Versatile Multilinked Aerial Robot with Tilting Propellers: Design, Modeling, Control and State Estimation for Autonomous Flight and Manipulation

Moju Zhao, Tomoki Anzai, Fan Shi, Toshiya Maki, Takuzumi Nishio,
Keita Ito, Naoya Kuromiya, Kei Okada, Masayuki Inaba
Department of Mechano-Infomatics
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
`chou@jsk.t.u-tokyo.ac.jp`

Abstract

Multilinked aerial robot is one of the state-of-the-art works in aerial robotics, which demonstrates the deformability benefiting both maneuvering and manipulation. However, the performance in outdoor physical world has not yet been evaluated because of the weakness in the controllability and the lack of the state estimation for autonomous flight. Thus we adopt tilting propellers to enhance the controllability. The related design, modeling and control method are developed in this work to enable the stable hovering and deformation. Furthermore, the state estimation which involves the time synchronization between sensors and the multilinked kinematics is also presented in this work to enable the fully autonomous flight in the outdoor environment. Various autonomous outdoor experiments, including the fast maneuvering for interception with target, object grasping for delivery, and blanket manipulation for firefighting are performed to evaluate the feasibility and versatility of the proposed robot platform. To the best of our knowledge, this is the first study for the multilinked aerial robot to achieve the fully autonomous flight and the manipulation task in outdoor environment. We also applied our platform in all challenges of the 2020 Mohammed Bin Zayed International Robotics Competition, and ranked third place in Challenge 1 and sixth place in Challenge 3 internationally, demonstrating the reliable flight performance in the fields.

1 Introduction

In recent years, the development of aerial robot in field robotics has been greatly enhanced, and applications are diverse, ranging from autonomous exploration [Michael et al., 2012] and data collection [Ore and Detweiler, 2018] to provision of commercial services such as cinematography [Bonatti et al., 2020]. Furthermore, the advanced grasping ability has been focused, which enables the fully autonomous delivery without the human interference. In the last MBZIRC 2017 challenge, many teams succeeded to use their multirotor aerial robots equipped with electromagnetic gripper to pick up and deliver the treasure autonomously [Spurný et al., 2019, Bähnemann et al., 2019, Beul et al., 2019, Lee et al., 2019]. However, these task-specific grippers can only pick up the magnetic object, indicating the lack of versatility in grasping.

In order to achieve a versatile aerial robot platform without an additional gripper or manipulator, the multilinked structure which enables grasping and manipulation by self-deformation has been proposed in our

previous works [Zhao et al., 2017, Zhao et al., 2018, Shi et al., 2020]. Furthermore, the adoption of tilting propellers for the enhancement on the controlability in a model composed from eight links is proposed [Anzai et al., 2018]. However, such an over-actuated model has a relatively large inertia indicating the difficulty to perform aggressive maneuvering. In addition, the large number of mechanical components would also increase the maintenance cost. Therefore, the integration of tilting propellers with a four-link model which is the minimum configuration for the multilinked platform is investigated in this work for the achievement of various autonomous task as shown in Fig. 1. The main issues addressed in this work are the design of the multilinked structure with the tilting propellers, the modeling and control for such special under-actuated system, and the state estimation for the fully autonomous flight in the fields.
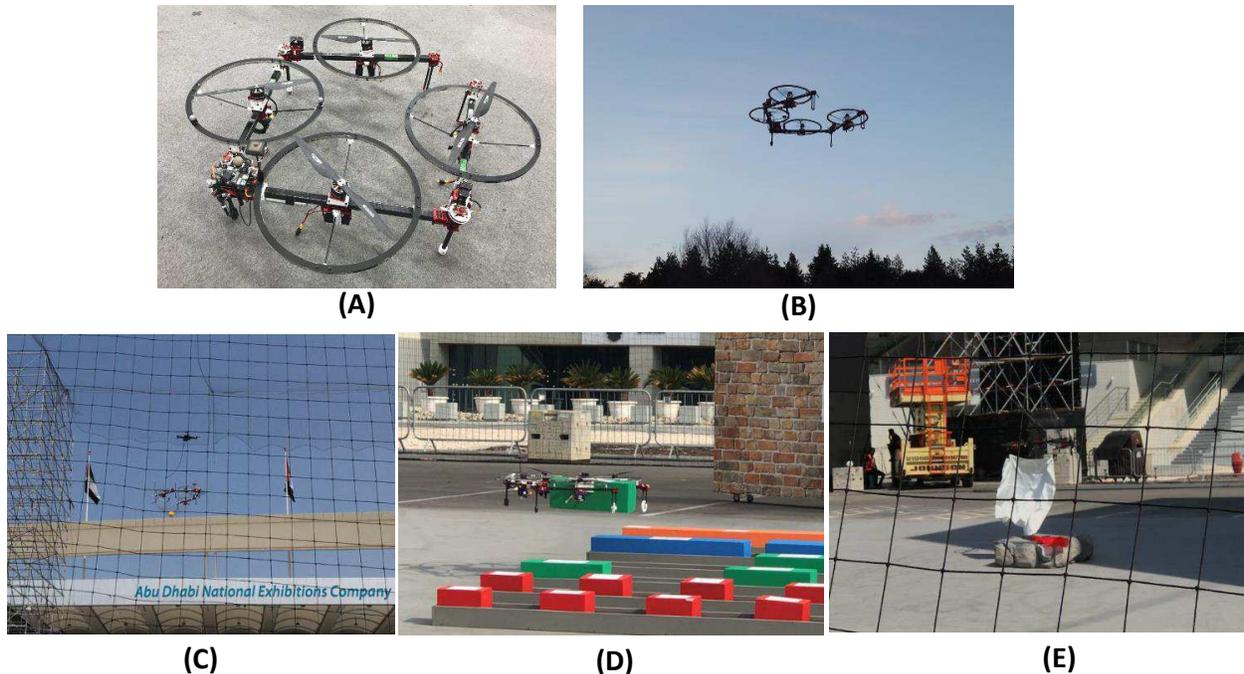
Figure 1: (**A**): the versatile multilinked aerial robot platform with tilting propellers. (**B**): fully autonomous flight in outdoor environment. (**C**) ∼ (**E**): participation in MBZIRC 2020 which involved the interception with fast moving target, grasping and delivering object, and firefighting by manipulating blanket.

1.1 Related Works

1.1.1 Deformable Aerial Robot

Deformability is one of the cutting-edge studies in the field of aerial robot. Various deformable structures are developed based on the quad-rotor model. One of the advantages of deformability is an advanced maneuvering to pass through a narrow space by morphing [Riviere et al., 2018, Falanga et al., 2019, Zhao et al., 2017], leading to a potential for exploration in a confined environment such as disaster site. On the other hand, several modular structures are also proposed [Gabrich et al., 2018, Yang et al., 2018], which afford an advanced ability of aerial manipulation without using additional manipulator or gripper. However, in these works, there are more than four propellers in each module indicating the redundancy in structural design. Thus, a minimized modular structure which only contains a single propeller is presented in our previous works [Zhao et al., 2016], and a deformable platform with four links is developed to achieve the grasping ability [Zhao et al., 2018]. In this platform, there are three joints connecting four links. However, it is possible to further reduce the joint number to two which can still achieve the grasping ability. Such a simplified design can decrease the structural complexity and thus improve the robustness against crash.

1.1.2 Design and Control for Model with Tilting Propellers

Regarding the general (undeformable) multirotor model, the rotational axes of rotors are all parallel, leading to the under-actuation in terms of control since the collective thrust force is always vertical in the body frame, and thus the robot cannot track full pose in SE(3). Moreover, the torque in z axis of the body frame is solely dependent on the drag moment generated by propeller rotation, which is significantly weak compared with torques in other axes. This is the reason of the low controlability in yaw rotational motion. Thus, the tilting propeller design is introduced to overcome these difficulties, and several undeformable models composed from more than six tilting propellers are proposed to achieve full pose tracking [Rajappa et al., 2015,Brescianini et al., 2016,Park et al., 2018]. On the hand, the tilting propeller design is also adopted in the multilinked model by [Anzai et al., 2018], which is composed from eight links. However, the model is over-actuated resulting in a large inertia which is difficult to perform fast motion.

Then the integration of tilting propellers with a four-link model is studied. Prior to the multilinked structure, the tilting propeller design for a general quadrotor is presented by [Efraim et al., 2015], where rotors are all tilted inwards regarding the center of body to enhance the stability of the horizontal motion. However, the insufficiency of the torque in z axis still remains in such tilting design, since torque in z axis still depends solely on the drug moment. Then, alternative tilting design same with [Anzai et al., 2018] is applied in the four-link model by [Shi et al., 2019], where the propeller is tilted along the link direction. With this tilting design, an additional moment much larger than the drag moment can be generated resulted from the multiplication by the tilted thrust force and the distance to the center of gravity. On the other hand, the decision of the tilting angle is very important. The relationship between the tilting angle and the force efficiency is revealed by [Ryll et al., 2016]. However, there are more factors which should be considered, such as the quantity of the torque in z axis and the horizontal force generated by the tilted thrust force. In this work, a comprehensive investigation of the influence of the tilting angle on the statics, dynamics and the aerodynamics interference is presented.

In terms of the control for model with tilting propellers, several control methods for fully-actuated model have been proposed [Rajappa et al., 2015, Park et al., 2018, Anzai et al., 2018], which are not available for the under-actuated model with four tilting propellers . Then, a nonlinear model predictive control method is proposed by [Shi et al., 2019] to address the under-actuation, and the stability of the translational and yaw rotational motion is achieved. However, this control method highly depends on the accurate dynamics model, thus the model error is not allowed. Such condition would lead to a difficulty in stabilizing during the grasping task, since the model offset resulted from the additional inertial parameter of the grasped object can not be perfectly compensated. Thus, a more robust control method is developed in this work which is based on the cascaded control flow similar to [Zhao et al., 2016].

1.1.3 State Estimation

The autonomous flight is achieved by the state estimation fusing mutiple sensors. In addition to GPS which is a useful sensor to get a global position in outdoor environment, visual odometry (VO) and visual-inertial-odometry (VIO) are also the effective methods to obtain the robot motion [Qin et al., 2018, Forster et al., 2017, Bloesch et al., 2017, Mur-Artal et al., 2015, Sun et al., 2018]. Several VIO based state estimation methods used in MBZIRC 2017 are introduced by [Tzoumanikas et al., 2019, Bähnemann et al., 2019].

Sensors have measurement delays. Ignoring the measurement delay would significantly decrease the estimation performance. Therefore, a time-synchronized algorithm in sensor fusion is necessary. A time-synchronized extended Kalman filter framework which can handle multiple sensors with different measurement delays is first proposed by [Lynen et al., 2013]. In our work, a similar time-synchronized framework with a limited buffer is applied.

On the other hand, the consideration of the kinematics is significantly important in a multilinked model,

since the relative pose of each sensor changes according to the joint angles. Thus, the proper transformation for both input and output of extended Kalman filter is necessary to guarantee the accurate estimation and stable control during deformation.

1.2 Main Contribution

To the best of our knowledge, this is the first study for the multilinked aerial robot to achieve the fully autonomous flight and the manipulation task in outdoor environment. In short, the main contribution of this study which can benefit the field robotics community is summarized as follows:

- we design a multilinked aerial robot containing two joints and four tilting propellers to enhance the controlability in yaw rotational motion, and also introduce an optimal design method for the tilt angle.

- we derive the modeling of the under-actuated model with tilting propellers and also develop the control method. The stability analysis based on the Lyapunov theory is also presented.

- we develop the state estimation framework for the multilinked model which takes the kinematics and sensor measurement delay into account to enable autonomous flight in outdoor environment.

- we perform various experiments including fast maneuvering and aerial manipulation to demonstrate the feasibility of our design, modeling, control and state estimation method for fully autonomous flight.

1.3 Notation

All the symbols in this paper are explained at their first appearance. Boldface symbols (most are lowercase, e.g., 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, e.g., ${}^{\{C\}}r$ expresses r with reference to (w.r.t.) the frame $\{C\}$. Subscript are used to express a relation or attribute, e.g., ${}^{\{C\}}r_x$ represents the value of position on the x axis w.r.t. the frame $\{C\}$.

1.4 Organization

The remainder of this paper is organized as follows: the design and modeling for the multilinked structure along with the optimal design method for the propeller tilt angle are presented in Sec. 2. Then the control method and the state estimation are presented in Sec. 3 and Sec. 4 respectively, followed by the description on the hardware and software system for a real robot platform in Sec. 5. Last, we show the experiment results in Sec. 6 before concluding in Sec. 7.

2 Design and Modeling

Although the multilinked aerial robot with tilting propellers has been already developed in our previous work [Anzai et al., 2018, Shi et al., 2019, Shi et al., 2019], none of theses works state the concrete design methodology regarding the tilting propeller. Thus, in this section, we will reveal the influence of the tilt angle on the flight performance to further derive an optimal design method for a multilinked aerial robot.

2.1 Mechanical Design

The proposed multilinked aerial robot as shown in Fig. 2(A) is composed from three parts which are connected by two joints, namely $\boldsymbol{q} = [q_1 \ q_2]^{\mathrm{T}}$. Joints are actuated by servos and the rotating axes are parallel, resulting in an ability of two-dimensional deformation. A critical difference from the four-link model in our previous work is that the joint between link2 and link3 is replaced by a fixed connection which can greatly enhance the entire rigidity. Although the freedom-of-dimension (dof) of deformation decreases, such simplified design can still achieve various manipulation task such as grasping object by regarding the whole body like a gripper. For convenience in modeling, we still regard the central part as two separated links, namely link2 and link3.

Regarding each link module as shown in Fig. 2(B), there is a fixed angle $(-1)^i\beta$ to tilt the propeller for the enhancement of the controllabiltiy in yaw rotational motion. The tilting direction is opposite between neighboring links to increase the z axis torque in both direction. Besides, the propeller spins oppositely regarding neighboring propellers. Furthermore, the propeller duct is designed not only for safety, but also for grasping.
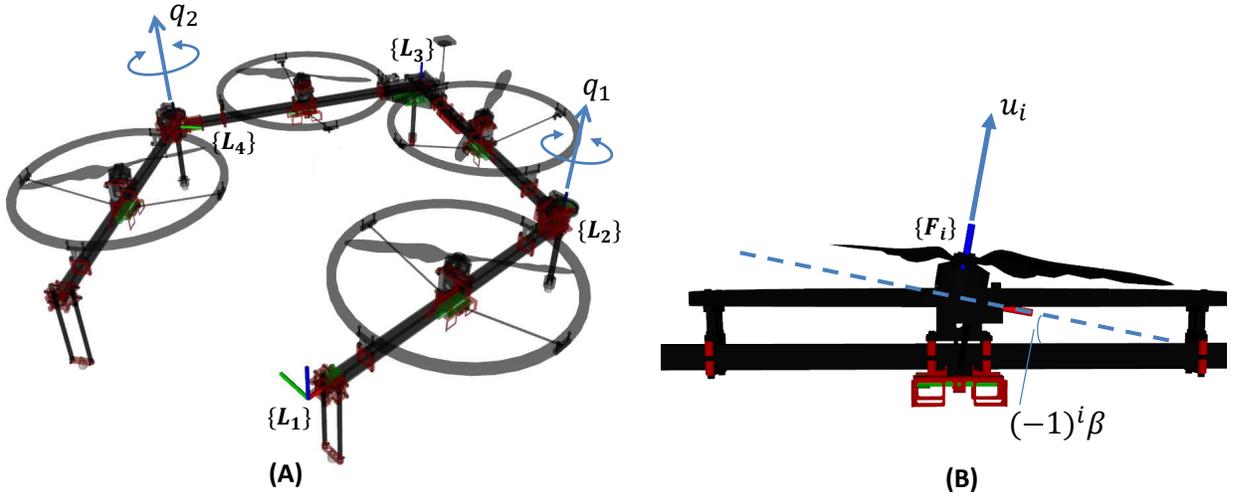
Figure 2: **A**: the mechanical structure of the multilinked aerial robot composed from four links (the central part is separated into two links: link2 and link3). $\{L_i\}$ corresponds to the origin of each link. **B**: the structure of the link module with the embedded propeller which has a fixed tilting angle $(-1)^i\beta$. u_i denotes the thrust force acting on the rotor frame $\{F_i\}$.

2.2 Basic Modeling

We first define an intermediate frame, namely $\{C\}$ as shown Fig. 3, which has following attributes:

$$^{\{L_1\}}\boldsymbol{p}_{\{C\}} \ = \ \frac{1}{m_\Sigma} \sum_i^4 m_{L_i} \, ^{\{L_1\}}\boldsymbol{p}_{C_i}(\boldsymbol{q}) \tag{1}$$

$$^{\{L_1\}}R_{\{C\}} \ = \ E_{3\times3} \tag{2}$$

where, $\{L_1\}$ is the frame of link1 which is the root of whole model, and $^{\{L_1\}}\boldsymbol{p}_{C_i}$ is the position of center of gravity (CoG) of i-th link which is variant because of the joint angles $\boldsymbol{q} \in \mathcal{R}^2$. m_Σ is the overall mass, i.e., $m_\Sigma = \sum m_{L_i}$. Eq. 1 and Eq. 2 indicate that the origin of $\{C\}$ is identical to the CoG of the whole model, and the direction of frame axes of $\{C\}$ is identical to that of $\{L_1\}$. Note that, the frame $\{C\}$ is not suitable to serve as the reference frame in flight control, and thus a further derived frame $\{CoG\}$ will be introduced in Sec. 3.
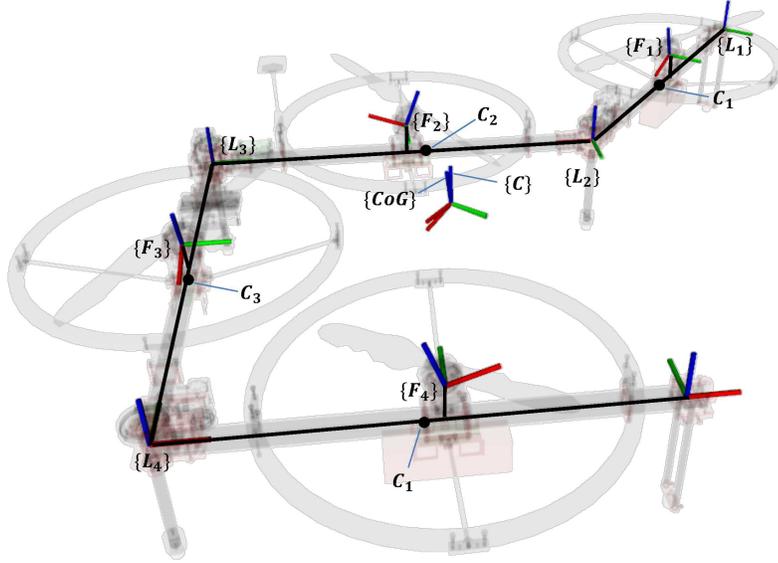
Figure 3: The kinematics model of the multilinked structure. The frame $\{C\}$ and $\{CoG\}$ have the same origin which are all identical to the center of gravity of the whole model. However, the orientation of the frame $\{C\}$ is identical to the frame $\{L_i\}$, while the orientation of the frame $\{CoG\}$ is specially defined in this work for flight control which is presented in Sec. 3.

Then, the wrench generated by each rotor w.r.t the frame $\{C\}$ can be written as follows:

$$
\begin{aligned}
{}^{\{C\}}\boldsymbol{f}_i &= {}^{\{C\}}R_{\{F_i\}}(\boldsymbol{q})u_i && (3) \\
{}^{\{C\}}\boldsymbol{\tau}_i &= {}^{\{C\}}\boldsymbol{p}_{\{F_i\}}(\boldsymbol{q}) \times {}^{\{C\}}\boldsymbol{f}_i = {}^{\{C\}}\hat{\boldsymbol{p}}_{\{F_i\}}(\boldsymbol{q}){}^{\{C\}}R_{\{F_i\}}(\boldsymbol{q})u_i && (4)
\end{aligned}
$$

where, $u_i \in \mathcal{R}$ is the thrust force generated by i-th rotor in the rotor frame $\{F_i\}$. Note that, $\hat{\ }$ denotes the operation from a vector to a skew-symmetric matrix.

Using Eq. 3 and Eq. 4, the total wrench can be summarized as:

$$
\left(\begin{array}{c} {}^{\{C\}}\boldsymbol{f} \\ {}^{\{C\}}\boldsymbol{\tau} \end{array} \right) = \sum_{i=1}^{N} \left(\begin{array}{c} {}^{\{C\}}\boldsymbol{f}_i \\ {}^{\{C\}}\boldsymbol{\tau}_i \end{array} \right) = \left(\begin{array}{c} Q'_{\text{tran}}(\boldsymbol{q}) \\ Q'_{\text{rot}}(\boldsymbol{q}) \end{array} \right) \boldsymbol{u} = Q'(\boldsymbol{q})\boldsymbol{u} \tag{5}
$$

where $\boldsymbol{u} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^{\mathrm{T}}$.

A hovering state is the zero equilibrium that total wrench expressed by Eq. 5 balances with gravity. However the allocation matrix $Q'(\boldsymbol{q}) \in \mathcal{R}^{6\times 4}$ reveals the difficulty to use four dof input to manipulate six dof output independently, which is also the reason of the under-actuation. Therefore, in most of the case, it is unable to find $\boldsymbol{u}$ that satisfies following condition: $\begin{bmatrix} 0 & 0 & m_\Sigma g & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} = Q'(\boldsymbol{q})\boldsymbol{u}$. However, it is possible to find a solution $\boldsymbol{u}_s$ for following relaxed equations:

$$
\begin{aligned}
\|{}^{\{C\}}\boldsymbol{f}\| &= \|Q'_{\text{tran}}\boldsymbol{u}_s\| = m_\Sigma g && (6) \\
\boldsymbol{0} &= Q'_{\text{rot}}\boldsymbol{u}_s && (7)
\end{aligned}
$$

It is notable that $\boldsymbol{u}_s$ is the hovering thrust vector, since $Q'_{\text{tran}}\boldsymbol{u}_s$ can be converted to $\begin{bmatrix} 0 & 0 & m_\Sigma g \end{bmatrix}^\text{T}$ by a certain rotational transformation. To obtain $\boldsymbol{u}_s$, we first find a thrust vector $\boldsymbol{u}'$ which only balances with unit z axis and torque:

$$\boldsymbol{u}' = \left(\begin{array}{c} Q'_{\text{tran}_z}(\boldsymbol{q}) \\ Q'_{\text{rot}}(\boldsymbol{q}) \end{array} \right)^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\text{T} \tag{8}$$

where, $Q'_{\text{tran}_z}(\boldsymbol{q}) \in \mathcal{R}^{1\times4}$ is the third row vector of $Q'_{\text{tran}}(\boldsymbol{q})$.

