinger et al., 2011; Orsag et al., 2017; Yüksel et al., 2015) or to the side (Scholten et al., 2013; Alexis et al., 2016; Fumagalli et al., 2014) of the body performs the physical interaction. Given the under-actuation of the unidirectional propellers, compensating the reaction wrench using the thrust force is challenging, and most of the tasks accomplished so far only involve simple grasping or precise end-effector pose control with an  $N$ -DoF arm manipulator. Then, a fully- or over-actuated model with tilted (Ryll et al., 2019; Tognon et al., 2019; Trujillo et al., 2019) rotors are adopted to enhance the controllability against the reaction

wrench. However, if the attached arm manipulator is relatively long, the reaction wrench can still induce a large rotational load on the moment control of the multirotor. Moreover, the gravity of a long arm manipulator can also induce a large rotational load on the multirotor when the CoG of the arm is far from that of the multirotor. Although the contact force can be handled by larger propellers in greater, such a bulky platform limits mobility in confined environments. Then several omnidirectional multirotor structures are presented (Brescianini and D'Andrea 2016; Kamel et al., 2018; Park et al., 2016). The entire body can be considered as an end-effector (Bodie et al., 2019; Brescianini and D'Andrea 2018; Park et al., 2018), and thus, joint actuators for the manipulator are not required, leading to another advantage of weight-saving. Furthermore, the thrust forces generated by the rotors are distributed over the robot body, which can handle a larger rotational load compared with the coupled platform. However, a common disadvantage of such omnidirectional designs is the large wasted internal force.

Modular aerial robots can be considered as an effective solution to overcome the shortcomings of both the coupled platform and the omnidirectional aerial robot because rotors are highly distributed across all units, and the highly flexible rotor configuration can considerably suppress the internal force. A general two dimensional structure composed of unidirectional rotors and joints was developed by Zhao et al. (2018b) and Gabrich et al. (2018). Both models demonstrate the ability to grasp objects using the entire body. The main difference between these designs is that the former has an actuated joint between two neighboring units and a single rotor embedded in each unit, whereas the latter has a free joint connecting two neighboring quadrotors. Similarly, Nguyen and Alexis (2021) developed a chain of quadrotors connected by rigid rods with three-DoF joints, and attaches



**Figure 1.** Versatile articulated aerial robot DRAGON. (a) Ability to transform from the normal form in air. (b) Performing aerial manipulation to move a plate away. (c) Grasping various objects using the lateral thrust force generated from the rotor vectoring apparatus.

an additional finger-like end-effector on the rigid rod to perform manipulations such as valve-turning. However, the DoF of the end-effector pose in these types of modular structure is limited because each module unit is not omnidirectional and thus the end-effector cannot track an arbitrary pose. Then, a serial chain connecting omnidirectional aerial robots is presented by Park et al. (2019), which enables manipulation using the end of the chain (Yang et al., 2018) like a single arm manipulator. However, the root of this robot is fixed to the environment. Moreover, the omnidirectional module contains eight propellers (Park et al., 2018), which shows limited mobility in confined environments. To achieve a minimum number of rotors for the module unit, a two-DoF rotor vectoring apparatus was proposed in our previous work (Zhao et al., 2018a), which utilizes two servo motors to change the rotor direction. This vectorable apparatus can greatly benefit the articulated robot DRAGON in avoiding the internal force, which is a crucial issue in omnidirectional structured composed of tilted but fixed propellers. Another advantage of DRAGON is the ability to squeeze through small openings by deforming like a snake (Zhao et al., 2018c). The traversal can be kept slow and stable, which is safer for the exploration in unknown environments than the motions described in Loianno et al. (2017) and Falanga et al. (2017) in which the ballistic motion through vertical gaps is only available in well-known environments.

For the modeling and control of DRAGON, Zhao et al. (2021) revealed several types of singular configurations and also developed a flight control methodology to solve these singularities by using the thrust force from the rotor vectoring apparatus. In this previous work, the dynamics of actuators (especially the vectoring apparatus) were omitted. Thus, all actuators were assumed to be able to track the target position or speed instantaneously. However, this assumption can cause the rotational oscillation regarding the entire dynamics due to the non-negligible rotational inertia of the vectoring actuators, and thus renders the flight unstable. In terms of actuator dynamics, Faessler et al. (2017) proposed a control framework for general multirotor which combines the rotor dynamics model in an optimal control manner. For the vectoring apparatus, Allenspach et al. (2020) presented a control method for a special omnidirectional multirotor composed of tiltable rotors with one-DoF vectoring apparatuses. In this control method, the derivatives of the thrust force and vectoring angle are treated as the control input, which enables the involvement of actuator dynamics. Inspired from these works, we are motivated to reveal the mechanism of oscillation caused by the two-DoF thrust vectoring in our robot model, and further develop a control method to suppress the influence of the actuator's inertia on the entire dynamics by using a special control allocation strategy.

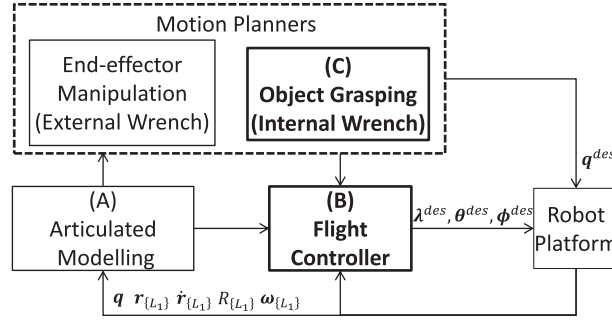
Regarding aerial grasping, general methods depend on either the joint torque (Zhao et al. 2018b; Shi et al. 2020) or the rotor drag moment (Falanga et al., 2019; Gabrich et al., 2018); however, the grasping force by these

paradigms is limited because the joint servos embedded in the aerial robots are relatively small and the rotor drag moment is also considerably weak. The special rotor vectoring apparatus in our proposed robot DRAGON can directly transit the thrust force to the grasping force, as shown in Figure 1(C), in which the rotors are tilted inward to yield sufficient lateral force to grasp objects, rather than using the joint torques. This lateral force can be considered as an internal wrench which should not influence the stability of the flight. Thus, the integration of such internal wrenches into the control flight is an important issue. Another important issue is the online thrust-level planning to generate the desired internal wrenches for grasping. In the optimization method proposed in Zhao et al. (2020) which plans the kinematic motion (e.g., joint angles) in the air, the thrust force and joint torque are considered in an equality constraint that guarantees the balance with gravity and external wrenches. However, the thrust force for grasping belongs to the null space of this equilibrium, which indicates that the equality constraint used in this previous work cannot handle such internal wrenches. Moreover, the planning targets differ in the kinematic motion and the thrust planning. Therefore, it is necessary to design a totally different optimization problem and its online solution for the thrust-level grasping planning.

In terms of object manipulation, the styles can be categorized into grasping type and graspless type (Aiyama et al., 1993) depending on the contact condition. The former type is manipulation by holding, leading to force closer during manipulation. Most of the object manipulation by aerial robots belongs to this type (Kim et al., 2015; Lippiello et al., 2016; Orsag et al., 2017). In contrast, graspless manipulation is more difficult, owing to the contact between the target object and the environment which yields the external wrench (e.g., friction). In this work, we apply the proposed control method to perform graspless manipulation on object that has more than two DoFs in motion (i.e., both the translational and rotational motion) to evaluate the integration of external wrenches in our control framework.

An overview of this work is depicted as Figure 2, and the main contributions can be summarized as follows:

1. We reveal the mechanism of the oscillation caused by the two-DoF vectoring apparatus during flight, and further develop an extended flight control method based on Zhao et al. (2021) to suppress the oscillation. This control method can also handle the integration with the external wrench for manipulation and internal wrench for grasping.
2. We present an online thrust-level planning method for bimanual object grasping by using the vectorable thrust force as internal wrench instead of joint torque. A closed-form solution to the thrust optimization problem is developed to enable realtime planning and integration with the proposed control method.



**Figure 2.** Overview of this work. The blocks with bold font are the main contributions proposed in this work. (A) kinematic and dynamic modeling for our articulated model presented in Section 2; (B) flight control method using the vectorable thrust force presented in Section 3; (C) planning method for bimanual object grasping using the vectorable thrust force as internal wrench presented in Section 4. The proposed flight control method is also able to handle the external wrench for end-effector object manipulation of which the planning method is based on our previous work (Zhao et al., 2020). The details on arrows will be explained in each section, whereas related symbols will be explained in Section 2.

3. We perform challenging flight, object manipulation, and object grasping experiments using quad-type DRAGON to demonstrate the feasibility of the proposed control and planning methods. To the best of our knowledge, this is the first time to successfully use the thrust force to the grasping motion.

## 1.2. Notation

All the symbols in this paper are explained at their first appearance. Boldface symbols (most are lowercase, e.g.,  $\mathbf{r}$ ) denote vectors, whereas non-boldface symbols (e.g.,  $m$  or  $I$ ) denote either scalars or matrices. A coordinate regarding a vector or a matrix is denoted by a left superscript, for example,  ${}^{\{A\}}\mathbf{r}$  expresses  $\mathbf{r}$  with reference to (w.r.t.) the frame  $\{A\}$ . We define  $\{W\}$  as a unique frame to represent the inertial reference frame. Then subscript are used to express: a target frame for a vector or matrix, for example,  ${}^{\{W\}}\mathbf{r}_{\{A\}}$  represents the 3D position of the frame  $\{A\}$  w.r.t  $\{W\}$ ; or/and a relation or attribute for a scalar, for example,  ${}^{\{W\}}r_{\{A\}x}$  represents the scalar position of the frame  $\{A\}$  along the  $x$  axis of the frame  $\{W\}$ . As a special case,  ${}^{\{A\}}\mathbf{v}_{\{A\}}$  refers to a vector (e.g., linear velocity, angular velocity, and linear acceleration) of the frame  $\{A\}$  w.r.t the frame  $\{W\}$  as seen by the frame  $\{A\}$ .

## 1.3. Organization

The remainder of this paper is organized as follows. The brief introduction to robot design and the modeling of kinematics and dynamics are described in Section 2. The flight control method using the two-DoF vectoring thrust force is presented in Section 3. Then, the online thrust-level planning object grasping using the vectorable thrust in Section 4. Finally, we show the experimental results using our quad-type DRAGON in Section 5 involving the flight in challenging configuration, along with object manipulation and grasping. The conclusion is presented in Section 6.

## 2. Design, modeling

In this section, we first explain the basic mechanical design and the kinematics model for the articulated robot DRAGON, and then derive the vectorable thrust model associated with vectoring angles. The method to quantify the controllability of the vectorable thrust force is also presented. We further present the dynamics model for actuator, which is followed by the approximation of the multilinked dynamics model under a quasi-static assumption on the joint motion.

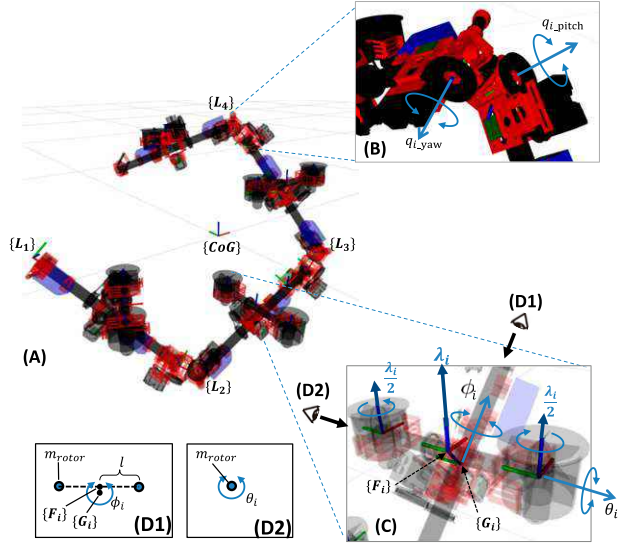
### 2.1. Mechanical design and vectorable thrust model

The kinematics model of DRAGON is depicted in Figure 3. As shown in Figure 3(B), each joint module is composed from two orthogonal joint axes actuated by independent servo motors, whereas a two-DoF force vectoring apparatus is embedded in each link module that contains two rotating axes as shown in Figure 3(C).  $\phi$  and  $\theta$  denote the first roll axis and the second pitch axis, respectively. Besides, a dual-rotor module is developed to counteract the drag moment and gyroscopic moment. It is notable that we count the pair of rotors as an integrated rotor that generates a combined uni-directional thrust  $\lambda \in [0, \lambda_{\max}]$  for each link module.

Based on this kinematic model as shown in Figure 3, the force  ${}^{\{CoG\}}\mathbf{f}_i$  and torque  ${}^{\{CoG\}}\boldsymbol{\tau}_i$  related to the  $i$ -th rotor module can be written as

$$\begin{aligned} {}^{\{CoG\}}\mathbf{f}_i &= \lambda_i^{\{CoG\}} R_{\{L_i\}}(\mathbf{q}) {}^{\{L_i\}} R_{\{G_{i-roll}\}}(\phi_i) {}^{\{G_i\}} R_{\{F_i\}}(\theta_i) \mathbf{b}_3, \\ &= \lambda_i^{\{CoG\}} \mathbf{u}_i \end{aligned} \quad (1)$$

$$\begin{aligned} {}^{\{CoG\}}\boldsymbol{\tau}_i &= \lambda_i^{\{CoG\}} \mathbf{p}_{\{F_i\}}(\mathbf{q}, \phi_i) \times {}^{\{CoG\}} \mathbf{u}_i, \\ &= \lambda_i^{\{CoG\}} \mathbf{v}_i \end{aligned} \quad (2)$$



**Figure 3.** (A) Kinematics model of the aerial robot DRAGON,  $\{L_i\}$  is a frame attached to the start point of the  $i$ -th link, and the  $x$  axis is aligned with the direction of link rod. (B) two DoF joint module composed of two orthogonal joint axes ( $q_{i\_yaw}$ ,  $q_{i\_pitch}$ ); (C) two DoF force vectoring apparatus ( $\theta_i$ ,  $\phi_i$ ).  $\{G_i\}$  is a frame attached to the origin of vectoring apparatus, and the  $x$  axis is aligned with the  $x$  axis of  $\{L_i\}$  and rotates around it with  $\phi_i$ .  $\{F_i\}$  is a frame in the middle of the dual rotors, and the  $z$  axis, where the combining the thrust force  $\lambda_i$  is attached, is parallel to the rotor rotation axis and is tilted from the  $z$  axis of  $\{G_i\}$  with  $\theta_i$ ; (D1)/(D2) schematic diagrams of the vectoring actuator's inertia.

where  $\mathbf{b}_3 = [0 \ 0 \ 1]^T$ , and  ${}^{\{A\}}R_{\{B\}}$  denotes a rotation matrix of the frame  $\{A\}$  w.r.t. the frame  $\{B\}$ . Definition of the frame  $\{L_i\}$ ,  $\{G_i\}$  and  $\{F_i\}$  can be found in Figure 3.  $\mathbf{p}_{\{F_i\}}^{\{CoG\}}$  in equation (2) is the position of the frame  $\{F_i\}$ , which depends on the joint angles  $\mathbf{q} \in \mathcal{R}^{N_j}$  ( $N_j$ : number of joints in the whole model) as well as the vectoring roll angle  $\phi_i$ , because there is an offset from  $\{G_i\}$  to  $\{F_i\}$  as shown in Figure 3(c).

Then the total wrench in the CoG frame can be given by

$$\begin{bmatrix} \{CoG\} \mathbf{f} \\ \{CoG\} \boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N_r} \{CoG\} \mathbf{f}_i \\ \sum_{i=1}^{N_r} \{CoG\} \boldsymbol{\tau}_i \end{bmatrix} = \mathbf{Q} \boldsymbol{\lambda} \quad (3)$$

$$\mathbf{Q} = \begin{bmatrix} \{CoG\} \mathbf{u}_1 & \{CoG\} \mathbf{u}_2 & \dots & \{CoG\} \mathbf{u}_{N_r} \\ \{CoG\} \mathbf{v}_1 & \{CoG\} \mathbf{v}_2 & \dots & \{CoG\} \mathbf{v}_{N_r} \end{bmatrix} \quad (4)$$

$$\boldsymbol{\lambda} = [\lambda_1 \ \lambda_2 \ \dots \ \lambda_{N_r}]^T$$

where  $N_r$  is the number of rotors.  $\{CoG\} \mathbf{u}_i$ ,  $\{CoG\} \mathbf{v}_i$ , and  $\lambda_i$  correspond to equations (1) and (2).

## 2.2. Controllability of vectorable thrust force

To quantify the controllability of two-DoF vectoring apparatus, the feasible control volume (Park et al., 2018) is introduced for our robot model.

**2.2.1. Feasible control force polygon.** The vectoring angles have a smaller influence on the translational motion along the  $z$  axis than that along other axes because all rotors must be vertical to counteract gravity during flight. Then, to quantify the controllability of the  $x$  and  $y$  axes, a normalized feasible control force polygon  $\mathcal{V}_F$  is introduced to address the nonlinearity due to the rotation matrix operation in equation (1)

$$\mathcal{V}_F := \left\{ \mathbf{f}_{xy} \in \mathcal{R}^2 \mid \mathbf{f}_{xy} = \sum_{i=1}^{N_r} \sum_{j=1}^{N_F} \gamma_{ij} \mathbf{u}_{xyij}, \ -1 \leq \gamma_{ij} \leq 1 \right\} \quad (5)$$

where  $\mathbf{u}_{xyij}$  denotes the special unit vector, which has two types for two-DoF vectoring rotor ( $N_F = 2$ ):  $\mathbf{u}_{xyi1} = [1, 0]^T$  and  $\mathbf{u}_{xyi2} = [0, 1]^T$ .  $\gamma_{ij}$  denotes the normalized range for  $\mathbf{u}_{xyij}$ , which takes a value in  $[-1 \ 1]$  corresponding to the full range of a vectoring angle in  $[-\pi \ \pi]$ .

For convenience,  $\mathbf{u}_{xyij}$  is rearranged in a combined set  $\mathcal{I}_F := \{1, 2, \dots, N_r N_F\}$ , and the distance from the origin to each edge of the polygon can be written using the rearranged unit vector  $\mathbf{u}_{xyi}$  as follows

$$d_i^{f_{xy}} = \sum_{j \in \mathcal{I}_F} \|\mathbf{u}_{xyi} \times \mathbf{u}_{xyj}\|, \quad i \in \mathcal{I}_F \quad (6)$$

where  $\cdot \times \cdot$  here denotes the cross product of the 2D vectors.

Finally, we introduce the guaranteed minimum control force  $f_{\min}$ , which corresponds to an inscribed circle of  $\mathcal{V}_F$

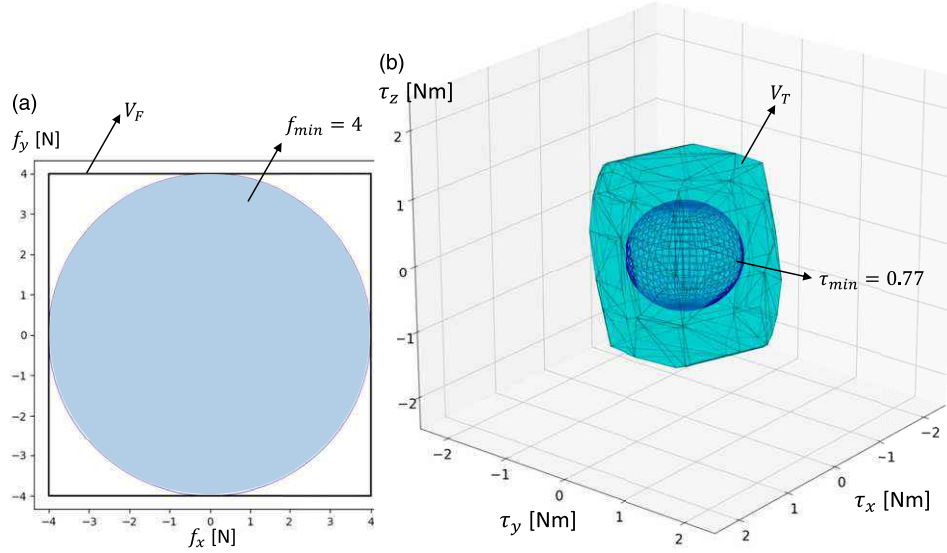
$$f_{\min} = \min d_i^{f_{xy}} \quad i \in \mathcal{I}_F \quad (7)$$

Then,  $f_{\min}$  can be used as a criterion to evaluate the controllability of the force for the arbitrary robot pose. Figure 4(a) shows the feasible control force polygon and circle of the guaranteed minimum control force. It is notable that  $\mathcal{V}_F$  would be always a square regardless of the robot pose, and thus  $f_{\min}(\mathbf{q}, R_{\{L_1\}}) = N_r$  is always established. However, if the vectoring angles  $\phi$  are fixed, the  $\mathcal{V}_F$  will shrink significantly, and  $f_{\min}$  will also decrease greatly. This truncated operation is used in our flight control method, which is presented in Section 3.

**2.2.2. Feasible control torque convex.** Similar to the feasible control force polygon, a normalized feasible control torque convex  $\mathcal{V}_T$  can be given by

$$\mathcal{V}_T := \left\{ \boldsymbol{\tau} \in \mathcal{R}^3 \mid \boldsymbol{\tau} = \sum_{i=1}^{N_r} \sum_{j=1}^{N_T} \gamma_{ij} \mathbf{v}'_{ij}, \ -1 \leq \gamma_{ij} \leq 1 \right\} \quad (8)$$

where  $\mathbf{v}'_{ij}$  has three types for the two-DoF vectoring rotor ( $N_T = 3$ ):  $\mathbf{v}'_{i1} = \mathbf{p}_{\{F_i\}} \times [1, 0, 0]^T$ ,  $\mathbf{v}'_{i2} = \mathbf{p}_{\{F_i\}} \times [0, 1, 0]^T$  and  $\mathbf{v}'_{i3} = \mathbf{p}_{\{F_i\}} \times [0, 0, 1]^T$ , which are similar to equation (2); however,  $\mathbf{u}_i$  is replaced by the three orthogonal vectors above to resolve the nonlinearity due to the rotation matrix operation in equation (2), and  $\gamma_{ij}$  is the normalized range for  $\mathbf{v}'_{ij}$ . Again, we rearrange  $\mathbf{v}'_{ij}$  in another combined set



**Figure 4.** Example of feasible control volume with the robot configuration of Figure 3(A). (a) feasible control force polygon and minimum control circle in 2D. (b) feasible control torque volume and the minimum control sphere in 3D.

$\mathcal{I}_T := \{1, 2, \dots, N_r N_T\}$  which yields  $\mathbf{v}'_i$ . Then the distance from the origin to the plane of convex  $\mathcal{V}_T$  along a normal vector  $\mathbf{v}'_i \times \mathbf{v}'_j$  and the guaranteed minimum control torque can be written as

$$d_{ij}^r = \sum_{k \in \mathcal{I}_T} \left\| \frac{(\mathbf{v}'_i \times \mathbf{v}'_j)^\top}{\|\mathbf{v}'_i \times \mathbf{v}'_j\|} \mathbf{v}'_k \right\|, \quad i, j \in \mathcal{I}_T \quad (9)$$

$$\tau_{\min} = \min d_{ij}^r, \quad i, j \in \mathcal{I}_T \quad (10)$$

Figure 4(b) shows the feasible control torque convex and the guaranteed minimum control torque sphere for the robot pose in Figure 3(A). As discussed in the feasible control force polygon, if the vectoring angles  $\boldsymbol{\phi}$  are fixed, the guaranteed minimum control torque  $\tau_{\min}$  will decrease significantly, which is a criterion to determine singularity in the flight control method presented in Section 3.

### 2.3. Dynamics of actuators

**2.3.1. Rotor.** The actual control command to the rotor is the rotor speed, and the rotor dynamics can be approximated as following first-order damped system as proposed in Faessler et al. (2017)

$$\lambda_i = c_F \Omega_i^2 \quad (11)$$

$$\dot{\Omega}_i = \frac{1}{T_\Omega} (\Omega_i^{des} - \Omega_i) \quad (12)$$

where  $\Omega_i$  and  $\Omega_i^{des}$  are the actual and desired rotor speeds, and  $c_F$  and  $T_\Omega$  are the speed-force ratios and the time constant of rotor motion. In our model, the inertia of the rotor is sufficiently small due to the small ducted design. Thus,  $T_\Omega$  can be considered sufficiently small ( $T_\Omega \approx 0$ ). This

leads to the further approximation:  $\lambda_i \approx \lambda_i^{des}$ , which allows us to use  $\lambda_i$  directly as control input.

**2.3.2. Vectoring apparatus.** The vectoring apparatus are composed from two servo motors which perform position control to track the desired vectoring angles  $\phi_i$  and  $\theta_i$ , respectively.

To properly reflect the influence of actuator inertia on the dynamics, we use a second-order system to model a servo motor with a general PD control

$$I_\phi \ddot{\phi}_i = \tau_\phi = I_\phi (k_{\phi,p} (\phi_i^{des} - \phi_i) - k_{\phi,d} \dot{\phi}_i) \quad (13)$$

$$I_\theta \ddot{\theta}_i = \tau_\theta = I_\theta (k_{\theta,p} (\theta_i^{des} - \theta_i) - k_{\theta,d} \dot{\theta}_i) \quad (14)$$

where  $I_\phi$ ,  $I_\theta$  are the inertial moments around the  $\phi$  and  $\theta$  rotating angle, respectively.  $\phi_i^{des}$ ,  $\theta_i^{des}$  are the desired vectoring angles.  $k_{*,p}$ ,  $k_{*,d}$  are the P and D control gains for each servo motor.

As shown in Figure 3 (D1) and (D2), the CoG of the two rotors, are aligned along the vectoring axis of  $\theta_i$ . Thus, the inertial moment can be considered significantly small (i.e.,  $I_\theta \approx 0$ ). In contrast, since the CoG of the two rotors are at opposite sides of the vectoring axis of  $\phi_i$ , the inertia moment around  $\phi_i$  can be approximated as

$$I_\phi = 2m_{\text{rotor}} l^2 \quad (15)$$

where  $m_{\text{rotor}}$  is the rotor module mass and  $l$  is the distance from the rotating axis to the rotor as shown in Figure 3 (D1).

In our previous work (Zhao et al., 2021), the influence of inertial moment  $I_\phi$  is also considered significantly small. However, we found the inertial moment can affect the flight rotational control significantly. Thus in this work, we

provide a comprehensive analysis on the influence of inertial moment  $I_\phi$  on the dynamics of the robot body in Section 2.4, and also reveal the mechanism of oscillation due to this inertial moment in Section 3.1.

#### 2.4. Dynamics of multilinked model

The dynamic model of the proposed multilinked robot w.r.t. the entire CoG frame  $\{CoG\}$  can be written as follows

$$\{{}^W\}\dot{\mathbf{P}}_\Sigma = \{{}^W\}R_{\{CoG\}}\{{}^{CoG}\}\mathbf{f} - m_\Sigma \mathbf{g} + \sum_{i=1}^{N_{ex}} \{{}^W\}\mathbf{f}_{ex_i} \quad (16)$$

$$\{{}^{CoG}\}\dot{\mathcal{L}}_\Sigma = \{{}^{CoG}\}\boldsymbol{\tau} + \sum_{i=1}^{N_{ex}} (\{{}^{CoG}\}\boldsymbol{\tau}_{ex_i} + \{{}^{CoG}\}\mathbf{p}_{ex_i}(\mathbf{q}) \times \{{}^{CoG}\}\mathbf{f}_{ex_i}) \quad (17)$$

$$M_J(\mathbf{q})\ddot{\mathbf{q}} + c(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau}_q + \sum_{i=1}^{N_{ex}} J_{ex_i}^T \begin{bmatrix} \mathbf{f}_{ex_i} \\ \boldsymbol{\tau}_{ex_i} \end{bmatrix} + \sum_{i=1}^{N_r} J_{r_i}^T \mathbf{f}_i + \sum_{i=1}^{N_s} J_{s_i}^T m_{s_i} \mathbf{g} \quad (18)$$

where the first equation denotes the dynamic motion of entire linear momentum which is described in the inertial frame  $\{W\}$ , whereas the second equation denotes the dynamic motion of the entire rotational momentum which is described in the CoG frame of the entire multibody model (i.e.,  $\{CoG\}$ ). The third equation corresponds to the joint motion.  $\{{}^W\}P_\Sigma$  and  $\{{}^{CoG}\}\mathcal{L}_\Sigma$  are the total linear and angular momentum, respectively, which are both affected by the joint angles, vectoring angles, and their velocities (i.e.,  $\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\phi}, \dot{\boldsymbol{\phi}}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}$ ). We refer the reader to [Appendix A](#), where we provide the detailed derivation of these momentums.  $\{{}^{CoG}\}\mathbf{f}$  and  $\{{}^{CoG}\}\boldsymbol{\tau}$  are the total wrench described in (3).  $\{{}^W\}\mathbf{f}_{ex_i}$  and  $\{{}^W\}\boldsymbol{\tau}_{ex_i}$  are the external force and torque w.r.t. the world frame  $\{W\}$ , and  $N_{ex}$  is the number of external wrenches.  $\mathbf{g}$  is a three-dimensional vector expressing gravity.  $\{{}^{CoG}\}\mathbf{p}_{ex_i}(\mathbf{q})$  is the position of the acting point of the  $i$ -th external wrench w.r.t. the robot CoG frame and is determined by the joint angles  $\mathbf{q}$ . In the last equation (18),  $M_J(\mathbf{q})$  denotes the inertial matrix, whereas  $c(\mathbf{q}, \dot{\mathbf{q}})$  is the term related to the centrifugal and Coriolis forces in joint motion. The symbols “r” and “s” stand for “rotor” and “segment” in multilinks. It is notable that there is no explicit reference frame described in equation (18) and the frame of the root link  $\{L_1\}$  is generally used.  $J_{ex_i} \in \mathcal{R}^{6 \times N_J}$ ,  $J_{r_i} \in \mathcal{R}^{3 \times N_J}$ , and  $J_{s_i} \in \mathcal{R}^{3 \times N_J}$  are the Jacobian matrices for the frames of the  $i$ -th interaction point, the  $i$ -th rotor, and the  $i$ -th segment’s CoG, respectively.  $\boldsymbol{\tau}_q \in \mathcal{R}^{N_J}$  is the vector of joint torque and  $\mathbf{f}_i$  is the vectoring thrust force corresponding to equation (1).

The entire dynamics model as shown in equations (16)–(18) is highly complex owing to the joint motion and the dynamics of vectoring actuators. Thus, the realtime feedback control based on such a nonlinear model is significantly difficult with an onboard computational

resource. For the joint motion, a crucial assumption is proposed in our work to simplify the dynamics, that is, all the joints are actuated slowly by servos motors ( $\dot{\mathbf{q}} \approx \boldsymbol{\theta}; \ddot{\mathbf{q}} \approx \boldsymbol{\theta}$ ). Then, the joint velocity and acceleration can be considered zero during the joint motion. Under these assumptions, the original dynamic model can be approximated as follows

$$m_\Sigma \{{}^W\}\ddot{\mathbf{r}}_{\{CoG\}}(\mathbf{q}) = \{{}^W\}R_{\{CoG\}}\{{}^{CoG}\}\mathbf{f} - m_\Sigma \mathbf{g} + \sum_{i=1}^{N_{ex}} \{{}^W\}\mathbf{f}_{ex_i} \quad (19)$$

$$\begin{aligned} \{{}^{CoG}\}I_\Sigma(\mathbf{q})\{{}^{CoG}\}\dot{\boldsymbol{\omega}} + \{{}^{CoG}\}\boldsymbol{\omega} \times \{{}^{CoG}\}I_\Sigma(\mathbf{q})\{{}^{CoG}\}\boldsymbol{\omega} + \Delta\dot{\mathcal{L}}_{A_p} \\ = \{{}^{CoG}\}\boldsymbol{\tau} + \sum_{i=1}^{N_{ex}} (\{{}^{CoG}\}\boldsymbol{\tau}_{ex_i} + \{{}^{CoG}\}\mathbf{p}_{ex_i}(\mathbf{q}) \times \{{}^{CoG}\}\mathbf{f}_{ex_i}) \end{aligned} \quad (20)$$

$$\mathbf{0} = \boldsymbol{\tau}_q + \sum_{i=1}^{N_{ex}} J_{ex_i}^T \begin{bmatrix} \mathbf{f}_{ex_i} \\ \boldsymbol{\tau}_{ex_i} \end{bmatrix} + \sum_{i=1}^{N_r} J_{r_i}^T \mathbf{f}_i + \sum_{i=1}^{N_s} J_{s_i}^T m_{s_i} \mathbf{g} \quad (21)$$

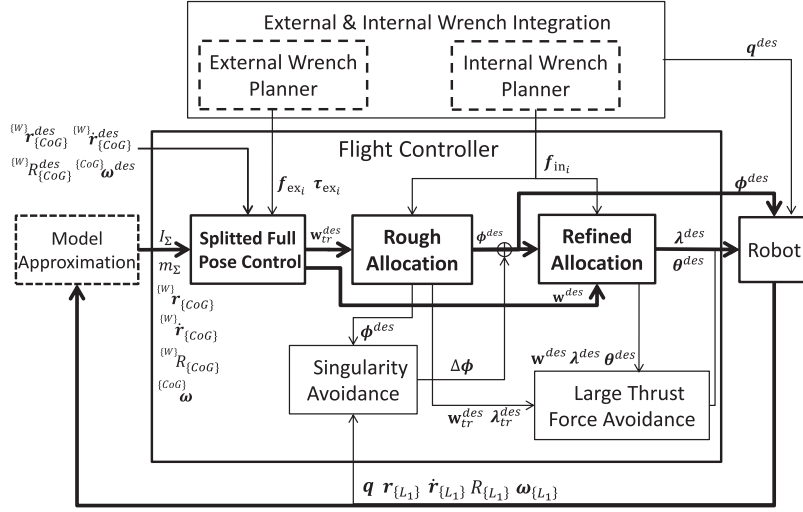
$$\Delta\dot{\mathcal{L}}_{A_p} = \sum_{i=1}^{N_r} \{{}^{CoG}\}R_{\{G_i\}} \begin{bmatrix} I_\phi \ddot{\phi}_i \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad (22)$$

where  $\{{}^W\}\mathbf{r}_{\{CoG\}}$ ,  $\{{}^W\}R_{\{CoG\}}$ , and  $\{{}^{CoG}\}\boldsymbol{\omega}$  are the position, attitude, and angular velocity of the CoG frame calculated based on the forward-kinematics from the root link states (i.e.,  $\{{}^W\}\mathbf{r}_{\{L_1\}}$ ,  $\{{}^W\}R_{\{L_1\}}$ , and  $\{{}^{L_1}\}\boldsymbol{\omega}$ ) with joint angles  $\mathbf{q}$ .  $m_\Sigma$  is the total mass and  $\{{}^W\}\mathbf{r}_{\{CoG\}}(\mathbf{q})$  is the CoG position.  $\{{}^{CoG}\}I_\Sigma(\mathbf{q})$  is the total inertia tensor only influenced by the joint angles  $\mathbf{q}$ . Please refer to [Appendix B](#) for the detail of the deviation from equations (16) and (17) to (19) and (20).

The essential difference compared with our previous work is the consideration of  $\Delta\dot{\mathcal{L}}_{A_p}$  in equation (20), which denotes the influence of the dynamics of vectoring actuators on that of the whole body.  $I_\phi$  is the inertial moment around the vectoring angle around  $\phi_i$  as shown in equation (13), which was ignored in our previous control method. In Section 3, we will reveal the oscillation caused by this neglect in the previous control method, and subsequently introduce the improved method to handle this problem.

### 3. Flight control

In this section, we describe the flight control method for vectorable thrust force as shown in [Figure 5](#). The output of the control framework is the desired thrust forces  $\lambda^{des}$  and the vectoring angles  $\boldsymbol{\phi}^{des}$  and  $\boldsymbol{\theta}^{des}$ . We first reveal the mechanism of oscillation owing to the rotational inertia of the vectoring actuator, and then present a two-stage control allocation to effectively avoid such an oscillation and enhance the flight stability. This framework further enables the integration of external wrench for manipulation and internal wrench for grasping. Furthermore, we also introduce two special strategies to address challenging configurations.



**Figure 5.** Overview of the control framework developed in this work, which is a part of the whole system as depicted in Figure 2. The main flow, which is from “Model Approximation” to “Robot”, is drawn in thick arrows and blocks. “Model Approximation” is presented in Section 2. “Splitted Full Pose Control” is presented in Section 3.2, whereas two blocks regarding the allocation are presented in Section 3.3. The integration of external and internal wrenches (Section 3.5) is developed to enable aerial manipulation and grasping. Besides, two special strategies are presented in Section 3.6 to address challenge configurations (“Singularity Avoidance” and “Large Thrust Force Avoidance”).

### 3.1. Oscillation caused by vectoring actuators

We first briefly introduce the conventional control method proposed in Zhao et al. (2021) that may cause the oscillation. This control method contains two parts: the full pose control and the control allocation.

**3.1.1. Full pose control.** For the approximated dynamics equations (19) and (20), feedback control based on a common PID control is used in Zhao et al. (2021) as follows

$$\begin{aligned} \{{CoG}\} \mathbf{f}^{des} &= m_{\Sigma}^{\{W\}} R_{\{CoG\}}^{\top} (K_{f,p} \mathbf{e}_r + K_{f,i} \int \mathbf{e}_r + K_{f,d} \dot{\mathbf{e}}_r) \\ &+ \{{W}\} R_{\{CoG\}}^{\top} \mathbf{f}_{ff} \end{aligned} \quad (23)$$

$$\mathbf{e}_r = \{{W}\} \mathbf{r}_{\{CoG\}}^{des} - \{{W}\} \mathbf{r}_{\{CoG\}} \quad (24)$$

$$\mathbf{f}_{ff} = m_{\Sigma} \mathbf{g} - \sum_{i=1}^{N_{ex}} \{{W}\} \mathbf{f}_{ex_i} \quad (25)$$

where  $K_{f,*}$  are the PID gain diagonal matrices.

The attitude control follows the SO(3) control method proposed by Lee et al. (2010)

$$\begin{aligned} \{{CoG}\} \boldsymbol{\tau}^{des} &= I_{\Sigma} (K_{\tau,p} \mathbf{e}_R + K_{\tau,i} \int \mathbf{e}_R + K_{\tau,d} \dot{\mathbf{e}}_R) \\ &+ \boldsymbol{\omega} \times I_{\Sigma} \boldsymbol{\omega} + \boldsymbol{\tau}_{ff} \end{aligned} \quad (26)$$

$$\mathbf{e}_R = \frac{1}{2} [R^{\top} R^{des} - R^{des\top} R]^{\vee} \quad (27)$$

$$\mathbf{e}_{\omega} = R^{\top} R^{des} \boldsymbol{\omega}^{des} - \boldsymbol{\omega} \quad (28)$$

$$\boldsymbol{\tau}_{ff} = - \sum_{i=1}^{N_{ex}} (\{{CoG}\} \boldsymbol{\tau}_{ex_i} + \{{CoG}\} \mathbf{p}_{ex_i} \times \{{CoG}\} \mathbf{f}_{ex_i}) \quad (29)$$

where  $[\bullet]^{\vee}$  is the inverse of a skew map, and  $R := \{{W}\} R_{\{CoG\}}$ ,  $\boldsymbol{\omega} := \{{CoG}\} \boldsymbol{\omega}$  for convenience.

Then, the desired wrench can be summarized as follows

$$\mathbf{w}^{des} = \begin{bmatrix} \{{CoG}\} \mathbf{f}^{des} \\ \{{CoG}\} \boldsymbol{\tau}^{des} \end{bmatrix} \quad (30)$$

**3.1.2. Control allocation.** The control objective is to calculate the desired thrust  $\lambda^{des}$  and the desired vectoring angles  $\boldsymbol{\phi}^{des}$ ,  $\boldsymbol{\theta}^{des}$  from the desired CoG wrench  $\mathbf{w}^{des}$  from equation (30). For an over-actuated model (i.e.,  $N_r > 2$ ), there is infinite solution of  $(\lambda, \boldsymbol{\phi}, \boldsymbol{\theta})$  according to equation (3). A straightforward method for optimal geometry allocation is proposed in our previous work (Zhao et al., 2021) as follows

$$\min_{\lambda, \boldsymbol{\theta}, \boldsymbol{\phi}} \|\lambda\|^2 \quad (31)$$

$$s.t. \quad \mathbf{w}^{des} = Q(\boldsymbol{\theta}, \boldsymbol{\phi}) \lambda \quad (32)$$

where  $Q$  is derived from equation (4).

**3.1.3. Oscillation of vectoring actuator.** The dynamics of vectoring actuator have no explicit influence on the translational dynamics of equation (19), and the positional stability by equations (23) and (31) is presented in Appendix C; whereas, the dynamics of vectoring actuator has a relatively complex influence on the rotational dynamics of equation (20) because of the non-negligible rotational



moment  $I_\phi$  in equation (22). To provide a more clear insight into the mutual influence, we first focus on the dynamics of a single vectoring actuator related to  $\phi_i$  as shown in equation (13). Furthermore, we introduce a special angular velocity  $\omega_{L_0}$  for the robot body which ensures the conservation of entire angular momentum and assume there is only  $\phi_i$  that performs vectoring. With the assumption of no thrust force generated, the conservation of entire angular momentum can be established as follows

$$(I_\Sigma - I_\phi)\omega_{L_0} + I_\phi(\omega_{L_0} + \dot{\phi}) = 0 \quad (33)$$

where  $I_\Sigma$  is the overall rotational inertia. For convenience, we omit the subscript  $i$  for  $\phi_i$ .

Then  $\omega_{L_0}$  can be considered as the angular velocity that has no influence on the overall rotational dynamics of equation (20).

We further perform the Laplace Transform for the dynamics of  $\phi$  from equation (13)

$$s^2\Phi(s) = k_{\phi,p}\Phi^{des}(s) - (k_{\phi,p}\Phi(s) + k_{\phi,d}s\Phi(s)) \quad (34)$$

To obtain  $\Phi^{des}(s)$ , it is necessary to consider the nonlinear mapping from the desired wrench  $\mathbf{w}^{des}$  to the desired vectoring angle  $\phi^{des}$ . However, equations (31) and (32) only involves geometry allocation. Thus, the characteristics in frequency domain should not change after such allocation. Focusing on the desired torque obtained from equation (26), following Laplace transform can be established for  $\Phi^{des}(s)$

$$\Phi^{des}(s) = \beta \left( k_{\tau,p}E_R(s) - k_{\tau,d} \left( \Omega_{L_0}(s) + \Omega'(s) \right) \right) \quad (35)$$

where  $\beta$  is a constant that denote the nonlinear map from torque to vectoring angle in frequency domain.  $E_R(s)$  is the Laplace transform of the one-dimensional rotational error which can be converted from  $e_R$ .  $\Omega_{L_0}(s)$  and  $\Omega'(s)$  are the Laplace transform of  $\omega_{L_0}$  defined in equation (33) and the residual one-dimensional angular velocity, respectively.  $k_{\tau,p}$  and  $k_{\tau,d}$  are the one-dimensional feedback gains converted from  $K_{\tau,p}$  and  $K_{\tau,d}$ . The I control term in equation (26) is

ignored, because it is only responsible for the steady-state model error.

Performing the Laplace transform for equation (33) and substituting the result along with equation (35) into equation (34), we can finally obtain the Laplace transform for the actuator dynamics as follows

$$\Phi(s) = G(s)U(s) \quad (36)$$

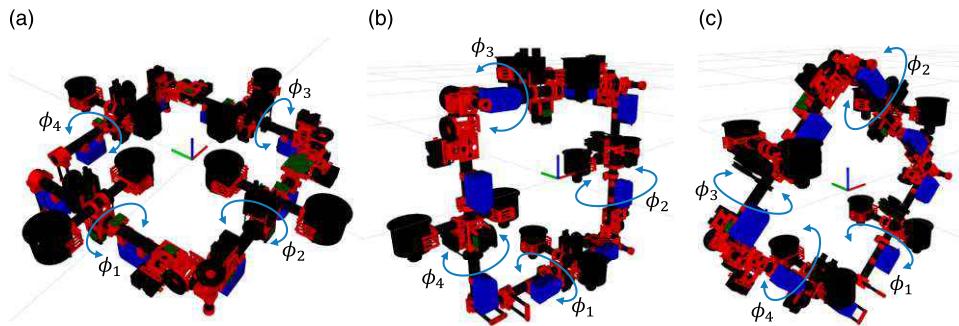
$$U(s) = k_{\tau,p}E_R(s) - k_{\tau,d}s\Omega'(s) \quad (37)$$

$$G(s) = \frac{k_{\phi,p}\beta}{s^2 + (k_{\phi,d} - k')s + k_{\phi,p}} \quad (38)$$

$$k' = k_{\phi,p}k_{\tau,d}\beta \frac{I_\phi}{I_\Sigma} \quad (39)$$

To ensure the stability of  $\Phi^{des}(s)$ , all poles of the transfer function  $G(s)$  should be negative, which means  $k_{\phi,d} - k' > 0$  in equation (38) should be guaranteed. For considerably small actuator rotational moment  $I_\phi$ , it is easy to achieve all negative poles; however, for non-negligible rotational moment as shown in equation (15), the absolute value of poles may decrease because of equation (39). If there is a pole that closes to zero, then the actuator will become oscillated, and propagate the oscillation to the robot body according to equation (33), which is the mechanism of resonance.

Here, we reconsider the role of  $\beta$  in equation (35). It can be considered as an influence factor of the desired torque on the vectoring angle. This indicates that  $\beta$  would change according to the robot pose as shown in Figure 6. Smaller  $\beta$  denotes less influence of the desired torque on  $\phi$ , but larger influence on other actuators (e.g., the thrust force  $\lambda$ ). Therefore, under the pose of Figure 6(a),  $\beta$  is zero for all  $\phi_i$ . However, under the pose of Figure 6(b),  $\beta$  has a large value for  $\phi_1$  and  $\phi_3$  because these angles significantly contribute to the torque around the  $x$  axis (the red axis). Furthermore, under the pose of Figure 6(c),  $\beta$  is non-zero for all  $\phi_i$  because all  $\phi_i$  contribute to the torque control. Consequently, these vectoring actuator amplifies the oscillation on the



**Figure 6.** Different robot poses to show different  $\beta$  of equation (35). (a) horizontal robot pose.  $\beta$  for all  $\phi_i$  is zero, because none of  $\phi_i$  contributes to the torque control. (b) vertical robot pose where the first and third links are horizontal.  $\beta$  for  $\phi_2$  and  $\phi_4$  is zero. However,  $\beta$  for  $\phi_1$  and  $\phi_3$  has a large value, because these angles significantly contribute to the torque around the  $x$  axis (the red axis). (c) vertical robot pose where all links have a tilt angle of 45 deg from the  $x$  axis.  $\beta$  for all  $\phi_i$  is non zero, because all  $\phi_i$  contribute to the torque control.

robot body, which leads to the singularity of this pose in the conventional control method. This fact reveals the reason why the stable flight under Figure 6(a) and (b) can be achieved by the conventional control method in (Zhao et al., 2021), but the flight under Figure 6(c) is unstable (which will be presented in Section 5.2.2).

To conclude, we attribute the reason of the resonance between the vectoring actuator and the robot body to the non-negligible rotational moment of vectoring actuator as shown in equations (38) and (39). The influence of such rotational moment depends on the robot pose as shown in Figure 6.

### 3.2. Splitted full pose control

Our goal is to ensure the stability of  $G(s)$  in equation (38) to be stable. If there is a way to remove the terms related to  $k_{\tau,d}$  in equation (35),  $k'$  in equation (38) will be zero, and  $G(s)$  will be always stable with proper actuator control gains ( $k_{\phi,p}, k_{\phi,d}$ ). Therefore, in this work, we propose a splitted full pose control based on equations (23) and (26) as follows

$$\{{CoG}\} \mathbf{f}^{des} = \{{CoG}\} \mathbf{f}_{tr}^{des} + m_{\Sigma}^{\{W\}} R_{\{CoG\}}^T K_{f,d} \dot{\mathbf{e}}_r \quad (40)$$

$$\{{CoG}\} \mathbf{f}_{tr}^{des} = m_{\Sigma}^{\{W\}} R_{\{CoG\}}^T (K_{f,p} \mathbf{e}_r + K_{f,i} \int \mathbf{e}_r) + \mathbf{f}_{ff} \quad (41)$$

$$\{{CoG}\} \boldsymbol{\tau}^{des} = \{{CoG}\} \boldsymbol{\tau}_{tr}^{des} + I_{\Sigma} K_{\tau,d} \mathbf{e}_{\omega} + \boldsymbol{\omega} \times I_{\Sigma} \boldsymbol{\omega} \quad (42)$$

$$\{{CoG}\} \boldsymbol{\tau}_{tr}^{des} = I_{\Sigma} (K_{\tau,p} \mathbf{e}_R + K_{\tau,i} \int \mathbf{e}_R) + \boldsymbol{\tau}_{ff} \quad (43)$$

where  $\{{CoG}\} \mathbf{f}_{tr}^{des}$  and  $\{{CoG}\} \boldsymbol{\tau}_{tr}^{des}$  are truncated, which both have no D control terms.

Then, the truncated desired wrench can be given by

$$\mathbf{w}_{tr}^{des} = \begin{bmatrix} \{{CoG}\} \mathbf{f}_{tr}^{des} \\ \{{CoG}\} \boldsymbol{\tau}_{tr}^{des} \end{bmatrix} \quad (44)$$

It is notable that in an ideal hovering state,  $\mathbf{w}_{tr}^{des} = \mathbf{w}^{des}$ , because the robot is static.