Using Eq. 8, $\boldsymbol{u}_s$ can be given by

$$\boldsymbol{u}_s = \frac{m_\Sigma g}{\|Q'_{\text{tran}}(\boldsymbol{q})\boldsymbol{u}'\|}\boldsymbol{u}' \tag{9}$$

2.3 Design of Tilting Angle for Propeller

Regarding the tilting propeller as shown in Fig. 2(B), the thrust force can be divided into the vertical component $cos(\beta)u_i$ and the horizontal component $sin(\beta)u_i$. The relationship between the vertical component and the energy efficiency is revealed by [Ryll et al., 2016], whereas the influence of the horizontal component on the yaw rotational motion is qualitatively discussed in [Shi et al., 2019]. However, there are other critical factors, such as the influence on the horizontal motion, should be also taken into account to design the tilting angle. In this work, we present four factors (i.e., hovering thrust, torque in z axis, horizontal force and aerodynamic interference) associated with the tilting propeller, which are further integrated into an optimal problem to obtain the best tilting angle. Note that, we apply the model specification of the actual platform as shown in Fig. 11 to quantify the model parameters (e.g., the link length, mass).

2.3.1 Hovering Thrust

The hovering thrust $\boldsymbol{u}_s$ calculated from Eq. 9 is the thrust vector in hovering situation, which corresponds to the energy efficiency in most of the situation. We explicitly add the joint angles $\boldsymbol{q}$ and the propeller tilt angle β as the variables of $\boldsymbol{u}_s$, and define the maximum and minimum elements as $u_{s_{\max}}(\boldsymbol{q}, \beta)$ and $u_{s_{\min}}(\boldsymbol{q}, \beta)$, respectively. Then, we analysis the influence of tilt angle β on $u_{s_{\min}}$ and $u_{s_{\max}}$ by changing the angle from 0 rad to 0.8 rad and fixing the joint angles (i.e. $\boldsymbol{q} = [\frac{\pi}{2} \ \frac{\pi}{2}]^\text{T}$ rad) as shown in Fig. 4(A). Both plots demonstrate the monotonous increase that are identical to the behaviors of $\frac{u_{s_{\min}}(\boldsymbol{q},0)}{cos(\beta)}$ and $\frac{u_{s_{\max}}(\boldsymbol{q},0)}{cos(\beta)}$. On the other hand, we also fix the tilt angle (i.e., $\beta = 10$ deg), and change the joint angles (i.e., $q_i \in \left[-\frac{\pi}{2} \ \frac{\pi}{2}\right]$) as shown in Fig. 4(B) and (C). From the plot results, we can confirm that the gap between the $u_{s_{\max}}(\boldsymbol{q})$ and $u_{s_{\min}}(\boldsymbol{q})$ becomes larger when the joint angle q_i is close to $-\frac{\pi}{2}$ rad, implying the force efficiency and the stability become worse under such form. Furthermore, the hovering thrust should be also within the valid range of the thrust force, i.e., $u_{s_{\min}}(\boldsymbol{q}) \geq 0, u_{s_{\max}}(\boldsymbol{q}) \leq u_{\max}$, which is an important factor to clarify the valid deformation range.

2.3.2 Torque in Z Axis

The weakness of torque in z axis compared with other axes (i.e., $\tau_z \sim 0.1\tau_{xy}$) is one of the critical issues regarding the general multirotor, since torque in z axis can only be generated from the rotor's drug moment which is significantly small . This leads to the insufficient robustness against the yaw rotational disturbance [Anzai et al., 2017], and also induces the thrust force saturation which further influences the stability in other axes.
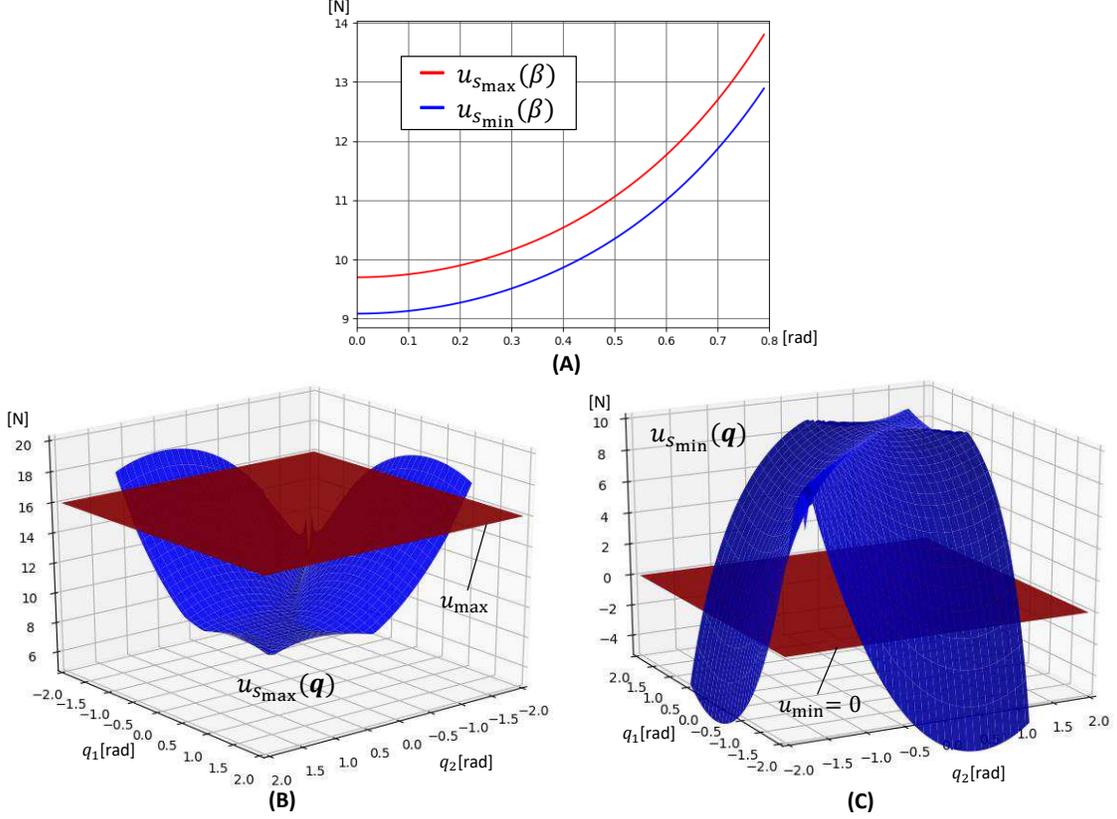
Figure 4: **(A)**: the behaviors regarding the maximum and the minimum components of the hovering thrust vector $\boldsymbol{u}_s$ along the change in the tilt angle β with fixed joint angles (i.e., $q_i = \frac{\pi}{2}$ rad). **(B)** the behavior of the maximum component of $\boldsymbol{u}_s$ along the change in the joint angles (i.e., $q_i \in \left[-\frac{\pi}{2}\ \frac{\pi}{2}\right]$) with fixed tilt angle (i.e., $\beta = 10$ deg). The red plane denotes the upper limit of the thrust that the rotor can generate (i.e., 16 N). **(C)** the behavior of the minimum component of $\boldsymbol{u}_s$ along the change in the joint angles with fixed tilt angle (i.e., $\beta = 10$ deg). The red plane denotes the lower limit of thrust (i.e., 0 N).

The minimum and maximum of the z axis torque can be obtained by solving following problems:

$$\tau_{z_{\min}}(\boldsymbol{q}, \beta) = \min_{\boldsymbol{u}} Q'_{\mathrm{rot}_z}(\boldsymbol{q}, \beta)\boldsymbol{u} \tag{10}$$

$$s.t. \quad 0 \le u_i \le u_{\max}$$

$$\tau_{z_{\max}}(\boldsymbol{q}, \beta) = \max_{\boldsymbol{u}} Q'_{\mathrm{rot}_z}(\boldsymbol{q}, \beta)\boldsymbol{u} \tag{11}$$

$$s.t. \quad 0 \le u_i \le u_{\max}$$

where, $Q'_{\mathrm{rot}_z}(\boldsymbol{q}, \beta)$ is the third row vector of the matrix $Q'_{\mathrm{rot}}(\boldsymbol{q}, \beta)$.

Again, we first analysis the influence of tilt angle β on these values $\tau_{z_{\min}}$ and $\tau_{z_{\max}}$ by changing the angle from 0 rad to 0.8 rad and fixing the joint angles (i.e. $\boldsymbol{q} = [\frac{\pi}{2}\ \frac{\pi}{2}]^{\mathrm{T}}$ rad) as shown in Fig. 5(A). Both absolute values monotonically increases because z axis torque generated by tilting propeller are associated with the sine of the tile angle. We then fix the tilt angle (i.e., $\beta = 10$ deg) and change the joint angles (i.e., $q_i \in \left[-\frac{\pi}{2}\ \frac{\pi}{2}\right]$) as shown in Fig. 5(B) and (C). When $q_i \approx -\frac{\pi}{2}$, the absolute value of either $\tau_{z_{\max}}(\boldsymbol{q})$ or $\tau_{z_{\min}}(\boldsymbol{q})$ is relatively small, leading to the relatively weak controlability regarding the yaw rotational motion under such robot form.
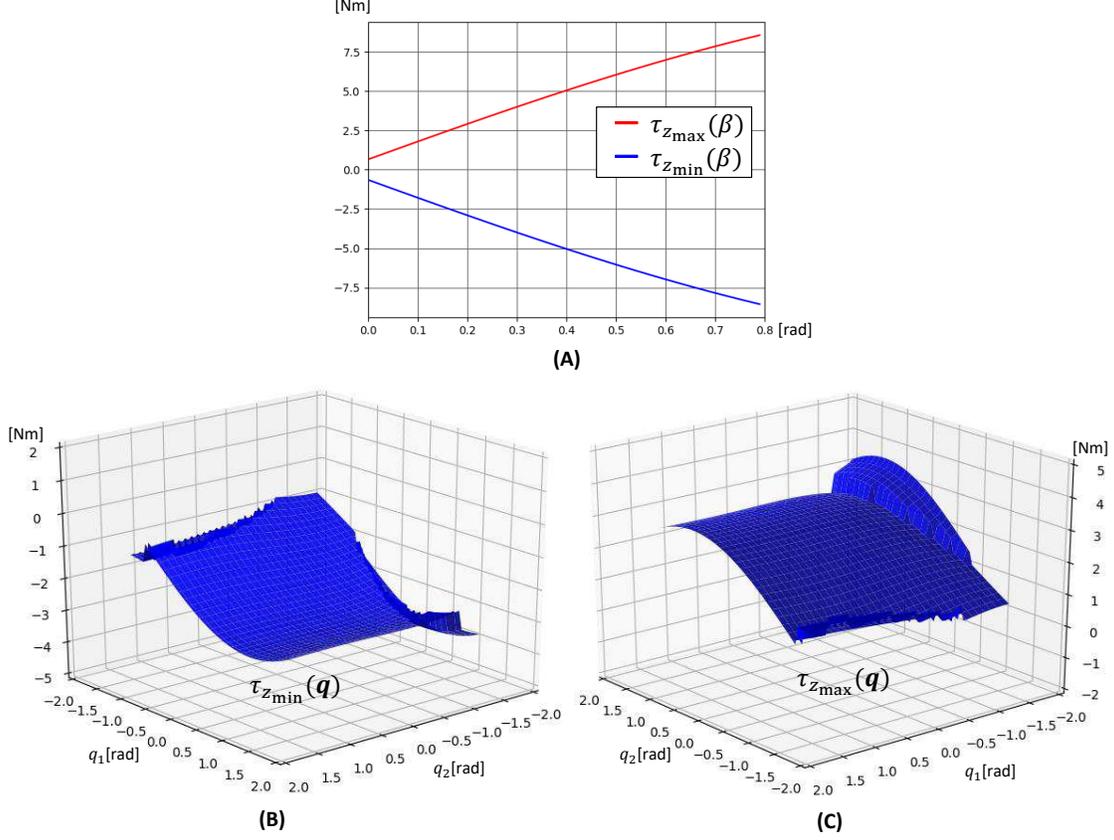
Figure 5: **(A)**: the behaviors of the maximum and the minimum of the z axis torque along the change in the tilt angle with fixed joint angles (i.e., $q_i = \frac{\pi}{2}$ rad). **(B)** the behavior of the minimum of z axis torque along the change in the joint angles (i.e., $q_i \in \left[-\frac{\pi}{2} \ \frac{\pi}{2}\right]$) with fixed tilt angle (i.e., $\beta = 10$ deg). **(C)** the behavior of the maximum of z axis torque along the change in the joint angles (i.e., $q_i \in \left[-\frac{\pi}{2} \ \frac{\pi}{2}\right]$) with fixed tilt angle (i.e., $\beta = 10$ deg).

2.3.3 Horizontal Force

The horizontal force $\boldsymbol{f}_{xy} \in \mathcal{R}^2$ corresponds to the first two elements of ${}^{\{C\}}\boldsymbol{f}$ which can be calculated from Eq. 5. For fully-/over-actuated model, the horizontal force is beneficial to the control in horizontally translational motion. However, for an under-actuated model, the horizontal motion is controlled by changing the robot attitude, and thus the intervention of this force would break the position stability. To clarify the Given a desired torque ${}^{\{C\}}\boldsymbol{\tau}^{\text{des}}$ obtained from a attitude controller, a derived horizontal force $\boldsymbol{f}_{xy}$ can be calculated by following equation:

$$\boldsymbol{f}_{xy}(\boldsymbol{q}, \beta, {}^{\{C\}}\boldsymbol{\tau}^{\text{des}}) = Q'_{\text{tran}_{xy}}(\boldsymbol{q}, \beta)Q'^{\#}_{\text{rot}}(\boldsymbol{q}, \beta){}^{\{C\}}\boldsymbol{\tau}^{\text{des}} \tag{12}$$

where the matrix operation $(\cdot)^{\#}$ denotes the MP inverse which corresponds to the minimum norm of $\boldsymbol{u}$ to generate ${}^{\{C\}}\boldsymbol{\tau}^{\text{des}}$. Also note that, $Q'_{\text{tran}_{xy}}(\boldsymbol{q}) \in \mathcal{R}^{2 \times 4}$ corresponds to the top two rows of Q'_{tran}.

To simplify the influence of the rotational motion in the translational motion, a fixed desired torque ${}^{\{C\}}\boldsymbol{\tau}^{\text{des}} = [1\ 1\ 1]^{\text{T}}$ is introduced. Then it is possible to validate the relationship between the norm of the horizontal force $\|\boldsymbol{f}_{xy}(\boldsymbol{q}, \beta)\|$ and the tilt angle and joint angles. Again, we first analysis the influence of tilt angle β on $\|\boldsymbol{f}_{xy}(\boldsymbol{q}, \beta)\|$ with fixed joint angles (i.e. $\boldsymbol{q} = [\frac{\pi}{2} \ \frac{\pi}{2}]^{\text{T}}$ rad) as shown in Fig. 6(A), which demonstrates a monotonous increase and also an increase in inclination. We consider that $Q'_{\text{tran}_{xy}}(\boldsymbol{q}, \beta)$ and $Q'^{\#}_{\text{rot}}(\boldsymbol{q}, \beta)$ has

the element of $sin(\beta)$ and $\frac{1}{cos(\beta)}$ respectively, thus $\|\boldsymbol{f}_{xy}(\beta)\| \propto tan(\beta)$. Then, we fix the tilt angle (i.e., $\beta = 10$ deg), and change the joint angles (i.e., $q \in \left[-\frac{\pi}{2} \ \frac{\pi}{2}\right]$) as shown in Fig. 6(B). $\|\boldsymbol{f}_{xy}(q)\|$ should be as small as possible. However, the value divergence when both q_1 and q_2 are close to $-\frac{\pi}{2}$ rad, since this is a singular form where all propeller are aligned on the same line.
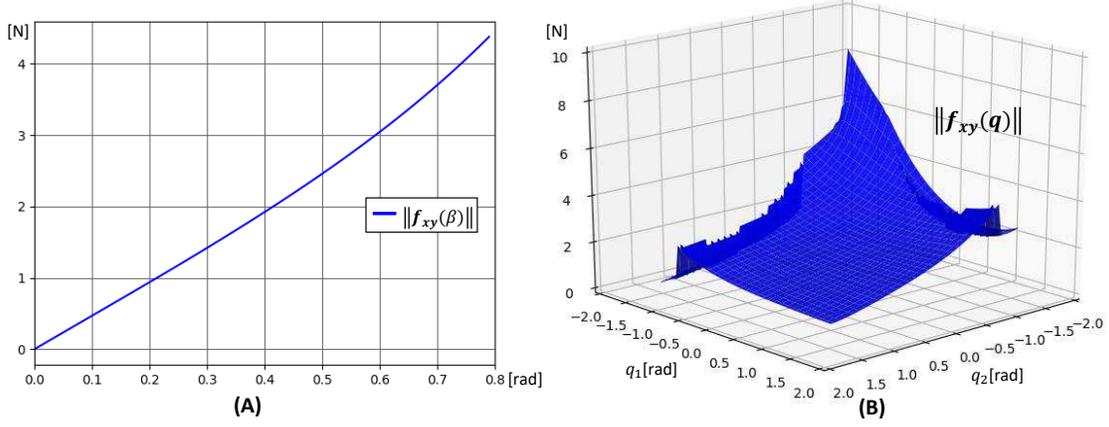
Figure 6: **(A)**: the behavior of the norm of the horizontal force $\|\boldsymbol{f}_{xy}(\beta)\|$ calculated from Eq. 12 with $\boldsymbol{\tau}^{\mathrm{des}} = [1\ 1\ 1]^{\mathrm{T}}$ and fixed joint angles (i.e., $q_i = \frac{\pi}{2}$ rad) along the change in the tilt angle. **(B)** the behavior of the norm of the horizontal force $\|\boldsymbol{f}_{xy}(q)\|$ along the change in the joint angles with fixed tilt angle (i.e., $\beta = 10$ deg).

2.3.4   Aerodynamic Interference

Is is also necessary to consider the aerodynamic interference from the tilting propeller, since the airflow form propeller hits on the downstream link rod as shown Fig. 2(B). The influence range can be given by

$$l_{\mathrm{air}}(\beta) \approx \frac{D_{\mathrm{prop}}}{cos(\beta)} \tag{13}$$

where, D_{prop} is the diameter of the propeller. It is also notable that, when the tilt angle increases, the airflow will act not only on the link rod but also on the side component, such as, the duct. Thus, it is necessary to suppress such aerointerference.

2.3.5   Optimization Problem

Our goal is to propose a design method which takes above four factors into account to find an optimal tilt angle for multilinked aerial robot. Although we only validate the influence of the change in the tilt angle β under a single form (i.e., $q_i = \frac{\pi}{2}$ rad), it can be considered that the behavior in other forms is identical. Thus, the optimization problem to find the best tilt angle is designed under $q_i = \frac{\pi}{2}$ rad. Then, a maximization of the following weighted sum function is introduced:

$$\max_{\beta} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}(\beta) \tag{14}$$

$$\boldsymbol{x} = \begin{bmatrix} -u_{s_{\max}}(\beta) + u_{s_{\min}}(\beta) & \tau_{z_{\max}}(\beta) - \tau_{z_{\min}}(\beta) & -\|\boldsymbol{f}_{xy}(\beta)\| & -l_a(\beta) \end{bmatrix}^{\mathrm{T}}$$

where, $\boldsymbol{w}$ is the weight vector to normalize each component in $\boldsymbol{x}(\beta)$. Given a robot specification from the actual platform as shown Fig. 11, the optimal tilt angle is 0.1745 rad ($= 10$ deg). This is the result which puts more weight on the suppression of the horizontal force $\boldsymbol{f}_{xy}(q, \beta, {}^{\{C\}}\boldsymbol{\tau}^{\mathrm{des}})$ to guarantee the position stability. More discussion on the stability and tilt angle will be presented in the Sec. 3. Note that, a

larger tilt angle (i.e. 20 deg) is applied in other works [Shi et al., 2019, Shi et al., 2019]. This is due to the different robot specification (e.g., no battery in the experimental validation). Furthermore, the nonlinear model predict control is applied in those works, which can utilize such horizontal force for position control even in under-actuation model, but with the expense of the high computational cost.

2.4 Valid Deformation Range

In addition to the hovering thrust, the controlability regarding the rotational motion is another important factor to check the validity of a form. For torque in z axis, the maximum and minimum value should be positive and negative respectively, i.e., $\tau_{z_{\max}}(\boldsymbol{q}) > 0$, $\tau_{z_{\min}}(\boldsymbol{q}) < 0$. However, these constraints only corresponds to the validity of yaw rotational motion. It is necessary to check the comprehensive motion in all three axes. Therefore an extended quantity called feasible control torque convex $\mathcal{V}_T(\boldsymbol{q})$ [Park et al., 2018, Anzai et al., 2019] is applied to validate the rotational controllability which is given by

$$\mathcal{V}_T(\boldsymbol{q}) := \{^{\{C\}}\boldsymbol{\tau}(\boldsymbol{q}) \in \mathcal{R}^3 |^{\{C\}}\boldsymbol{\tau}(\boldsymbol{q}) = \sum_{i=0}^{N} u_i \boldsymbol{v}_i(\boldsymbol{q}), 0 \leq u_i \leq u_{\max}\} \tag{15}$$

where, $\boldsymbol{v}_i(\boldsymbol{q}) = {}^{\{C\}}\hat{\boldsymbol{p}}_{\{F_i\}}(\boldsymbol{q}){}^{\{C\}}R_{\{F_i\}}(\boldsymbol{q})$ according to Eq. 4.