### 3.3. Two-stage control allocation

The truncated desired wrench is only used for calculating the desired vectoring angles of  $\boldsymbol{\phi}$  to avoid the oscillation as described in Section 3.1. Therefore, a two-stage allocation strategy is developed in this work, which first calculates the proper vectoring angles  $\boldsymbol{\phi}$  based on the truncated desired wrench  $\mathbf{w}_{tr}^{des}$ , and then calculates the rest of the control input  $\boldsymbol{\theta}$  and  $\boldsymbol{\lambda}$  based on the complete wrench  $\mathbf{w}^{des}$ .

**3.3.1. Rough allocation based on truncated desired wrench.** We first calculate the desired vectoring angles  $\boldsymbol{\phi}$  (along with  $\boldsymbol{\theta}$  and  $\boldsymbol{\lambda}$ ) based on  $\mathbf{w}_{tr}^{des}$

$$\min_{\boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{\phi}} \|\boldsymbol{\lambda}\|^2$$

$$s.t. \quad \mathbf{w}_{tr}^{des} = Q(\boldsymbol{\theta}, \boldsymbol{\phi}) \boldsymbol{\lambda} \quad (45)$$

Zhao et al. (2021) proposed an iterative solution for this nonlinear optimization problem; however, the rigorous convergence proof is not presented. Therefore, we develop an enhanced iterative solution using gradient to guarantee the convergence to the optimal (or at least suboptimal) solution in this work. We first follow the method proposed by Kamel et al. (2018), which applies the vectored forces  $\mathbf{f}_i \in \mathcal{R}^3$  from equation (1) as an intermediate variable and introduces a combined vector  $\mathbf{F} = \begin{bmatrix} \{{L}_1\} \mathbf{f}_1^T & \{{L}_2\} \mathbf{f}_2^T \\ \dots & \{{L}_{N_r}\} \mathbf{f}_{N_r}^T \end{bmatrix}^T$ . Then the above optimization problem can be modified as follows

$$\min_{\mathbf{F}} \|\mathbf{F}\|^2 \quad (46)$$

$$s.t. \quad \mathbf{w}_{tr}^{des} = \tilde{Q} \mathbf{F} \quad (47)$$

$$\tilde{Q} = \begin{bmatrix} \tilde{Q}_{col_1} & \tilde{Q}_{col_2} & \dots & \tilde{Q}_{col_{N_r}} \end{bmatrix} \quad (48)$$

$$\tilde{Q}_{col_i} = \begin{bmatrix} E_{3 \times 3} \\ [\{{CoG}\} \mathbf{p}_{\{F_i\}} \times] \end{bmatrix} \{{CoG}\} R_{\{L_i\}} \quad (49)$$

where  $E_{3 \times 3}$  is a  $3 \times 3$  identity matrix and  $[\cdot \times]$  denotes the skew symmetric matrix of a three dimensional vector.

Then, the closed-form for equations (46) and (47) and the desired thrust and vectoring angles can be directly given by

$$\mathbf{F} = \tilde{Q}^{\#} \mathbf{w}_{tr}^{des} \quad (50)$$

$$\lambda_i = \|\{{L}_i\} \mathbf{f}_i\| \quad (51)$$

$$\phi_i = \tan^{-1} \left( \frac{-\{{L}_i\} f_{iy}}{\{{L}_i\} f_{iz}} \right) \quad (52)$$

$$\theta_i = \tan^{-1} \left( \frac{\{{L}_i\} f_{ix}}{-\{{L}_i\} f_{iy} \sin(\phi_i) + \{{L}_i\} f_{iz} \cos(\phi_i)} \right) \quad (53)$$

where  $(\cdot)^{\#}$  denotes the MP-pseudo-inverse of a full-rank matrix.

As a unique mechanical feature of the two-DoF vectoring apparatus depicted in Figure 3(c), the result of vectoring angles  $\boldsymbol{\phi}$  and  $\boldsymbol{\theta}$  from equations (52) and (53) will deviate the position  $\{{CoG}\} \mathbf{p}_{\{F_i\}}$  in equation (49) because of the offset from the frame  $\{G_{i\_roll}\}$  to frame  $\{F_i\}$ . Then, the results of equations (51)–(53) will no longer satisfy the constraint equation (47) because  $\tilde{Q}$  has changed.

Then, we introduce the residual term for equation (45) using the results of equations (51)–(53), and also compute the derivative with respect to  $\boldsymbol{\lambda}$ ,  $\boldsymbol{\phi}$ , and  $\boldsymbol{\theta}$

$$\boldsymbol{\epsilon} = \mathbf{w}_{tr}^{des} - Q(\boldsymbol{\theta}, \boldsymbol{\phi}) \boldsymbol{\lambda} \quad (54)$$

$$\delta\epsilon = J_w [\delta\lambda \quad \delta\phi \quad \delta\theta]^\top \quad (55)$$

$$J_w = [J_{w/\lambda} \quad J_{w/\phi} \quad J_{w/\theta}] \quad (56)$$

$$J_{w/\lambda} = -Q(\theta, \phi) \in \mathcal{R}^{6 \times N_r} \quad (57)$$

$$J_{w/\phi} = \frac{\partial \mathbf{w}}{\partial \phi} - \frac{\partial(Q(\theta, \phi)\lambda)}{\partial \phi} \in \mathcal{R}^{6 \times N_r} \quad (58)$$

$$J_{w/\theta} = \frac{\partial \mathbf{w}}{\partial \theta} - \frac{\partial(Q(\theta, \phi)\lambda)}{\partial \theta} \in \mathcal{R}^{6 \times N_r} \quad (59)$$

The partial derivative elements in equations (58) and (59) can be calculated from the multilinked kinematics model.

Our objective is to find a solution of  $\lambda$ ,  $\phi$ , and  $\theta$  to achieve zero  $\epsilon$ . Then we start from the initial values  $\lambda_0$ ,  $\phi_0$ , and  $\theta_0$  calculated from equations (51)–(53), and perform the following linear iteration with the objective of minimizing

$$\sum_{i=1}^{N_r} \|\delta\lambda_i\|^2 + \|\delta\phi_i\|^2 + \|\delta\theta_i\|^2 \text{ at each iteration}$$

$$[\lambda_k, \phi_k, \theta_k]^\top = [\lambda_{k-1}, \phi_{k-1}, \theta_{k-1}]^\top + J_w^\# \epsilon_{k-1} \quad (60)$$

$$\epsilon_k = \mathbf{w}_{tr}^{des} - Q(\theta_k, \phi_k)\lambda_k \quad (61)$$

where  $J_w^\#$  is the psuedo-inverse matrix of  $J_w$ , and  $k \in [0, 1, 2, \dots]$  is the iteration number. Note that the most computationally intensive operation in this iteration process is the calculation of the inverse matrix with a size of the  $6 \times 6$  for  $J_w^\#$ , which can be solved instantaneously for realtime control. In most of the cases, it only requires 2 or 3 iterations to get a sufficiently small value of  $\epsilon$  (i.e.,  $\|\epsilon\| \sim 10^{-6}$ ). This may be attributed to the initial values  $\lambda_0$ ,  $\phi_0$ , and  $\theta_0$  being relatively close to the convergent solution. We define the convergent solution as  $\lambda_{tr}^{des}$ ,  $\phi_{tr}^{des}$ , and  $\theta_{tr}^{des}$ , which are independent of the D control terms in equations (40) and (42). Thus, we treat  $\phi_{tr}^{des}$  as the final control input to the robot (i.e.,  $\phi^{des} = \phi_{tr}^{des}$ ), which can perfectly enable  $k'$  in equation (38) to be zero (and thus avoid the oscillation of the vectoring actuators and the whole body).

**3.3.2. Refined allocation for all control terms.** Given that  $\lambda_{tr}^{des}$ ,  $\phi_{tr}^{des}$ , and  $\theta_{tr}^{des}$  can only track the truncated desired wrench  $\mathbf{w}_{tr}^{des}$ , it is necessary to further modify  $\lambda$  and  $\theta$  to track the complete desired wrench  $\mathbf{w}^{des}$  which contains the D control terms. With the fixed vectoring angles of  $\phi^{des} (= \phi_{tr}^{des})$ ,  $\lambda$  and  $\theta$  are refined by an iteration similar to equations (60) and (61)

$$[\lambda_k, \theta_k]^\top = [\lambda_{k-1}, \theta_{k-1}]^\top + J_w^\# \epsilon'_{k-1} \quad (62)$$

$$\epsilon'_k = \mathbf{w}^{des} - Q(\theta_k, \phi^{des})\lambda_k \quad (63)$$

$$J'_w = [J_{w/\lambda} \quad J_{w/\theta}] \quad (64)$$

For fast convergence, the initial states  $\lambda_0$  and  $\theta_0$  are calculated from following rough allocation

$$\tilde{\mathbf{F}} = \tilde{Q}^\# \mathbf{w}^{des} \quad (65)$$

$$\lambda_{0,i} = \left\| \{G_{i-roll}\} \tilde{\mathbf{f}}_i \right\| \quad (66)$$

$$\theta_{0,i} = \tan^{-1} \left( \frac{\{G_{i-roll}\} \tilde{\mathbf{f}}_i(0)}{\{G_{i-roll}\} \tilde{\mathbf{f}}_i(1)} \right) \quad (67)$$

$$\tilde{Q}' = \begin{bmatrix} \tilde{Q}'_{col1} & \tilde{Q}'_{col2} & \dots & \tilde{Q}'_{colN_r} \end{bmatrix} \quad (68)$$

$$\tilde{Q}'_{col_i} = \begin{bmatrix} E_{3 \times 3} \\ [\{CoG\} \mathbf{p}_{\{F_i\}} \times] \end{bmatrix} \{CoG\} R_{\{G_{i-roll}\}} (\phi_i^{des}) \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (69)$$

where  $\{G_{i-roll}\} \tilde{\mathbf{f}}_i \in \mathcal{R}^2$  is the truncated vectored force w.r.t. the frame of  $\{G_{i-roll}\}$ , and  $\tilde{\mathbf{F}}$  is a combined vector:

$$\tilde{\mathbf{F}} = \begin{bmatrix} \{G_{1-roll}\} \tilde{\mathbf{f}}_1^\top & \{G_{2-roll}\} \tilde{\mathbf{f}}_2^\top & \dots & \{G_{N_r-roll}\} \tilde{\mathbf{f}}_{N_r}^\top \end{bmatrix}^\top \cdot \lambda_{0,i}$$

$\theta_{0,i}$ , and  $\phi_i^{des}$  are the  $i$ -th elements of  $\lambda_0$ ,  $\theta_0$ , and  $\phi^{des}$ , respectively.

Finally, we obtain the convergent  $\lambda$ ,  $\theta$  from equations (62) and (63), which are the desired control input  $\lambda^{des}$  and  $\theta^{des}$ .

### 3.4. Stability analysis

The analysis on complete stability is presented in Appendix D in detail, which demonstrates that the trackability to the desired position and attitude can be achieved by the propose two-stage control allocation.

In the proposed control method, the dynamics of vectoring actuator is not explicitly considered; however, this method can still suppress the influence of vectoring actuator on the flight stability. This significantly reduces the complexity of the control framework. Such implicit consideration of actuator dynamics is the crucial strengthen compared to other explicit control methods (Allenspach et al., 2020; Faessler et al., 2017).

### 3.5. Integration with external and internal wrenches

**3.5.1. External wrench.** An external wrench is the combination of force  $\{W\} \mathbf{f}_{ex_i}$  and torque  $\{W\} \boldsymbol{\tau}_{ex_i}$  exerted by the environment or object during the manipulation by an end-effector and is integrated into the control framework by equations (25) and (29).  $\mathbf{f}_{ff}$  and  $\boldsymbol{\tau}_{ff}$  in those equations are constant against  $\lambda$ ,  $\phi$ , and  $\theta$ , except for the torque caused the external force  $\{CoG\} \mathbf{p}_{ex_i} \times \{CoG\} \mathbf{f}_{ex_i}$  in equation (29). Because the robot CoG is influenced by the vectoring angles  $\phi$  and  $\theta$ ,  $\frac{\partial \{CoG\} \mathbf{p}_{ex_i}}{\partial \phi}$  and  $\frac{\partial \{CoG\} \mathbf{p}_{ex_i}}{\partial \theta}$  are not zero. Those partial

derivatives should be considered when calculating  $\frac{\partial \mathbf{w}}{\partial \phi}$  and  $\frac{\partial \mathbf{w}}{\partial \theta}$  in equations (58) and (59).

The stability with the external wrench is also presented in Appendix D in detail, which demonstrates the ability to compensate the external wrench that performs as a step input, and keep tracking the desired pose at the same time.

**3.5.2. Internal wrench.** An internal wrench is a force or torque which has no influence on the entire CoG motion (i.e., equations (19) and (20)), but has influence on the joint motion equation (21). We define the internal wrench as a list of the additional vectored thrust forces of  $\mathcal{R}^3$ :  $[\mathbf{f}_{in_1}, \mathbf{f}_{in_2}, \dots, \mathbf{f}_{in_{N_r}}]$  which have the following attributes

$$\sum_{i=1}^{N_r} \{CoG\} \mathbf{f}_{in_i} = \mathbf{0} \quad (70)$$

$$\sum_{i=1}^{N_r} \{CoG\} \mathbf{p}_{\{F_i\}} \times \{CoG\} \mathbf{f}_{in_i} = \mathbf{0} \quad (71)$$

$$\sum_{i=1}^{N_r} J_{T_i}^T \{CoG\} \mathbf{f}_{in_i} \neq \mathbf{0} \quad (72)$$

where  $\{CoG\} \mathbf{p}_{\{F_i\}}$  is the position of the frame of the  $i$ -th rotor w.r.t the robot CoG frame, and  $J_{T_i}$  is the Jacobian matrix for this frame. Note that  $f_{ex_i}$  is an external force exerted on the end-effector, whereas  $f_{in_i}$  is a vectored thrust force generated by a rotor.

Given the above attributes, this internal wrench can contribute to grasping object with the two ends of multilinks instead of the joint torque, which is a unique characteristic of the robot DRAGON. The calculation of the thrust forces  $\mathbf{f}_{in_i}$  is presented in Section 4. Here, we discuss the integration of such an internal wrench into the control framework.

Since the desired CoG wrench  $\mathbf{w}^{des}$  and the truncated wrench  $\mathbf{w}_{tr}^{des}$  cannot reflect the internal wrench, we modify equation (51) with the following linear combination

$$\lambda_i = \left\| \{L_i\} \mathbf{f}'_i \right\| \quad (73)$$

$$\{L_i\} \mathbf{f}'_i = \{L_i\} \mathbf{f}_i + \{L_i\} \mathbf{f}_{in_i} \quad (74)$$

where  $\{L_i\} \mathbf{f}_{in_i}$  is the vectored force related to the internal wrench w.r.t. the  $i$ -th link frame  $\{L_i\}$ . Then,  $\phi_i$  and  $\theta_i$  are calculated from equations (52) and (53) using  $\{L_i\} \mathbf{f}'_i$  instead of  $\{L_i\} \mathbf{f}_i$ .

This linear combination can further cause the deviation of the desired wrench as indicated in equation (54). However, the proposed iteration process of equations (60) and (61) can ensure the convergence to the desired wrench

of  $\mathbf{w}_{tr}^{des}$  despite the modification of equation (73). The same modification is also applied in equation (66) for the refined allocation with  $\mathbf{w}^{des}$ .

### 3.6. Challenging configurations

In our proposed two-stage allocation method, there are three types of control input:  $\lambda, \phi, \theta \in \mathcal{R}^{N_r}$ ; however, the vectoring angles  $\phi$  are not used in the second (refined) allocation. This indicates that the DoF of control input at the second stage of allocation decreases from  $3N_r$  to  $2N_r$ . In certain configurations, such a decrease can lead to an undesirable increase in the command value (e.g., large  $\lambda$ ). Furthermore, the decision of  $\phi$  at the first stage of allocation also results in singularities.

**3.6.1. Avoidance of large thrust force.** The proposed refined allocation as stated in Section 3.3.2 may result in relatively large thrust force under certain robot configurations where the controllability is relatively low and may exceed the range that rotor can generate. In other cases, since the D control terms in equations (40) and (42) contain the linear and angular velocity which are sensitive to the mechanical vibration, undesirable fluctuations may occur because of the refined allocation.

To avoid the invalid command for the thrust force, we first evaluate the difference between the thrust forces  $\lambda_{tr}^{des}$  from the rough allocation and  $\lambda^{des}$  from the refined allocation. A large gap between the two vectors implies that the truncated motion related to  $\mathbf{w}_{tr}^{des}$  has a significantly different direction from the non-truncated motion. If the gap  $\lambda_{tr}^{des} - \lambda^{des}$  is larger than a certain threshold,  $\lambda^{des}$  must be modified to suppress the gap within the threshold. To achieve this, the SR-inverse method (Nakamura and Hanafusa 1986) is applied

$$\tilde{\mathbf{F}} = \tilde{\mathbf{Q}}^* (\mathbf{w}^{des} - \mathbf{w}_{tr}^{des}) + \tilde{\mathbf{F}}_{tr} \quad (75)$$

$$\tilde{\mathbf{Q}}^* = \tilde{\mathbf{Q}}'^T (\tilde{\mathbf{Q}}' \tilde{\mathbf{Q}}'^T + \epsilon_{sr} E_{6 \times 6})^{-1} \quad (76)$$

$$\tilde{\mathbf{F}}_{tr} = \left[ \begin{array}{ccc} \{G_{1\_roll}\} \tilde{\mathbf{f}}_{tr,1}^T & \{G_{2\_roll}\} \tilde{\mathbf{f}}_{tr,2}^T & \dots & \{G_{tr,Nr\_roll}\} \tilde{\mathbf{f}}_{tr,Nr}^T \end{array} \right]^T$$

where  $\{G_{1\_roll}\} \tilde{\mathbf{f}}_{tr,i} \in \mathcal{R}^2$  is a truncated vector thrust force which can be obtained from the inverse operations of equations (66) and (67) using  $\theta_{tr,i}^{des}$  and  $\lambda_{tr,i}^{des}$ .  $\epsilon_{sr}$  is a rate for the SR-inverse.

To avoid a heuristic decision on the dimensionless quantity  $\epsilon_{sr}$ , we also introduce a search method for  $\epsilon_{sr}$  with two basic criteria:  $\|\lambda_{tr}^{des} - \lambda^{des}\|_\infty$  and  $\|\mathbf{w}_{tr}^{des} - \mathbf{w}^{des}\|$ , where the first one is the L-Infinity norm.  $\epsilon_{sr} \rightarrow \infty$  corresponds to  $\|\lambda_{tr}^{des} - \lambda^{des}\|_\infty \rightarrow 0$ , whereas  $\epsilon_{sr} \rightarrow 0$  corresponds to  $\|\mathbf{w}_{tr}^{des} - \mathbf{w}^{des}\| \rightarrow 0$ . An incremental process summarized as Alg. 1 is proposed to find the desirable control input. Since

the deviation of the target wrench has a higher priority, we start from a value not too small (e.g., 1) for  $\epsilon_{sr}$ , and gradually decrease it by half. Two thresholds, namely,  $\Delta\lambda$  and  $\Delta w$  are introduced, which are easier to determine than  $\epsilon_{sr}$ . The results of Alg. 1 are the final desired control input (i.e.,  $\lambda^{des}$  and  $\theta^{des}$ ), which replace the output from equations (62) and (63). Thus, if  $\|\lambda^{des} - \lambda_{tr}^{des}\|_{\infty} > \Delta\lambda$ , the arrow from ‘‘Refined Allocation’’ to ‘‘Robot’’ in Figure 5 will disappear, and the arrow from the ‘‘Large Thrust Force Avoidance’’ to ‘‘Robot’’ will be activated.

---

**Algorithm 1** Modification of  $\lambda^{des}$ ,  $\theta^{des}$  by iterative SR-inverse

---

```

 $\epsilon_{sr} \leftarrow 1$ 
while  $\|\lambda^{des} - \lambda_{tr}^{des}\|_{\infty} > \Delta\lambda$  or  $\|\mathbf{w}^{des} - \mathbf{w}_{tr}^{des}\| > \Delta w$ 
do
   $\tilde{\mathbf{F}} \leftarrow (75)$ 
   $\lambda \leftarrow (66)$  with  $\tilde{\mathbf{F}}$ 
   $\theta \leftarrow (67)$  with  $\tilde{\mathbf{F}}$ 
   $\epsilon_{sr} \leftarrow \frac{\epsilon_{sr}}{2}$ 
end while
 $\lambda^{des} \leftarrow \lambda$ 
 $\theta^{des} \leftarrow \theta$ 
return  $\lambda^{des}$ ,  $\theta^{des}$ 

```

---

**3.6.2. Singular configuration.** In the two-stage allocation strategy, the desired vectoring angles  $\phi$  that are determined at the first stage can significantly affect the decision on the desired vectoring angles  $\theta$  and thrust forces  $\lambda$  at the second stage. For instance, if all links of robot are aligned on a vertical plane, the desired  $\phi$  will render all rotors almost vertical to balance gravity with minimum thrust force. This makes rotors significantly difficult to generate the lateral force that is perpendicular to the vertical plane, because  $\theta$  determined at the second stage cannot contribute to this lateral force in most of the time. To quantify the influence of the early decision of  $\phi$  on the controllability, we introduced a truncated feasible control force polygon  $\mathcal{V}'_F$  and torque convex  $\mathcal{V}'_T$  with fixed  $\phi$  as follows

$$\mathcal{V}'_F := \left\{ \mathbf{f}_{xy} \in \mathcal{R}^2 \mid \mathbf{f}_{xy} = \sum_{i=1}^{N_r} \gamma_{\theta} \mathbf{u}_{\theta_i}, -1 \leq \gamma_{\theta} \leq 1 \right\} \quad (77)$$

$$\mathcal{V}'_T := \left\{ \boldsymbol{\tau} \in \mathcal{R}^3 \mid \boldsymbol{\tau} = \sum_{i=1}^{N_r} \sum_{j=1}^2 \gamma_{ij} \mathbf{v}''_{ij}, -1 \leq \gamma_{ij} \leq 1 \right\} \quad (78)$$

where  $\mathbf{u}_{\theta_i}$  and  $\mathbf{v}''_{ij}$  ( $j = 1, 2$ ) are defined as follows

$$\mathbf{u}_{\theta_i} = [\cos(\phi_i^{des} + \psi_{\{L_i\}}), -\sin(\phi_i^{des} + \psi_{\{L_i\}})]^T \quad (79)$$

$$\mathbf{v}''_{ij} = {}^{\{CoG\}} \mathbf{P}_{\{F_i\}} \times \left( {}^{\{CoG\}} \mathbf{R}_{\{L_i\}} R_{\{G_{i-roll}\}} (\phi_i^{des}) [1, 0, 0]^T \right) \quad (80)$$

$$\mathbf{v}''_{i2} = {}^{\{CoG\}} \mathbf{P}_{\{F_i\}} \times \left( {}^{\{CoG\}} \mathbf{R}_{\{L_i\}} R_{\{G_{i-roll}\}} (\phi_i^{des}) [0, 0, 1]^T \right) \quad (81)$$

where  $\phi_i^{des}$  is the desired vectoring angle obtained from the first stage of allocation as presented in Section 3.3.1, whereas  $\psi_{\{L_i\}}$  indicates the yaw direction of the  $i$ -th link.

Figure 7 shows  $\mathcal{V}'_F$  and  $\mathcal{V}'_T$  in the case of the vertical pose shown in Figure 6(c).  $\phi_i^{des}$  is calculated based on Section 3.3.1 with  $\mathbf{w}_{tr}^{des} = [0 \ 0 \ m_{\Sigma}g \ 0 \ 0 \ 0]^T$ . It can be confirmed that the torque controllability around the  $x$  and  $z$  axes is significantly small. As discussed in Section 3.1.3, the pose of Figure 6(c) is a singularity in the conventional control method due to the oscillation caused by vectoring actuators; however, this pose is still singular in the proposed control method due to the early decision of  $\phi$  in two-stage allocation strategy. To avoid such singularity, additional constraints related to  $\mathcal{V}'_F$  and  $\mathcal{V}'_T$  should be considered in the first (rough) allocation. This leads to following extended optimization problem for rough allocation

$$\min_{\lambda, \theta, \phi} \|\lambda\|^2$$

$$s.t. \quad \mathbf{w}_{tr}^{des} = \mathcal{Q}(\theta, \phi)\lambda$$

$$f'_{\min}(\phi) > \underline{f} \quad (82)$$

$$\tau'_{\min}(\phi) > \underline{\tau} \quad (83)$$

where  $f'_{\min}(\phi)$  and  $\tau'_{\min}(\phi)$  are the guaranteed minimum control force and torque of  $\mathcal{V}'_F$  and  $\mathcal{V}'_T$  that are obtained from equations (7) and (10).  $\underline{f}$  and  $\underline{\tau}$  are the lower bounds for the force and torque controllability.