We further introduce the guaranteed minimum control torque $\tau_{\min}(\boldsymbol{q})$ which has following property:

$$\|^{\{C\}}\boldsymbol{\tau}(\boldsymbol{q})\| \leq \tau_{\min}(\boldsymbol{q}) \Rightarrow \boldsymbol{\tau}(\boldsymbol{q}) \in \mathcal{V}_T(\boldsymbol{q}), \tag{16}$$

Using the distance $d_{ij}^{\tau}(\boldsymbol{q})$ which is from the origin to a plane of convex $\mathcal{V}_T$ along its normal vector $\frac{\boldsymbol{v}_i \times \boldsymbol{v}_j}{\|\boldsymbol{v}_i \times \boldsymbol{v}_j\|}$, the guaranteed minimum control torque can be given by:

$$d_{ij}^{\tau}(\boldsymbol{q}) = \sum_{k=0}^{N} \max(0, \frac{(\boldsymbol{v}_i \times \boldsymbol{v}_j)^{\mathrm{T}}}{\|\boldsymbol{v}_i \times \boldsymbol{v}_j\|} \boldsymbol{v}_k), \tag{17}$$

$$\tau_{\min}(\boldsymbol{q}) = \min_{i,j \in \mathcal{I}} d_{ij}^{\tau}(\boldsymbol{q}). \tag{18}$$

where, $\mathcal{I} := \{1, 2, \cdots, N\}$. It is obvious that the condition to guarantee the torque controlability is $\tau_{\min}(\boldsymbol{q}) > 0$.

Fig. 7(A) shows relationship between $\tau_{\min}(\boldsymbol{q})$) and joint angles $\boldsymbol{q}$, while Fig. 7(B)-(D) demonstrate several distinguish cases of $\mathcal{V}_T$ and $\tau_{\min}$ with differential forms. It can be confirmed that the form of Fig. 7(D) stretches the convex and thus $\tau_{\min}$ becomes significantly small. However, the shrink direction is not perfectly identical to the z axis, which indicates the importance to apply $\tau_{\min}$ instead of only evaluating z axis (i.e., $\tau_{z_{\max}}(\boldsymbol{q}), \tau_{z_{\min}}(\boldsymbol{q})$).

To summarize, the valid range of the joint angles can be given by:

$$\mathcal{D}_q := \{\boldsymbol{q} \in \mathcal{R}^2 | u_{s_{\min}}(\boldsymbol{q}) \geq u_{\mathrm{thre}}, u_{s_{\max}}(\boldsymbol{q}) \leq u_{\max} - u_{\mathrm{thre}}, \tau_{\min}(\boldsymbol{q}) \geq \tau_{\mathrm{thre}}\} \tag{19}$$

where, u_{thre} and τ_{thre} are the positive thresholds to provide certain control margin.

3   Control

An improved control framework as shown in Fig. 8 based on our previous work [Zhao et al., 2016] has been developed in this work. The dynamics and control is described in a specially defined frame $\{CoG\}$. Subsequently the model approximation is introduced to simply the dynamics, which is followed by a cascaded control flow.
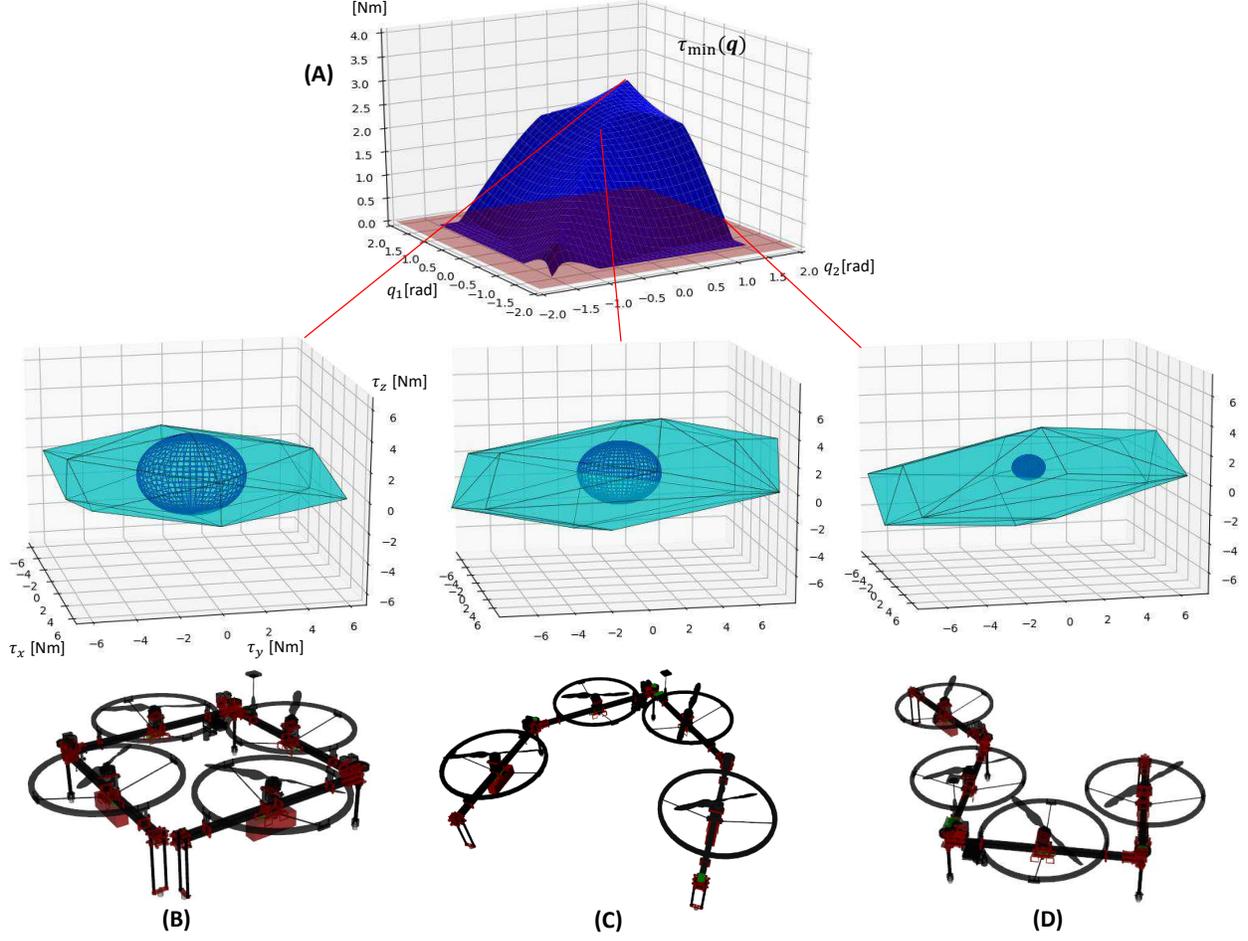
Figure 7: **(A)**: the behavior of the guaranteed minimum control torque $\tau_{\min}(\boldsymbol{q})$ along the change in joint angles. **(B)-(D)**: feasible control torque convex $\mathcal{V}_T$ and the sphere of which the radius is $\tau_{\min}(\boldsymbol{q})$ under different robot forms as shown in the bottom images (i.e., **B**: $\boldsymbol{q} = \in \left[\frac{\pi}{2} \; \frac{\pi}{2}\right]$, **C**: $\boldsymbol{q} = \in \left[\frac{\pi}{4} \; \frac{\pi}{4}\right]$, **D**: $\boldsymbol{q} = \in \left[-\frac{\pi}{4} \; \frac{\pi}{2}\right]$).

3.1 Definition of CoG Frame

As shown in Eq. 6, in most cases, the frame $\{C\}$ could not be level in hovering state. Then, a new frame which is always level in hovering state is introduced to to reduce the nonlinearity of the dynamics and control. We define such a frame as $\{CoG\}$.

In terms of the relation between the frame of $\{CoG\}$ and $\{C\}$, there must exist a rotation matrix $^{\{CoG\}}R_{\{C\}}$ which satisfies following transformation:

$$^{\{CoG\}}R_{\{C\}}Q'_{\text{tran}}\boldsymbol{u}_s = {}^{\{CoG\}}R_{\{C\}}{}^{\{C\}}\boldsymbol{f}_s = m_\Sigma\boldsymbol{g} \tag{20}$$

where, $\boldsymbol{g} = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^{\mathrm{T}}$, and $\boldsymbol{u}_s$ is the hovering thrust vector calculated from Eq. 9.

Then, Euler angles $\boldsymbol{\alpha} = \begin{pmatrix} \alpha_x & \alpha_y & \alpha_z \end{pmatrix}^{\mathrm{T}}$ are introduced to represent $^{\{CoG\}}R_{\{C\}}$:

$$^{\{CoG\}}R_{\{C\}} = R_Y(\alpha_y)R_Y(\alpha_x) \tag{21}$$

$$\alpha_x = tan^{-1}({}^{\{C\}}f_{s_y}, {}^{\{C\}}f_{s_z}) \tag{22}$$

$$\alpha_y = tan^{-1}(-{}^{\{C\}}f_{s_x}, \sqrt{{}^{\{C\}}f_{s_y}^2 + {}^{\{C\}}f_{s_z}^2}) \tag{23}$$
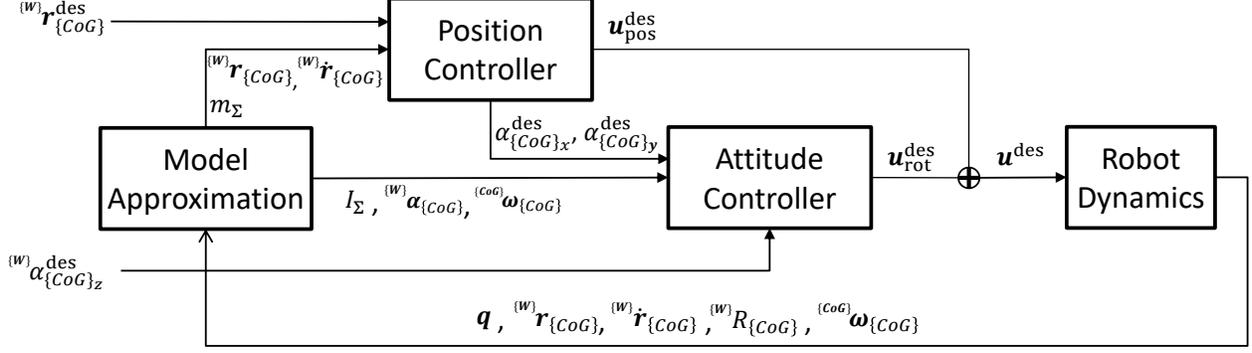
Figure 8: The proposed control framework for multilinked aerial robot which involves the dynamics model approximation and a cascaded control flow including position and attitude controller.

where, $R_X(\cdot), R_Y(\cdot)$ are the special rotation matrix which only rotates along x and y axis, respectively.

Finally, the allocation matrix Q w.r.t the frame $\{CoG\}$ can be given by

$$Q(\boldsymbol{q}) = \left(\begin{array}{c} Q_{\mathrm{tran}}(\boldsymbol{q}) \\ Q_{\mathrm{rot}}(\boldsymbol{q}) \end{array} \right) = \left(\begin{array}{c} {}^{\{CoG\}}R_{\{C\}}Q'_{\mathrm{tran}}(\boldsymbol{q}) \\ {}^{\{CoG\}}R_{\{C\}}Q'_{\mathrm{rot}}(\boldsymbol{q}) \end{array} \right) \tag{24}$$

3.2 Dynamics

As stated in our previous work [Zhao et al., 2018], we assume the joint motion is sufficiently slow. Then the multilinked model can be regarded as a time-variant rigid body. Thus, the dynamics regarding the CoG frame can be simplified as follows:

$$m_{\Sigma}({}^{\{W\}}\ddot{\boldsymbol{r}}_{\{CoG\}} + \boldsymbol{g}) = {}^{\{W\}}R_{\{CoG\}}{}^{\{CoG\}}\boldsymbol{f} + \boldsymbol{\Delta}_{\mathrm{tran}} = {}^{\{W\}}R_{\{CoG\}}Q_{\mathrm{tran}}(\boldsymbol{q})\boldsymbol{u} + \boldsymbol{\Delta}_{\mathrm{tran}} \tag{25}$$

$${}^{\{CoG\}}I_{\Sigma}(\boldsymbol{q}){}^{\{CoG\}}\dot{\boldsymbol{\omega}}_{\{CoG\}} + {}^{\{CoG\}}\boldsymbol{\omega}_{\{CoG\}} \times {}^{\{CoG\}}I_{\Sigma}(\boldsymbol{q}){}^{\{CoG\}}\boldsymbol{\omega}_{\{CoG\}} = {}^{\{CoG\}}\boldsymbol{\tau} + \boldsymbol{\Delta}_{\mathrm{rot}} = Q_{\mathrm{rot}}(\boldsymbol{q})\boldsymbol{u} + \boldsymbol{\Delta}_{\mathrm{rot}} \tag{26}$$

where ${}^{\{W\}}\boldsymbol{r}_{\{CoG\}}$ and ${}^{\{W\}}R_{\{CoG\}}$ are the position and attitude of the frame $\{CoG\}$ w.r.t the world frame $\{W\}$, respectively. These states can be calculated based on the forward-kinematics from the states of the root link (i.e., ${}^{\{W\}}\boldsymbol{r}_{\{L_1\}}$, ${}^{\{W\}}\dot{\boldsymbol{r}}_{\{L_1\}}$ and ${}^{\{W\}}R_{\{L_1\}}$), while the angular velocity can be obtained by ${}^{\{CoG\}}\dot{\boldsymbol{\omega}}_{\{CoG\}} = {}^{\{CoG\}}R_{\{C\}}{}^{\{L_1\}}\boldsymbol{\omega}_{\{L_1\}}$. The total inertial matrix ${}^{\{CoG\}}I_{\Sigma}(\boldsymbol{q})$ w.r.t the frame $\{CoG\}$ can be also calculated from the forward-kinematics process. We explicitly consider the unstructured but fixed uncertainties $\boldsymbol{\Delta}_{\mathrm{tran}}$ and $\boldsymbol{\Delta}_{\mathrm{rot}}$ (e.g., the model offset caused by the additional weight in object grasping task) to show the compensation ability of the proposed control framework.

We further assume that most of the flight tasks are performed in near-hover condition. Thus, the approximation between the differential of Euler angles and angular velocity ${}^{\{W\}}\dot{\boldsymbol{\alpha}}_{\{CoG\}} \approx {}^{\{CoG\}}\boldsymbol{\omega}_{\{CoG\}}$ is available, since ${}^{\{W\}}\alpha_{\{CoG\}_x} \approx 0$, ${}^{\{W\}}\alpha_{\{CoG\}_y} \approx 0$. Then the rotational dynamics expressed in Eq. 26 can be further linearized in the near-hover condition:

$$ {}^{\{CoG\}}I_{\Sigma}(\boldsymbol{q}){}^{\{W\}}\ddot{\boldsymbol{\alpha}}_{\{CoG\}} + {}^{\{CoG\}}\boldsymbol{\omega}_{\{CoG\}} \times {}^{\{CoG\}}I_{\Sigma}(\boldsymbol{q}){}^{\{CoG\}}\boldsymbol{\omega}_{\{CoG\}} = Q_{\mathrm{rot}}(\boldsymbol{q})\boldsymbol{u} + \boldsymbol{\Delta}_{\mathrm{rot}} \tag{27}$$

Finally, Eq. 25 and Eq. 27 are the fundamental dynamics which are used in the proposed control framework. It is also notable that $Q_{\mathrm{tran}}(\boldsymbol{q})\boldsymbol{u}$ in Eq. 25 implies the full influence on all translational axes without rotation ${}^{\{W\}}R_{\{CoG\}}$. Such special property is the main difference from a general quadrotor [Kumar and Michael, 2012] and is also the crucial issue to solve in flight control.

3.3 Attitude Control

We first present the attitude control part in the cascaded control flow as shown in Fig. 8. In order to compensate the unstructured but fixed uncertainties $\boldsymbol{\Delta}_{\text{rot}}$, the integrated control term is required. Although the general PID control can be applied, the optimal control with integral control, called LQI [Young and Willems, 1972], is more suitable for our model, since it is possible to design the cost function in the optimal control framework.

We first rewrite the rotational dynamics Eq. 27 as follows:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} + D(\boldsymbol{\Delta}_{\text{rot}} - I^{-1}\boldsymbol{\omega} \times I\boldsymbol{\omega}) \tag{28}$$

$$\boldsymbol{y} = C\boldsymbol{x} \tag{29}$$

$$\boldsymbol{x} \in R^6, \quad \boldsymbol{u} \in R^4, \quad \boldsymbol{y} \in R^3$$

where

$$\boldsymbol{x} = \begin{bmatrix} {}^{\{W\}}\alpha_{\{CoG\}_x} & {}^{\{W\}}\dot{\alpha}_{\{CoG\}_x} & {}^{\{W\}}\alpha_{\{CoG\}_y} & {}^{\{W\}}\dot{\alpha}_{\{CoG\}_y} & {}^{\{W\}}\alpha_{\{CoG\}_z} & {}^{\{W\}}\dot{\alpha}_{\{CoG\}_z} \end{bmatrix}^{\text{T}}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} ; B = \begin{bmatrix} \boldsymbol{0} \\ Q_{\text{rot}_x}(\boldsymbol{q}) \\ \boldsymbol{0} \\ Q_{\text{rot}_y}(\boldsymbol{q}) \\ \boldsymbol{0} \\ Q_{\text{rot}_z}(\boldsymbol{q}) \end{bmatrix}^{\text{T}} ; C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} ; D = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note that, $Q_{\text{rot}}(\boldsymbol{q}) = \begin{bmatrix} Q_{\text{rot}_x}(\boldsymbol{q})^{\text{T}} & Q_{\text{rot}_y}(\boldsymbol{q})^{\text{T}} & Q_{\text{rot}_z}(\boldsymbol{q})^{\text{T}} \end{bmatrix}^{\text{T}}$.

We then introduce a tracking error $\boldsymbol{e}$ between the desired value and the system output, along with its integral value:

$$\dot{\boldsymbol{e}}_{I_\alpha} = \boldsymbol{y}^{\text{des}} - \boldsymbol{y} = C\boldsymbol{x}^{\text{des}} - C\boldsymbol{x} = C\boldsymbol{e}_x \tag{30}$$

Using Eq. 30, Eq. 28 and Eq. 29 can be extended as follows:

$$\dot{\bar{\boldsymbol{x}}} = \bar{A}\bar{\boldsymbol{x}} + \bar{B}\boldsymbol{u} + \bar{D}(\boldsymbol{\Delta}_{\text{rot}} - I^{-1}\boldsymbol{\omega} \times I\boldsymbol{\omega}) \tag{31}$$

$$\bar{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{e}_x \\ \boldsymbol{e}_{I_\alpha} \end{bmatrix}; \quad \bar{A} = \begin{bmatrix} A & 0_{8\times3} \\ C & 0_{3\times3} \end{bmatrix}; \quad \bar{B} = \begin{bmatrix} -B \\ 0_{3\times4} \end{bmatrix}; \quad \bar{D} = \begin{bmatrix} -D \\ 0_{3\times3} \end{bmatrix}$$

In the cost function of a general optimal control, there are two basic terms: the term regarding the state to change the convergence performance and the term regarding the control input to restrict the input magnitude. However, in this work, the suppression of the force generated by the thrust force, especially the horizontal force (i.e., Eq. 12) should be taken into account in the attitude control. Then an original cost function for this model is designed as follows:

$$J = \int_0^\infty \left(\bar{\boldsymbol{x}}^{\text{T}} M \bar{\boldsymbol{x}} + \tilde{\boldsymbol{u}}^{\text{T}} N \tilde{\boldsymbol{u}} \right) dt \tag{32}$$

$$N = W_1 + Q_{\text{tran}}^{\text{T}}(\boldsymbol{q}) W_2 Q_{\text{tran}}(\boldsymbol{q}) \tag{33}$$

where, the second term in Eq. 33 corresponds to the minimization of the norm of the force generate by the attitude control, since $\|{}^{\{CoG\}}\boldsymbol{f}\|^2 = {}^{\{CoG\}}\boldsymbol{f}^{\text{T}}{}^{\{CoG\}}\boldsymbol{f} = u^{\text{T}} Q_{\text{tran}}(\boldsymbol{q})^{\text{T}} Q_{\text{tran}}(\boldsymbol{q}) u$. The diagonal weight matrices M, W_1 and W_2 balance the performance of the convergence to the desired state, the suppression of the control input and the suppression of the translational force generated by the attitude control.

Then, a constant feed-back gain matrix K_x can be obtained by solving the related algebraic Riccati equation (ARE) driven by the equation Eq. 31 and the cost function Eq. 32. Finally, the desired control input regarding the attitude control $\boldsymbol{u}_{\text{att}}^{\text{des}}$ can be given by:

$$\boldsymbol{u}_{\text{att}}^{\text{des}} = K_x \bar{\boldsymbol{x}} + Q_{\text{rot}}^{\#}(\boldsymbol{q}) I^{-1}\boldsymbol{\omega} \times I\boldsymbol{\omega} \tag{34}$$

3.4 Position Control

The position control follows the method presented in [Lee et al., 2010, Goodarzi et al., 2013], which first calculates the desired total force based on the common PID control, then converts the desired total force to the desired thrust vector and the deisred roll and pitch angles.

The desired total force can be given by:

$$
\begin{aligned}
\boldsymbol{f}^{\text{des}} &= m_\Sigma(K_P\boldsymbol{e}_r + K_I\int(\dot{\boldsymbol{e}}_r + c\boldsymbol{e}_r)dt + K_D\dot{\boldsymbol{e}}_r + \ddot{\boldsymbol{r}}^{\text{des}}) - {}^{\{W\}}R_{\{CoG\}}Q_{\text{tran}}(\boldsymbol{q})Q^{\#}_{\text{rot}}(\boldsymbol{q})I^{-1}\boldsymbol{\omega}\times I\boldsymbol{\omega} \\
&= m_\Sigma(K_P\boldsymbol{e}_r + K_I\boldsymbol{e}_{I_r} + K_D\dot{\boldsymbol{e}}_r + \ddot{\boldsymbol{r}}^{\text{des}}) + \boldsymbol{\phi}
\end{aligned}
\tag{35}
$$

where $\boldsymbol{e}_r = \boldsymbol{r}^{\text{des}} - \boldsymbol{r}$, and c is a positive constant. Note that the offset term $\boldsymbol{\phi}$ is added to compensate the force caused by the attitude control (i.e., the second term of Eq. 34).

Then the desired roll and pitch angles can be also calculated from $\boldsymbol{f}^{\text{des}}$:

$$
\alpha^{\text{des}}_{\{CoG\}_x} = atan^{-1}(-\bar{f}_y, \sqrt{\bar{f}_x^2 + \bar{f}_z^2}) \tag{36}
$$

$$
\alpha^{\text{des}}_{\{CoG\}_y} = atan^{-1}(\bar{f}_x, \bar{f}_z) \tag{37}
$$

$$
\bar{\boldsymbol{f}} = R_z^{-1}(\alpha_z^{\text{des}})\boldsymbol{f}^{\text{des}}
$$

where, $R_Z(\cdot)$ is a special rotation matrix which only rotates along z axis. Note that these two values are subsequently transmitted to the attitude controller as shown in Fig. 8.

On the other hand, the desired collective thrust force can be calculated as follows:

$$
f_T^{\text{des}} = ({}^{\{W\}}R_{\{CoG\}}\boldsymbol{b}_3)^{\text{T}}\boldsymbol{f}^{\text{des}}, \tag{38}
$$

where, $\boldsymbol{b}_3$ is a unit vector $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\text{T}}$.

Using (38), the allocation from the collective thrust force to the thrust force vector can be performed as follows:

$$
\boldsymbol{u}^{\text{des}}_{\text{pos}} = \frac{\boldsymbol{u}_s}{m_\Sigma g}f_T^{\text{des}}, \tag{39}
$$

where, $\boldsymbol{u}_s$ is the thrust force vector under the zero equilibrium as expressed by Eq. 9, which only balances with the gravity force and thus does not affect rotational motion. Eventually, the final deisred thrust force for each rotor should be the sum of output from bthe attitude controller and the position control: $\boldsymbol{u}^{\text{des}} = \boldsymbol{u}^{\text{des}}_{\text{att}} + \boldsymbol{u}^{\text{des}}_{\text{pos}}$.

3.5 Stability Analysis

For a general multirotor, the exponential stability of the cascaded control method is well-studied by [Lee et al., 2010, Goodarzi et al., 2013]. However, the full influence $Q_{\text{tran}}(\boldsymbol{q})\boldsymbol{u}$ on all translational axes as shown in Eq. 25 makes the problem difficult. Thus, it is required to clarify the new stable condition regarding the feed-back control gains in Eq. 32 and Eq. 35.

3.5.1 Attitude Stability

Is it well-known that the LQI control framework guarantees the exponential stability. In order to show the stability of the complete system, we design a proper lyapunov function candidate for attitude control based on the error dynamics Eq. 31 as follows:

$$
\mathcal{V}_1 = \tfrac{1}{2}\bar{e}_{\text{rot}}^2 \tag{40}
$$

$$
\boldsymbol{e}_{\text{rot}} = \bar{\boldsymbol{x}} - \begin{bmatrix} \boldsymbol{0}_6 & \boldsymbol{\Delta}_{\text{rot}}\circ\frac{1}{Q_{\text{rot}}(\boldsymbol{q})K_{x_I}} \end{bmatrix}^{\text{T}}.
$$

where $(\cdot) \circ (\cdot)$ denote the element-wise multiplication of two vectors, and $K_{x_I} \in \mathcal{R}^{4\times3}$ corresponds to the right three columns of the gain matrix K_x. Thus $\boldsymbol{\Delta}_{\text{rot}} \circ \frac{1}{Q_{\text{rot}}^{\#}(\boldsymbol{q})K_{x_I}}$ is the convergent integral value of $\boldsymbol{e}_{I_\alpha}$.

Then, the time derivative of $\mathcal{V}_1$ is given by:

$$\begin{aligned}
\dot{\mathcal{V}}_1 &= \boldsymbol{e}_{\text{rot}}^{\text{T}}\dot{\boldsymbol{e}}_{\text{rot}} = \boldsymbol{e}_{\text{rot}}^{\text{T}}\dot{\bar{\boldsymbol{x}}} = \boldsymbol{e}_{\text{rot}}^{\text{T}}(\bar{A}\bar{\boldsymbol{x}} + \bar{B}\boldsymbol{u} + \bar{D}(\boldsymbol{\Delta}_{\text{rot}} - I^{-1}\boldsymbol{\omega} \times I\boldsymbol{\omega})) \\
&= \boldsymbol{e}_{\text{rot}}^{\text{T}}((\bar{A} + \bar{B}K_x)\bar{\boldsymbol{x}} + \bar{D}\boldsymbol{\Delta}_{\text{rot}})) = \boldsymbol{e}_{\text{rot}}^{\text{T}}(\bar{A} + \bar{B}K_x)\boldsymbol{e}_{\text{rot}} < 0.
\end{aligned} \tag{41}$$

$\dot{\mathcal{V}}_1$ should be always negative since all eigenvalues of $(\bar{A} + \bar{B}K)$ are negative, and thus the attitude control guarantees the exponential stability. However, the derivation of Eq. 41 is only valid in the near-hover state (i.e., $\boldsymbol{\omega} \approx \dot{\boldsymbol{\alpha}}$), otherwise the state equation Eq. 28 would not be established.

3.5.2 Complete Stability

The dynamics of position error can be given by:

$$m\ddot{\boldsymbol{e}}_r = -mK_P\boldsymbol{e}_r - mK_I\boldsymbol{e}_{I_r} - mK_D\dot{\boldsymbol{e}}_r + bmX + RQ'_{\text{tran}}\boldsymbol{e}_{\text{rot}} + \boldsymbol{\Delta}_{\text{tran}} + \boldsymbol{\Delta}'_{\text{rot}}, \tag{42}$$

where $\boldsymbol{\Delta}'_{\text{rot}} = Q_{\text{tran}}Q_{\text{rot}}^{\#}\boldsymbol{\Delta}_{\text{rot}}$, $Q'_{\text{tran}} = Q_{\text{tran}}K_x$ and $\boldsymbol{X} = \|\boldsymbol{f}^{\text{des}}\|(((R^{\text{des}}\boldsymbol{b}_3)^{\text{T}}R\boldsymbol{b}_3)R\boldsymbol{b}_3 - R^{\text{des}}\boldsymbol{b}_3)$. For convenience $\{^W\}R_{\{CoG\}}$ and m_Σ are simplified as R and m. We refer the reader to Appendix A, where the we provide the detailed derivation of Eq. 42.

Then the integral Lyapunov candidate $\mathcal{V} = \mathcal{V}_1 + \mathcal{V}_2$ for the complete system is written as follows:

$$\mathcal{V} = \mathcal{V}_1 + \mathcal{V}_2 \tag{43}$$

$$\mathcal{V}_2 = \tfrac{1}{2}\boldsymbol{e}_r^{\text{T}}K_P\boldsymbol{e}_r + \tfrac{1}{2}\|\dot{\boldsymbol{e}}_r\|^2 + c\boldsymbol{e}_r^{\text{T}}\dot{\boldsymbol{e}}_r + \tfrac{1}{2}(\boldsymbol{e}_{I_r} - \boldsymbol{\Delta} \circ \tfrac{1}{\boldsymbol{k}_{I_r}})^{\text{T}}K_{I_r}(\boldsymbol{e}_{I_r} - \boldsymbol{\Delta} \circ \tfrac{1}{\boldsymbol{k}_{I_r}}) \tag{44}$$

where, $\mathcal{V}_2$ is the Lyapunov candidate regarding the position error dynamics, and the gain matrix K_P and the positive constant c corresponds to Eq. 35. Also note that, $\boldsymbol{k}_{I_r} \in \mathcal{R}^3$ is the diagonal elements of K_{I_r}: $K_{I_r} = diag(\boldsymbol{k}_{I_r})$, and $\boldsymbol{\Delta} = \frac{\boldsymbol{\Delta}_{\text{tran}} + \boldsymbol{\Delta}'_{\text{rot}}}{m}$.

In order to guarantee $\mathcal{V} > 0$ and $\dot{\mathcal{V}} < 0$, following constraints should be satisfied

$$k_{P_{\min}} > \gamma k_{P_{\max}} \tag{45}$$

$$c < \min\{\tfrac{4(k_{P_{\min}} - \gamma k_{P_{\max}})(k_{D_{\min}} - \gamma k_{D_{\max}})}{k_{D_{\max}}^2(1+\gamma)^2 + 4(k_{P_{\min}} - \gamma k_{P_{\max}})}, k_{D_{\min}} - \gamma k_{D_{\max}}, \sqrt{k_{P_{\min}}}\} \tag{46}$$

$$\lambda_{\min}(W_1)\lambda_{\min}(W_2) > \tfrac{\sigma_{\max}^2(W_{12})}{4} \tag{47}$$

where,

$$W_1 = \frac{1}{2}\begin{bmatrix} c(k_{P_{\min}} - \gamma k_{P_{\max}}) & -\frac{ck_{D_{\max}}}{2}(1+\gamma) \\ -\frac{ck_{D_{\max}}}{2}(1+\gamma) & -c + k_{D_{\min}} - \gamma k_{D_{\max}} \end{bmatrix}, W_{12} = \begin{bmatrix} \frac{c(\sigma_{max}(Q'_{\text{tran}}) + O)}{m} \\ \frac{\sigma_{max}(Q'_{\text{tran}}) + B}{m} + k_{P_{\max}}\boldsymbol{e}_{r_{\max}} \end{bmatrix}, W_2 = -(\bar{A} + \bar{B}K_x)$$

Note that, $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ are the maximum and minimum eigenvalue of a matrix, while $\sigma_{max}(\cdot)$ denotes the maximum singular value of a matrix. $k_{P_{\max}}$, $k_{P_{\min}}$, $k_{D_{\max}}$ and $k_{D_{\min}}$ are the maximum and minimum elements in the gain matrix K_P and K_D, respectively. γ is a positive constant, i.e., $\gamma \leq sin(\|\boldsymbol{e}_\alpha\|) < 1$. while O is also a positive constant which satisfies Eq. 58. $\boldsymbol{e}_{r_{\max}}$ is the upper bound of the postion error. We refer the reader to Appendix B, where the we provide the detailed derivation of Eq. 45 $\sim$ Eq. 47.

According to Eq. 47, if we increase the propeller tilt angle, $\sigma_{\max}^2(W_{12})$ would increase because of the positive correlation with the sine of tilt angle. Then, it is necessary to increase either the attitude or position gains and further increase the left side of Eq. 47. However, the increase of the position gains would make the change of desired attitude more aggressive according to Eq. 36 and Eq. 37, and thus the violation of the near-hover assumption would be easier to occur. Therefore, changing the attitude gains should be more effective.

4   State Estimation

In this work, the state estimation for the multilinked aerial robot to achieve fully autonomous flight in outdoor environment is also developed. The crucial issue regarding the multilinked model is the necessity to transform the sensor value from each sensor frame to a common frame for the sensor fusion and then transform again to the frame $\{CoG\}$ for control. These transformation processes involve the joint angles q as shown in Fig. 9. On the other hand, the time synchronization is also considered in the extended Kalman filter to solve the delay of sensor measurement.
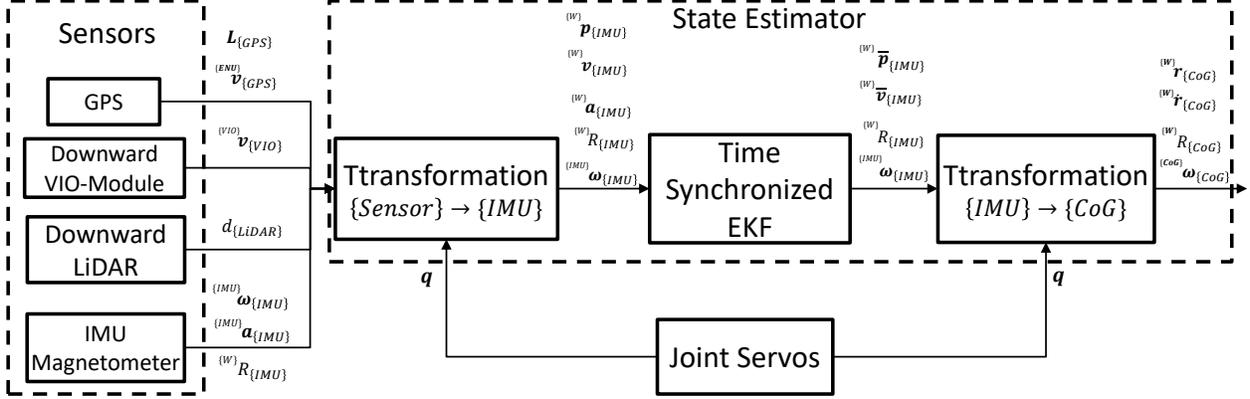
Figure 9: The framework of the state estimation for fully autonomous flight. Sensors provide sensor values w.r.t. each sensor frame. Then the transformation from each sensor frame to the frame $\{IMU\}$ is performed to unify these sensor values according to the kinematics involving joint angles q, which is followed by a time-synchronized extended Kalman filter to generate the fused odometry. Note that the symbols with bar are the estimated output of EKF. Finally the odometry is transformed to the frame $\{CoG\}$ for flight control.

4.1   Core Sensors

4.1.1   GPS

Global positioning system (GPS) mainly provides the latitude and longitude $L_{\{GPS\}} = [lat\ lon]^{\mathrm{T}}$ based on WGS84 [1], which can be straightforwardly converted into a metric position around a reference point with an east-north-up (ENU) coordinate. In other words, GPS can provide the global horizontal position, namely ${}^{\{W\}}p_{\{GPS\}xy}$ w.r.t a world frame $\{W\}$ of which the x and y axis coincide with the north and east direction, respectively. Such definition also benefits the integration with magnetometer. Furthermore, the latest GPS module can also provide the global velocity w.r.t ENU coordinate which is calculated according to relative motion between the GPS module and each satellite.

On the other hand, there are couple of problems with GPS module. First, the accuracy highly depends on the satellite number and the weather condition, which however can be solved by fuseing multiple sensors. Second, the delay of measurement is most significant compared with other sensors, which can reach 0.5 s. Given that such a delay would worsen the performance of the sensor fusion, the time-synchronized EKF framework is developed in this work.

4.1.2   Downward VIO-Module

A VIO Module can provide a 6DoF odometry from either a monocular or a stereo camera combined with IMU, which is call visual-inertial-odometry (VIO). In an outdoor environment, the visual processing from a

---

[1] WGS84: WORLD GEODETIC SYSTEM 1984

horizontal has the problem of the insufficient feature points. Then the downward view is applied. However, the density of valid feature points is still lower than an indoor environment. Thus the position estimated by a VIO module is omitted since it highly depends on the feature map, and only the relative velocity $^{\{VIO\}}\boldsymbol{v}_{\{VIO\}}$ is used in our framework.

4.1.3 Downward LiDAR

A downward Light Detection and Ranging (LiDAR) sensor measures one dimension distance $d_{\{LiDAR\}}$ from the sensor to the ground. In most cases, the outdoor flight is performed upon a flat ground and the robot orientation would not change significantly from the hovering state. Thus the height $^{\{W\}}p_{\{LiDAR\}_z}$ from the ground can be given by

$$^{\{W\}}p_{\{LiDAR\}_z} = [0\ 0\ 1]\,^{\{W\}}R_{\{LiDAR\}}\left[d_{\{LiDAR\}}\ 0\ 0\right]^{\mathrm{T}} \tag{48}$$

where, $^{\{W\}}R_{\{LiDAR\}}$ denotes the orientation of the LiDAR sensor. Note that the x axis of the frame $\{LiDAR\}$ coincides with the direction of light emission.

4.1.4 IMU and Magnetometer

The rotational motion can be estimated by the accelerometer and gyroscope in IMU with magnetometer. Among various effective estimation algorithms (e.g. [Madgwick et al., 2011, Marins et al., 2001]), the most computationally light algorithm, namely the complementary filter [Martin and Salaün, 2010], is applied in this work to calculate the orientation in a microcontroller. The orientation estimated by IMU and magnetometer is also based on the ENU coordinate. On the other hand, the angular velocity and the linear acceleration measured by IMU are also used in subsequent frame transformation and the extended Kalman filter as shown in Fig. 9. It is also notable that, the VIO algorithm can also provide the estimated orientation; however, IMU and Magnetometer are the most robust and reliable sensors in fields, and thus a separated orientation estimation only by these two sensors is designed as shown in Fig. 9.

4.2 Transformation from Sensor Frame to IMU Frame

The extended Kalman filter requires all sensor values to be described at a common frame. Then the frame $\{IMU\}$ is chosen as the common frame to avoid the necessity of the linear and angular accleration in the transformation process. The position and velocity provided by each sensor is converted with following rules:

$$^{\{W\}}\boldsymbol{p}_{\{IMU\}} = {}^{\{W\}}\boldsymbol{p}_{\{S\}} - {}^{\{W\}}R_{\{IMU\}}{}^{\{IMU\}}\boldsymbol{p}_{\{IMU\}\to\{S\}}(\boldsymbol{q}) \tag{49}$$

$$^{\{W\}}\boldsymbol{v}_{\{IMU\}} = {}^{\{W\}}R_{\{IMU\}}({}^{\{IMU\}}R_{\{S\}}(q){}^{\{S\}}\boldsymbol{v}_{\{S\}} - {}^{\{IMU\}}\boldsymbol{\omega}_{\{IMU\}} \times {}^{\{IMU\}}\boldsymbol{p}_{\{IMU\}\to\{S\}}(\boldsymbol{q})) \tag{50}$$

where, $\{S\}$ denotes the sensor frame, and $^{\{W\}}\boldsymbol{p}_{\{IMU\}\to\{S\}}(\boldsymbol{q})$ is the position vector from $\{IMU\}$ to $\{S\}$ which involves the joint angles $\boldsymbol{q}$. Note that, Eq. 49 is used for the conversion about the GPS and Downward LiDAR, while Eq. 50 is used for the conversion of the VIO module.

4.3 Time Synchronized Extended Kalman Filter

The estimation state $\boldsymbol{x}$ in EKF holds the position $^{\{W\}}\boldsymbol{p}_{\{IMU\}}$ and velocity $^{\{W\}}\boldsymbol{v}_{\{IMU\}}$ of the frame $\{IMU\}$ and the bias of the acceleration $\boldsymbol{b}_{\mathrm{acc}}$: $\boldsymbol{x} := \begin{bmatrix} ^{\{W\}}\boldsymbol{p}_{\{IMU\}}^{\mathrm{T}} & ^{\{W\}}\boldsymbol{v}_{\{IMU\}}^{\mathrm{T}} & \boldsymbol{b}_{\mathrm{acc}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. The input for prediction is the accleration obtained from IMU sensor $^{\{IMU\}}\boldsymbol{a}_{\{IMU\}}$, while the accleration bias $\boldsymbol{b}_{\mathrm{acc}}$ is modeled as random walk with their derivatives being white gaussian noise. Then the prediction model can be expressed as a simple dynamics only involving position, velocity and acceleration. On the other hand, the measurement

vector $\boldsymbol{z} := \begin{bmatrix} {}^{\{W\}}\boldsymbol{p}^{\mathrm{T}}_{\{IMU\}} & {}^{\{W\}}\boldsymbol{v}^{\mathrm{T}}_{\{IMU\}} \end{bmatrix}^{\mathrm{T}}$ contains the position and velocity which are obtained from sensors other than IMU, implying the observation model is a simple linear matrix.

An important issue in sensor fusion is time synchronization among sensors, since ignoring the measurement delay would significantly decrease in estimation performance. Therefore, a time-synchronized Kalman Filter framework is developed based on [Lynen et al., 2013] as shown in Fig. 10. The key of this framework is the FIFO structure which enables the correction in the past node. Given that the sensor value from IMU has the smallest delay, the arrival of new IMU state serves as the trigger to perform prediction. On the other hand, the correction process is performed on-demand when a new sensor value from other sensors is arrived as shown in Fig. 10. There two two cases of correction: (A) shows the case when a new sensor value with the latest timestamp $sensor1_k$ arrives which requires correction, whereas (B) shows the case when a further delayed sensor value $sensor2_k$ arrives which requires both re-prediction and re-correction to the latest sensor timestamp. In both of cases, re-prediction from the latest sensor timestamp to the latest imu timestamp is also required, and the covariance is not predicted during this phase since it is not necessary for the control. Thus, the successive prediction of covariance is performed on-demand during the correction phase to reduce the computational cost.
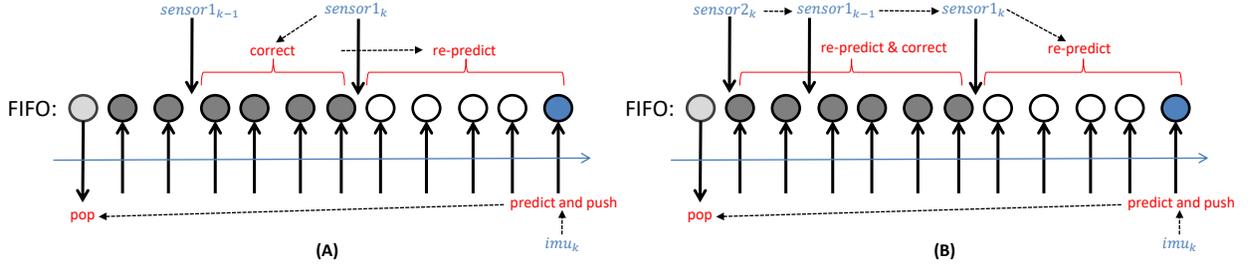
Figure 10: The time-synchronized extended Kalman filter framework which contains a FIFO buffer sorted by timestamp. The push motion is performed when a new imu value arrives, and the pop motion is performed in the same time to keep the buffer length fixed. There are two cases of correction with delayed sensor value. **(A)**: the case when the sensor value with latest sensor timestamp $sensor1_k$ arrives. **(B)**: the case when the further delayed sensor value which is older than the latest sensor timestamp $sensor2_k$ arrives. The dashed arrows indicate the process sequence when either a new imu value or a sensor value arrives.