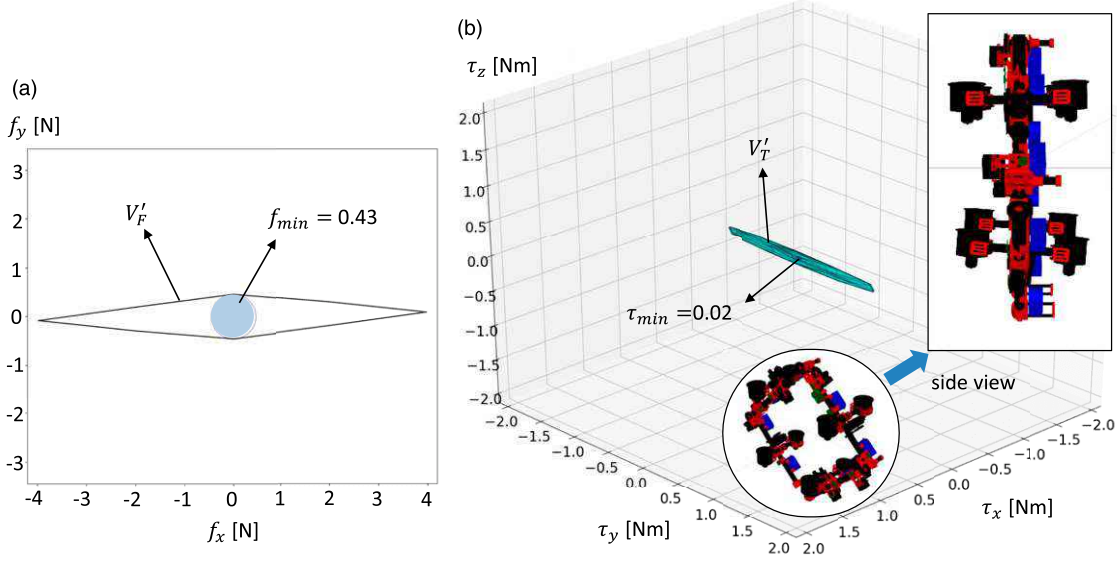
We literally solve the above problem with nonlinear optimization method (i.e., SQP) under the pose of Figure 6(c), and compare with the result that does not includes equations (82) and (83). Then, following phenomena are found from the comparison

$$\text{sgn}(\hat{\phi}_i^{des} - v_i) = \text{sgn}(\phi_i^{des} - v_i) \quad (84)$$

$$\hat{\phi}_i^{des} - v_i \propto \underline{\tau} \quad (85)$$

where  $v_i$  is the vectoring angle for  $\phi_i$  that make the  $i$ -th rotor entirely vertical ( $|v_i| = \frac{\pi}{2}$  rad in the case of Figure 6(c)).  $\hat{\phi}_i^{des}$  is the optimal result of the above extended optimization problem, whereas  $\phi_i^{des}$  is the optimal result excluding equations (82) and (83).  $\text{sgn}(\ast)$  is a function to get the sign of the variable. Note that the gradient of  $f'_{\min}(\phi)$  and  $\tau'_{\min}(\phi)$  required for SQP is presented in Anzai et al. (2019).

The inequality constraints of equations (82) and (83) not only significantly increase the calculation cost, but also prevent the usage of the fast convergence solution as presented in equations (46)–(61). Thus, to consider  $f'_{\min}(\phi)$  and



**Figure 7.** Singular configuration, where the robot is aligned on a vertical plane in a rhombus shape. All rotors are nearly vertical to counteract the gravity. (a) and (b) are the feasible control force polygon and torque convex without using the vectoring angles  $\phi$  (they are fixed at  $\phi^{des}$ ). The rotational controllability decreases significantly, which can only generate sufficient torque around the  $y$  axis if we do not use  $\phi$ . The guaranteed minimum control torque  $\tau_{min}$  decreases to 0.02 Nm, which clearly quantify this singularity.

$\tau_{min}(\phi)$  in a realtime control framework, we seek for a method that implicitly consider equations (82) and (83).

According to equation (85), there must exist a deviation vector  $\Delta\phi$  that satisfies  $f'_{min}(\mathbf{v} + \Delta\phi) > f_-$ ,  $\tau'_{min}(\mathbf{v} + \Delta\phi) > \underline{\tau}$ , and  $\mathbf{w}_{tr}^{des} = Q(\theta, \mathbf{v} + \Delta\phi)\lambda$ , where  $\mathbf{v}$  is the vector of  $v_i$ . Although it is difficult to search all elements of  $\Delta\phi$  simultaneously, we can first focus on a single element  $\Delta\phi_i$  and then find other elements by using equations (50)–(61). This is performed as follows:

1. First, according to equations (84) and (85), a constant offset  $\Delta\phi$  (e.g., 20 deg) is added for the  $\frac{N_r}{2}$ -th rotor

$$\phi_i^{des} = \phi_i^{des} + \Delta\phi \text{sgn}(\phi_i^{des} - v_i); \quad i = \frac{N_r}{2} \quad (86)$$

where  $\phi_i^{des}$  is the optimal result without equations (82) and (83).

2. Then, by fixing this vectoring angle, the rest of  $\phi^{des}$  can be recalculated with the same allocation sequence of equations (50)–(61) with a modification on equation (50) as follows

$$\mathbf{F}'' = \tilde{Q}^{i\#} \mathbf{w}_{lq}^{des} \quad (87)$$

$$\tilde{Q}'' = \begin{bmatrix} \tilde{Q}_{col_1}'' & \tilde{Q}_{col_2}'' & \cdots & \tilde{Q}_{col_{N_r}}'' \end{bmatrix} \quad (88)$$

$$\tilde{Q}_{col_i}'' = \begin{cases} \tilde{Q}'_{col_i}(\phi_i^{des}) & i = \frac{N_r}{2} \\ \tilde{Q}_{col_i} & \text{else} \end{cases} \quad (89)$$

where  $\tilde{Q}_{col_i}$  and  $\tilde{Q}'_{col_i}$  are defined in equations (49) and (69), respectively.

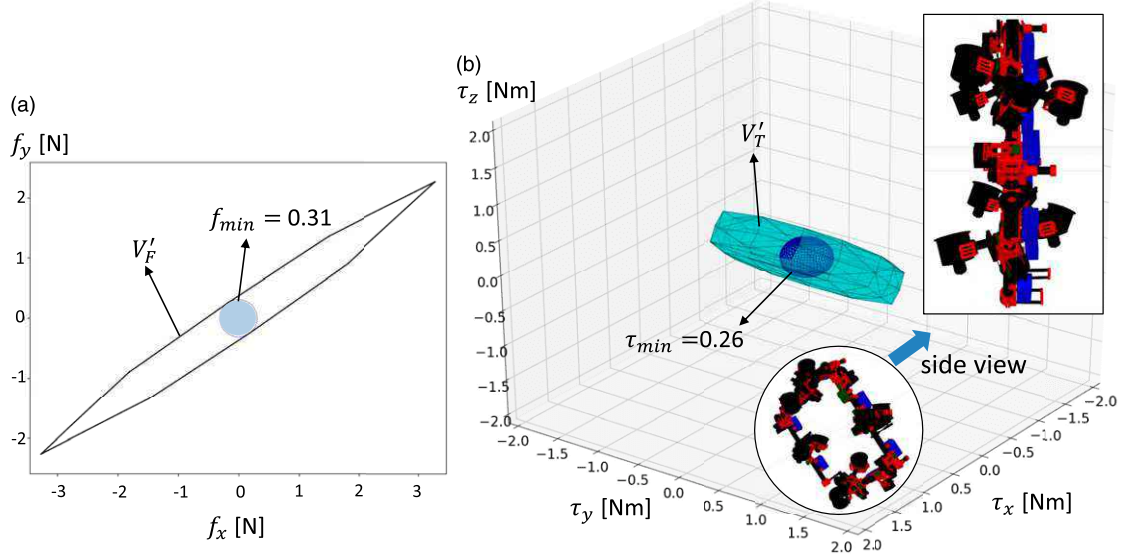
3. If  $f'_{min}(\phi^{des}) > f_-$ , and  $\tau'_{min}(\phi^{des}) > \underline{\tau}$  are not established, a larger  $\Delta\phi$  is introduced to repeat the process of equation (86) to update  $\phi^{des}$ . Such iteration process won't stop until the latest  $\phi^{des}$  satisfies the above constraints.

The deviation of  $\phi_i^{des} \rightarrow \phi_i^{des}$  ( $i = \frac{N_r}{2}$ ) induces a lateral force on the  $\frac{N_r}{2}$ -th rotor. Thus, other rotor should also generate proper lateral forces to satisfy  $\mathbf{w}_{tr}^{des} = Q(\theta, \phi^{des})\lambda$ , which is achieved by equations (50)–(61). Lateral forces by other rotors indicate the certain deviation from  $v_i$  for these rotors, which finally guarantees the satisfaction of equations (82) and (83).

It is also notable that the above calculation on  $\Delta\phi$  ( $= \phi^{des} - \phi^{des}$ ), which is related to the block of ‘‘Singularity Avoidance’’ in Figure 5, is only performed when  $\phi^{des}$  from the rough allocation does not satisfy equations (82) and (83). The enhanced controllability for the singular pose of Figure 7 can be confirmed in Figure 8. The guaranteed minimum control torque  $\tau_{min}$  increases from 0.02 Nm to 0.26 Nm, indicating the significant improvement in rotational controllability.

## 4. Object grasping

In this section, we present the unique object grasping ability of the robot DRAGON, which can use the vectored thrust force as an internal wrench instead of the joint torque. We first describe the statics for the multilinked model involving



**Figure 8.** Modified configuration for the singularity of Figure 7. All rotors are tilted vertically. (a) and (b) are the feasible control force polygon and torque convex with modified  $\phi^{des}$ . The rotational controllability is significantly enhanced, and the guaranteed minimum control torque  $\tau_{min}$  increases from 0.02 Nm to 0.26 Nm; whereas the guaranteed minimum control force  $f_{min}$  shows a slight decrease from 0.41 N to 0.31 N.

the equilibrium between the vector thrust force, contact force, and joint torque during grasping, which is extended from our previous work (Zhao et al., 2020).

Then, we derive the vectored thrust force for grasping based on this statics model. Finally, we present an online estimation method for the object mass to ensure the flight stability in an aerial grasping task as shown in Figure 9.

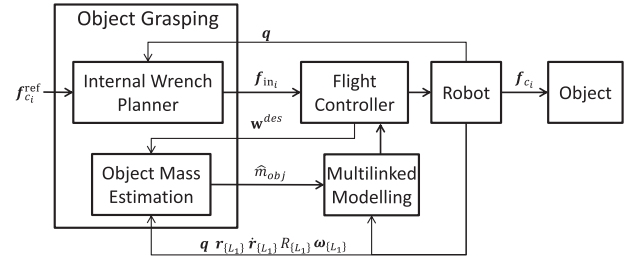
#### 4.1. Statics of multilinked model for grasping

The object is grasped by two ends of the multilinked model as shown in Figure 10. Then a truncated statics model based on equations (19)–(21), which focuses on grasping, can be given by

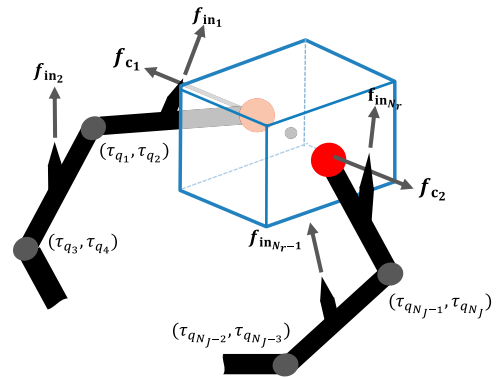
$$\begin{aligned} \sum_{i=1}^{N_r} \{CoG\} \mathbf{f}_{in_i} &= \mathbf{0} \\ \sum_{i=1}^{N_r} \{CoG\} \mathbf{p}_{\{F_i\}} \times \{CoG\} \mathbf{f}_{in_i} &= \mathbf{0} \\ \boldsymbol{\tau}_q + \sum_{i=1}^2 \mathbf{J}_{c_i}^T \mathbf{f}_{c_i} + \sum_{i=1}^{N_r} \mathbf{J}_{r_i}^T \mathbf{f}_{in_i} &= \mathbf{0} \end{aligned} \quad (90)$$

where  $\mathbf{f}_{in_i}$  and  $\tau_q$  are the thrust force and the joint torque generated by the robot, whereas  $\mathbf{f}_{c_i}$  is the contact force which is orthogonal to the object surface as shown in Figure 10. Equations (70) and (71) indicate that  $\mathbf{f}_{in_i}$  belongs to the null space of the thrust force that satisfies the balance with the gravity and external wrenches.

Considering the force closure of the object grasping, the contact forces can be further written as follows



**Figure 9.** Overview of the framework for object grasping using the vectored thrust force as internal wrench, which is a part of the whole system as depicted in Figure 2. Based on the statics of multilinked model (Section 4.1), the internal wrench planner calculates the desired vectored thrust force for grasping (Section 4.2). Online mass estimation method is also developed (Section 4.3) to update the robot configuration during grasping.



**Figure 10.** Statics model of bimanual object grasping.  $\mathbf{f}_{c_1}$  and  $\mathbf{f}_{c_2}$  are two contact forces.  $\mathbf{f}_{in_i}$  is the vectored thrust force which contributes to grasping as internal wrench, whereas  $\tau_{q_i}$  is joint torque for grasping.

$$\mathbf{f}_{c_1} = -\mathbf{f}_{c_2} = f_c \mathbf{n}_c \quad (91)$$

where  $f_c$  is the norm, and  $\mathbf{n}_c$  is a contact normal vector. It is notable that this pair of contact forces counteracts the mutual influence on the robot translation and rotational motion (no appearance in equations (70) and (71)).

Then the purpose of planning the grasp is to find the proper vectored thrust force  $\mathbf{f}_{in_i}$  and joint torque  $\tau_q$  to ensure sufficient contact force  $f_c$ .

## 4.2. Internal wrench planning

There are two different objectives for planning the vectored thrust force grasping: minimization of the joint torque or maximization of the contact force.

**4.2.1. Minimization of joint torques.** Based on the statics model of equation (90), the optimization problem with the objective of minimizing the joint torques along with thrust forces can be given by

$$\min_{\mathbf{F}_{in}, \tau_q} \mathbf{F}_{in}^T W_1 \mathbf{F}_{in} + \tau_q^T W_2 \tau_q \quad (92)$$

s.t. (70), (71), and (90)

$$\mathbf{F}_{in} = \begin{bmatrix} \mathbf{f}_{in_1}^T & \mathbf{f}_{in_2}^T & \cdots & \mathbf{f}_{in_{N_r}}^T \end{bmatrix}^T$$

where the first and second terms of equation (92) are introduced to balance the minimization of thrust force and joint torque by the diagonal weight matrices  $W_1$  and  $W_2$ .

A desired contact force  $\mathbf{f}_{c_i}^{\text{ref}}$  with a norm of  $f_c^{\text{ref}}$  is substituted into equation (90). Then, by substituting equation (90) into equation (92), we can obtain an optimization problem that is only related to  $\mathbf{F}_{in}$ . Then the closed-form solution for (92) can be derived using Lagrange multiplier

$$\mathbf{F}_{in} = -\left(E - \Phi A_2^T (A_2 \Phi A_2^T)^{-1} A_2\right) \Phi A_1^T W_2 \mathbf{b} \quad (93)$$

$$\Phi = (W_1 + A_1^T W_2 A_1)^{-1} \quad (94)$$

$$\mathbf{b} = \sum_{i=1}^2 J_{c_i}^T \mathbf{f}_{c_i}^{\text{ref}} \quad (95)$$

$$A_1 = \begin{bmatrix} J_{r_1}^T & J_{r_2}^T & \cdots & J_{r_{N_r}}^T \end{bmatrix} \quad (96)$$

$$A_2 = \begin{bmatrix} \hat{J}_1 & \hat{J}_2 & \cdots & \hat{J}_{N_r} \end{bmatrix}, \hat{J}_i = \begin{bmatrix} E_{3 \times 3} \\ \mathbf{p}_{\{F_i\}} \times \end{bmatrix} \quad (97)$$

where  $E$  in equation (93) is a  $3N_r \times 3N_r$  identity matrix.

The diagonal weight matrices  $W_1$  and  $W_2$  in equation (92) balance the effort of thrust force and joint torque on grasping.  $W_1 \gg W_2$  indicates that only the vectored thrust force is used to achieve grasping, and no joint torques is

used. In contrast,  $W_1 \ll W_2$  indicates that only joint torques are used for grasping, which is the case for general grippers. Given that the closed-form solution of equation (93) can be performed in realtime, no explicit boundaries for the thrust force  $\mathbf{f}_{in_i}$  are introduced as inequality constraints for equation (92). Instead, more weight for the thrust force (i.e.,  $W_1 > W_2$ ) is designed to avoid the saturation of thrust force. For our robot, the diagonal values of  $W_1$  and  $W_2$  are set as 4 and 1. Finally, the result of the vectored thrust forces  $\mathbf{f}_{in_i}$  from equation (93) is sent to the flight control framework as shown in Figure 9.

**4.2.2. Maximization of contact normal force.** Optimization problem depicted by equation (92) for realtime control is calculated with the known contact forces  $\mathbf{f}_{c_i}^{\text{ref}}$ ; however, it is also important to know the maximum contact normal force that can be generated by our multilinked aerial robot. Then the static model (90) can be further derived as follows

$$f_c J_c^T \mathbf{n}_c = -\tau_q - \sum_{i=1}^{N_r} J_{r_i}^T \mathbf{f}_{in_i} \quad (98)$$

$$J_c = J_{c_1} - J_{c_2} \quad (99)$$

The term  $J_c^T \mathbf{n}_c$  at the left side of equation (98) is constant because the contact points and joint angles are all fixed. Therefore, it is possible to obtain the maximum norm of the contact normal force  $f_c$  via following optimization problem

$$\max_{f_c, \mathbf{F}} f_c^2 \quad (100)$$

s.t. (70), (71), and (98)

$$f_{in_x}^2 + f_{in_y}^2 \leq \bar{f}^2 \quad (101)$$

$$0 \leq \tau_{q_i} \leq \tau_{\max} \quad (102)$$

Regarding the vectoring thrust force, although it is better to directly constraint the vectoring angles ( $\phi_i, \theta_i$ ) and force scalar  $\lambda_i$  for each rotor, the constraint of equation (101) that restrains the lateral force  $[f_{in_x}, f_{in_y}]$  can be considered equivalent in the case of grasping. This is because the upper limit of the lateral force  $\bar{f}$  can be approximately obtained from the maximum tilted angle  $\bar{\zeta}$  (a combined vectoring angle of  $\phi$  and  $\theta$ ) and the thrust force in the hovering state  $f_{T_0}$ :  $\bar{f} = \sin(\bar{\zeta}) f_{T_0}$ .

Given that this optimization problem contains the non-linear constraint (98) and several inequalities, the closed-form solution is not available and the realtime calculation with an onboard computation resource is significantly difficult. Instead, the numerical validation of this optimization method will be explained in Section 5.

## 4.3. Robot model modification

Another important issue in grasping is the compensation of the inertia of grasping object. The grasped object can be



treated as one of part of robot. Then, the overall mass  $m_\Sigma$  in equation (19), inertia tensor  ${}^{CoG}I_\Sigma$  in equation (20), and the rotor position  ${}^{CoG}\mathbf{p}_{\{F_i\}}$  in equation (49) should be updated for flight stability during object grasping. Although the P and I control in equations (23) and (26) can passively compensate for the model error, the compensated term in P and I control terms will render the flight unstable again once the robot releases the object.

Compared with the rotational inertia of the object, the mass estimation is more important because it affects the rotor position  ${}^{CoG}\mathbf{p}_{\{F_i\}}$  w.r.t the CoG frame, and further influences the control allocation. We first estimate the external wrench exerted on the entire CoG by the momentum-based method (De Luca and Mattone 2005):

Rewriting equations (19) and (20) with external wrench gives

$$\dot{\mathbf{P}} = M\dot{\mathbf{v}} = J\mathbf{w} + \mathbf{w}_e - N \quad (103)$$

$$M = \begin{bmatrix} m_\Sigma E_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_\Sigma \end{bmatrix},$$

$$J = \begin{bmatrix} {}^W R_{\{CoG\}} & 0_{3 \times 3} \\ 0_{3 \times 3} & E_{3 \times 3} \end{bmatrix},$$

$$N = \begin{bmatrix} m_\Sigma \mathbf{g} \\ \boldsymbol{\omega} \times I_\Sigma \boldsymbol{\omega} \end{bmatrix}$$

where,  $\mathbf{P}$  and  $\mathbf{v}$  are the generalized momentum and velocity, respectively, and  $\mathbf{w}$  and  $\mathbf{w}_{ex}$  are the overall wrench generated from the rotor thrust and the external wrench, respectively. According to De Luca and Mattone (2005), the estimated overall external wrench can be written as follows

$$\widehat{\mathbf{w}}_{ex} = K_w \left[ \mathbf{P}(t) - \mathbf{P}(t_0) - \int_{t_0}^t (J\mathbf{w} + \widehat{\mathbf{w}}_{ex} - N) dt \right] \quad (104)$$

where,  $K_w \in \mathcal{R}^{6 \times 6}$  is a positive-definite diagonal observer matrix which has a smoothing function.

Then the estimated mass can be given by

$$\widehat{m}_{obj} = \widehat{\mathbf{w}}_{ex}(3)/g \quad (105)$$

where  $\widehat{\mathbf{w}}_{ex}(3)$  is the third element of the estimated external wrench.

Grasping an object from the ground can provide a long term estimation of equation (104) which can provide better convergence to the estimated mass. However, to grasp an object handed over in the air, instantaneous mass estimation and active compensation are necessary; otherwise the flight will become unstable. Therefore, we propose Alg. 2 to find the optimal estimation mass and update the dynamics model for control in a short time. While releasing the object, the

flight stability can be ensured by simply removing the object mass from the robot dynamics model.

---

### Algorithm 2 Estimation of the mass of grasped object

---

```

t ← 0; m_max ← 0
while t < t_max do
  m_obj ← (104), (105)
  if m_obj > m_max then
    m_max ← m_obj
    update m_Σ in (19), {}^{CoG}I_Σ in (20), and {}^{CoG}\mathbf{p}_{\{F_i\}}
    in (49) with m_obj
  end if
end while
m ← m_max
return m

```

---

## 5. Experiments

In this section, we first describe the robot used in each experiment. Then, we present the experimental results, including the evaluation of flight stability based on the proposed control method, object manipulation using an end-effector, and object grasping using the vectored thrust force. Videos of the experiments are available at <https://youtu.be/fJA4Rch0biE>.

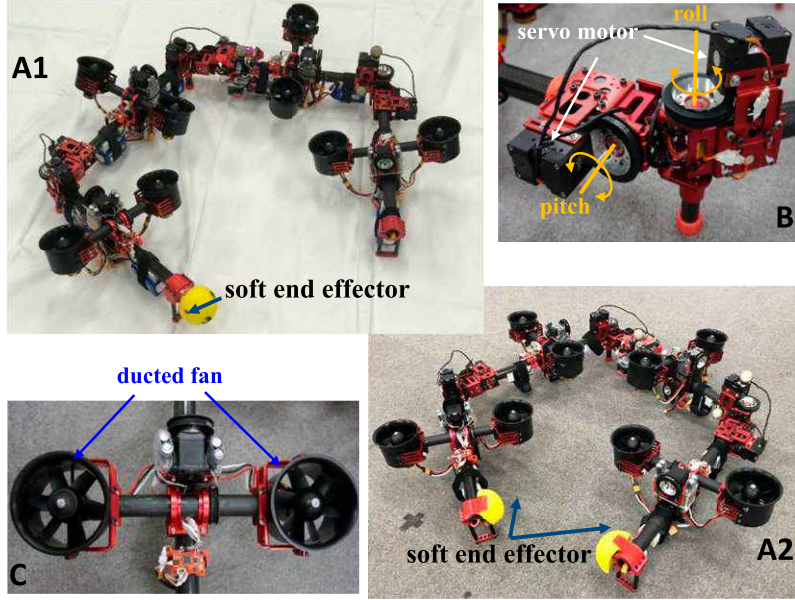
### 5.1. Robot platform

A quad-type DRAGON composed of four links was developed in our previous work (Zhao et al., 2018a). In order to perform manipulation and grasping, soft spherical end-effectors made of sponge were designed with the aim of obtaining sufficient friction and compliance at the interaction surface (two different designs were developed as shown in Figure 11 A1 and A2). The main specifications of this robot can be found in Table 1. An external motion capture system was applied in our experiment to obtain the state of the root link (i.e.,  ${}^W \mathbf{r}_{\{L_1\}}$ ,  ${}^W \dot{\mathbf{r}}_{\{L_1\}}$ ,  ${}^W R_{\{L_1\}}$ , and  ${}^W \boldsymbol{\omega}_{\{L_1\}}$ ), which were used to calculate the state of CoG motion based on forward kinematics. Furthermore, all processes related to motion planning and flight control were performed using the on-board compact computer (i.e., an Intel Atom x7-Z8700 Quad core CPU). A maximum joint speed (0.31 rad/s) was set to satisfy the quasi-static assumption in the presented control framework.