4.4 Transformation from IMU Frame to CoG Frame

The estimated odometry regarding the frame $\{IMU\}$ should be finally converted to the frame $\{CoG\}$ with following rule

$$ {}^{\{W\}}R_{\{CoG\}} = {}^{\{W\}}R_{\{IMU\}}{}^{\{IMU\}}R_{\{CoG\}}(\boldsymbol{q}) \tag{51} $$

$$ {}^{\{CoG\}}\boldsymbol{\omega}_{\{CoG\}} = {}^{\{CoG\}}R_{\{IMU\}}(\boldsymbol{q}){}^{\{IMU\}}\boldsymbol{\omega}_{\{IMU\}} \tag{52} $$

$$ {}^{\{W\}}\boldsymbol{p}_{\{CoG\}} = {}^{\{W\}}\bar{\boldsymbol{p}}_{\{IMU\}} + {}^{\{W\}}R_{\{IMU\}}{}^{\{IMU\}}\boldsymbol{p}_{\{IMU\}\to\{CoG\}}(\boldsymbol{q}) \tag{53} $$

$$ {}^{\{W\}}\boldsymbol{v}_{\{CoG\}} = {}^{\{W\}}\bar{\boldsymbol{v}}_{\{IMU\}} + {}^{\{W\}}R_{\{IMU\}}({}^{\{IMU\}}\boldsymbol{\omega}_{\{IMU\}} \times {}^{\{IMU\}}\boldsymbol{p}_{\{IMU\}\to\{S\}}(\boldsymbol{q})) \tag{54} $$

where, ${}^{\{W\}}\bar{\boldsymbol{p}}_{\{IMU\}}$ and ${}^{\{W\}}\bar{\boldsymbol{v}}_{\{IMU\}}$ are the estimated result from the time-synchronized extend EKF, and ${}^{\{IMU\}}R_{\{CoG\}}(\boldsymbol{q})$ is obtained from the forward-kinematics and Eq. 21. Finally the converted odometry regarding the frame $\{CoG\}$ is used as the feed-back state in the control system as shown in Fig. 8.

5 Platform

5.1 Mechanical Specification

The hardware decisions are led by the need to create the synergy among different tasks, which leads to an original hardware platform as shown in Fig. 11. The link rod of which the length is 0.6 m is make from the carbon square pipe, and cables pass inside the rod. Other white and red components are made from PLA and Aluminum respectively, which are used in different places according to the desired strength.

Two joints are actuated by servo motors (Dynamixel XH430-W350R [2]) of which the maximum torque are 4.2 Nm at 14.8 V. However, the maximum torque might be not sufficient for grasping task since large friction is required during delivery. Thus an enhanced joint design is developed in this work which applies pulleys (white disk-like parts in Fig. 11(C)) to further amplify the torque. The pulley ratio is 1:2 which enables double torque output. It is notable that, such design enables to employ different servos and different pulleys to achieve different joint torque characteristics. Such a configurable design is the main improvement compared with the servo-embedded structure as developed in our previous work [Zhao et al., 2018].

The propulsion system is built based on T-motor products (rotors: T-motor MN3510 KV360[3]; ESC: T-motor Air40A[4]; propeller: T-motor P14×4.8 Prop[5]) , and the maximum thrust generated by this propulsion system is 16 N. As shown in Fig. 11(D), the propeller and rotor are mounted at a PLA component of which the top surface is inclined at an angle of 10 deg for tilting. On the other hand, battery can be mounted below the propulsion system where there is a power cable connecting to the battery. Generally, a 6 s (22.2 V) Turnigy 1300 mAh battery is connected to each link module. Then the total weight of this basic platform with batteries is 3.4 Kg, which results in 15 min flight. However, it is possible to change the arrangement of batteries along with the number and capacity to different task, since all batteries are wired in parallel. In addition, a propeller protect duct with an aluminum honeycomb structure is employed not only for safety, but also for acting as gripper tip in grasping task.

5.2 System Architecture

5.2.1 Processors

The onboard computer as shown in Fig. 11(B) is UP Board[6] with Intel Atom CPU running Ubuntu and the robot operating system (ROS)[7] as middleware to execute the state estimation, flight control and task-specific motion planning as shown in Fig. 12. Other onboard computers with higher computational performance (e.g. Intel NUC[8]) can be alternatively used for the task requiring the vision or point cloud processing.

On the other hand, the original printed circuit board (PCB) called *Spinal* as shown in Fig. 11(E) is a micro controller unit (MCU) with a STM32F7 core to fulfill the real-time processing such as the attitude estimation and controller. As shown in Fig. 12, a IMU and Magnetometer unit (InvenSense MPU9250[9]) are embedded in *Spinal* to achieve the zero delay data transmission for attitude estimation to control. The message exchange between onboard computer and *Spinal* is achieved by the UART and rosserial based protocol[10].

Another type of original MCU called *Neuron* with STM32F4 core is designed to directly connect to actuators

---

[2]See http://www.robotis.us/dynamixel-xh430-w350-r.
[3]See https://store-en.tmotor.com/goods.php?id=337.
[4]See https://store-en.tmotor.com/goods.php?id=368.
[5]See https://store-en.tmotor.com/goods.php?id=380.
[6]https://www.aaeon.com/jp/p/up-board-computer-board-for-professional-makers.
[7]http://www.ros.org/.
[8]https://www.intel.com/content/www/us/en/products/boardskits/nuc.html.
[9]https://invensense.tdk.com/products/motion-tracking/9-axis/mpu-9250/
[10]http://wiki.ros.org/ja/rosserial

Figure 11: **(A)**: robot platform composed from three parts which are connected by two joints. For convenience, we regard the central part as two separated links, namely link2 and link3. The link length is 0.6m. **(B)**: the top of the connection part between link2 and link3. **(C)**: the top of the joint part which is actuated by a servo motor. **(D)**: the tilting propeller in each link. **(E)**: the bottom of the connection part between link2 and link3. The original micro controller units called *spinal* and *neuron*s are connected by Controller Area Network (CAN) for the internal communication between link modules.

(e.g., rotor ESC, joint servo) as shown in Fig. 12. An internal communication system based on Controller Area Network (CAN) [ISO, 1993] which is developed in our previous work [Zhao et al., 2018] connects *Spinal* and *Neuron*s, and an original message protocol inside CAN is designed to pass ROS messages from the computer to each actuators as shown in Fig. 12.

5.2.2 Sensors

The core sensors for localization are an embedded IMU & magnetometer, a GPS (u-blox M8 module[11]), a downward LiDAR (LedderOne[12]), and a downward-facing light VIO module (RealSense T265[13]) as shown in Fig. 11. Regarding the VIO module, RealSense T265 contains internal processor to calculate the visual odometry from an internal IMU and stereo fisheye cameras, which can significantly save the external computational resource. Furthermore, This device is relatively light (~ 80g) which is suitable for the aerial application. On the other hand, RTK-GPS is not applied in our platform since it can not afford the fully autonomous flight in wide area. Nevertheless, the proposed sensor employment can still promise the sufficient localization accuracy which will be shown in our outdoor experiments in Sec. 6.

6 Experiments

We evaluated our robot platform by experiments in both indoor and outdoor environment. The video of our evaluation can be found at https://youtu.be/LkDGP82sg1I. The control gain parameters for the real robot platform are summarized as follows:

---

[11]https://www.u-blox.com/en/product/neo-m8-series
[12]https://leddartech.com/lidar/leddarone/
[13]https://www.intelrealsense.com/tracking-camera-t265/
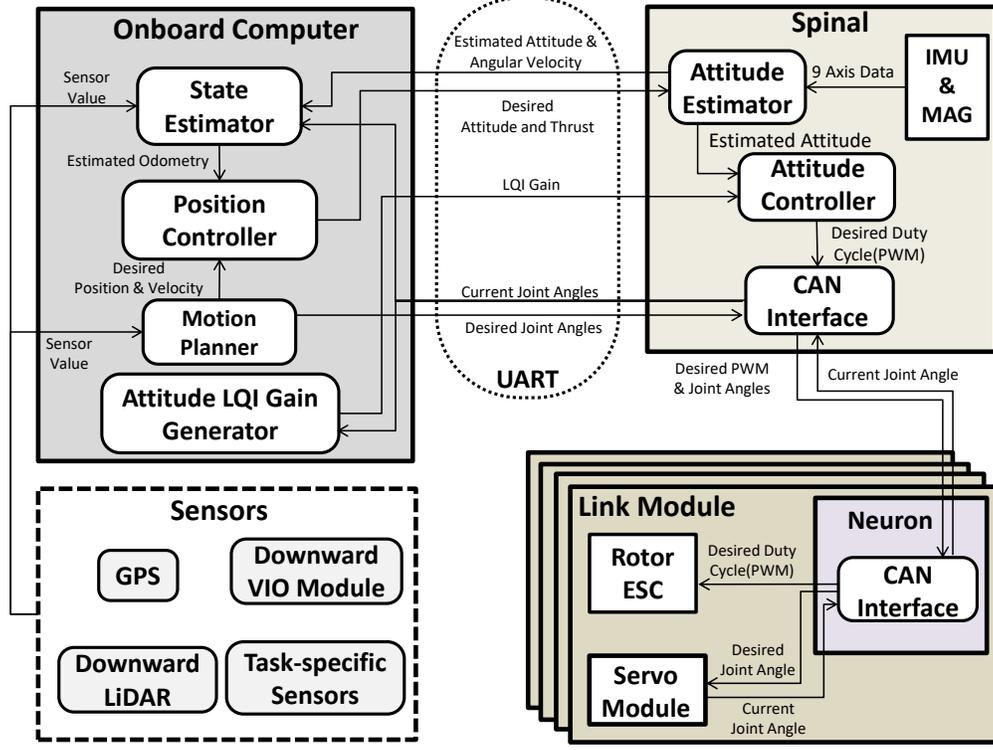
Figure 12: The system architecture to achieve fully autonomous flight.

Table 1: Control Gain Parameters for the Real Platform

Equation	Parameter	Value
Eq. 32	M	$diag(1100, 80, 1100, 80, 100, 50, 10, 10, 0.5)$
Eq. 33	W_1	$diag(1, 1, 1, 1)$
	W_2	$diag(100, 100, 100)$
Eq. 35	K_P	$diag(2.3, 2.3, 3.6)$
	K_I	$diag(0.02, 0.02, 3.4)$
	K_D	$diag(4.0, 4.0, 1.55)$

6.1 Indoor Experiments

In order to evaluate the feasibility of proposed control method, several indoor flight experiments were conducted to testify the stability during deformation, the compensation ability regarding the unknown model error, and the robustness against the strong disturbance. A motion capture system was employed instead of using the state estimator in all indoor experiments for the ground truth regarding robot odometry.

6.1.1 Deformation Stability

The flight stability during deformation was evaluated as shown in Fig. 13. During deformation, the form with $q_1 = -\frac{\pi}{4}$ rad, $q_2 = \frac{\pi}{2}$ rad (Fig. 13②) corresponds to the smallest $\tau_{\min}$ as shown in Fig. 7(D), while the form with $q_1 = q_2 = \frac{\pi}{4}$ rad (Fig. 13③) serves as the initial form to grasp an object. The overall tracking errors are relatively small as shown in Fig. 13(B); however, the deviation of the yaw motion (i.e., α_z) can be confirmed under the form of ② and ③. We consider this is due to the model error caused by the slight bending and torsion of the multilinked structure. Nevertheless, the deviation was slowly reduced by the

integral control in our proposed attitude control, and the stability regarding other axes was guaranteed regardless of the deformation. These results demonstrate the effectiveness of the proposed control method for the multilinked aerial robot with the tilting propeller. Furthermore, the robustness against the sudden impact was also evaluated as shown in Fig. 13④. The large external wrench caused by human kicks induced the temporal divergence in horizontal and yaw motion (41 s and 43 s in Fig. 13(B)); however, the hovering was recovered quickly, which showed the sufficient robustness of this platform.
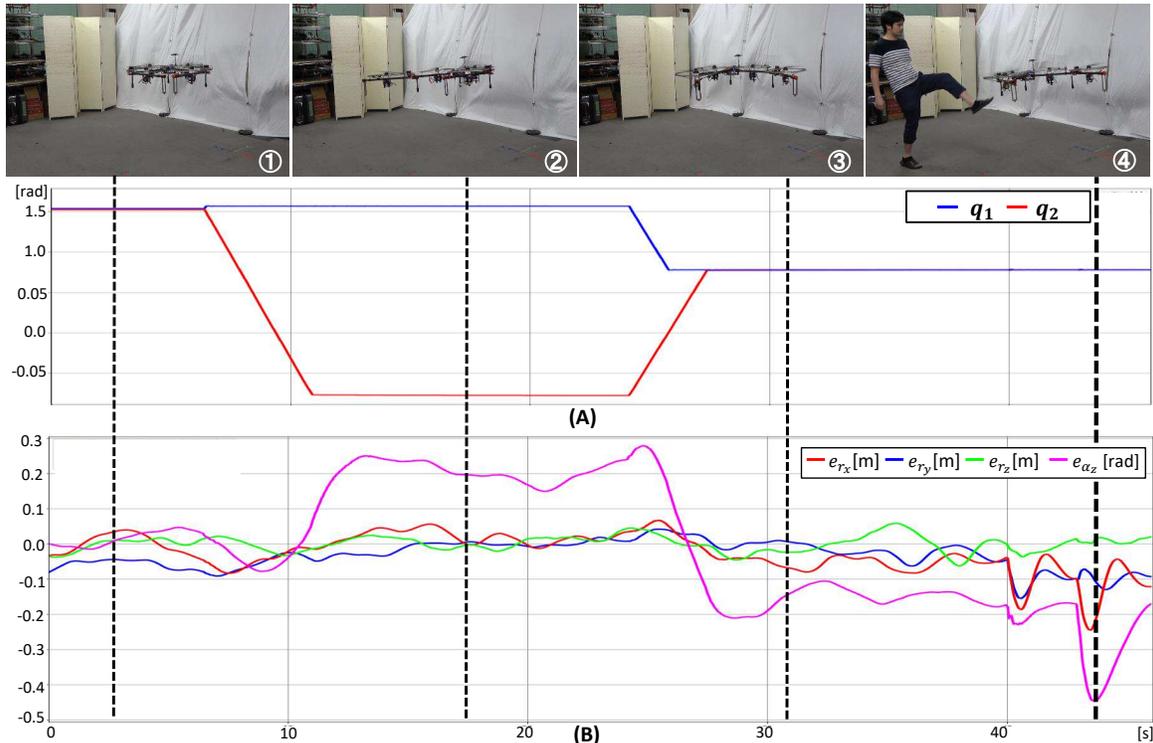
Figure 13: The evaluation of stable deformation (② and ③) in an indoor environment using a motion capture system, along with the robustness against large impact ④. (**A**): the change in joint angles during this experiment. (**B**): the tracking errors regarding the translational and yaw rotational motion.

6.1.2   Grasping and Releasing Object

Our multilinked robot can be regarded as an entire gripper to grasp an object using two end links. In order to increase the contact area, an extended gripper tip attached on the propeller duct was designed as shown in Fig. 21(A). Then an hovering experiment with a grasped object was conducted as shown in Fig. 14. The weight of grasped object is 1.0Kg, which was not contained in the dynamics model Eq. 25 and Eq. 27 for control system. Therefore, the temporal position divergence because of the deviation of the CoG frame occurred after takeoff as shown in Fig. 14①, which was recovered by the integral control as shown in Fig. 14②. Furthermore, a sudden ascending and deviation in x axis also occurred right after releasing the object as shown Fig. 14③. However, the robot was back to the desired point smoothly as shown Fig. 14④. Therefore, these result demonstrates the compensation ability of our control method regarding the unknown and large model error.

6.1.3   Opening Sheet by Deformation

The proposed multilinked structure can also be regarded as a manipulator to expand sheet, and one of the application using this motion is to cover a fire spot from the air which will be presented later in detail.
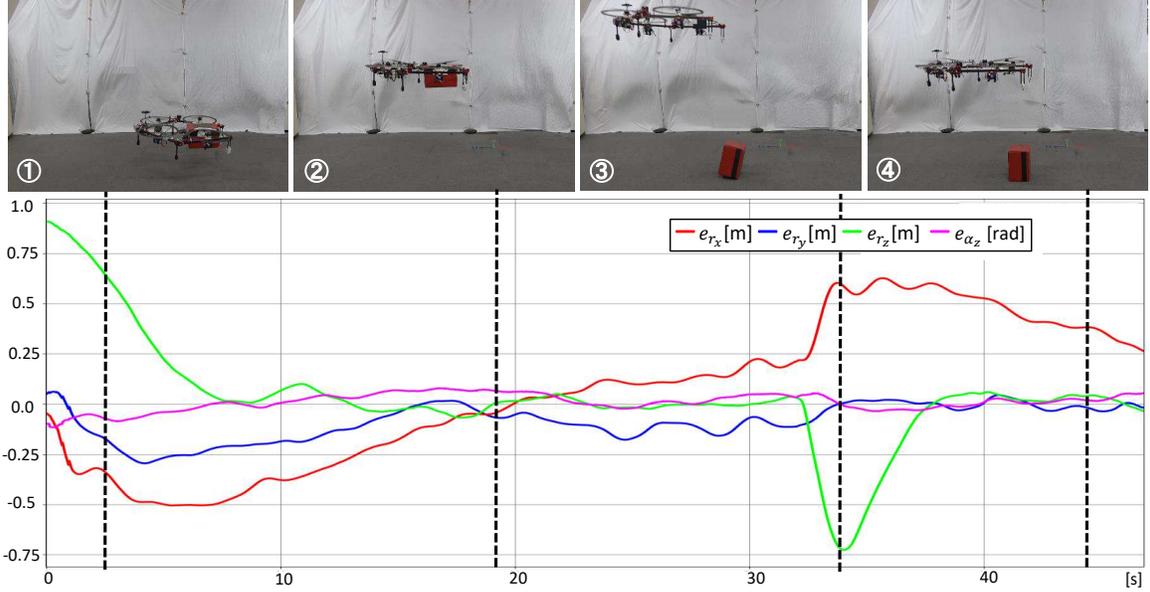
Figure 14: Grasping and dropping an object with unknown weight (the actual weight is 1.0 Kg) from takeoff phase. The tracking errors confirm the stable hovering, demonstrating the compensation ability of our control method regarding the unknown and large model error.

However, the aerodynamics disturbance caused by the airflow acting on the expanded sheet surface can significantly prevent the hovering stability. In the case of the robot model without the tilting propeller, such disturbance can easily induce the yaw control divergence because of the low controllability. Then, the improvement of stability achieved by the robot model with proposed tilting propeller was evaluated as shown in Fig. 15. The sheet of which the weight is 0.68 Kg and width is 1 m was folded in the beginning as shown in Fig. 15①, and then fully expanded as shown in Fig. 15②~⑤. The tracking errors as shown in Fig. 15(B) demonstrate the hovering stability when sheet was fully expanded. In addition to the aerodynamics disturbance, the CoG of the sheet would suddenly change at the expanding moment. Nevertheless, the stable flight during ②~④ was guaranteed, which also indicates the robustness against a large disturbance.

6.2 Outdoor Experiments

In the outdoor experiments, we first performed the fundamental hovering test with the onboards sensor to confirm the feasibility of the state estimation, which was followed by a circle trajectory tracking experiment to show the feasibility of the control method for a relatively fast and wide-range motion. Then, three task-specific experiments were conducted. Given than the main focus of this paper is the evaluation on the effectiveness of the proposed robot platform in fully autonomous flight involving aerial deformation, the specific target/environment detection method and motion planning algorithm for each task will be not presented in this paper; however the task-specific sensors and the platform customization for each task will be introduced in each experiment.