There are two LiHv batteries (1300 mAh, 22.8 V) at each link to provide power for a pair of rotors, and the maximum flight time is 3 min. The modeling and control parameters used in the experiments are summarized in Table 2.

### 5.2. Flight stability

We evaluated the feasibility of the enhanced flight control method presented in Section 3 by comparing with the



**Figure 11.** Quad-type DRAGON robot for performing the aerial manipulation and grasping. (A1/A2) Models for single end-effector manipulation and two-point grasping, respectively. The soft end-effector is composed of a sponge ball. (B) two DoF joint module composed of two identical units (torque estimation based on current). (C) Compact ducted fan rotor for vectoring control.

**Table 1.** Main specifications.

Attribute	Description
Link length	0.42 m
Total weight	7.6 kg
Max payload	3.4 kg
Max flight time	3 min
Max rotor thrust	30 N
Max joint torque	7.0 Nm
Max joint speed	0.31 rad/s
End effector	Sponge ball (radius: 0.03 mm)

performance obtained by the previous control method (Zhao et al., 2021). The crucial difference between the two methods is the use of the vectoring angles  $\phi$ .

**5.2.1. Aerial transformation.** We first testified the stability during aerial transformation from a normal form to the form

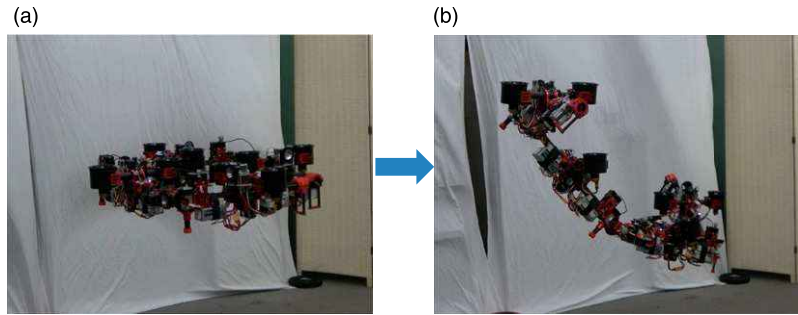
as shown in Figure 12:  $\mathbf{q} = \left[ 0 \quad \frac{\pi}{2} \quad 0 \quad \frac{\pi}{2} \quad 0 \quad \frac{\pi}{2} \right] \text{ rad} \rightarrow \left[ -0.6 \quad -\frac{\pi}{2} \quad -0.6 \quad 0 \quad 0 \quad \frac{\pi}{2} \right] \text{ rad}$ . The RMS of  $\mathbf{e}_r$  in

equation (23) were [0.07, 0.05, 0.03] m for the new method, and [0.09, 0.05, 0.03] m for the previous method; whereas the RMS of  $\mathbf{e}_R$  in equation (26) were [0.04, 0.03, 0.04] rad for the new method, and [0.03, 0.03, 0.05] rad for the previous method. Both control methods ensured sufficiently small errors in the CoG position and rotation. However, more fluctuation can be observed from the vectoring angles and the onboard gyroscope as shown in Figure 13(b) and (d) compared with the results of Figure 13(a) and (c). This demonstrates that, because the previous control method

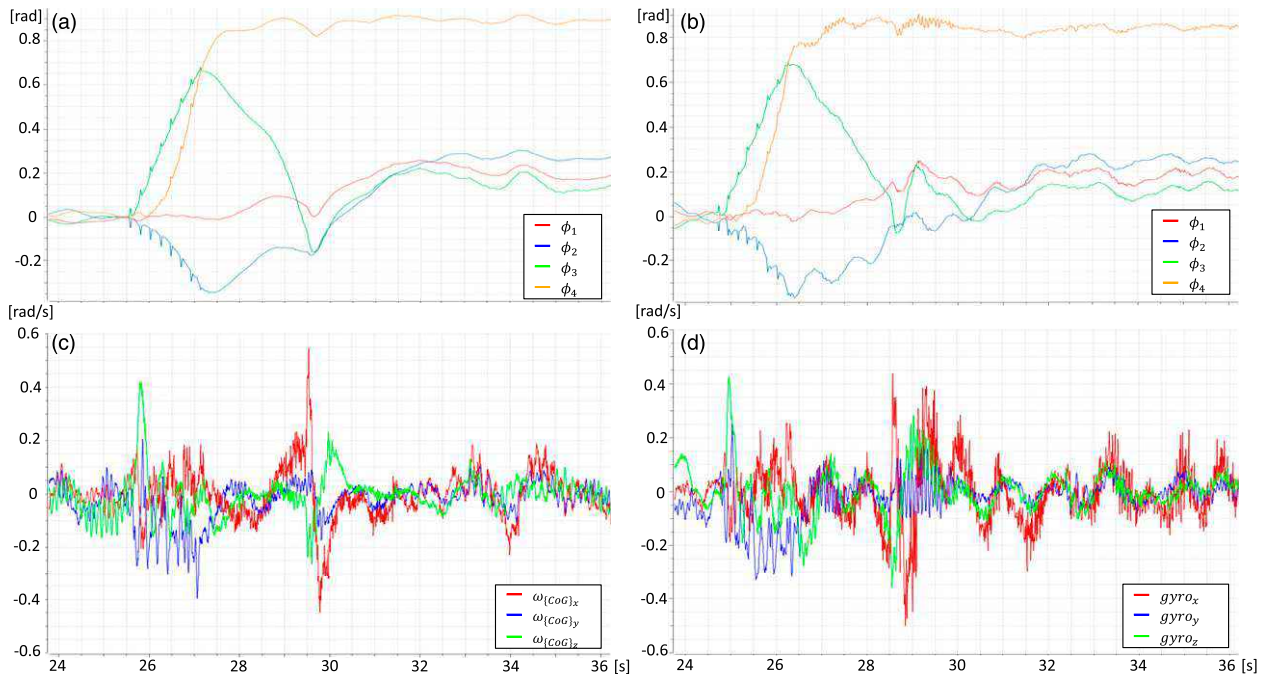
**Table 2.** Parameters for quad-type DRAGON.

Parameter	Value	Equation
$K_{f,p}$	diag (3.6, 3.6, 2.8)	(23)
$K_{f,i}$	diag (0.03, 0.03, 1.2)	
$K_{f,d}$	diag (4, 4, 2.8)	
$K_{T,p}$	diag (15, 15, 10)	(26)
$K_{T,p}$	diag (0.3, 0.3, 0.1)	
$K_{T,d}$	$5E_3 \times 3$	
$\Delta\lambda$	1.5	Alg. 1
$\Delta w$	0.05	
$\Delta\phi$	0.5	(86)
$\bar{e}_r$	0.32	(170)
$\bar{e}_\psi$	0.5	
$k_\psi$	1	
$W_1$	$0.001E_{12 \times 12}$	(173)
$W_2$	$E_3 \times 3$	
$W_1$	$4E_{12 \times 12}$	(92)
$W_2$	$E_6 \times 6$	
$\bar{f}$	30	(101)
$\tau_{\max}$	7.0	(102)
$K_w$	diag (5, 5, 5, 2.5, 2.5, 2.5)	(104)
$t_{\max}$	1	Alg. 2

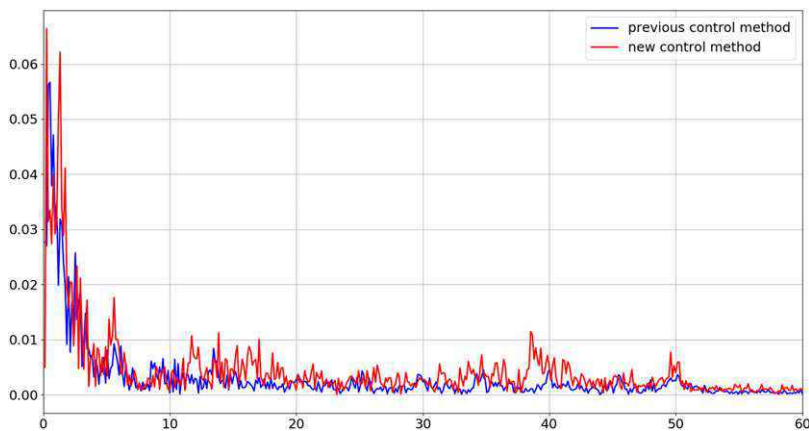
used the vectoring angles  $\phi$  for the D control terms, the dynamics of the vectoring apparatus, especially the rotational inertia, affected the dynamics of the whole body, and induced oscillation. This proved our analysis on the mechanism of oscillation caused by the vectoring actuators. We further analyzed the frequency spectrum of the gyroscope data depicted in Figure 14, which shows that the proposed new control methods can suppress the oscillation across all frequency ranges. This result demonstrated the feasibility of the new



**Figure 12.** Aerial transformation from the normal form to a unique form. The flight performance by the proposed control method (Section 3) is compared with the previous control method (Zhao et al., 2021).



**Figure 13.** Comparison the flight performance between the proposed control method (a c and e) and the previous control method (b d and f) during the transformation as shown in Figure 12. (a)/(b): changes in vectoring angles  $\phi$ , the forth angle  $\phi_4$  in (b) shows a rapid fluctuation during 28 s–30 s. (c)/(d): changes in onboard gyroscope data. (d) Shows larger fluctuation.



**Figure 14.** Frequency spectrum of onboard gyroscope data of Figure 13(c) and (d).

control method in terms of suppressing the influence of the actuator dynamics on the flight stability.

**5.2.2. Stable flight under challenging configurations.** The configuration shown in Figure 15 is a singularity in the previous method control (Zhao et al., 2021) as discussed in



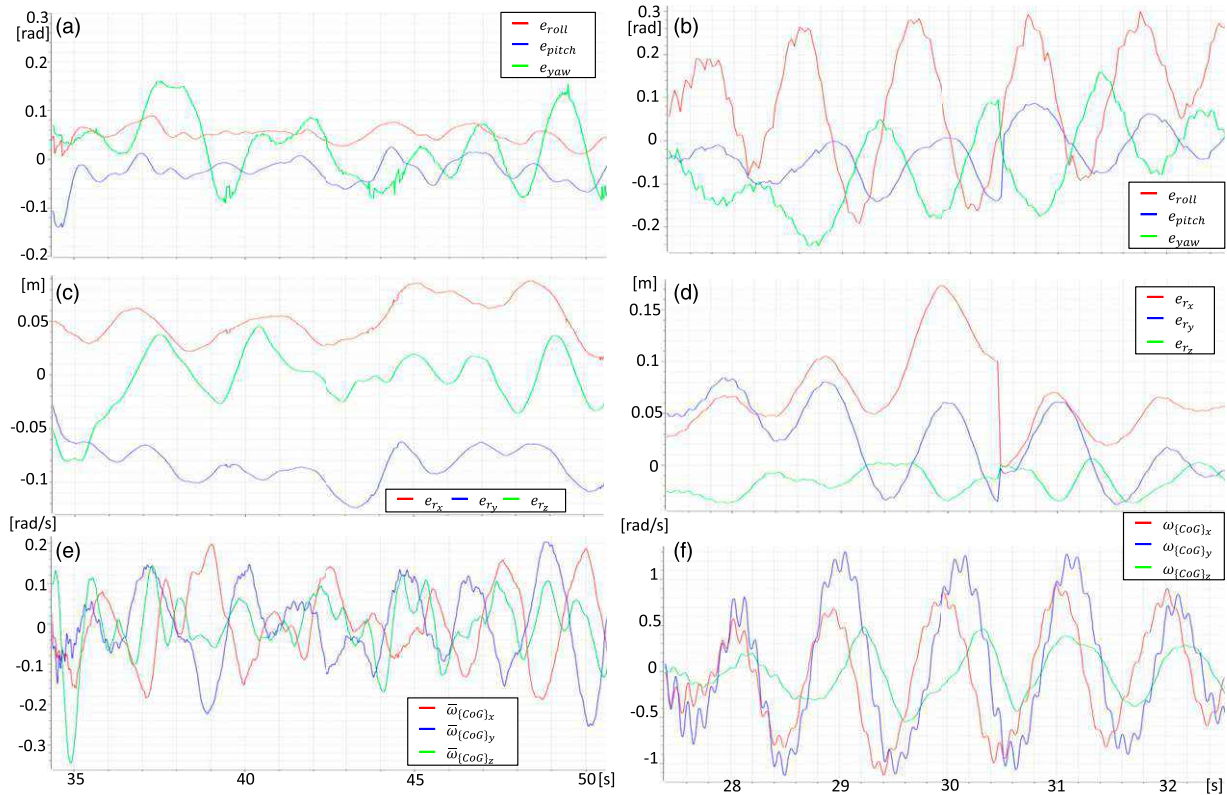
**Figure 15.** Stable hovering under the challenge configuration by the proposed control method. This configuration is a singularity in the previous control method (Zhao et al., 2021). The related video can be found in <https://youtu.be/fJA4Rch0biE>.

Section 3.1.3. This can be confirmed from Figure 16(b), (d) and (f), where the rotational motion oscillated significantly around the  $x$  axis (axes are depicted in Figure 6(c)). In contrast, the proposed control method can solve this singularity as discussed in Figure 8. Thus, stable hovering under this challenging configuration has been achieved as shown in Figure 15 and Figure 16(a), (c) and (e), and the RMS of  $e_r$  and  $e_R$  were  $[0.06, 0.08, 0.02]$  m, and  $[0.05, 0.03, 0.07]$  rad, respectively. This result further demonstrated the effectiveness of the control framework of Figure 5 in suppressing the influence of vectoring actuator on the flight stability.

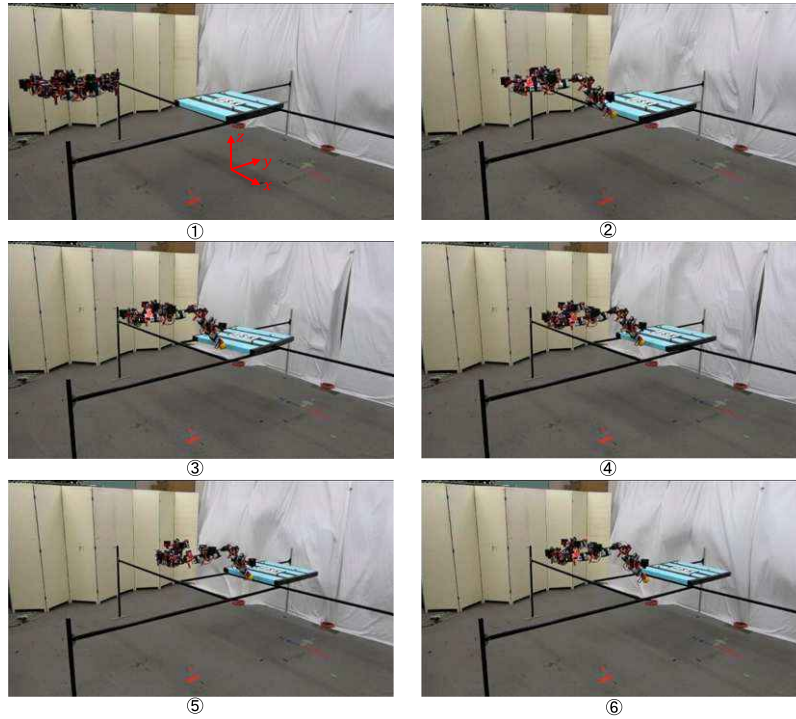
### 5.3. Object manipulation

The framework of the grasplless manipulation is presented in Appendix E. We extended the previous work (Zhao et al., 2020) by adding a feedback loop with the object pose to update the end-effector position and the desired external wrench in realtime.

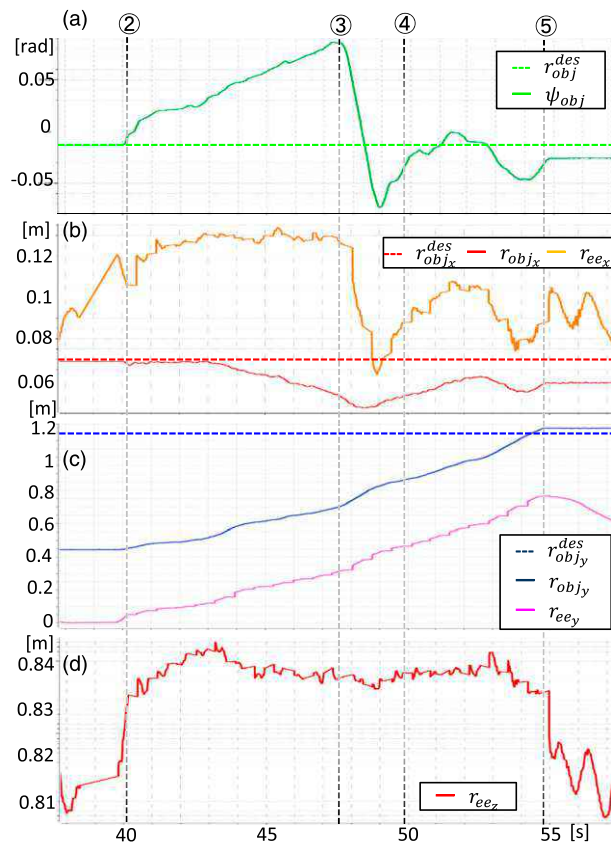
Manipulation of a rectangular plate was evaluated using our robot. The target plate was considered a hatch cover (e.g., manhole cover) that was required to be moved by the robot. Since there was no rail under the plate, the motion of this plate belonged to  $SE(2)$ , which indicated the necessity of orientation manipulation. Several parameters were assumed be known in the online manipulation planning phase.



**Figure 16.** Comparison the flight performance between the proposed control method (a, c, and e) and the previous control method (b, d, and f) under the challenging pose as shown in Figure 15. (a)/(b) Changes in the rotational errors which are converted to RPY Euler angles ( $e_R \rightarrow [e_{roll}, e_{pitch}, e_{yaw}]$ ). (c)/(d) Changes in the positional errors  $e_r$ . (e)/(f): changes in the angular velocity of the  $\{CoG\}$  frame.



**Figure 17.** Moving the hatch plate from the side. The orientation of plate was kept constant. Experiment video is available at <https://youtu.be/fJA4Rch0biE>.



**Figure 18.** Plots related to Figure 17. (a) Object orientation  $\psi_{obj}$ . The target angle  $\psi_{obj}^{des}$  is the initial value. (b) Position of the end-effector and the object along the  $x$  axis of the inertia frame as depicted in Figure 17. The target object position  $r_{obj_x}^{des}$  is the initial value. (c) Position of the end-effector and the object along the  $y$  axis of the inertia frame. The target object position  $r_{obj_y}^{des}$  is 1.15 m. (d) Position of the end-effector along the  $z$  axis of the inertia frame.

For instance, the plate width was 0.8 m, and the weight plate was 1 kg. The friction force ( $f_{fric}$  in equation (177)) was measured, which was around 3.5 N. The object pose was measured from the motion capture system.

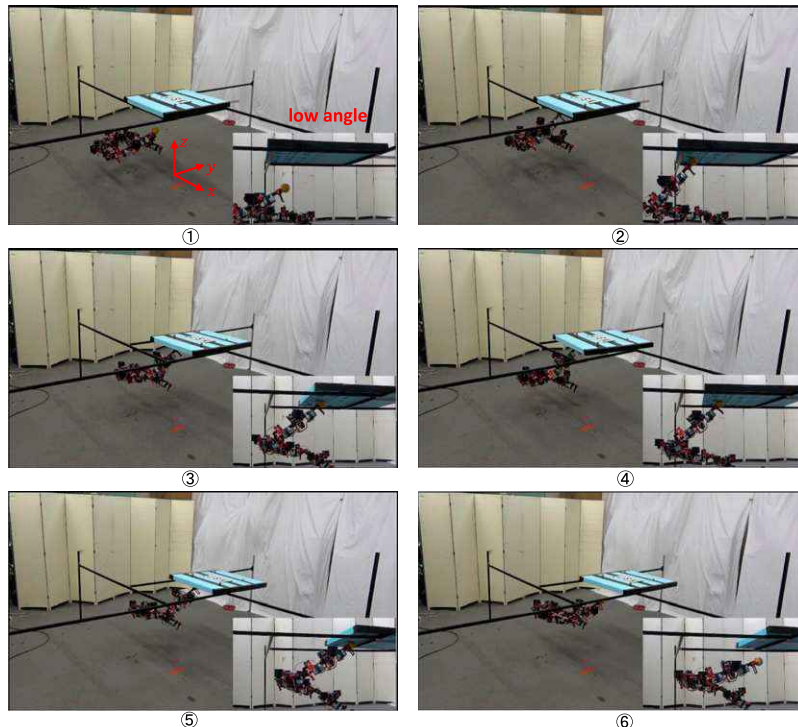
Two different experiments were performed according to the contact point on the plate.

**5.3.1. Case1: Manipulation at the side of the plate.** In this case, the contact point is set as the side of the plate with a thickness of 0.05 m, as shown in Figure 17. The approach trajectory to this contact point (① and ② in Figure 17) was planned in realtime according to the object pose, which was based on the planning method proposed in Zhao et al. (2020). The contact between the end-effector and the object side demonstrated the precise end-effector control by the proposed control method. Then, during ②–⑤, the plate was pushed by the end-effector, and the goal pose of object for end-effector position control (equations (171) and (172)) was  $r_{obj}^{\{des\}} = [0.07 \ 1.15] \text{ m}$ ,  $\psi_{obj}^{\{des\}} = 0 \text{ rad}$ , which was a pose moving parallel from the initial pose

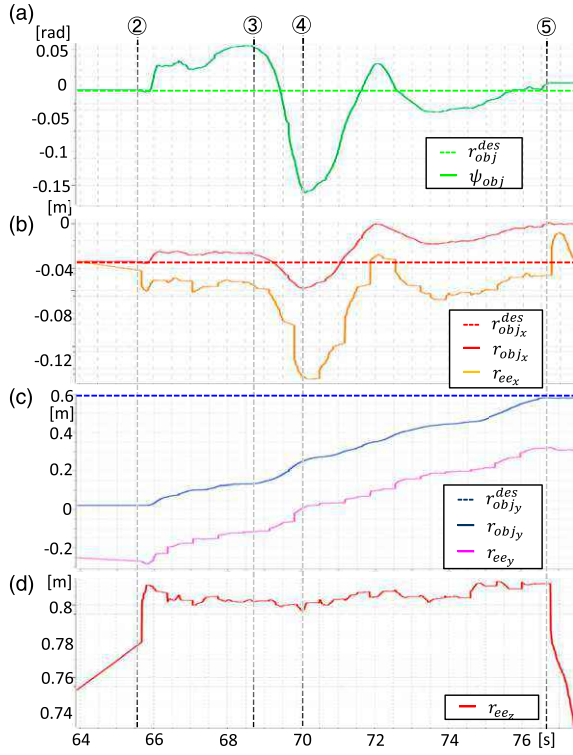
along the  $y$  axis. In terms of the external force for manipulation, since the contact force was parallel to the friction force, the reaction force  $f_{\eta}$  in equation (177) was set as zero. Finally, the plate was successfully moved away from the hatch, and the orientation of the plate was kept almost constant during the manipulation. As shown in Figure 18(a), the deviation of the plate orientation  $\psi_{obj}$  once increased up to 0.08 rad. However, the end-effector position  $r_{ee_x}$  was updated to compensate for this deviation as shown in Figure 18(b), and the rotational deviation was convergent and close to zero at the end. The object position also reached the goal position as shown in Figure 18(b) and (c). The small change in the end-effector height  $r_{ee_z}$  as depicted in Figure 18(d) also showed the steady contact between object and end-effector during manipulation. Although the previous motion planning method (Zhao et al., 2020) can also achieve physical interaction, there was no closed loop to control the object pose. Thus, the plate was moved with an undesirable orientation. The comparison of the manipulation

**Table 3.** Comparison of object manipulation performance.

Experiment	Case1	Case2	Case3
Control method	Zhao et al. (2018a)	Proposed method (Section 3)	Proposed method (Section 3)
Planning method	Zhao et al. (2020)	Zhao et al. (2020)	Appendix E
Final error of Side contact (Figure 17)	0.70	0.26	0.03
$ \psi_{obj} - \psi_{obj}^{des} $ [rad] Bottom contact (Figure 19)	0.20	0.10	0.02



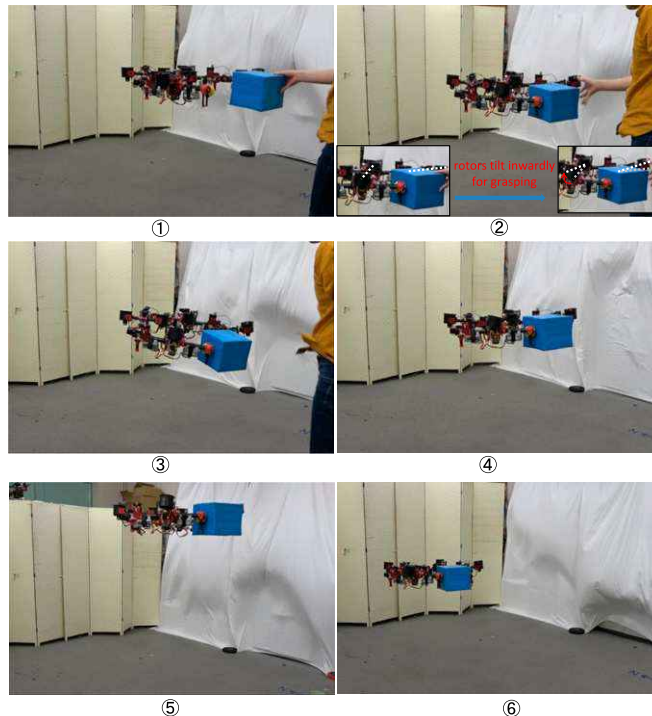
**Figure 19.** Moving the plate from the bottom. The orientation of plate was kept constant. Experiment video is available at <https://youtu.be/EJA4Rch0biE>.



**Figure 20.** Plots related to Figure 19. (a) Object orientation  $\psi_{obj}$ . The target angle  $\psi_{obj}^{des}$  is the initial value. (b) Position of the end-effector and the object along the  $x$  axis of the inertia frame as depicted in Figure 19 ①. The target object position  $r_{obj_x}^{des}$  is the initial value. (c) Position of the end-effector and the object along the  $y$  axis of the inertia frame. The target object position  $r_{obj_y}^{des}$  is 0.6 m. (d) Position of the end-effector along the  $z$  axis of the inertia frame.

performance using different flight control and motion planning methods is summarized in Table 3. A significant improvement in rotational manipulation of the object can be confirmed. This experimental result demonstrated the feasibility of the proposed online motion planning for manipulating an object in more than two DoFs.

**5.3.2. Case2: Manipulation at the bottom of the plate.** In this case, the robot was under the plate as shown in Figure 19. Then, the movement strategy was modified to pull the bottom surface of the plate. As such, it was necessary to impose a certain contact normal force (i.e.,  $f_\eta$  in equation (177)) on the surface to ensure steady contact. The contact point was set close to the edge of the plate; therefore, this force was set as half of the plate weight ( $\|f_\eta\| = \frac{10}{2} N$ ) to reduce the contact area between the plate and the black frames. The goal pose of object for end-effector position control (equations (171) and (172)) was  $r_{obj}^{des} = [-0.04 \ 0.6] m$ ,  $\psi_{obj}^{des} = 0$  rad. A relatively large deviation of  $-0.15$  rad in rotational error can be confirmed in ④ of in Figure 19 and in Figure 20(a), which then decreased quickly owing to the end-effector position control and force control as presented in Appendix E. Finally, the plate was successfully moved away from the hatch as shown in Figure 20(b) and (c), and the final orientation of the plate was nearly identical to the initial value as shown in Figure 20(a). The superior performance can be also confirmed by comparing with the results obtained from the previous method (Zhao et al., 2020), as shown in Table 3.



**Figure 21.** Grasping a box-shape object (0.3 kg) using vectored thrust force instead of joint torque. The object mass was unknown and was estimated and compensated by robot based on Alg. 2. The experiment video is available at <https://youtu.be/fJA4Rch0biE>.

Two experiments required the robot to manipulate the object from the differential direction and using a differential robot pose, which demonstrated not only the successful integration of external wrench into our propose control method, but also the dexterity of our articulated model in manipulation.

#### 5.4. Object grasping

**5.4.1. Evaluation of the maximum grasping force.** According to the optimization problem as presented in equations (100)–(102), we first investigated the maximum contact force that could be generated in the model shown in Figure 11(A2). The joint angles  $\mathbf{q}$  were first calculated based on a given target object of width 0.1 m. Since the grasping was performed in a two dimensional manner, all of the pitch joints (see Figure 11(B)) were set as zero. In addition, the center yaw joint  $q_{2\_yaw}$  was fixed as  $\frac{\pi}{2}$  rad, and the four points (i.e., two contact points, first and third joint points) formed a trapezoid. Then, the three yaw joint angles were calculated as  $[1.53, \frac{\pi}{2}, 1.36]$  rad. Besides, the boundary of the vectoring force  $\bar{f}$  was set as 10 N, which rendered the maximum tilted angle  $\bar{\psi}$  of approximately 0.45 rad in the hovering state. Finally, the maximum contact force  $f_c$  in equation (98) was 21.3 N, which was almost twice that of the maximum value generated by the joint torques alone (i.e., 10.7 N). Although applying a lower gearing can produce much greater joint torques (and thus grasping forces), a major disadvantage is the difficulty of handling the high torque load by a compact gearbox. Furthermore, our rotor vectoring apparatus demonstrates the possibility of replacing the joint servo with a fully passive joint structure for grasping. However, development of the control methodology to handle multi-rigid-body dynamics based on passive joints is necessary, which is one of the challenges of future work.

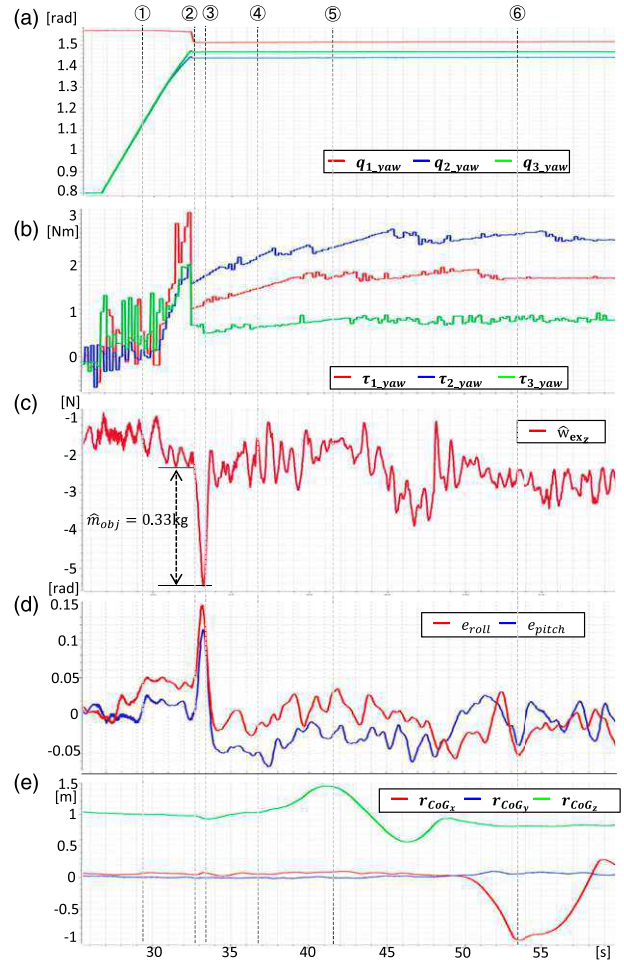
**5.4.2. Grasping and transporting objects.** As mentioned in Section 4, calculating the maximum contact force for grasping involves extensive computation. Therefore, to achieve online grasping planning, calculation of the vectoring thrust force for grasping from equation (93) to relax the load of joint torques was performed on the onboard computer.

The target object was handed over to the robot when the robot form was opened ( $\mathbf{q} = [0, \frac{\pi}{4}, 0, \frac{\pi}{2}, 0, \frac{\pi}{4}]$  rad). After a trigger was sent to the robot, the robot gradually increased the joint angles of  $q_{1\_yaw}$  and  $q_{3\_yaw}$ . Meanwhile, the robot performed contact detection by checking the joint torque on  $q_{2\_yaw}$  with a preset threshold. The joint torque was estimated from the current of each servo motor. As observed from the general hovering test, the change in the joint torque on  $q_{2\_yaw}$  was less than 1.5 Nm. Therefore, the threshold for contact detection was set as 1.8 Nm. Once the contact was detected, the additional vectoring force  $\mathbf{f}_{in}$  is continuously calculated from equation (93) with the actual joint angles and added to the proposed control framework as shown in

Figure 9. To generate enough contact force on the object, the desired norm of the contact force  $\|\mathbf{f}_c^{ref}\|$  was set as 10 N. It is also notable that the object mass was unknown, and thus mass estimation based on Alg. 2 was also performed after grasping

The experiments were performed with two different objects:

- **Box.** The box mass was 0.32 kg and the width was 0.2 m, as shown in Figure 21. The robot detected contact with the object in ②. Then the joint angles of  $q_{i\_yaw}$  were fixed at  $[1.51, 1.47, 1.44]$  rad as shown in Figure 22(a). The robot instantaneously calculated and applied the vectored thrust force  $\mathbf{f}_{in}$  for grasping based on the joint angles. The rotor then tilted inwardly to use the thrust force as an internal wrench for grasping as depicted in Figure 21 ②. According to the grasping planning method detailed in equations (92)



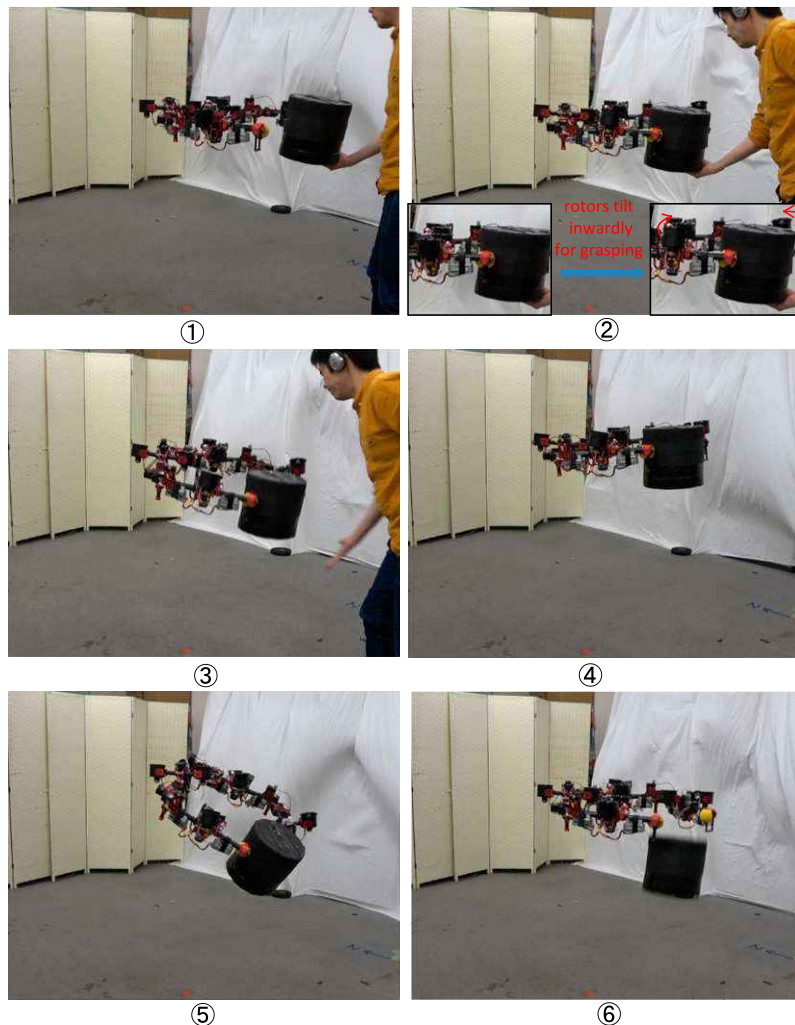
**Figure 22.** Plots related to Figure 21. (a) Changes in the joint angles of  $q_{i\_yaw}$  from the pre-grasping phase (①) to the grasping phase (②→③) and to the transportation phase (④→⑥). (b) Changes in the joint torques for  $q_{i\_yaw}$ . (c) Change in the estimated wrench  $\hat{w}_{ext}$  for object mass estimation. (d) Changes in the robot rotational errors which are converted to RPY Euler angles. (e) Changes in the robot position  $r_{\{CoG\}}$ .



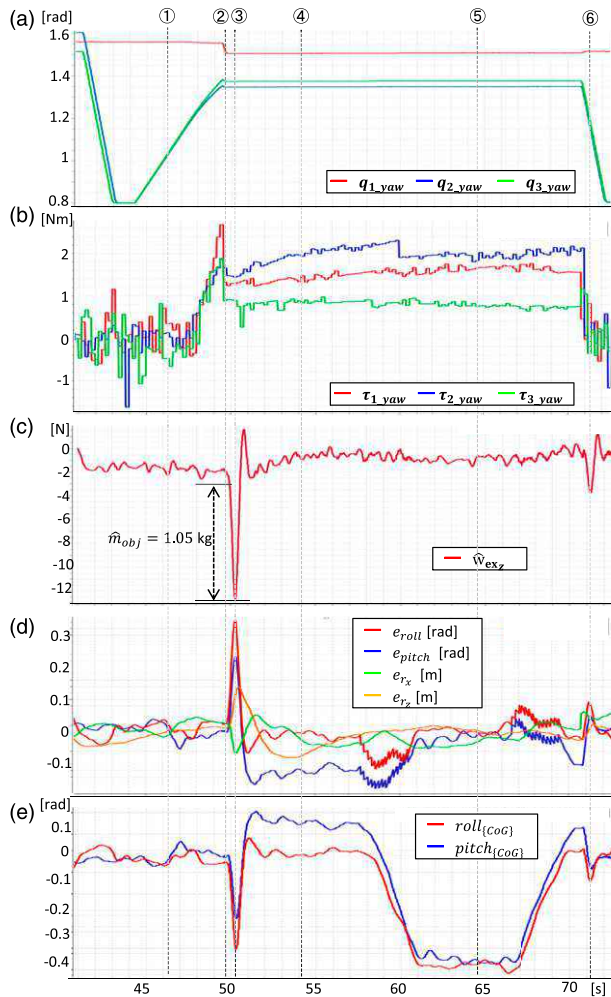
and (93), if we only use the joint torques to grasp, the necessary torques for  $q_{i\_yaw}$  would be [4.22, 5.83, 1.4] Nm, where the joint of  $q_{2\_yaw}$  required a relatively large load. However, with the help of the additional vectoring force for grasping, these joint torques could be significantly reduced to [2.95, 3.26, 0.78] Nm. During grasping, the average of the actual joint torque are [1.65, 2.56, 0.77] Nm as shown in Figure 22(b), which were all smaller than the planned (predicted) values. Nevertheless, the steady grasp can be confirmed from Figure 21, which involves a variety of motion such as ascending/descending (38–48 s in Figure 22(d)) and horizontal movement (50–60 s in Figure 22(e)). In terms of object mass estimation, after the object was released by human hand at ②, the robot orientation diverged quickly as shown in Figure 21 ③ and Figure 22(d), as the object mass was unknown at first. However, Alg. 2 was performed at the same time to estimate the object mass and update the robot model, which contributed to the rapid

recovery from this divergence (less than 1 s). Besides, the estimation result was 0.33 kg which was significantly close to the ground truth of 0.32 kg. It is notable that there was an offset and drift in the estimated wrench  $\widehat{w}_{ex}$ , as shown in Figure 22(c) because of the model error inside the dynamics modeling; however, the relative change in the estimated wrench demonstrated high reliability in mass estimation.

- **Cylinder.** The weight was 1.0 kg (much heavier than the box-shape object), and its diameter was 0.125 m as shown in Figure 23. Most of the phenomena were similar to those of the box case, such as the suppression of joint torque for grasping as depicted in Figure 24(b). However, since the object was relatively heavy, after the object was released from human hand, the relatively large divergence in the robot height and position can be confirmed from Figure 24(d). Nevertheless, the object mass was rapidly estimated and compensated by the robot, and the flight stability was recovered quickly within 1 s. Besides, the mass estimation result was



**Figure 23.** Grasping a cylinder-shape object (1 kg) using vectored thrust force instead of joint torque. The object mass was unknown and was estimated and compensated by robot based on Alg. 2. The experiment video is available at <https://youtu.be/fIA4Rch0biE>.



**Figure 24.** Plots related to Figure 23. (a) Changes in the joint angles of  $q_{i\_yaw}$  involving the pre-grasping phase (①), the grasping phase (②→④), the rotating phase (⑤), and release phase (⑥). (b) Changes in the joint torques for  $q_{i\_yaw}$ . (c) Change in the estimated wrench  $\hat{w}_{exz}$  for object mass estimation. (d) Changes in the robot rotational errors which are converted to RPY Euler angles. (e) Changes in the robot position  $r_{\{CoG\}}$ .

1.05 kg which was also considerably close to the ground truth of 1 kg. It is also notable that, during the aggressive rotational divergence, the object was held steadily by the robot, which indicating the effectiveness of the vectored thrust force for grasping. After the robot was back to the hovering state, we then actively changed the robot orientation as shown in Figure 23 ⑤ and 58–70 s in Figure 24(e). The object can be still held stably by the robot, which demonstrated the potential to change the orientation of a large object by grasping-based manipulation.

These two experiments demonstrated not only the effectiveness of the proposed grasping planning method, but also the feasibility of the proposed control method in terms of handling an additional vectored thrust force as the internal wrench.

## 6. Conclusions and future work

In this paper, we presented a comprehensive framework including the flight control, object manipulation, and grasping for an articulated aerial robot named DRAGON that has rotors embedded in each link and is able to change the force direction via a vectoring mechanism. First, we revealed the mechanism of oscillation caused by the rotational inertia of vectoring actuators, and further developed a two-stage control allocation to suppress the influence of actuators dynamics on the flight stability. This control framework can also handle the external wrench for manipulation and internal wrench for grasping. Second, we proposed online thrust-level optimization method for object grasping, which utilizes the vectorable thrust force rather than the joint torque as an internal wrench. The related experiments, including (1) flight stability evaluation; (2) manipulating a plate that has surface contact with the environment; and (3) grasping an object using the vectored thrust force instead of the joint torque, demonstrated the feasibility of each control and planning methods, and the consistency of the entire framework.

One key issue that remains in this work is the recognition of the target object by using onboard sensors such as an RGB sensor. Further, the estimation of the external wrench during object manipulation% is another crucial issue. A straightforward measurement by the force sensor attached to the end-effector is an effective solution; however, a hybrid external wrench observer (Shi et al., 2019) which combines the momentum-based method presented in Section 4.3 and the distributed IMU system can be applied to our articulated robot to avoid the additional payload owing to the force sensor. Last but not the least, the efficiency of the thrust usage in DRAGON is considerably high due to the omnidirectional operation of the individual rotor; however, the energy efficiency of the entire system is lower than that of a general multirotor of the same size, because there are much more actuators and metal components for this articulated model, leading to a significantly shorter flight duration compared to the general multirotor. In terms of power supply, wired charging from a tether can be an effective solution. From the perspective of mechanical design, free joints for the link connection can be investigated to reduce the overall weight, and further, it is necessary to develop a control methodology that can handle non-approximated multibody dynamics in the air.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by JSPS KAKENHI Grant Number JP19H04188.

### ORCID iD

Moju Zhao  <https://orcid.org/0000-0001-8361-5825>

## References

- Aiyama Y, Inaba M and Inoue H (1993) Pivoting: a new method of grasplless manipulation of object by robot fingers. In: Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '93), Yokohama, Japan, 26–30 July 1993, pp. 136–143.
- Alexis K, Darivianakis G, Burri M, et al. (2016) Aerial robotic contact-based inspection: planning and control. *Autonomous Robots* 40(4): 631–655.
- Allenspach M, Bodie K, Brunner M, et al. (2020) Design and optimal control of a tiltrotor micro-aerial vehicle for efficient omnidirectional flight. *The International Journal of Robotics Research* 39(10–11): 1305–1325.
- Anzai T, Zhao M, Murooka M, et al. (2019) Design, modeling and control of fully actuated 2d transformable aerial robot with 1 dof thrust vectorable link module. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019, pp. 2820–2826.
- Bodie K, Brunner M, Pantic M, et al. (2019) An omnidirectional aerial manipulation platform for contact-based inspection. In: Proceedings of Robotics: Science and Systems (RSS), Freiburg/Breisgau, Germany, 22–26 June 2019.
- Bonatti R, Wang W, Ho C, et al. (2020) Autonomous aerial cinematography in unstructured environments with learned artistic decision-making. *Journal of Field Robotics* 37(4): 606–641.
- Bonyan Khamseh H, Janabi-Sharifi F, Abdessameud A, et al. (2018) Aerial manipulation – A literature survey. *Robotics and Autonomous Systems* 107: 221–235.
- Brescianini D and D'Andrea R (2016) Design, modeling and control of an omni-directional aerial vehicle. In: Proceedings of 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016, pp. 3261–3266.
- Brescianini D and D'Andrea R (2018) Computationally efficient trajectory generation for fully actuated multirotor vehicles. *IEEE Transactions on Robotics* 34(3): 555–571.
- de Luca A and Mattone R (2005) Sensorless robot collision detection and hybrid force/motion control. In: 2005 IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain, 18–22 April 2005, pp. 999–1004.
- Doitsidis L, Weiss S, Renzaglia A, et al. (2012) Optimal surveillance coverage for teams of micro aerial vehicles in gps-denied environments using onboard vision. *Autonomous Robots* 33(1): 173–188.
- Faessler M, Falanga D, Scaramuzza D, et al. (2017) Thrust mixing, saturation, and body-rate control for accurate aggressive quadrotor flight. *IEEE Robotics and Automation Letters* 2(2): 476–482.
- Falanga D, Kleber K, Mintchev S, et al. (2019) The foldable drone: a morphing quadrotor that can squeeze and fly. *IEEE Robotics and Automation Letters* 4(2): 209–216.
- Falanga D, Mueggler E, Faessler M, et al. (2017) Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision. In: Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Marina Bay Sands, Singapore, 29 May–3 June 2017. pp. 5774–5781.
- Floreano D and Wood RJ (2015) Science, technology and the future of small autonomous drones. *Nature* 521: 460–466.
- Fumagalli M, Naldi R, Macchelli A, et al. (2014) Developing an aerial manipulator prototype: physical interaction with the environment. *IEEE Robotics Automation Magazine* 21(3): 41–50.
- Gabrich B, Saldaña D, Kumar V, et al. (2018) A flying gripper based on cuboid modular robots. In: Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018, pp. 7024–7030.
- Heredia G, Jimenez-Cano A, Sanchez I, et al. (2014) Control of a multirotor outdoor aerial manipulator. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, IL, 14–18 September 2014, pp. 3417–3422.
- Kamel M, Verling S, Elkhatib O, et al. (2018) The Voliro omniorientational hexacopter: an agile and maneuverable tilttable-rotor aerial vehicle. *IEEE Robotics Automation Magazine* 25(4): 34–44.
- Kim S, Seo H and Kim HJ (2015) Operating an unknown drawer using an aerial manipulator. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, 26–30 May 2015, pp. 5503–5508.
- Kumar V and Michael N (2012) Opportunities and challenges with autonomous micro aerial vehicles. *International Journal of Robotics Research* 31(11): 1279–1291.
- Lee T, Leok M and McClamroch NH (2010) Geometric tracking control of a quadrotor UAV on SE(3). In: 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, 15–17 December 2010, pp. 5420–5425.
- Lippiello V, Cacace J, Santamaria-Navarro A, et al. (2016) Hybrid visual servoing with hierarchical task composition for aerial manipulation. *IEEE Robotics and Automation Letters* 1(1): 259–266.
- Loianno G, Brunner C, McGrath G, et al. (2017) Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and IMU. *IEEE Robotics and Automation Letters* 2(2): 404–411.
- Mellinger D, Lindsey Q, Shomin M, et al. (2011) Design, modeling, estimation and control for aerial grasping and manipulation. In: Proceedings of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, 25–30 September 2011, pp. 2668–2673.
- Michael N, Shen S, Mohta K, et al. (2012) Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics* 29: 832–841.
- Nakamura Y and Hanafusa H (1986) Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measurement, and Control* 108(3): 163–171.
- Nguyen H and Alexis K (2021) Forceful aerial manipulation based on an aerial robotic chain: Hybrid modeling and control. *IEEE Robotics and Automation Letters* 6(2): 3711–3719.
- Orsag M, Korpela C, Bogdan S, et al. (2017) Dexterous aerial robots-mobile manipulation using unmanned aerial systems. *IEEE Transactions on Robotics* 33(6): 1453–1466.

Ozaslan T, Loianno G, Keller J, et al. (2017) Autonomous navigation and mapping for inspection of penstocks and tunnels with mavcs. *IEEE Robotics and Automation Letters* 2(3): 1740–1747.