6.2.1 Hovering and Circle Trajectory Tracking

The autonomous hovering at a height of 3 m was first conducted as shown in Fig. 16(A). The relatively small tracking errors as shown in Fig. 16(B) demonstrate the stable flight around a desired point. Although there was no ground truth provided in this experiment to evaluate the accuracy of the proposed state estimation, the convergence of flight control can indirectly confirm the performance of state estimation, since the uncertainty of state estimation would induce the divergence of flight control. However, a relatively
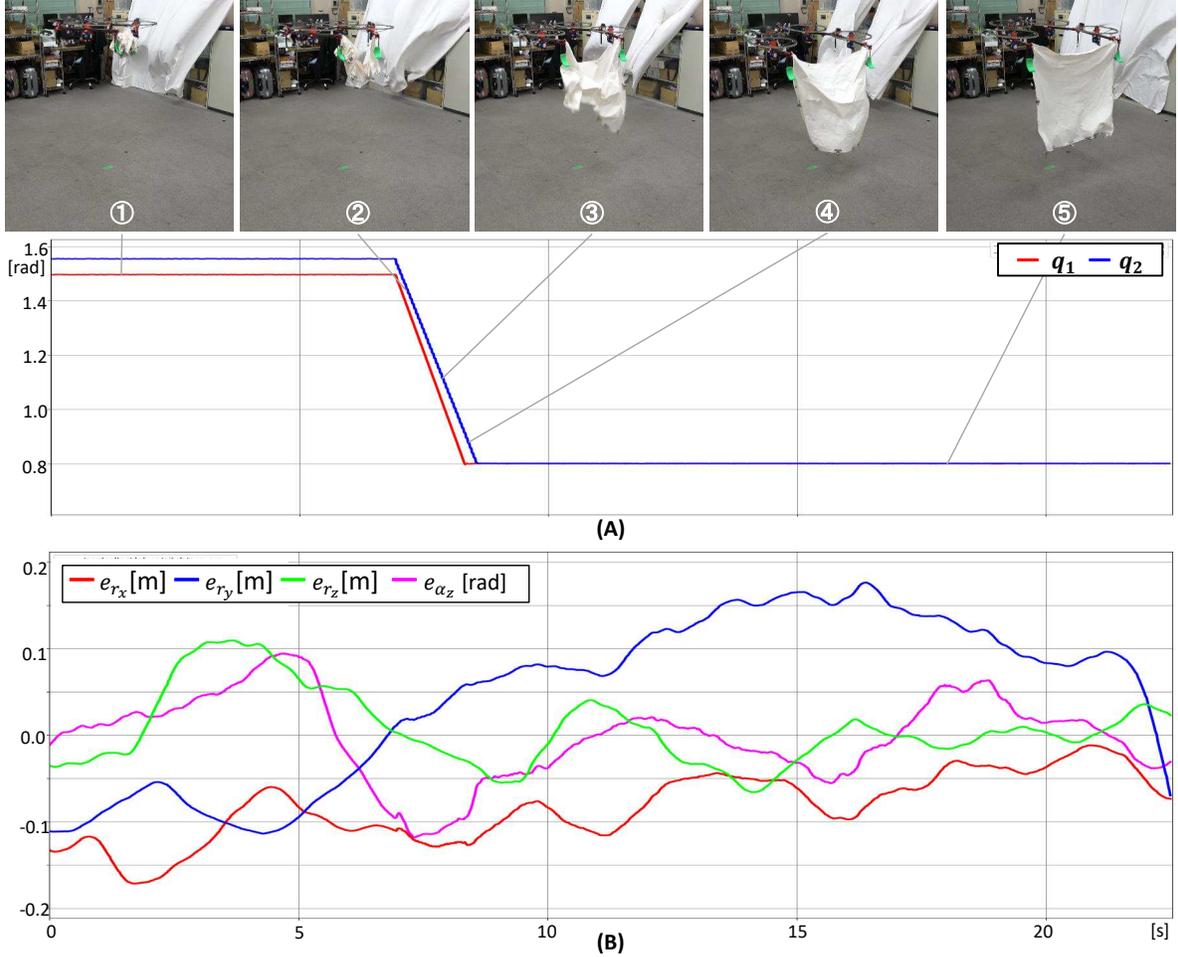
Figure 15: The stable sheet expanding motion by deformation (① ∼ ⑤). **(A)**: the change in joint angles during this experiment. **(B)**: the tracking errors regarding the translational and yaw rotational motion. The video of comparison with a robot model without the tilting propeller can be found at https://youtu.be/LkDGP82sg1I.

constant offset in x and y axes might always exist, since the global position in x and y axes for the state estimation framework is only available from GPS sensor ($^{\{W\}}\boldsymbol{p}_{\{GPS\}_{xy}}$ in Fig. 9) which generally contains a certain offset in positioning. Nevertheless, most of the outdoor application involves visual servoing which can guarantee the expected tracking performance towards a desired position.

Then, the feasibility of the control system to track a relatively wide and aggressive trajectory was evaluated. The desired trajectory is a circle of which the radius is 8 m and the height is 4 m as shown in Fig. 17, and the desired tracking velocity was gradually increased from 0.5 m/s to 3.0 m/s over 3 laps. As shown in Fig. 18(A), the maximum horizontal error reached 1m at a desired tracking velocity of 3.0m. Such a large deviation is due to the insufficient proportional control in Eq. 35 when the motion becomes aggressive. However, increasing K_P in Eq. 35 might induce unexpected vibration in hovering flight. Thus, in order to guarantee the tracking performance, the maximum horizontal velocity is limited to 2.0m/s in most of tasks. In comparison with the position tracking, the velocity tracking performance demonstrated a better result as shown in Fig. 18(D), implying the potential to perform an aggressive maneuvering. On the other hand, the expected tracking performance on z and yaw motion around the fixed desired values (i.e., $r_z^{\mathrm{des}} = 4$ m, $\alpha_z^{\mathrm{des}} = 0$ rad) can be confirmed from Fig. 18(B) and (C).
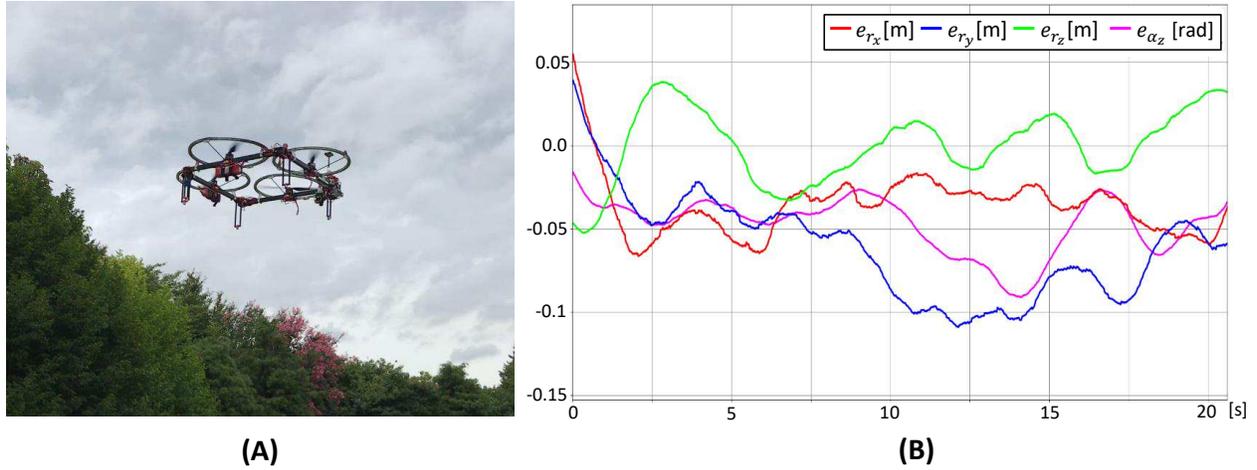
Figure 16: **(A)** the hovering test at a height of 3 m in outdoor environment using the onboard sensors and developed state estimation method. **(B)** the tracking errors regarding the translational and yaw rotational motion.
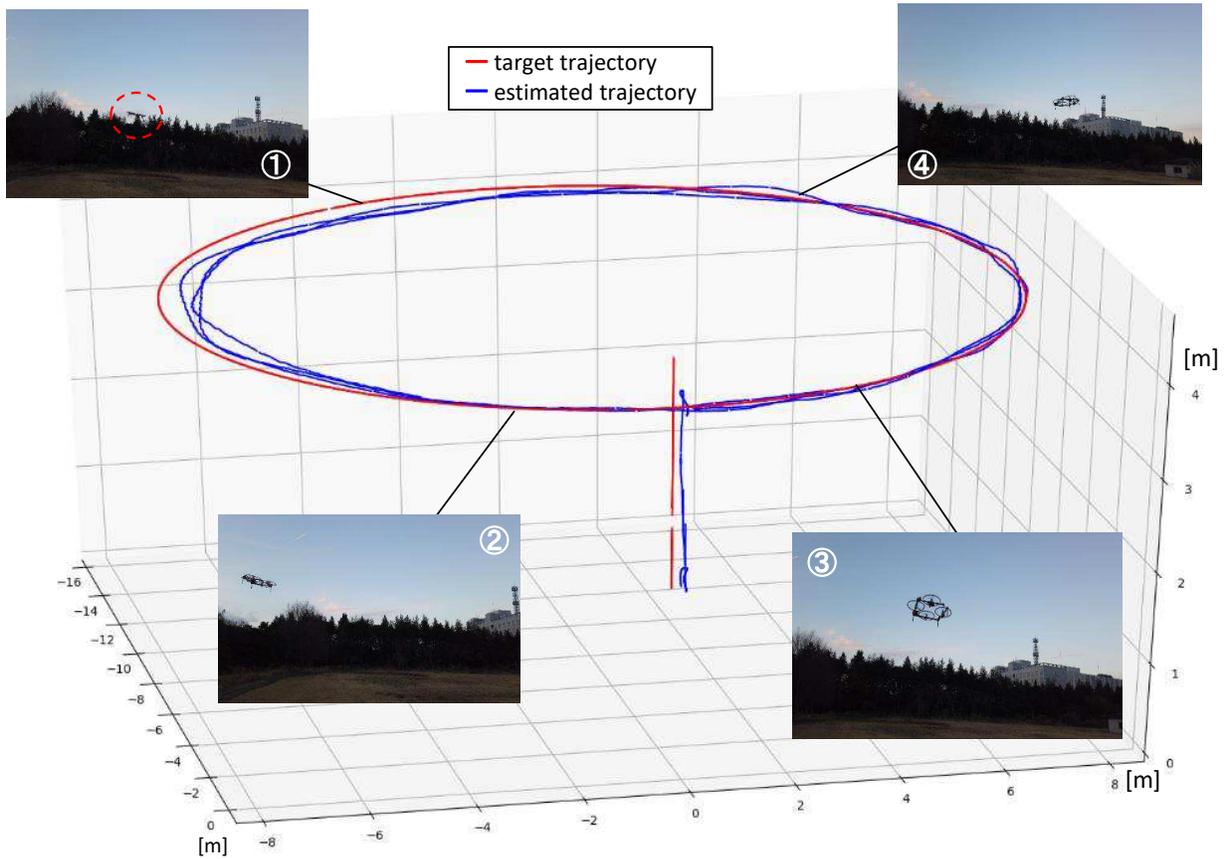
Figure 17: The tracking performance in outdoor environment which tracked a circle trajectory of which the radius is 8m and the height is 4m. The tracking desired velocity was gradually increased from 0.5 m/s to 3.0 m/s over three laps.
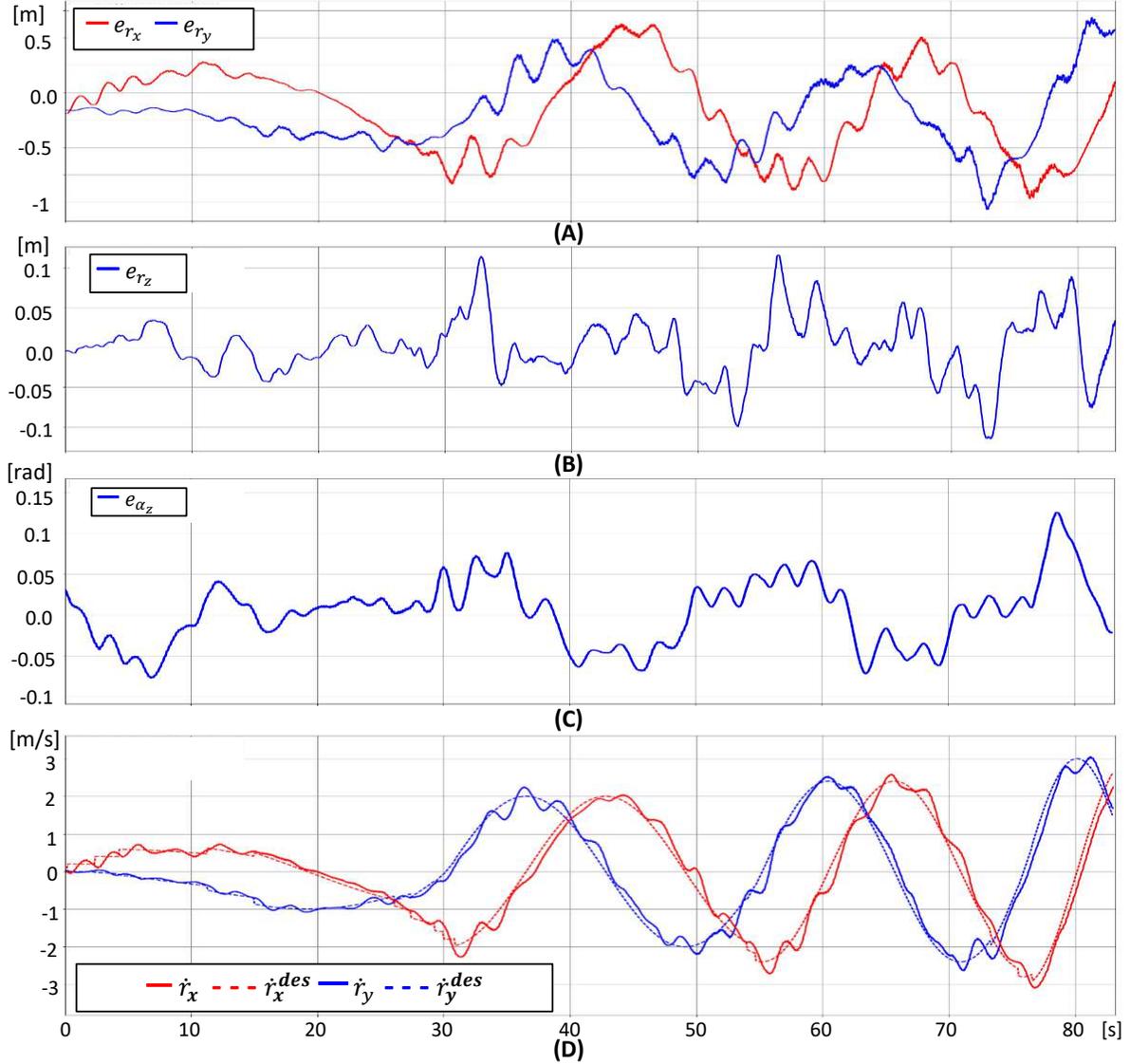
Figure 18: **(A)** $\sim$ **(C)**: the tracking errors (i.e., $e_{r_x}, e_{r_y}, e_{r_z}, e_{\alpha_z}$) regarding experiment as shown in Fig. 17. **(D)**: the velocity tracking performance. The desired velocity was gradually increased from 0.5 m/s to 3.0 m/s.

6.2.2 Interception with a Fast Flying Target

In order to evaluate the performance of our developed platform on an aggressive task, an experiment to intercept a fast flying target was conducted. Regarding the task-specific sensor, a front-facing monocular camera (ELP-SUSB1080P01-LC1100[14]) was equipped to detect the moving target as shown in Fig. 19(A), and a edge computing device (Google Coral[15]) is connected to the onboard computer (Intel NUC7I7DNHE[16]) to perform SSD detection [Liu et al., 2016] as shown in Fig. 19(B). On the other hand, a front net between two ends was also equipped to enable dropping or catching target.

In this experiment, the target is a yellow ball hung from a quadrotor. As shown Fig. 20, the moving trajectory

---

[14]http://www.webcamerausb.com/

[15]https://coral.ai/products/accelerator

[16]https://www.intel.com/content/www/us/en/products/boards-kits/nuc/kits/nuc7i7dnhe.html
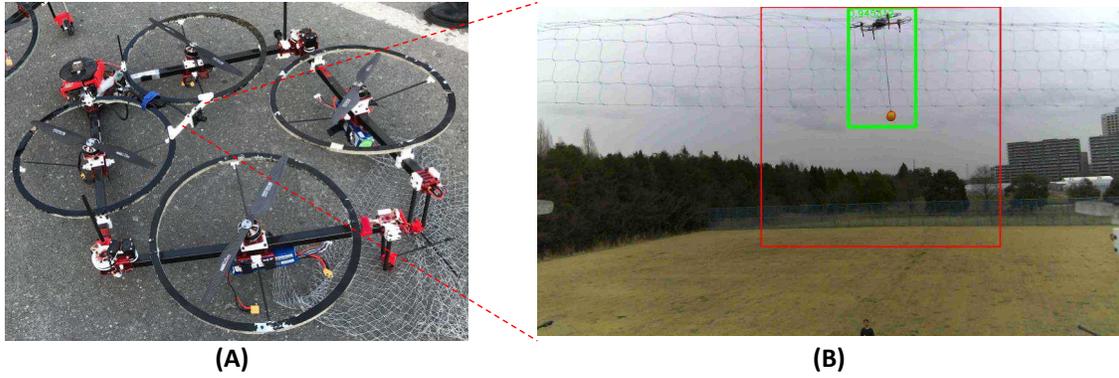
Figure 19: **(A)** the front-facing camera for the moving target detection and the net between two ends to perform dropping and catching. **(B)** the image detection on the moving target (a moving quadrotor with a hung yellow ball) by using the edge computing device Google Coral.
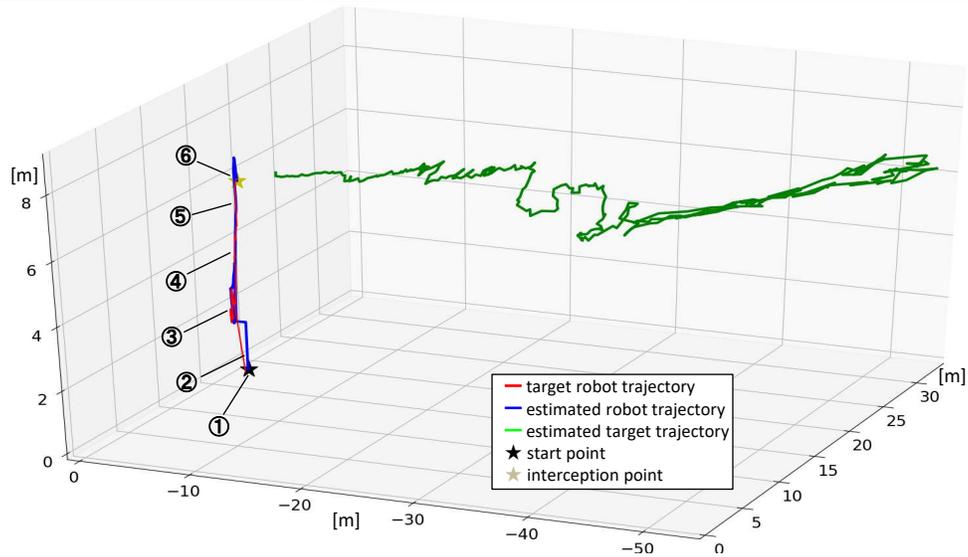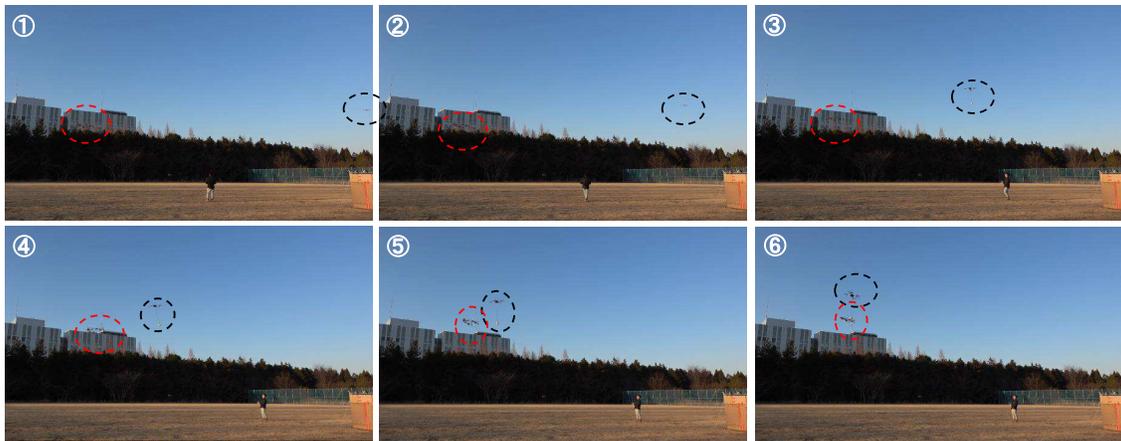
Figure 20: The trajectory of our robot along with the estimated trajectory of the target in the autonomous interception task as shown in ①~⑥. Our robot increased the height from 3 m to 8 m, and succeeded to intercept the target. The duration from starting ascending to interception was 6 s.

of the this quadrotor is straight line when viewed from above, and the flight height changes between 6 m to 10 m. The moving speed is 5 m/s. On the other hand, our robot was waiting at a height of 3 m at the beginning. Once the target was detected with a certain duration, the desired interception point could be predicted based on the estimated target trajectory. Subsequently, the desired interception motion for the robot was planned and further fulfilled. As shown in Fig. 20③~⑥ and the plotted trajectory, the robot was ascending quickly to reach the same height with the yellow ball, and an adjustment in horizontal motion was also performed simultaneously. Finally our robot succeeded to hit the yellow ball by the net and drop it from the moving quadrotor, which took 6 s from starting ascending to the interception. The accurate of the target detection and position projection was relatively low leading to the unreliable estimated trajectory when the target was far from the robot (the green trajectory in Fig. 20). However, the estimation of the target height was relatively reliable, which enables early ascending and thus promises the good visual servoing when the target becomes closer. The success of interception as shown in Fig. 20 confirmed the feasibility of the proposed control system and state estimation to perform an aggressive task involving fast ascending and quick horizontal motion.

6.2.3 Searching, Grasping and Delivering a Brick

One of the novel characteristics of our platform is the manipulation ability. Then a task to grasp object from the ground and subsequently deliver to a designated location was performed. The grasping strategy was the same as shown in Fig. 14, which used two ends to contact with the object surfaces. A RBGD sensor (RealSense D435i[17]) was equipped as shown in Fig. 21(B) to detect the ground object from both color image and point cloud as shown in Fig. 21(C) and (D). The onboard processor to process the image and point cloud is LattePanda Alpha 864s[18].
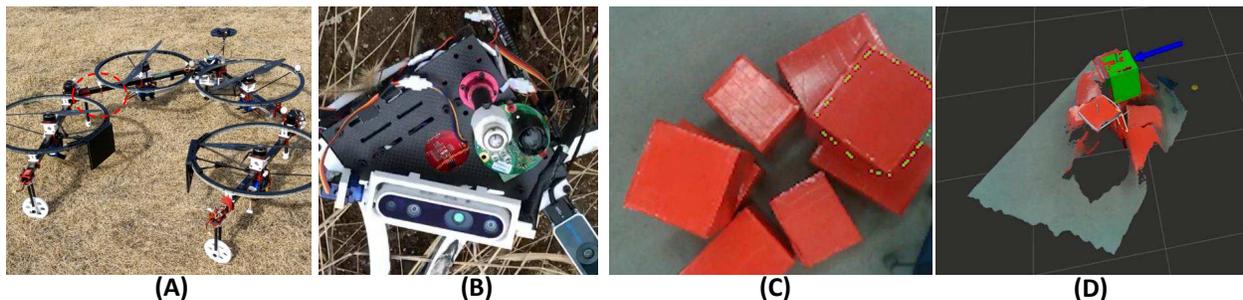
Figure 21: **(A)**: the customized platform which can grasp object by using two black tips attached on the propeller duct. **(B)**: the downward RGBD sensor (RealSense D435i) for the ground object detection. **(C)**: the downward image from the sensor. **(D)**: the downward point cloud from the sensor, where the detected object is highlighted by green color.