Park S, Her J, Kim J, et al. (2016) Design, modeling and control of omni-directional aerial robot. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016, pp. 1570–1575.

Park S, Lee J, Ahn J, et al. (2018) ODAR: aerial manipulation platform enabling omnidirectional wrench generation. *IEEE/ASME Transactions on Mechatronics* 23(4): 1907–1918.

Park S, Lee Y, Heo J, et al. (2019) Pose and posture estimation of aerial skeleton systems for outdoor flying. In: 2019 International Conference on Robotics and Automation (ICRA), Montreal, Canada, 20–24 May 2019, pp. 704–710.

Riviere V, Manecy A, Viollet S, et al. (2018) Agile robotic fliers: a morphing-based approach. *Soft Robotics* 5(5): 541–553.

Ryll M, Muscio G, Pierri F, et al. (2019) 6D interaction control with aerial robots: the flying end-effector paradigm. *International Journal of Robotics Research* 38(9): 1045–1062.

Scholten JJJ, Fumagalli M, Stramigioli S, et al. (2013) Interaction control of an UAV endowed with a manipulator. In: Proceedings of 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013, pp. 4910–4915.

Shi F, Zhao M, Anzai T, et al. (2019) External wrench estimation for multilink aerial robot by center of mass estimator based on distributed imu system. In: 2019 International Conference on Robotics and Automation (ICRA), Montreal, Canada, 20–24 May 2019, pp. 1891–1897.

Shi F, Zhao M, Murooka M, et al. (2020) Aerial regrasping: Pivoting with transformable multilink aerial robot. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 200–207.

Tognon M, Chávez HAT, Gasparin E, et al. (2019) A truly-redundant aerial manipulator system with application to push-and-slide inspection in industrial plants. *IEEE Robotics and Automation Letters* 4(2): 1846–1851.

Trujillo MA, Martínez-de Dios JR, Martín C, et al. (2019) Novel aerial manipulator for accurate and robust industrial NDT contact inspection: a new tool for the oil and gas inspection industry. *Sensors* 19(6): 1305.

Yang H, Park S, Lee J, et al. (2018) LASDRA: large-size aerial skeleton system with distributed rotor actuation. In: Proceedings of 2018 IEEE International Conference on Robotics and Automation, Brisbane, Australia, 21–25 May 2018, pp. 7017–7023.

Yüksel B, Mahboubi S, Secchi C, et al. (2015) Design, identification and experimental testing of a light-weight flexible-joint arm for aerial physical interaction. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, 26–30 May 2015, pp. 870–876.

Zhao M, Anzai T, Shi F, et al. (2018a) 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 Robotics and Automation Letters* 3(2): 1176–1183.

Zhao M, Kawasaki K, Anzai T, et al. (2018b) Transformable multirotor with two-dimensional multilinks: modeling, control, and whole-body aerial manipulation. *International Journal of Robotics Research* 37(9): 1085–1112.

Zhao M, Okada K, Inaba M, et al. (2021) Enhanced modeling and control for multilinked aerial robot with two dof force vectoring apparatus. *IEEE Robotics and Automation Letters* 6(1): 135–142.

Zhao M, Shi F, Anzai T, et al. (2018c) Flight motion of passing through small opening by DRAGON: transformable multilinked aerial robot. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018, pp. 4735–4742.

Zhao M, Shi F, Anzai T, et al. (2020) Online motion planning for deforming maneuvering and manipulation by multilinked aerial robot based on differential kinematics. *IEEE Robotics and Automation Letters* 5(2): 1602–1609.

## Appendix A. Linear and angular momentum of the multilinked model

This appendix presents the derivation of linear and angular momentum  ${}^{\{W\}}P_{\Sigma}$  and  ${}^{\{CoG\}}\mathcal{L}_{\Sigma}$  in equations (16) and (17).

$${}^{\{W\}}P_{\Sigma} = \sum_{i=1}^{N_r} \left( {}^{\{W\}}P_{L_i} + {}^{\{W\}}P_{A_{r_i}} + {}^{\{W\}}P_{A_{p_i}} \right) \quad (106)$$

$${}^{\{W\}}\mathcal{L}_{\Sigma} = \sum_{i=1}^{N_r} \left( {}^{\{W\}}\mathcal{L}_{L_i} + {}^{\{W\}}\mathcal{L}_{A_{r_i}} + {}^{\{W\}}\mathcal{L}_{A_{p_i}} \right) \quad (107)$$

where  $\mathcal{L}_i$ ,  $A_{r_i}$ , and  $A_{p_i}$  stand for the  $i$ -th link component, the component between the vectoring angles of  $\phi_i$  and  $\theta_i$ , and the component of dual rotors, respectively, as shown in Figure 3.

Then, the linear and angular momentums for each component can be further given by

$${}^{\{W\}}P_{s_i} = m_{s_i} {}^{\{W\}}\dot{\mathbf{r}}_{s_i} \quad (108)$$

$${}^{\{CoG\}}\mathcal{L}_{s_i} = {}^{\{CoG\}}\mathbf{p}_{s_i} \times (m_{s_i} {}^{\{CoG\}}\dot{\mathbf{p}}_{s_i}) + {}^{\{CoG\}}R_{\{s_i\}I_{s_i}} {}^{\{CoG\}}R_{\{s_i\}}^T {}^{\{CoG\}}\boldsymbol{\omega}_{\{s_i\}} \quad (109)$$

$$s \in \{L, A_r, A_p\}$$

where  ${}^{\{W\}}\mathbf{r}_{s_i}$  in equation (108) is the CoG position of each segment  $s_i$  w.r.t. the frame  $\{W\}$ . Equation (109) denotes two part of angular momentum: the offset from the origin of the frame  $\{CoG\}$  to the CoG of segment, and the segment inertia term.  ${}^{\{CoG\}}\mathbf{p}_{s_i}$  is the CoG position of each segment w. r.t the frame  $\{CoG\}$ .  ${}^{\{CoG\}}R_{\{s_i\}}$  is the orientation of the local frame for each segment w.r.t. the frame  $\{CoG\}$ , and  ${}^{\{CoG\}}\boldsymbol{\omega}_{\{s_i\}}$  is the angular velocity of the local frame w.r.t. the frame  $\{CoG\}$ .  $I_{s_i}$  is the constant inertia tensor expressed in the local frame  $\{S_i\}$ .

It is also notable that the joint angles  $\mathbf{q}$  and vectoring angles  $\boldsymbol{\phi}$ ,  $\boldsymbol{\theta}$  have influence on the position of  $\{{}^W\}\mathbf{r}_{s_i}$  and  $\{{}^{CoG}\}\mathbf{p}_{s_i}$ . Thus, the derivatives of these positions  $\{{}^W\}\dot{\mathbf{r}}_{s_i}$  and  $\{{}^{CoG}\}\dot{\mathbf{p}}_{s_i}$ , and the angular velocity  $\{{}^{CoG}\}\boldsymbol{\omega}_{\{s_i\}}$  are also determined by the joint velocities  $\dot{\mathbf{q}}$ , and the vectoring velocities  $\dot{\boldsymbol{\phi}}$ ,  $\dot{\boldsymbol{\theta}}$ .

## Appendix B. Approximation of the dynamics of multilinked model

This appendix presents the derivation from equations (16) and (17) to equations (19) and (20).

### Entire translational dynamics

We focus on the derivative of the linear momentum

$$\{{}^W\}\dot{\mathbf{P}}_{s_i} = m_{s_i} \{{}^W\}\ddot{\mathbf{r}}_{s_i} \quad (110)$$

We discuss the above derivation with different types of  $s_i$ :

• **Link module  $L_i$ .** Since we assume the joint angles move slowly (i.e.,  $\dot{\mathbf{q}} \approx \mathbf{0}$ ,  $\ddot{\mathbf{q}} \approx \mathbf{0}$ ), the second derivative of the link position is only effected by the joint angles

$$\{{}^W\}\mathbf{P}_{L_i} = m_{L_i} \{{}^W\}\ddot{\mathbf{r}}_{L_i}(\mathbf{q}) \quad (111)$$

where  $\{{}^W\}\mathbf{r}_{L_i}(\mathbf{q})$  is the CoG position of link module w.r.t. the frame  $\{W\}$ .

• **Vectoring module  $A_{r_i}$ .** The offset from the vectoring axis  $\phi_i$  to the CoG position of  $A_{r_i}$  is significantly small. Then

$\{{}^W\}\mathbf{P}_{A_{r_i}}$  can be given by

$$\begin{aligned} \{{}^W\}\dot{\mathbf{P}}_{A_{r_i}} &= m_{A_{r_i}} \left( \{{}^W\}\ddot{\mathbf{r}}_{\{G_i\}}(\mathbf{q}) + \{{}^W\}\Delta\ddot{\mathbf{r}}_{A_{r_i}}(\phi_i, \dot{\phi}_i, \ddot{\phi}_i) \right), \\ &\approx m_{A_{r_i}} \{{}^W\}\ddot{\mathbf{r}}_{\{G_i\}}(\mathbf{q}) \end{aligned} \quad (112)$$

where  $\mathbf{r}_{\{G_i\}}(\mathbf{q})$  is the position of the frame  $\{G_i\}$  which is attached to the vectoring axis  $\phi_i$  as shown in Figure 3(b), and  $\{{}^W\}\Delta\mathbf{r}_{A_{r_i}}(\phi_i, \dot{\phi}_i, \ddot{\phi}_i)$  is the offset from the vectoring axis to the CoG, which can be ignored.

• **Vectoring module  $A_{p_i}$ .**  $\{{}^W\}\dot{\mathbf{P}}_{A_{p_i}}$  can be given by

$$\begin{aligned} \{{}^W\}\dot{\mathbf{P}}_{A_{p_i}} &= m_{A_{p_i}} \left( \{{}^W\}\ddot{\mathbf{r}}_{\{G_i\}}(\mathbf{q}) + \{{}^W\}\ddot{\mathbf{r}}_{\{G_i\} \rightarrow \{F_i\}}(\phi_i, \dot{\phi}_i, \ddot{\phi}_i) \right. \\ &\quad \left. + \{{}^W\}\Delta\ddot{\mathbf{r}}_{A_{p_i}}(\theta_i, \dot{\theta}_i, \ddot{\theta}_i) \right) \\ &\approx m_{A_{p_i}} \{{}^W\}\ddot{\mathbf{r}}_{\{G_i\}}(\mathbf{q}) \end{aligned} \quad (113)$$

where  $\mathbf{r}_{\{G_i\} \rightarrow \{F_i\}}(\phi_i, \dot{\phi}_i, \ddot{\phi}_i)$  is the offset from the frame  $\{G_i\}$  to the frame  $\{F_i\}$  that is attached to the vectoring axis  $\theta_i$  as shown in Figure 3(b), and  $\{{}^W\}\Delta\mathbf{r}_{A_{p_i}}(\theta_i, \dot{\theta}_i, \ddot{\theta}_i)$  is the offset from the frame  $\{F_i\}$  to the CoG position of module

$A_{p_i}$ . These offset can be also ignored due to their relatively small values.

Then, the derivative of total linear momentum can be given by

$$\begin{aligned} \{{}^W\}\dot{\mathbf{P}}_{\Sigma} &= \sum_{i=1}^{N_r} (m_{L_i} \{{}^W\}\ddot{\mathbf{r}}_{L_i}(\mathbf{q}) + m_{A_{r_i}} \{{}^W\}\ddot{\mathbf{r}}_{\{G_i\}}(\mathbf{q}) \\ &\quad + m_{A_{p_i}} \{{}^W\}\ddot{\mathbf{r}}_{\{G_i\}}(\mathbf{q})) = m_{\Sigma} \{{}^W\}\ddot{\mathbf{r}}_{\{CoG\}}(\mathbf{q}), \end{aligned} \quad (114)$$

where  $m_{\Sigma}$  is the total mass of the entire robot and  $\{{}^W\}\mathbf{r}_{\{CoG\}}(\mathbf{q})$  is the position of the entire CoG point. This eventually gives the result of equation (19).

### Entire rotational dynamics

We first define a function  $\dot{\mathcal{L}}(m, I, \mathbf{p})$  that can be described as

$$\dot{\mathcal{L}}(m, I, \mathbf{p}) := m \widehat{\mathbf{p}}^{\top \{CoG\}} \dot{\boldsymbol{\omega}} + I^{\{CoG\}} \dot{\boldsymbol{\omega}} + \{CoG\} \boldsymbol{\omega} \times I^{\{CoG\}} \boldsymbol{\omega} \quad (115)$$

where  $m \in \mathcal{R}$ ,  $\mathbf{p} \in \mathcal{R}^3$ , and  $I \in \mathcal{R}^{3 \times 3}$  are the mass, the CoG position, and the inertia tensor of a rigid body. The hat map  $\wedge$  transforms a vector in  $\mathcal{R}^3$  to a  $3 \times 3$  skew-symmetric matrix.  $\{CoG\} \boldsymbol{\omega}$  is the angular velocity of the entire CoG frame  $\{CoG\}$  expressed in the same frame.

We then discuss the derivation of equation (109) according to the different types of  $s_i$ :

• **Link module  $L_i$ .** Since we assume the joint angles move slowly (i.e.,  $\dot{\mathbf{q}} \approx \mathbf{0}$ ,  $\ddot{\mathbf{q}} \approx \mathbf{0}$ ), the angular velocities of all links module are identical to that of the entire CoG frame:  $\{CoG\} \boldsymbol{\omega}_{\{L_i\}} \approx \{CoG\} \boldsymbol{\omega}$ .

Then the derivative of the angular momentum of equation (109) can be given by

$$\begin{aligned} \{CoG\} \dot{\mathcal{L}}_{L_i} &= \dot{\mathcal{L}}(m_{L_i}, \{CoG\} I_{L_i}(\mathbf{q}), \{CoG\} \mathbf{p}_{L_i}(\mathbf{q})), \\ \{CoG\} I_{L_i}(\mathbf{q}) &= \{CoG\} R_{\{L_i\}}(\mathbf{q}) I_{L_i} \{CoG\} R_{\{L_i\}}^{\top}(\mathbf{q}) \end{aligned} \quad (116)$$

where  $I_{L_i}$  is the constant inertia tensor of a link described in its own coordinate.

• **Vectoring module  $A_{r_i}$ .** The angular velocity of this module can given by

$$\{CoG\} \boldsymbol{\omega}_{\{G_i\}} = \{CoG\} \boldsymbol{\omega} + \{CoG\} \boldsymbol{\eta}_{\phi_i} \dot{\phi}_i \quad (117)$$

where  $\boldsymbol{\eta}_{\phi_i} \in \mathcal{R}^3$  is the unit vector that denotes the direction of vectoring axis of  $\phi_i$ .

Then the derivative of the angular momentum of equation (109) can be given by

$$\{{}^{CoG}\dot{\mathcal{L}}_{A_{r_i}} = \dot{\mathcal{L}}(m_{A_{r_i}}, \{{}^{CoG}\mathbf{I}_{A_{r_i}}(\mathbf{q}), \{{}^{CoG}\mathbf{p}_{A_{r_i}}(\mathbf{q})) + \Delta\dot{\mathcal{L}}_{A_{r_i}} \quad (118)$$

$$\begin{aligned} \Delta\dot{\mathcal{L}}_{A_{r_i}} &= m_{A_{r_i}} \{{}^{CoG}\hat{\mathbf{P}}_{A_{r_i}} \{{}^{CoG}\Delta\hat{\mathbf{P}}_{A_{r_i}}^T \{{}^{CoG}\boldsymbol{\eta}_{\phi_i} \ddot{\phi}_i \\ &\quad + \{{}^{CoG}\mathbf{I}_{A_{r_i}} \{{}^{CoG}\boldsymbol{\eta}_{\phi_i} \ddot{\phi}_i \\ &\quad + \boldsymbol{\omega}_{\{G_i\}} \times \{{}^{CoG}\mathbf{I}_{A_{r_i}} \{{}^{CoG}\boldsymbol{\omega}_{\{G_i\}} \\ &\quad - \{{}^{CoG}\boldsymbol{\omega} \times \{{}^{CoG}\mathbf{I}_{A_{r_i}} \{{}^{CoG}\boldsymbol{\omega}, \end{aligned} \quad (119)$$

where  $\{{}^{CoG}\Delta\hat{\mathbf{P}}_{A_{r_i}}$  is the offset from the frame origin to the CoG position that can be ignored, and  $\mathbf{I}_{A_{r_i}}$  is inertial tensor of this vectoring module that is significantly small. Therefore,  $\Delta\dot{\mathcal{L}}_{A_{r_i}}$  can be ignored:  $\Delta\dot{\mathcal{L}}_{A_{r_i}} \approx \mathbf{0}$ .

• **Vectoring module.**  $A_{p_i}$ . The angular velocity of this module can be given by

$$\{{}^{CoG}\boldsymbol{\omega}_{\{F_i\}} = \{{}^{CoG}\boldsymbol{\omega} + \{{}^{CoG}\boldsymbol{\eta}_{\phi_i} \dot{\phi}_i + \{{}^{CoG}\boldsymbol{\eta}_{\theta_i} \dot{\theta}_i \quad (120)$$

where  $\boldsymbol{\eta}_{\theta_i} \in \mathcal{R}^3$  is the unit vector that denotes the direction of vectoring axis of  $\theta_i$ .

Then the derivative of the angular momentum of equation (109) can be given by

$$\{{}^{CoG}\dot{\mathcal{L}}_{A_{p_i}} = \dot{\mathcal{L}}(m_{A_{p_i}}, \{{}^{CoG}\mathbf{I}_{A_{p_i}}(\mathbf{q}), \{{}^{CoG}\mathbf{p}_{A_{p_i}}(\mathbf{q})) + \Delta\dot{\mathcal{L}}_{A_{p_i}} + \Delta\dot{\mathcal{L}}'_{A_{p_i}} \quad (121)$$

$$\begin{aligned} \Delta\dot{\mathcal{L}}_{A_{p_i}} &= m_{A_{p_i}} \{{}^{CoG}\hat{\mathbf{P}}_{A_{p_i}} \Delta\hat{\mathbf{P}}_{A_{p_i}}^T \{{}^{CoG}\boldsymbol{\eta}_{\phi_i} \ddot{\phi}_i \\ &\quad + \{{}^{CoG}\mathbf{I}_{A_{p_i}} \{{}^{CoG}\boldsymbol{\eta}_{\phi_i} \ddot{\phi}_i \\ &\quad + \{{}^{CoG}\boldsymbol{\omega}_{\{F_i\}} \times \{{}^{CoG}\mathbf{I}_{A_{p_i}} \{{}^{CoG}\boldsymbol{\omega}_{\{F_i\}} \\ &\quad - \{{}^{CoG}\boldsymbol{\omega} \times \{{}^{CoG}\mathbf{I}_{A_{p_i}} \{{}^{CoG}\boldsymbol{\omega} \end{aligned} \quad (122)$$

$$\begin{aligned} \Delta\dot{\mathcal{L}}'_{A_{p_i}} &= m_{A_{p_i}} \{{}^{CoG}\hat{\mathbf{P}}_{A_{p_i}} \Delta\hat{\mathbf{P}}_{A_{p_i}}^T \{{}^{CoG}\boldsymbol{\eta}_{\theta_i} \ddot{\theta}_i \\ &\quad + \{{}^{CoG}\mathbf{I}_{A_{p_i}} \{{}^{CoG}\boldsymbol{\eta}_{\theta_i} \ddot{\theta}_i \end{aligned} \quad (123)$$

$$\{{}^{CoG}\mathbf{I}_{A_{p_i}} = \{{}^{CoG}\mathbf{R}_{\{F_i\}} \text{diag}(I_\phi, 0, I_\phi) \{{}^{CoG}\mathbf{R}_{\{F_i\}}^T \quad (124)$$

where  $\{{}^{CoG}\Delta\hat{\mathbf{P}}_{A_{p_i}}$  is the offset from the frame origin to the CoG position that can be ignored.  $\text{diag}(\dots)$  forms a diagonal matrix with the variables, and  $I_\phi$  is the inertial moment around the vectoring angle  $\phi$ . Zero moment around the vectoring axis  $\theta_i$  and the zero offset of  $\{{}^{CoG}\Delta\hat{\mathbf{P}}_{A_{p_i}}$  lead to  $\Delta\dot{\mathcal{L}}'_{A_{p_i}} \approx \mathbf{0}$ . However,  $\Delta\dot{\mathcal{L}}_{A_{p_i}}$  cannot be ignored because of  $I_\phi$ , and thus the last three terms of equation (122) are non-zero. The sum of the last two terms is related  $\dot{\phi}_i \{{}^{CoG}\boldsymbol{\omega}$  and  $\dot{\theta}_i \{{}^{CoG}\boldsymbol{\omega}$ , which can be considered relatively smaller than  $\dot{\phi}_i$  or  $\dot{\theta}_i$  because of the multiplication between two angular velocities. Therefore, equation (122) can be further approximated as follows

$$\Delta\dot{\mathcal{L}}_{A_{p_i}} = \{{}^{CoG}\mathbf{R}_{\{G_i\}} \begin{bmatrix} I_\phi \ddot{\phi}_i \\ 0 \\ 0 \end{bmatrix} \quad (125)$$

Finally, the derivative of the total angular momentum can be given by

$$\begin{aligned} \{{}^{CoG}\dot{\mathcal{L}}_\Sigma &= \sum_{i=1}^{N_s} \dot{\mathcal{L}}(m_{s_i}, \{{}^{CoG}\mathbf{I}_{s_i}(\mathbf{q}), \{{}^{CoG}\mathbf{p}_{s_i}(\mathbf{q})) + \sum_{i=1}^{N_r} \Delta\dot{\mathcal{L}}_{A_{p_i}} \\ &= \{{}^{CoG}\mathbf{I}_\Sigma(\mathbf{q}) \{{}^{CoG}\dot{\boldsymbol{\omega}} \\ &\quad + \{{}^{CoG}\boldsymbol{\omega} \times \{{}^{CoG}\mathbf{I}_\Sigma(\mathbf{q}) \{{}^{CoG}\boldsymbol{\omega} + \Delta\dot{\mathcal{L}}_{A_p} \end{aligned} \quad (126)$$

$$\{{}^{CoG}\mathbf{I}_\Sigma(\mathbf{q}) = \sum_{i=1}^{N_s} \left( m_{s_i} \{{}^{CoG}\hat{\mathbf{q}}_{s_i}(\mathbf{q}) \{{}^{CoG}\hat{\mathbf{q}}_{s_i}^T(\mathbf{q}) + \{{}^{CoG}\mathbf{I}_{s_i}(\mathbf{q}) \right) \quad (127)$$

which gives the result of equation (20).

## Appendix C. Stability analysis of conventional positional control

This appendix presents the stability analysis on the translational dynamics (19) controlled by the conventional PID controller shown in equation (23) with actuators that have the dynamics of equations (12)–(14). The following derivation flow will be also used in Appendix D to analyze the stability of the proposed control method presented in Section 3.2 and Section 3.3.

Because  $T_\Omega$  in equation (12) and  $I_\theta$  in equation (14) are significantly small, following approximation is available

$$\lambda_i \approx \lambda_i^{des} \quad (128)$$

$$\theta_i \approx \theta_i^{des} \quad (129)$$

Then, the dynamics of the total force  $\{{}^{CoG}\mathbf{f}$  (see equation (3)) is associated with that of the vectoring angles  $\phi$ . Although there is a complex nonlinear mapping between these two physical quantities as shown in equations (31) and (32), they should have the similar behavior in frequency domain that all belong to the second-order system. We further decouple  $\{{}^{CoG}\mathbf{f}$  into three independent dimensions, and perform the Laplace transform for each dimension as follows

$$F(s) = \frac{k_{\phi,p}}{s^2 + k_{\phi,d}s + k_{\phi,p}} F^{des}(s) \quad (130)$$

where  $F(s)$  is the Laplace transform of arbitrary dimension of  $\{{}^{CoG}\mathbf{f}$ , and  $F^{des}(s)$  is the Laplace transform of arbitrary dimension of the desired value  $\{{}^{CoG}\mathbf{f}^{des}$  in equation (23).  $k_{\phi,p}$  and  $k_{\phi,d}$  are the gains corresponding to equation (13).

For simplicity, we assume  ${}^{\{W\}}R_{\{CoG\}}$  to be an identity matrix. Then, the Laplace transform of (19) can be given by

$$m_{\Sigma}s^2R(s) = F(s) + F_{ex}(s) \quad (131)$$

$$F_{ex}(s) = \sum_{i=1}^{N_{ex}} F_{ex_i}(s) \quad (132)$$

where  $R(s)$  is the Laplace transform of the arbitrary dimension of  ${}^{\{W\}}r_{\{CoG\}}$ , and  $F_{ex_i}(s)$  is the Laplace transform of the arbitrary dimension of  ${}^{\{W\}}f_{ex_i}$ . We ignore the Laplace transform for gravity  $g$  which can be included in the external force  ${}^{\{W\}}f_{ex_i}$  as a step input.