In this autonomous task, the robot was first required to find the ground red bricks with a weight of 1.0 Kg by moving to several designated waypoints as shown in Fig. 22①~ ②. Once the bricks were detected and target brick was selected, the robot fully landed to grasp the target brick as shown in Fig. 22③. Subsequently, the robot took off again to deliver the brick to the wall as shown in Fig. 22④. A relatively large tracking error can be confirmed from the deviation between the blue and red trajectories as shown in Fig. 22 during this delivery phase. This is because the control system was switched to a velocity control mode (i.e., $K_P = 0$ in Eq. 35), and thus the position error was allows in this phase. Nevertheless, the robot converged to a desired position once the control system was switched back to the potions control model as shown in Fig. 22⑤. Finally, the robot placed the brick on the top of the wall with a height of 2 m (Fig. 22⑥) by performing the channel detection using the downward RGBD sensor. A heuristic solution to switch off the downward LiDAR while flying upon the wall was applied to solve the height gap problem in height estimation. The success of the whole task showed the feasibility of the developed platform to autonomously gasp, deliver

---

[17]https://www.intelrealsense.com/depth-camera-d435i/
[18]https://www.lattepanda.com/products/lattepanda-alpha-864s.html
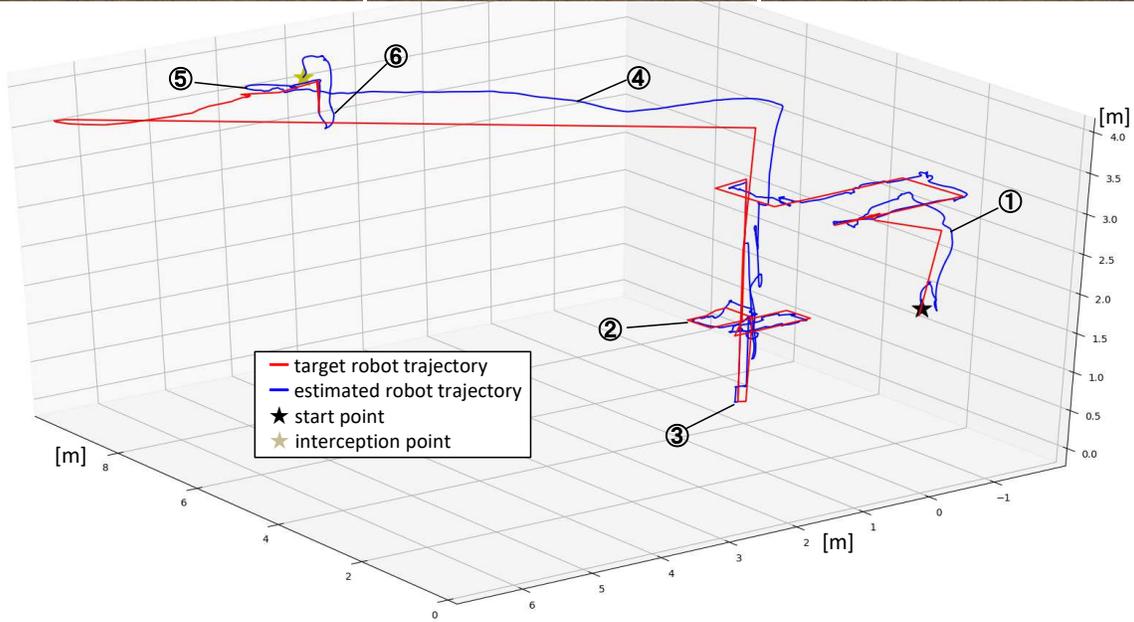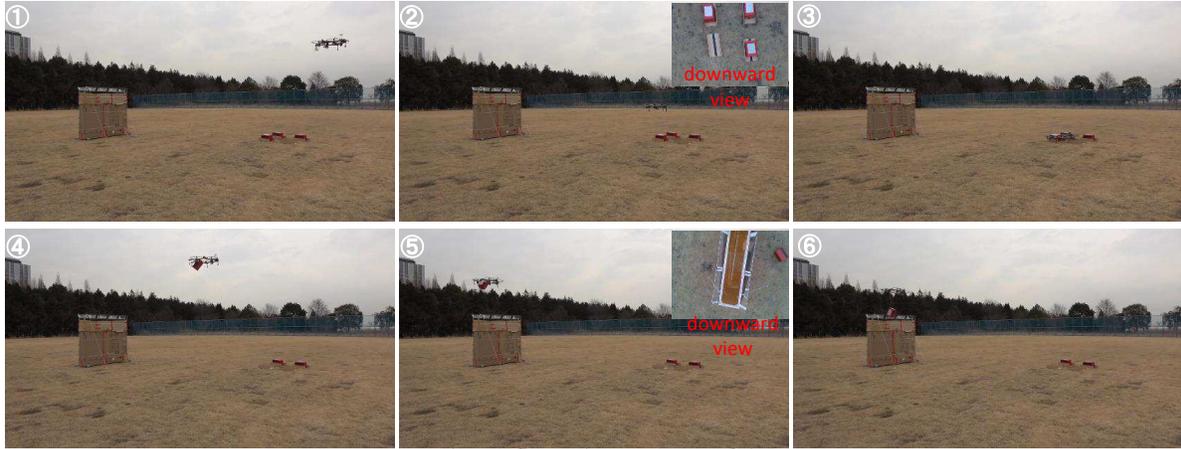
Figure 22: The trajectory of our robot in the autonomous task involving brick searching, grasping, delivering and placing as shown in ①~⑥. Our robot increased the height from 3 m to 8 m, and succeeded to intercept the target. The duration from starting ascending to interception was 6 s. The robot succeeded to grasp and place a brick with a weight 0f 1.0 Kg to a wall with a height of 2 m.

object, along with the effectiveness of state estimation and the flight control in a task involving landing and takeoff on the way.

6.2.4 Firefighting by Using Blanket

In comparison to water, using a blanket does not require accurate shooting to distinguish fire, and our platform can expand a blanket by openning joints as shown in Fig. 15. A hotplate was prepared to serve as a fire spot as shown in Fig. 23(A) and it is assumed that this virtual fire can be extinguished by the covering motion as shown in Fig. 23(B). To detect the heat from the hotplate, a downward thermal sensor (FLIR Radiometric Lepton Dev Kit[19]) was equipped as shown in Fig. 23(C) and (D). In this task, the image processing cost is relatively low, thus the default onboard computer, UP Board[6] was selected.

Similar to the object grasping and deliver task as shown in Fig. 22, the firefighting task also started with an

[19]https://www.sparkfun.com/products/15948

autonomous search by patrolling between several designated waypoints as shown in Fig. 24①~ ④. Again, a relatively large tracking error can be confirmed from the blue and red trajectories in Fig. 25 during this patrol phase, which is due to the same reason as explained in the second task. Once the heat source was found by the thermal sensor as shown in the sub-images inside Fig. 24⑤, the expanding motion followed by a rapid descending was performed as shown in Fig. 24⑥~ ⑨. The success of the whole task demonstrated the feasibility of the developed platform to autonomous patrol in a relatively wide area (i.e. 12 m × 20 m as shown in Fig. 25). Moreover, the proposed state estimation by fusing multiple sensors guaranteed the estimation quality and also the flight stability even after the blanket was expanded and thus large occlusion area occurred which decreased the accuracy of the output from VIO module.
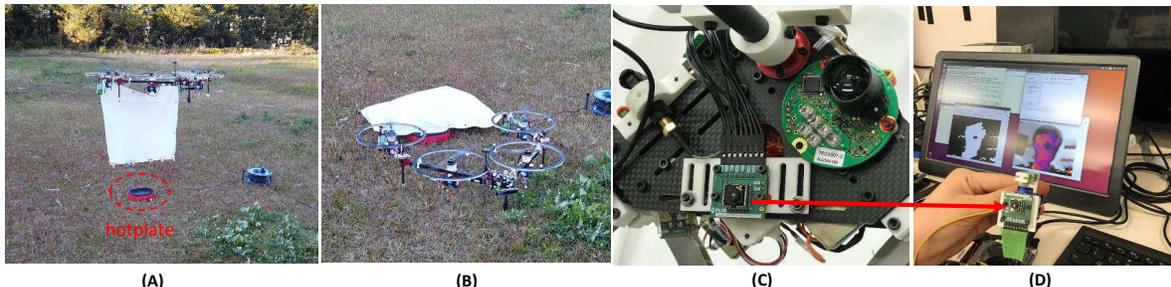
Figure 23: **(A) - (B)**: firefighting with a blanket expanded by our robot. The hotplate is regarded as a virtual fire spot. **(C) - (D)** the downward thermal sensor (FLIR Radiometric Lepton Dev Kit) for the ground heat source detection.

6.3 Result in MBZIRC 2020

At the competition, we participated in all challenges with multilinked aerial robots in fully autonomous mode as shown in Fig. 1(C)~(E), and the platforms were customized as shown in Fig. 19, Fig. 21 and Fig. 23 for each challenge, respectively.

The performance in the competition was not as ideal or as good as our experiments because we did not have enough time to debug and fine-tune our system by the time of the competition. But nonetheless, we were still able to rank third place in Challenge 1 (one of the three teams to succeed to intercept the moving target) and sixth place in Challenge 3. Regarding Challenge 2, we succeeded to grasp, deliver and place a brick to the goal in rehearsal, which would deserve a high score in the real challenge. Those results sufficiently demonstrated not only the feasibility of the proposed methods for desgin, control and state estimation to achieve the fully autonomous maneuvering, but also the versatility of our proposed deformable aerial robot in different tasks.

7 Conclusion and Lessons Learned

In this paper, we presented a multilinked aerial robot platform comprised from two joints and four tilting rotor. The design, modeling and control method has been developed to achieve the flight stability, while the state estimation based on the multilinked kinematics has been also developed to enable the fully autonomous flight in the fields. Various on-site experiments, including the fast maneuvering for target interception, the aerial grasping for delivery and the blanket manipulation for firefighting, has been evaluated to demonstrate the feasibility of the proposed platform.

This robot platform has been also evaluated in MBZIRC 2020 which is a highly demanding and successful event for us. An on-site testbed similar to the real competition also helped us to properly improve our platform from daily experiments. Furthermore, unification of the platform foundation in all challenges and suppression of task-specific customization are the strategic advantages in our development, which leads to the synergy between different tasks such as the efficiency of the robot operation and maintenance. In addition
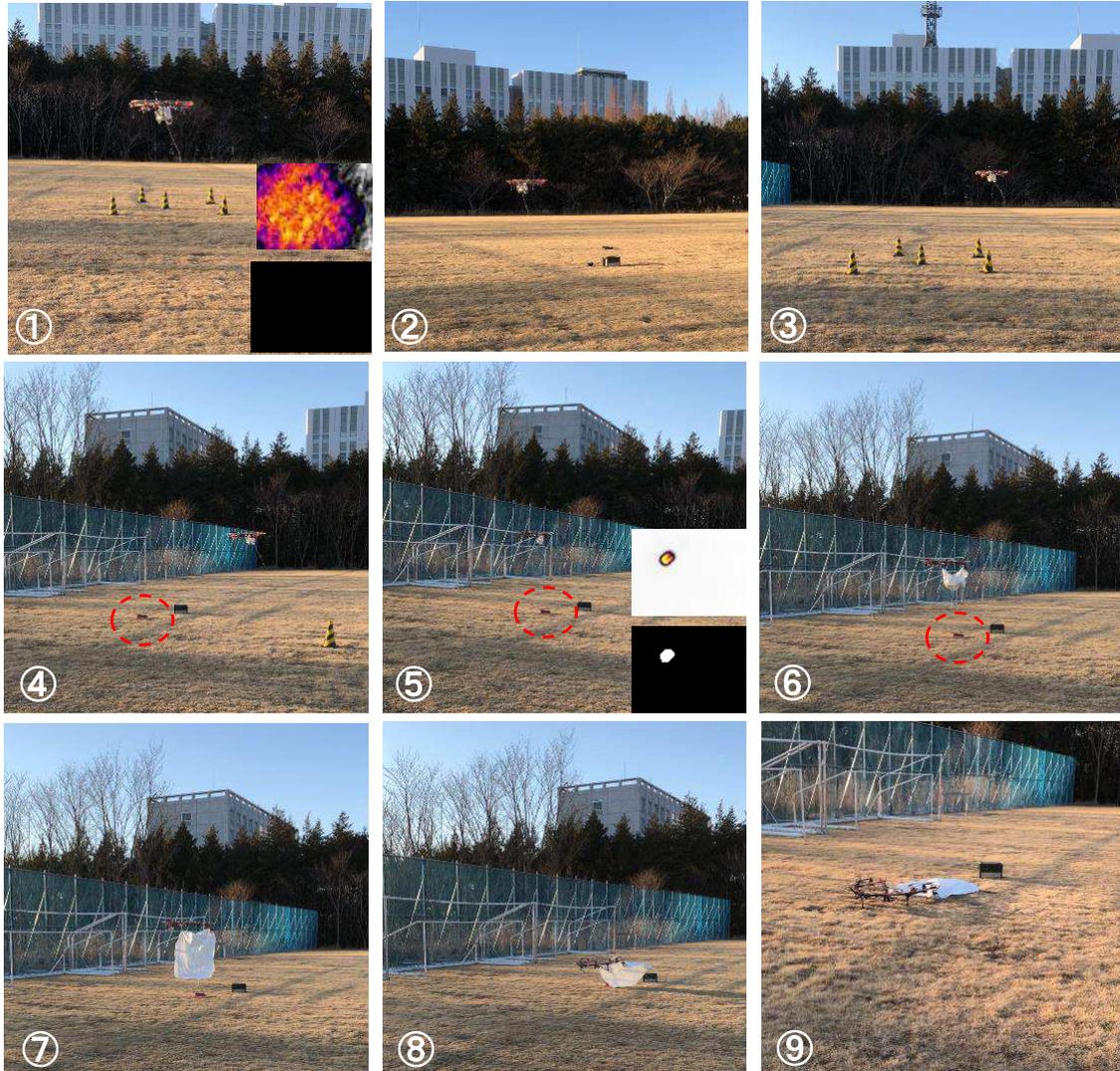
Figure 24: The autonomous task to find a heat source and perform firefighting motion by covering a blanket. The sub images in ① and ⑤ correspond to the colored thermal images and the thresholded images by temperature. The hotplate surrounded in a dashed circle was found at ⑤, and then the covering motion was performed ⑥ ~ ⑨.

to the versatility, our robot platform also demonstrates the novelty in aerial manipulation, which can be considered as an important contribution to the field robotics community.

Several open issues remain to be address in future work. First, an improved position control method should be developed to enable more aggressive maneuvering. Second, the measurement delay of a sensor which is affected by the whether and location was manually tuned by human in this work. Hence, a temporal calibration method should be developed to autonomously identify the measurement delay, and further improve the accuracy of the state estimation. Last but not least, a cooperative control system should be investigated to achieve the manipulation or transportation of a large object by using multiple robots.
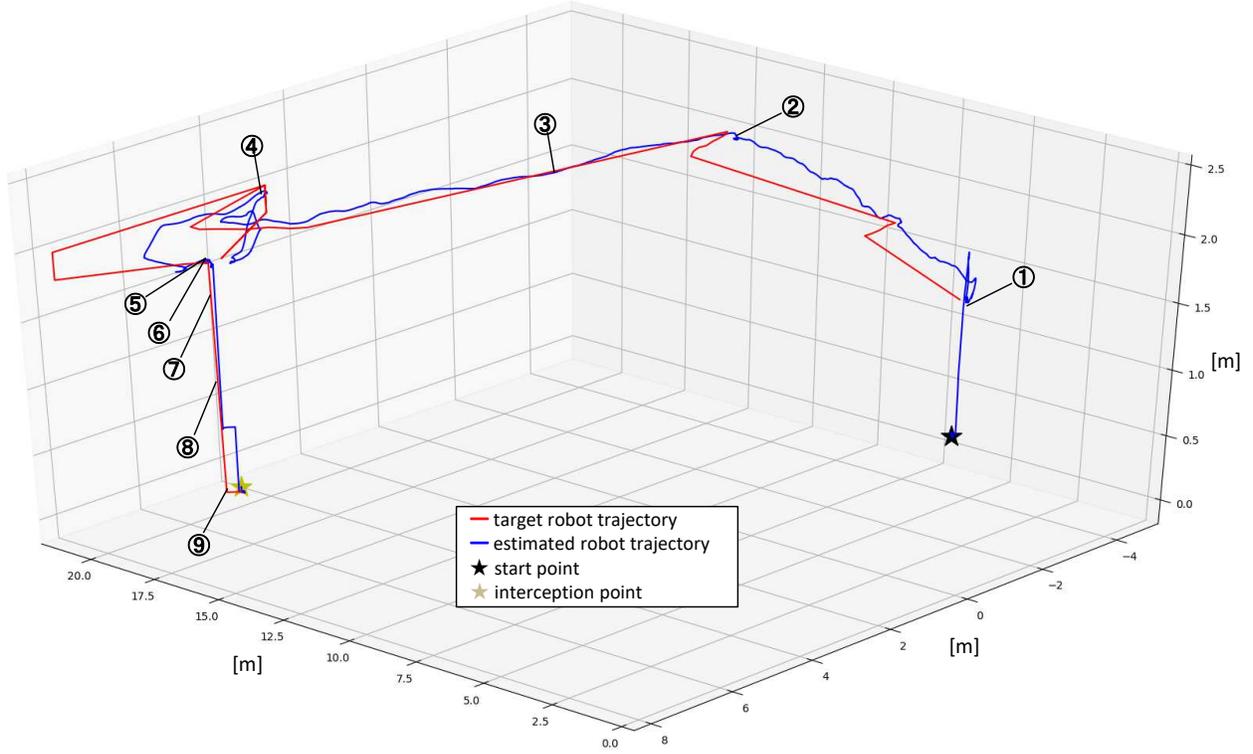
Figure 25: The trajectory of our robot related to the autonomous firefighting task as shown in Fig. 24.

Appendix A: Postion Error Dynamics

The position error dynamics is given by:

$$
\begin{aligned}
m\ddot{\boldsymbol{e}}_r &= m\ddot{\boldsymbol{r}} - m\ddot{\boldsymbol{r}}^{\text{des}} \\
&= m\boldsymbol{g} + R(f_T^{\text{des}}\boldsymbol{b}_3 + Q_{\text{tran}}\boldsymbol{u}_{\text{att}}^{\text{des}}) - m\ddot{\boldsymbol{r}}^{\text{des}} + \boldsymbol{\Delta}_{\text{tran}} \\
&= m\boldsymbol{g} + ((R\boldsymbol{b}_3)^{\text{T}}\boldsymbol{f}^{\text{des}})R\boldsymbol{b}_3 + RQ_{\text{tran}}\boldsymbol{u}_{\text{att}}^{\text{des}} - m\ddot{\boldsymbol{r}}^{\text{des}} + \boldsymbol{\Delta}_{\text{tran}} \\
&= m\boldsymbol{g} + \|\boldsymbol{f}^{\text{des}}\|((R^{\text{des}}\boldsymbol{b}_3)^{\text{T}}R\boldsymbol{b}_3)R\boldsymbol{b}_3 + RQ_{\text{tran}}\boldsymbol{u}_{\text{att}}^{\text{des}} - m\ddot{\boldsymbol{r}}^{\text{des}} + \boldsymbol{\Delta}_{\text{tran}} \\
&= m\boldsymbol{g} + \|\boldsymbol{f}^{\text{des}}\|R^{\text{des}}\boldsymbol{b}_3 - \|\boldsymbol{f}^{\text{des}}\|R^{\text{des}}\boldsymbol{b}_3 + \|\boldsymbol{f}^{\text{des}}\|((R^{\text{des}}\boldsymbol{b}_3)^{\text{T}}R\boldsymbol{b}_3)R\boldsymbol{b}_3 + RQ_{\text{tran}}\boldsymbol{u}_{\text{att}}^{\text{des}} - m\ddot{\boldsymbol{r}}^{\text{des}} + \boldsymbol{\Delta}_{\text{tran}} \\
&= m\boldsymbol{g} + \|\boldsymbol{f}^{\text{des}}\|R^{\text{des}}\boldsymbol{b}_3 + \|\boldsymbol{f}^{\text{des}}\|(((R^{\text{des}}\boldsymbol{b}_3)^{\text{T}}R\boldsymbol{b}_3)R\boldsymbol{b}_3 - R^{\text{des}}\boldsymbol{b}_3) + RQ_{\text{tran}}\boldsymbol{u}_{\text{att}}^{\text{des}} - m\ddot{\boldsymbol{r}}^{\text{des}} + \boldsymbol{\Delta}_{\text{tran}} \\
&= m\boldsymbol{g} + \|\boldsymbol{f}^{\text{des}}\|R^{\text{des}}\boldsymbol{b}_3 + \boldsymbol{X} + RQ_{\text{tran}}\boldsymbol{u}_{\text{att}}^{\text{des}} - m\ddot{\boldsymbol{r}}^{\text{des}} + \boldsymbol{\Delta}_{\text{tran}} \quad (55)
\end{aligned}
$$

where $\boldsymbol{X} = \|\boldsymbol{f}^{\text{des}}\|(((R^{\text{des}}\boldsymbol{b}_3)^{\text{T}}R\boldsymbol{b}_3)R\boldsymbol{b}_3 - R^{\text{des}}\boldsymbol{b}_3)$. Note that, $^{\{W\}}R_{\{CoG\}}$ and m_Σ are simplified as R and m for convenience.