Similarly, the Laplace transform of equation (23) can be given by

$$F^{des}(s) = m_{\Sigma} \left( k_{f,p} + \frac{k_{f,i}}{s} + k_{f,d}s \right) (R^{des}(s) - R(s)) - F_{ex}(s) \quad (133)$$

where  $R^{des}(s)$  is the Laplace transform of arbitrary dimension of  ${}^{\{W\}}r_{\{CoG\}}^{des}$ .

Then substituting equations (133) and (130) into equation (131), the transfer function regarding  $R(s)$  can be written as

$$R(s) = G_R(s)R^{des}(s) + G_{F_{ex}}(s)F_{ex}(s) \quad (134)$$

$$G_R(s) = \frac{k_{\phi,p}(k_{f,d}s^2 + k_{f,p}s + k_{f,i})}{D(s)} \quad (135)$$

$$G_{F_{ex}}(s) = \frac{s^3 + k_{\phi,d}s^2}{m_{\Sigma}D(s)} \quad (136)$$

$$D(s) = s^5 + k_{\phi,d}s^4 + k_{\phi,p}(s^3 + k_{f,d}s^2 + k_{f,p}s + k_{f,i}) \quad (137)$$

The necessary and sufficient condition to have a stable transfer function of  $G_R(s)$  and  $G_{F_{ex}}(s)$  is to ensure all poles of  $D(s)$  to be negative. This can be achieved by choosing proper feedback gains for position control ( $k_{f,p}$ ,  $k_{f,i}$ , and  $k_{f,d}$ ) and vectoring actuator ( $k_{\phi,p}$ ,  $k_{\phi,d}$ ). Generally,  $k_{\phi,p}$  and  $k_{\phi,d}$  are first chosen according to equation (13) to ensure stable servoing performance for rotor vectoring. Then, the PID control gains  $k_{f,p}$ ,  $k_{f,i}$ , and  $k_{f,d}$  are selected to achieve the stable position control.

We further analyze the position error  $e_r$  in the steady state. Assuming step input for both the desired position  ${}^{\{W\}}r_{\{CoG\}}$  and the external forces  ${}^{\{W\}}f_{ex_i}$ , the Laplace transform of  $e_r$  can be given by

$$\begin{aligned} E_r(s) &= R^{des}(s) - R \\ &= (1 - G_R(s))R^{des}(s) - G_{F_{ex}}(s)F_{ex}(s) \\ &= (1 - G_R(s))\frac{1}{s} - G_{F_{ex}}(s)\frac{1}{s} \\ &= \frac{s^4 + k_{\phi,d}s^3 + k_{\phi,p}s^2}{D(s)} - \frac{s^2 + k_{\phi,d}s}{m_{\Sigma}D(s)} \end{aligned} \quad (138)$$

According to the final value theorem, the position error in steady state can be obtained by

$$e_s = \lim_{s \rightarrow 0} sE_r(s) = 0 \quad (139)$$

which demonstrates each dimension (axis) of  ${}^{\{W\}}r$  can converge to the desired value.

## Appendix D. Stability analysis of proposed control method

This appendix presents the stability analysis on the whole dynamics controlled by the proposed control method presented in Section 3.2 and Section 3.3.

### Position control

Because the vectoring angles  $\phi$  is responsible for the truncated control terms presented in equation (41), the residual D control term is related to other types of actuators that have fast dynamics as presented in equations (128) and (129). Therefore, equation (130) should be reformulated as follows

$$F(s) = \frac{k_{\phi,p}}{s^2 + k_{\phi,d}s + k_{\phi,p}} F_{ir}^{des}(s) + F_D^{des}(s) \quad (140)$$

$$F_{ir}^{des}(s) = m_{\Sigma} \left( k_{f,p} + \frac{k_{f,i}}{s} \right) (R^{des}(s) - R(s)) - F_{ex}(s) \quad (141)$$

$$F_D^{des}(s) = m_{\Sigma} k_{f,d} s (R^{des}(s) - R(s)) \quad (142)$$

Then the transfer function regarding  $R(s)$  can be re-written as follows

$$R(s) = G'_R(s)R^{des}(s) + G'_{F_{ex}}(s)F_{ex}(s) \quad (143)$$

$$G'_R(s) = \frac{N'(s)}{D'(s)} \quad (144)$$

$$G'_{F_{ex}}(s) = \frac{s^3 + k_{\phi,d}s^2}{m_{\Sigma}D'(s)} \quad (145)$$

$$N'(s) = k_{f,d}(s^4 + k_{\phi,d}s^3) + k_{\phi,p}(k_{f,d}s^2 + k_{f,p}s + k_{f,i}) \quad (146)$$

$$D'(s) = s^5 + k_{\phi,d}s^4 + k_{\phi,p}s^3 + N'(s) \quad (147)$$

If poles of  $D'(s)$  are all negative, the transfer functions of both  $G'_R(s)$  and  $G'_{\tau_{\text{ex}}}(s)$  will be stable because they are strictly proper. This can be achieved by choosing proper control position gains as already discussed in [Appendix C](#). Further, the zero position error in the steady state can be confirmed by substituting equation (143) into equations (138) and (139). This not only demonstrates the trackability to a constant desired position, but also the stability against the external force  $\{^W\}f_{\text{ex}_i}$ .

### Attitude control

We focus on the analysis on the stability for tracking a constant desired attitude, and assuming the error from this desired attitude is relatively small. Then, we introduce RPY Euler angles  $\alpha$  to denote the deviation from the desired attitude ( $\{^{CoG}\}R^{des\top}\{^{CoG}\}R$ ). We further use the approximation of  $\omega \approx \dot{\alpha}$  to linearize equations (20), (42), and (43) as follows

$$I_{\Sigma}\ddot{\alpha} + \Delta\dot{\mathcal{L}}_{A_p} = \tau + \tau_{\text{ex}} \quad (148)$$

$$\tau^{des} = \tau_{tr}^{des} + I_{\Sigma}K_{\tau,d}\dot{e}_{\alpha} \quad (149)$$

$$\tau_{tr}^{des} = I_{\Sigma}\left(K_{\tau,p}e_{\alpha} + K_{\tau,i}\int e_{\alpha}\right) - \tau_{\text{ex}} \quad (150)$$

$$\tau_{\text{ex}} = \sum_{i=1}^{N_{\text{ex}}}(\tau_{\text{ex}_i} + p_{\text{ex}_i} \times f_{\text{ex}_i}) \quad (151)$$

$$\Delta\dot{\mathcal{L}}_{A_p} = \sum_{i=1}^{N_r} R_{\{F_i\}} \begin{bmatrix} I_{\phi}\ddot{\phi}_i \\ 0 \\ 0 \end{bmatrix}$$

where  $e_{\alpha} = \alpha^{des} - \alpha$ . Note that  $\alpha^{des}$  should be zero, because  $\alpha$  itself denotes the error to the desired attitude. However, a small step input of  $\alpha^{des}$  can denote the behavior to track a new desired attitude that is close to the last one.  $\dot{\alpha} \times I_{\Sigma}\ddot{\alpha}$  in equations (148) and (149) can canceled each other, and thus is omitted.

We then decouple equations (148) and (149) into three independent dimensions, and perform the Laplace transform for each dimension as follows

$$\bar{I}_{\Sigma}s^2A(s) + I_{\phi}s^2\bar{\Phi}(s) = T(s) + T_{\text{ex}}(s) \quad (152)$$

$$T_{tr}^{des}(s) = \bar{I}_{\Sigma}\left(k_{\tau,p} + \frac{k_{\tau,i}}{s}\right)(A^{des}(s) - A(s)) - T_{\text{ex}}(s) \quad (153)$$

$$T_D^{des}(s) = \bar{I}_{\Sigma}k_{\tau,d}s(A^{des}(s) - A(s)) \quad (154)$$

$$\bar{\Phi}(s) = \sum_{i=1}^{N_r} \gamma_i \Phi_i(s) \quad (155)$$

where  $A(s)$ ,  $A^{des}(s)$ ,  $T(s)$ , and  $T_{\text{ex}}(s)$  are the Laplace transform of arbitrary dimension of  $\alpha$ ,  $\alpha^{des}$ ,  $\tau$ , and  $\tau_{\text{ex}}(s)$ , respectively.  $\bar{I}_{\Sigma} \in \mathcal{R}$  is a decoupled inertial moment of  $I_{\Sigma}$ .  $\gamma_i \in [-1, 1]$  denotes the influence rate of each vectoring angle  $\phi_i$  due to  $R_{\{F_i\}}$ .

To further derive the transfer function for  $\Phi_i(s)$  in  $\bar{\Phi}(s)$ , we approximate the nonlinear mapping from  $\tau_{tr}^{des}$  to  $\phi_i^{des}$  (see Section 3.3.1) as a linear conversion in frequency domain

$$\Phi_i^{des}(s) = \beta_i T^{des}(s) \quad (156)$$

It is notable that similar mapping from  $f_{tr}^{des}$  to  $\phi_i^{des}$  should be also introduced. Since  $f_{tr}^{des}$  converges to a constant value when  $t \rightarrow \infty$  (which is obvious from the result of equation (143)), both  $f_{tr}^{des}$  and  $\dot{f}_{tr}^{des}$  converge to zero. Thus, we can ignore the influence of  $f_{tr}^{des}$ .

By substituting equation (156) into the Laplace transform of (13), equation (155) can be further derived as

$$\bar{\Phi}(s) = \frac{ck_{\phi,p}}{s^2 + k_{\phi,d}s + k_{\phi,p}} T_{tr}^{des}(s), \quad c = \sum_{i=1}^{N_r} \gamma_i \beta_i \quad (157)$$

Similar to equation (140), the transfer function from  $T^{des}(s)$  to  $T(s)$  can be written as

$$T(s) = \frac{k_{\phi,p}}{s^2 + k_{\phi,d}s + k_{\phi,p}} T_{tr}^{des}(s) + T_D^{des}(s) \quad (158)$$

Substituting equations (153), (154), (157), and (158) into equation (152), the transfer function for  $A(s)$  can be summarized as follows

$$A(s) = G_A(s)A^{des}(s) + G_{T_{\text{ex}}}(s)T_{\text{ex}}(s) \quad (159)$$

$$G_A(s) = \frac{N(s)}{D(s)} \quad (160)$$

$$G_{T_{\text{ex}}}(s) = \frac{(1 + ck_{\phi,p}I_{\phi})s^3 + k_{\phi,d}s^2}{I_{\Sigma}D(s)} \quad (161)$$

$$N(s) = k_{\tau,d}(s^4 + k_{\phi,d}s^3) + k_{\phi,p}(k_{\tau,d}s^2 + k_{\tau,p}s + k_{\tau,i}) - k_{\phi,p}cI_{\phi}(k_{\tau,p}s^3 + k_{\tau,i}s^2) \quad (162)$$

$$D(s) = s^5 + k_{\phi,d}s^4 + k_{\phi,p}s^3 + N(s) \quad (163)$$

The necessary condition to ensure stability of both  $G_A(s)$  and  $G_{T_{\text{ex}}}(s)$  is to ensure  $a_i > 0$  for  $D(s) = \sum_{i=0}^5 a_i s^i$ . Then  $a_3$  and  $a_2$  can be given by

$$a_2 = k_{\phi,p}(k_{\tau,d} - cI_{\phi}k_{\tau,i}) \quad (164)$$

$$a_3 = k_{\phi,d}k_{\tau,d} + k_{\phi,p} - cI_{\phi}k_{\phi,p}k_{\tau,p} \quad (165)$$



To provide the upper bound of the  $c$  for equations (164) and (165),  $\beta_i$  in equation (156) should be estimated. The moment generated by vectoring a rotor can be given as follows

$$\tau_i = f_i d_i \approx \frac{m_{\Sigma} g \sin(\phi_i)}{N_r} d_i \approx \frac{m_{\Sigma} g \phi_i}{N_r} L \quad (166)$$

This approximation is performed around a hovering state, where each rotor generates thrust force that is approximately equal to  $\frac{m_{\Sigma} g}{N_r}$ . Thus, the lateral force by vectoring is equal to  $\frac{m_{\Sigma} g \sin(\phi_i)}{N_r}$ , which is further approximated to  $m_{\Sigma} g \phi_i$ , due to the small value of  $\phi_i$ .  $L$  is the length of link which approximates the distance from the robot CoG to each rotor. Therefore,  $\beta_i$  can be approximated as

$$\beta_i = \frac{\phi_i}{\sum_{i=1}^{N_r} \tau_i} \approx \frac{\phi_i}{N_r \tau_i} = \frac{1}{m_{\Sigma} g L} \quad (167)$$

Then  $c$  in equation (157) can be approximated as follows

$$c I_{\phi} = I_{\phi} \sum_{i=1}^{N_r} \gamma_i \beta_i \leq I_{\phi} \sum_{i=1}^{N_r} \beta_i = \frac{I_{\phi} N_r}{m_{\Sigma} g L} \approx \frac{I_{\phi}}{m_L g L} \quad (168)$$

where  $m_L$  is the average weight of a link unit which also includes the vectoring actuators. Given  $I_{\phi} \ll m_L L^2$ ,  $c I_{\phi} < 1$  is available. Therefore,  $a_2 > 0$  is always established. If  $k_{\phi, d} k_{r, d} > k_{\phi, p} k_{r, p}$ ,  $a_3 > 0$  will be also valid.

An effective strategy to find valid gains to make all poles of (163) negative is to, first set  $c$  zero for searching gains under a similar condition with position control; then set  $c$  as (168) and use Routh-Hurwitz stability criterion to check the stability of each poles. If there are any unstable poles, we can repeat the first step to enhance the stability of poles.

With stable  $G_A(s)$  and  $G_{T_{ex}}(s)$ ,  $\alpha$  is stable in the near hovering state. Further, the zero rotational error in the steady state can be confirmed by applying (138) and (139) for  $A(s)$  with (159). This not only demonstrates the trackability to a constant desired attitude, but also the stability against the external torque  $\tau_{ex}$ .

## Appendix E. Planning method for object manipulation

In this appendix, we present an online motion planning method for grasplless manipulation in a feedback manner, which is based on our previous work (Zhao et al., 2020). The total framework is depicted in Figure 25. We first describe the end-effector position controller involving the feedback of object state. Then, we present the realtime planning method for the entire robot body (i.e., the root pose and joint angles) to track the desired end-effector position. Further, we derive the external wrench required for object manipulation, which is compensated by the proposed flight control method presented in Section 3.

### Position control of end-effector

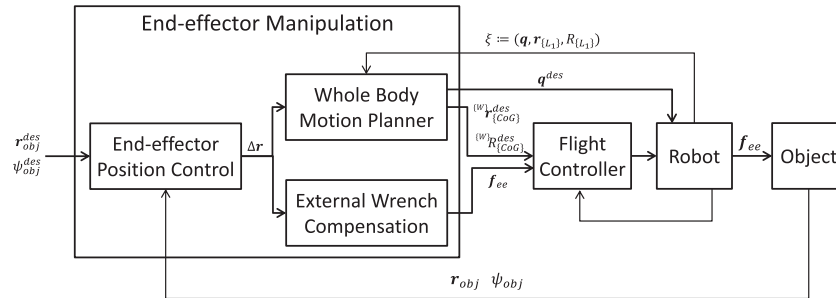
We assume that the target object has surface contact with the environment, and thus, object motion is restricted to SE(2): two DoFs of translation motion on the contact surface, and one DoF of rotational motion around the surface normal. Then grasplless manipulation with a single end-effector can be depicted as Figure 26. The end-effector can only exert two DoFs from the three-dimensional force along the surface, which indicates it is impossible to control all three DoFs of the object motion. However, it is still possible to manipulate the object rotation using the end-effector force in cooperation with the friction force. Thus, one DoF of the force is for rotational manipulation, and the other DoF is for translational manipulation. Then the goal of manipulation is to reach (or keep) a desired orientation angle  $\psi_{obj}^{des}$  and move to a desired position  $\mathbf{r}_{obj}^{des}$  as close as possible, as depicted in Figure 26.

The outer feedback loop for controlling the end-effector position to move an object from  $\mathbf{r}_{obj}(t)$ ,  $\psi_{obj}(t)$  to  $\mathbf{r}_{obj}^{des}$ ,  $\psi_{obj}^{des}$  can be given as follows.

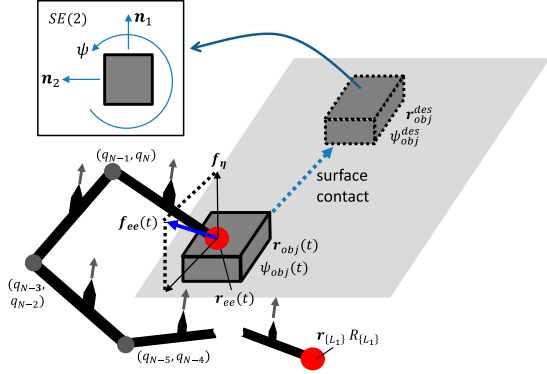
$$\mathbf{r}_{ee}(t+1) = \mathbf{r}_{ee}(t) + \Delta \mathbf{r}(t) \quad (169)$$

$$\Delta \mathbf{r}(t) = \mathbf{n}_1 \text{sat}\left(e_r(t), \bar{\epsilon}_r\right) + \mathbf{n}_2 k_{\psi} \text{sat}\left(e_{\psi}(t), \bar{\epsilon}_{\psi}\right) \quad (170)$$

$$\mathbf{e}_r(t) = \left(\mathbf{r}_{obj}^{des} - \mathbf{r}_{obj}(t)\right) \cdot \mathbf{n}_1 \quad (171)$$



**Figure 25.** Overview of the framework for grasplless manipulation. The position of the end-effector is controlled by the feedback of object pose. Then the robot configuration (i.e., the root pose and joint angles) is updated in realtime during manipulation. Meanwhile, the external wrench to move the object is calculated for compensation.



**Figure 26.** Grasps manipulation by a single end-effector to move an object which has surface contact with environment. The goal is to move the object to a desired position  $r_{obj}^{des}$  with a desired orientation angle  $\psi_{obj}^{des}$ . External force  $f_{ee}(t)$  contains the manipulation force and the constraining force  $f_n$ .

$$e_\psi(t) = \psi_{obj}^{des} - \psi_{obj}(t) \quad (172)$$

where  $\mathbf{n}_1$  is a unit vector which is aligned to the traveling direction of the object as shown in Figure 26, whereas  $\mathbf{n}_2$  is another orthogonal unit vector.  $k_\psi$  is the feedback gain for controlling the rotation angle of the object.  $sat(*, *)$  is a saturation function, and  $\bar{e}_r$  and  $\bar{e}_\psi$  are the upper limit for translational and rotational errors, respectively.  $r_{ee}(t)$  is the three dimensional position of the end-effector at time  $t$ . Note that there is no gain for the translational error  $e_r$ , because  $e_r$  and  $r_{ee}(t)$  have the same metric unit.

This feedback control for the end-effector position cannot ensure that the object reaches the desired position  $r_{obj}(t)$ , because it allows the translational error along the direction of  $\mathbf{n}_2$  (i.e.,  $(r_{obj}^{des} - r_{obj}(t_\infty)) \cdot \mathbf{n}_2 = 0$ ). In contrast, the desired orientation can be achieved (i.e.,  $e_\psi(t_\infty) = 0$ ). This manipulation style can contribute to pushing an object towards a wall.

### Whole body planning for end-effector manipulation

The operation state of manipulation is the Cartesian motion of the end-effector; whereas the configuration state is an integrated vector  $\xi$  composed of the root pose and joint angles  $\xi := (\{{}^W R_{\{L_1\}}, \{{}^W r_{\{L_1\}}, \mathbf{q}\})$ . The first two components denote the orientation and position of the root link  $\{L_1\}$  w.r.t the world frame  $\{W\}$ .

For motion planning of such a multilinked model, we follow the method based on differential kinematics presented in Zhao et al. (2020). Differential kinematics is used to plan a continuous trajectory for a multilinked model by solving the differential motion in multiple iterations. The end-effector position control of (169) can be considered as the desired differential motion of the end-effector. Then it is possible to convert the motion of the end-effector to the

kinematic motion of the robot by the following planning iteration:

1. Find the optimal small motion  $\delta\xi_t$  at time  $t$  satisfying the following local optimization problem:

$$\min_{\delta\xi(t)} \delta\xi(t)^\top W_1 \delta\xi(t) + \delta\mathbf{x}_{ee}(t)^\top W_2 \delta\mathbf{x}_{ee}(t) \quad (173)$$

$$\text{s.t. } \underline{\mathbf{b}}_i \leq A_i \left( \xi(t), \delta\xi(t) \leq \bar{\mathbf{b}}_i \right) \quad (174)$$

where  $\xi(t)$  is the current state in the configuration space. In equation (173), the first term is introduced to minimize the change in  $\xi(t)$  at each step, which corresponds to the velocity of the joints and the root pose. The second term corresponds to the Cartesian motion error of the end-effector which is given by

$$\delta\mathbf{x}_{ee}(t) = J_{ee}(\xi) \delta\xi(t) - \Delta\mathbf{r}(t) \quad (175)$$

where  $\Delta\mathbf{r}(t)$  is defined in equation (170), and  $J_{ee}(\xi) \in \mathcal{R}^{3 \times (N_f + 6)}$  is the Jacobian matrix regarding the position of the end-effector. Generally, the Cartesian motion error is set to zero (i.e.,  $\delta\mathbf{x}_{ee}(t) = 0$ ) in most of the planning cases, and then an equality constraint is required; however, setting equality constraints can easily lead to unexpected results (e.g., large  $\delta\xi_k$ ). Thus, we provide a soft constraint to minimize the residual term shown in the second term of equation (173). The positive definite diagonal weight matrices  $W_1$  and  $W_2$  are introduced to balance the two terms in equation (173). In terms of the constraints shown in equation (174),  $A_i$  ( $i = 1, 2, \dots$ ) are the linearized matrices for inequality constraints, and  $\underline{\mathbf{b}}_i$  and  $\bar{\mathbf{b}}_i$  are the lower and upper limit vectors for these inequality constraints. All inequality constraints and the derivation of the linearized matrices  $A_i$  are presented in Zhao et al. (2020). Among the constraints, we also consider the thrust force and joint torque within their valid ranges. This is achieved by making a quasi-static assumption for the manipulation motion. Then, the static thrust force can be calculated from the joint angles and the root pose. Further, the joint torque can be also retrieved from equation (21). Since these static values depend only on the kinematics model, the corresponding Jacobian matrix can be derived regarding  $\delta\xi_t$ , and thus, the inequality constraints for thrust force and joint torque are also available. The detail can be also found in Zhao et al. (2020).

2. Update the state in the configuration space by adding the resulting infinitesimal motion

$$\xi(t+1) = \xi(t) + \delta\xi(t) \quad (176)$$

where  $\xi(t+1)$  is the next desired configuration for the robot ( $\xi^{des} := \xi(t+1)$ ). Then, the desired robot CoG pose ( ${}^{\{W\}}\mathbf{r}_{\{CoG\}}^{des}, {}^{\{W\}}R_{\{CoG\}}$ ) can be calculated from the desired root pose ( ${}^{\{W\}}\mathbf{r}_{\{L_1\}}^{des}, {}^{\{W\}}R_{\{L_1\}}$ ) and the desired joint angles  $\mathbf{q}^{des}$  based on forward kinematics. As shown in [Figure 25](#), the desired CoG pose are then sent to the flight control framework presented in [Section 3](#), and the desired joint angles  $\mathbf{q}^{des}$  are sent to the joint actuators.

### External wrench compensation

It is also important to exert proper force to move an object, because the reaction (external) force also influences the flight stability. Although the reaction force can be compensated by the P and I control in the flight control framework, such passive compensation significantly degrades the tracking performance of the robot (and the end-effector) trajectory. Thus, an additional force must be exerted by the end-effector for active compensation. We assume that the object moves with a constant velocity, then the end-effector is required to exert a three-dimensional force to

counteract the friction owing to the surface contact. Then the desired reaction force can be calculated as follows

$$\mathbf{f}_{ee}(t) = -f_{fric} \frac{\Delta \mathbf{r}(t)}{\|\Delta \mathbf{r}(t)\|} + \mathbf{f}_\eta \quad (177)$$

where  $f_{fric}$  is the scalar of the friction force.  $\mathbf{f}_\eta$  is the reaction force from the surface to ensure steady contact between the end-effector and the object as shown in [Figure 26](#).

This reaction force is updated at every iteration of equations (169)–(172), and is substituted into equations (25) and (29) for active compensation. Generally, the actual friction force is measured by a force sensor attached to the end-effector. However, we assume that coefficient ratio of the object material and the object mass are well known, and thus, the friction force can be estimated feed-forwardly. The offset from the actual external force value, which can be considered as a steady-state error, can be compensated by the proposed flight control method as presented in [Appendix D](#).