Substituting Eq. 35 into Eq. 55, the further derivation can be given by

$$
\begin{aligned}
m\ddot{\boldsymbol{e}}_r &= -mK_P\boldsymbol{e}_r - mK_I\boldsymbol{e}_{I_x} - mK_D\dot{\boldsymbol{e}}_r + \boldsymbol{X} + RQ_{\text{tran}}K_x\bar{\boldsymbol{x}} + \boldsymbol{\Delta}_{\text{tran}} \\
&= -mK_P\boldsymbol{e}_r - mK_I\boldsymbol{e}_{I_r} - mK_D\dot{\boldsymbol{e}}_r + \boldsymbol{X} + RQ'_{\text{tran}}\boldsymbol{e}_{\text{rot}} + \boldsymbol{\Delta}_{\text{tran}} + \boldsymbol{\Delta}'_{\text{rot}} \quad (42)
\end{aligned}
$$

where $\boldsymbol{\Delta}'_{\text{rot}} = Q_{\text{tran}}Q^{\#}_{\text{rot}}\boldsymbol{\Delta}_{\text{rot}}$ and $Q'_{\text{tran}} = Q_{\text{tran}}K_x$. Note that R is omitted from $\boldsymbol{\Delta}'_{\text{rot}}$, since such converted fixed uncertainty from rotational space should be also constant in translational space regardless of the change of R. For instance, the offset of the CoG origin which induces a moment in rotational motion should be converted to be a constant force in translational force.

Appendix B: Complete Stability

As shown in Eq. 44, the Lyapunov candidate regarding the position error dynamics is given by

$$\mathcal{V}_2 = \frac{1}{2}e_r^{\mathrm{T}}K_P e_r + \frac{1}{2}\|\dot{e}_r\|^2 + ce_r^{\mathrm{T}}\dot{e}_r + \frac{1}{2}(e_{I_r} - \boldsymbol{\Delta}\circ\frac{1}{\boldsymbol{k}_{I_r}})^{\mathrm{T}}K_{I_r}(e_{I_r} - \boldsymbol{\Delta}\circ\frac{1}{\boldsymbol{k}_{I_r}}) \tag{44}$$

Then, the time derivative can be given by

$$
\begin{aligned}
\dot{\mathcal{V}}_2 &= e_r^{\mathrm{T}}K_P\dot{e}_r + \dot{e}_r^{\mathrm{T}}\ddot{e}_r + c\dot{e}_r^{\mathrm{T}}\dot{e}_r + ce_r^{\mathrm{T}}\ddot{e}_r + \dot{e}_I^{\mathrm{T}}K_{I_r}(e_I - \boldsymbol{\Delta}\circ\frac{1}{\boldsymbol{k}_{I_r}}) \\
&= c\|\dot{e}_r^{\mathrm{T}}\|^2 + e_r^{\mathrm{T}}K_P\dot{e}_r + (\dot{e}_r^{\mathrm{T}} + ce_r^{\mathrm{T}})\ddot{e}_r + (\dot{e}_r^{\mathrm{T}} + ce_r^{\mathrm{T}})(K_{I_r}e_{I_r} - \boldsymbol{\Delta}) \\
&= c\|\dot{e}_r^{\mathrm{T}}\|^2 + e_r^{\mathrm{T}}K_P\dot{e}_r + (\dot{e}_r^{\mathrm{T}} + ce_r^{\mathrm{T}})(-K_P e_r - K_{I_r}e_{I_r} - K_D\dot{e}_r + \frac{\boldsymbol{X} + RQ'_{\mathrm{tran}}\boldsymbol{e}_{\mathrm{rot}}}{m} + \boldsymbol{\Delta} + K_{I_r}e_{I_r} - \boldsymbol{\Delta}) \\
&= -ce_r^{\mathrm{T}}K_P e_r + \dot{e}_r^{\mathrm{T}}(C - K_D)\dot{e} - ce_r^{\mathrm{T}}K_D\dot{e}_r + (\dot{e}_r^{\mathrm{T}} + ce_r^{\mathrm{T}})\frac{\boldsymbol{X} + RQ'_{\mathrm{tran}}\boldsymbol{e}_{\mathrm{rot}}}{m} \tag{56}
\end{aligned}
$$

where, $C = cE_{3\times 3}$.

Then the upper bound of $\dot{\mathcal{V}}_2$ can be given by:

$$
\begin{aligned}
\dot{\mathcal{V}}_2 &\leq -ck_{P_{\min}}\|e_r\|^2 + (c - k_{D_{\min}})\|\dot{e}_r\|^2 + ck_{D_{\max}}\|e_r\|\|\dot{e}_r\| + (\|\dot{e}_r\| + c\|e_r\|)\frac{\|\boldsymbol{X}\| + \|RQ'_{\mathrm{tran}}\boldsymbol{e}_{\mathrm{rot}}\|}{m} \\
&\leq -ck_{P_{\min}}\|e_r\|^2 + (c - k_{D_{\min}})\|\dot{e}_r\|^2 + ck_{D_{\max}}\|e_r\|\|\dot{e}_r\| + (\|\dot{e}_r\| + c\|e_r\|)\frac{\|\boldsymbol{X}\| + \sigma_{max}(Q'_{\mathrm{tran}})\|\boldsymbol{e}_{\mathrm{rot}}\|}{m},
\end{aligned} \tag{57}
$$

where, $k_{P_{\min}}$ and $k_{D_{\min}}$ are the minimum element in the gain matrix K_P and K_D respectively, while $k_{D_{\max}}$ is the maximum element in the gain matrix K_D. Also note that $\sigma_{max}(\cdot)$ denotes the maximum singular value of a matrix.

In terms of $\boldsymbol{X}$, $((R^{\mathrm{des}}\boldsymbol{b}_3)^{\mathrm{T}}R\boldsymbol{b}_3)R\boldsymbol{b}_3 - R^{\mathrm{des}}\boldsymbol{b}_3 = (R\boldsymbol{b}_3)\times((R\boldsymbol{b}_3)\times(R^{\mathrm{des}}\boldsymbol{b}_3))$, and $\|(R\boldsymbol{b}_3)\times((R\boldsymbol{b}_3)\times(R^{\mathrm{des}}\boldsymbol{b}_3))\|$ represents the sine angle between $R\boldsymbol{b}_3$ and $R^{\mathrm{des}}\boldsymbol{b}_3$. Besides we assume the following constraint is available for translational motion:

$$\|mK_{I_r}\boldsymbol{e}_{I_r} + m\ddot{\boldsymbol{r}}^{\mathrm{des}} - RQ_{\mathrm{tran}}Q_{\mathrm{rot}}^{\#}I^{-1}\boldsymbol{\omega}\times I\boldsymbol{\omega}\| < O \tag{58}$$

Thus following constraints are available for X:

$$
\begin{aligned}
\|\boldsymbol{X}\| &\leq (mk_{P_{\max}}\|\boldsymbol{e}_{\mathrm{r}}\| + mk_{D_{\max}}\|\dot{\boldsymbol{e}}_{\mathrm{r}}\| + O)\gamma; \gamma\in(0,1) \tag{59} \\
\|\boldsymbol{X}\| &\leq (mk_{P_{\max}}\|\boldsymbol{e}_{\mathrm{r}}\| + mk_{D_{\max}}\|\dot{\boldsymbol{e}}_{\mathrm{r}}\| + O)\|\boldsymbol{e}_{\mathrm{rot}}\| \tag{60}
\end{aligned}
$$

Substituting Eq. 59 and Eq. 60 into Eq. 56:

$$
\begin{aligned}
\dot{\mathcal{V}}_2 \leq{}& -ck_{P_{\min}}\|\boldsymbol{e}_r\|^2 + (c - k_{D_{\min}})\|\dot{\boldsymbol{e}}_r\|^2 + ck_{D_{\max}}\|\boldsymbol{e}_r\|\|\dot{\boldsymbol{e}}_r\| + (\|\dot{\boldsymbol{e}}_r\| + c\|\boldsymbol{e}_r\|)\frac{\sigma_{max}(Q'_{\mathrm{tran}})}{m}\|\bar{\boldsymbol{e}}_{\mathrm{rot}}\| \\
&+ \frac{O}{m}\|\boldsymbol{e}_{\mathrm{rot}}\|(\|\dot{\boldsymbol{e}}_r\| + c\|\boldsymbol{e}_r\|) + k_{P_{\max}}\|\boldsymbol{e}_{\mathrm{rot}}\|\|\dot{\boldsymbol{e}}_r\|\|\boldsymbol{e}_r\| + \gamma ck_{P_{\max}}\|\boldsymbol{e}_r\|^2 + \gamma ck_{D_{\max}}\|\dot{\boldsymbol{e}}_r\|\|\boldsymbol{e}_r\| + \gamma k_{D_{\max}}\|\dot{\boldsymbol{e}}_r\|^2
\end{aligned} \tag{61}
$$

Regarding the third term in Eq. 61, namely, $k_{P_{\max}}\|\boldsymbol{e}_{\mathrm{rot}}\|\|\dot{\boldsymbol{e}}_r\|\|\boldsymbol{e}_r\|$, a upper bound for the position error is introduced to reduce the order: $\|\boldsymbol{e}_r\| < \boldsymbol{e}_{r_{\max}}$. Finally, the upper bound of $\dot{\mathcal{V}}_2$ can be given by

$$
\begin{aligned}
\dot{\mathcal{V}}_2 \leq{}& (-ck_{P_{\min}} + \gamma ck_{P_{\max}})\|\boldsymbol{e}_r\|^2 + (c - k_{D_{\min}} + \gamma k_{D_{\max}})\|\dot{\boldsymbol{e}}_r\|^2 + (1 + \gamma)ck_{D_{\max}}\|\boldsymbol{e}_r\|\|\dot{\boldsymbol{e}}_r\| \\
&+ (\|\dot{\boldsymbol{e}}_r\| + c\|\boldsymbol{e}_r\|)\frac{\sigma_{max}(Q'_{\mathrm{tran}}) + O}{m}\|\boldsymbol{e}_{\mathrm{rot}}\| + k_{P_{\max}}\boldsymbol{e}_{r_{\max}}\|\dot{\boldsymbol{e}}_r\|\|\boldsymbol{e}_{\mathrm{rot}}\| \tag{62}
\end{aligned}
$$

Then, the integral Lyapunov candidate for the complete system as shown in Eq. 43 is rewritten with $z_1 = \begin{bmatrix} \|e_r\| & \|\dot{e}_r\| \end{bmatrix}^T \in \mathsf{R}^2$, $z_2 = \|e_{\mathrm{rot}}\|$. The lower bound of $\mathcal{V}$ can be given by

$$z_1^T M_1 z_1 + \mathcal{V}_1 + \mathcal{V}_I \leq \mathcal{V} \tag{63}$$

where,

$$\mathcal{V}_I = \frac{1}{2}(e_{I_r} - \boldsymbol{\Delta} \circ \frac{1}{\boldsymbol{k}_{I_r}})^T K_{I_r}(e_{I_r} - \boldsymbol{\Delta} \circ \frac{1}{\boldsymbol{k}_{I_r}}), \quad M_1 = \frac{1}{2}\begin{bmatrix} k_{P_{\min}} & -c \\ -c & 1 \end{bmatrix}$$

Note that, M should be positive-defined to guarantee the $\mathcal{V}_I \geq 0$. This leads to the constraint about the constant c: $c < \sqrt{k_{P_{\min}}}$.

The time derivative of $\dot{\mathcal{V}}$ is given by

$$\begin{aligned} \dot{\mathcal{V}} &\leq -z_1^T W_1 z_1 + z_1^T W_{12} z_2 - e_{\mathrm{rot}}^T W_2 e_{\mathrm{rot}} \\ &\leq -\lambda_{\min}(W_1)\|z_1\|^2 + \sigma_{\max}(W_{12})\|z_1\|z_2 - \lambda_{\min}(W_2)z_2^2 \end{aligned} \tag{64}$$

where,

$$W_1 = \frac{1}{2}\begin{bmatrix} c(k_{P_{\min}} - \gamma k_{P_{\max}}) & -\frac{ck_{D_{\max}}}{2}(1+\gamma) \\ -\frac{ck_{D_{\max}}}{2}(1+\gamma) & -c + k_{D_{\min}} - \gamma k_{D_{\max}} \end{bmatrix}, W_{12} = \begin{bmatrix} \frac{c(\sigma_{max}(Q'_{\mathrm{tran}})+O)}{m} \\ \frac{\sigma_{max}(Q'_{\mathrm{tran}})+O}{m} + k_{P_{\max}}e_{r_{\max}} \end{bmatrix}, W_2 = -(\bar{A} + \bar{B}K_x)$$

$\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ are the maximum and minimum eigenvalue of a matrix. Note that all eigenvalues of W_2 are negative.

Define $z = \begin{bmatrix} \|z_1\| & z_2 \end{bmatrix}^T \in \mathsf{R}^2$, then Eq. 64 can be further summarized as follows:

$$\begin{aligned} \dot{\mathcal{V}} &\leq -z^T W z \\ &\leq -\lambda_{\min}(W)\|z\|^2 \end{aligned} \tag{65}$$

where the matrix $W \in \mathcal{R}^{2\times 2}$ is given by,

$$W = \frac{1}{2}\begin{bmatrix} \lambda_{\min}(W_1) & -\frac{\sigma_{\max}(W_{12})}{2} \\ -\frac{\sigma_{\max}(W_{12})}{2} & \lambda_{\min}(W_2) \end{bmatrix}$$

In order to guarantee $\dot{\mathcal{V}} < 0$, W is required to be positive-defined. In other words, all eigenvalues should be positive. This derives following constraints:

$$\lambda_{\min}(W_1) > 0 \tag{66}$$

$$\lambda_{\min}(W_1)\lambda_{\min}(W_2) > \frac{\sigma_{\max}^2(W_{12})}{4} \tag{47}$$

Note that Eq. 66 also implies W_1 should be positive defined. Therefore, the constraints regarding positive constant c and control gains can be given by

$$k_{P_{\min}} > \gamma k_{P_{\max}} \tag{45}$$

$$c < \min\{\frac{4(k_{P_{\min}} - \gamma k_{P_{\max}})(k_{D_{\min}} - \gamma k_{D_{\max}})}{k_{D_{\max}}^2(1+\gamma)^2 + 4(k_{P_{\min}} - \gamma k_{P_{\max}})}, k_{D_{\min}} - \gamma k_{D_{\max}}, \sqrt{k_{P_{\min}}}\} \tag{46}$$

With the given constraints Eq. 47 $\sim$ Eq. 46, the zero equilibrium of the tracking errors is exponentially stable with respect to $e_r, \dot{e}_r, e_x$, and the integral terms e_{I_x}, e_{I_r} are uniformly bounded.

References

Anzai, T. et al. (2018). Aerial grasping based on shape adaptive transformation by halo: Horizontal plane transformable aerial robot with closed-loop multilinks structure. In *Proceedings of 2018 IEEE International Conference on Robotics and Automation*, pages 6990–6996.

Anzai, T. 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)*, pages 2820–2826.

Anzai, T., Zhao, M., Chen, X., Shi, F., Kawasaki, K., Okada, K., and Inaba, M. (2017). Multilinked multirotor with internal communication system for multiple objects transportation based on form optimization method. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5977–5984.

Bähnemann, R., Pantic, M., Popović, M., Schindler, D., Tranzatto, M., Kamel, M., Grimm, M., Widauer, J., Siegwart, R., and Nieto, J. (2019). The eth-mav team in the mbz international robotics challenge. *Journal of Field Robotics*, 36(1):78–103.

Beul, M., Nieuwenhuisen, M., Quenzel, J., Rosu, R. A., Horn, J., Pavlichenko, D., Houben, S., and Behnke, S. (2019). Team nimbro at mbzirc 2017: Fast landing on a moving target and treasure hunting with a team of micro aerial vehicles. *Journal of Field Robotics*, 36(1):204–229.

Bloesch, M., Burri, M., Omari, S., Hutter, M., and Siegwart, R. (2017). Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research*, 36(10):1053–1072.

Bonatti, R., Wang, W., Ho, C., Ahuja, A., Gschwindt, M., Camci, E., Kayacan, E., Choudhury, S., and Scherer, S. (2020). Autonomous aerial cinematography in unstructured environments with learned artistic decision-making. *Journal of Field Robotics*, 37(4):606–641.

Brescianini, D. et al. (2016). Design, modeling and control of an omni-directional aerial vehicle. In *Proceedings of 2016 IEEE International Conference on Robotics and Automation*, pages 3261–3266.

Efraim, H., Shapiro, A., and Weiss, G. (2015). Quadrotor with a dihedral angle: on the effects of tilting the rotors inwards. *Journal of Intelligent & Robotic Systems*, 80:313–324.

Falanga, D., Kleber, K., Mintchev, S., Floreano, D., and Scaramuzza, D. (2019). The foldable drone: A morphing quadrotor that can squeeze and fly. *IEEE Robotics and Automation Letters*, 4(2):209–216.

Forster, C., Zhang, Z., Gassner, M., Werlberger, M., and Scaramuzza, D. (2017). Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265.

Gabrich, B., Saldana, D., Kumar, V., and Yim, M. (2018). A flying gripper based on cuboid modular robots. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7024–7030.

Goodarzi, F., Lee, D., and Lee, T. (2013). Geometric nonlinear pid control of a quadrotor uav on se(3). In *2013 European Control Conference (ECC)*, pages 3845–3850.

ISO (1993). Road vehicles - interchange of digital information - controller area network (can) for high-speed communication. In *ISO 11898*.

Kumar, V. and Michael, N. (2012). Opportunities and challenges with autonomous micro aerial vehicles.

Lee, J., Shim, D. H., Cho, S., Shin, H., Jung, S., Lee, D., and Kang, J. (2019). A mission management system for complex aerial logistics by multiple unmanned aerial vehicles in mbzirc 2017. *Journal of Field Robotics*, 36(5):919–939.

Lee, T., Leok, M., and McClamroch, N. H. (2010). Geometric tracking control of a quadrotor uav on se(3). In *49th IEEE Conference on Decision and Control (CDC)*, pages 5420–5425.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer Vision – ECCV 2016*, pages 21–37. Springer International Publishing.

Lynen, S., Achtelik, M. W., Weiss, S., Chli, M., and Siegwart, R. (2013). A robust and modular multi-sensor fusion approach applied to mav navigation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3923–3929.

Madgwick, S. O. H., Harrison, A. J. L., and Vaidyanathan, R. (2011). Estimation of imu and marg orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–7.

Marins, J. L. C., Yun, X., Bachmann, E. R., McGhee, R. B., and Zyda, M. J. (2001). An extended kalman filter for quaternion-based orientation estimation using marg sensors. *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, 4:2003–2011 vol.4.

Martin, P. and Salaün, E. (2010). The true role of accelerometer feedback in quadrotor control. In *2010 IEEE International Conference on Robotics and Automation*, pages 1623–1629.

Michael, N., Shen, S., Mohta, K., Mulgaonkar, Y., Kumar, V., Nagatani, K., Okada, Y., Kiribayashi, S., Otake, K., Yoshida, K., Ohno, K., Takeuchi, E., and Tadokoro, S. (2012). Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29:832–841.

Mur-Artal, R., Montiel, J. M. M., and Tards, J. D. (2015). Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163.

Ore, J.-P. and Detweiler, C. (2018). Sensing water properties at precise depths from the air. *Journal of Field Robotics*, 35(8):1205–1221.

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

Qin, T., Li, P., and Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020.

Rajappa, S., Ryll, M., Blthoff, H. H., and Franchi, A. (2015). Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers. In *Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4006–4013.

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

Ryll, M., Bicego, D., and Franchi, A. (2016). Modeling and control of fast-hex: A fully-actuated by synchronized-tilting hexarotor. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1689–1694.

Shi, F. et al. (2019). Achievement of online agile manipulation task for aerial transformable multilink robot. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 221–228.

Shi, F. et al. (2019). Multi-rigid-body dynamics and online model predictive control for transformable multi-links aerial robot. *Advanced Robotics*, 33(19):971–984.

Shi, F., Zhao, M., Murooka, M., Okada, K., and Inaba, M. (2020). Aerial regrasping: Pivoting with transformable multilink aerial robot. pages 200–207.

Spurný, V., Báča, T., Saska, M., Pěnička, R., Krajník, T., Thomas, J., Thakur, D., Loianno, G., and Kumar, V. (2019). Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles. *Journal of Field Robotics*, 36(1):125–148.

Sun, K., Mohta, K., Pfrommer, B., Watterson, M., Liu, S., Mulgaonkar, Y., Taylor, C. J., and Kumar, V. (2018). Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters*, 3(2):965–972.

Tzoumanikas, D., Li, W., Grimm, M., Zhang, K., Kovac, M., and Leutenegger, S. (2019). Fully autonomous micro air vehicle flight and landing on a moving target using visualinertial estimation and model-predictive control. *Journal of Field Robotics*, 36(1):49–77.

Yang, H. et al. (2018). Lasdra: Large-size aerial skeleton system with distributed rotor actuation. In *Proceedings of 2018 IEEE International Conference on Robotics and Automation*, pages 7017–7023.

Young, P. C. and Willems, J. C. (1972). An approach to the linear multivariable servomechanism problem. *International Journal of Control*, 15(5):961–979.

Zhao, M. et al. (2016). Transformable multirotor with two-dimensional multilinks: modeling, control, and motion planning for aerial transformation. *Advanced Robotics*, 30(13):825–845.

Zhao, M. et al. (2018). Transformable multirotor with two-dimensional multilinks: Modeling, control, and whole-body aerial manipulation. *I. J. Robotics Res.*, 37(9):1085–1112.

Zhao, M., Kawasaki, K., Chen, X., Noda, S., Okada, K., and Inaba, M. (2017). Whole-body aerial manipulation by transformable multirotor with two-dimensional multilinks. In *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5175–5182.

Zhao, N., Luo, Y., Deng, H., and Shen, Y. (2017). The deformable quad-rotor: Design, kinematics and dynamics characterization, and flight performance validation. In *in Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2391–2396.