Attitude Tracking Control on SO(3) Group with Linearization on Moving Operating Point for Transporting Quadrotor

Nurman Setiawan, Gilang Nugraha Putu Pratama *Member, IAENG*,
Hendra Ari Winarno, Samiadji Herdjunanto, and Adha Imam Cahyadi, *Member, IAENG*

***Abstract*—Attitude tracking control for quadrotor is a challenging topic, simply due to its nonlinearity. In this article, we present an attitude tracking control for quadrotor on Special Orthogonal-3 group. The proposed control system consist of dynamics compensator and Proportional-Derivative (PD) controller. By implementing the PD controller, performance of a linear system can be adjusted as desired. However, quadrotor is a non-linear system, hence it must be linearized along references angle firstly. In addition, the quadrotor is designed to transport payload. This payload generates an additional uncertainty of inertia which is called inertia perturbation. Inertia perturbation affects the dynamics of overall quadrotor system. The numerical simulation and results verify that the reference angle can be tracked by employing a PD controller plus dynamics compensator for the quadrotor despite under inertia perturbation due to transport a payload.**

***Index Terms*—quadrotor, PD controller, dynamics compensator, SO(3), attitude tracking.**

I. Introduction

QUADROTOR is a cross-shaped unmanned aerial vehicle with 4 rotors mounted on the ends of each arm. It offers some interesting features such as hovering, flying at low altitudes, taking-off and landing vertically, and pretty good maneuverability. Those features make quadrotor gains more popularity as a good platform in robotics [1], [2]. Today, its serves are not limited for military and research purpose, but also including in social environment such as search and rescue mission, agriculture, mitigation, surveillance, transporting payloads, and aerial photography [3]–[8].

Despite its great potentials, the problems arise since the quadrotor is an underactuated system. The term underactuated system refers to any system that has more outputs to be controlled than its inputs. Technically speaking, a quadrotor has 6 degrees of freedom (DoF) with only 4 input signals, which are its rotors. It can change its attitude and move from one point to another by manipulating the angular speed of its rotors [9]. Furthermore, quadrotor is also a nonlinear system that vulnerable to instability. We can not make it stable just by depending on its mechanical structure. In order to stabilize it, such a proper algorithm need to be implemented [10].

Before designing a control algorithm for quadrotor, we have to get an insight into the attitude representation. It is necessary since it asserts the attitude and the translation along $x-$, $y-$, and $z-$axis that can be controlled.

There are some notable attitude representations, namely Euler angles and exponential coordinates [11]. Due to its simple approach and easy to understand, the Euler angles representation obtains more attention for practical use. It employs roll ϕ, pitch θ, and yaw ψ angle separately, thus it also called RPY (roll-pitch-yaw) representation [12]. In spite of its simplicity, the Euler angles representation has a critical drawback. Technically speaking, the inverse Wronskian matrix is employed to convert the angular body velocity into Euler angles representation. It should be noted that when the pitch angle reaches 90°, it causes some elements in the inverse Wronskian matrix to have infinite value, which we call a singularity. Moreover, this singularity leads to an undesired condition named gimbal lock. It should be hindered since the gimbal lock causes the quadrotor to loss one degree of freedom due to the roll and yaw axis lie align with each other [10].

On the other hand, the exponential coordinates offers a singularity-free representation of attitude. The reason why it is not as popular as the Euler angles representation is due to the fact that the exponential coordinates is rather complicated. Notable works for attitude control on exponential coordinates presents rigorous mathematical equations [13], [14]. Their works exhibit the superiority of the exponential coordinates over the Euler angles representation.

This paper presents attitude tracking control for quadrotor on the exponential coordinates while transporting the payloads. As mentioned before that quadrotor can be deployed for transporting payloads, even in a rough environment. In reality, it is not an easy task, since the payloads will cause inertia perturbations, by all means, the stability of quadrotor will be affected. Previously, Min [15] and Wang [16] have proposed a control algorithm for a quadrotor to overcome the perturbations due to the payloads. Both of them proposed adaptive control for quadrotor which serves quite good for transporting the payloads. Despite their good results, adaptive control seems to be too sophisticated in regards to the cost computationally high. Another thing that should be noted, they used the Euler angles representation which is vulnerable to singularity.

Another related work by Pratama [17]. They utilized the

Nurman Setiawan is a Lecturer of Department of Electrical Engineering Education, Universitas Negeri Yogyakarta , Indonesia (as a corresponding author; phone: 6285725806009; email: nurman.setiawan@uny.ac.id).

Gilang Nugraha Putu Pratama is an engineer in Beehive Drones, Indonesia (email: gilang.n.p.pratama@gmail.com).

Hendra Ari Winarno is a Lecturer of Department of Electrical Engineering, Universitas Muhammadiyah Gresik, Indonesia (email: hendra.ari.winarno@umg.ac.id).

Samiadji Herdjunanto is a Lecturer of Department of Electrical and Information Technology, Universitas Gadjah Mada, Indonesia (email: samiadji@ugm.ac.id).

Adha Imam Cahyadi is a Lecturer of Department of Electrical and Information Technology, Universitas Gadjah Mada, Indonesia (email: adha.imam@ugm.ac.id).

PD controller for a quadrotor to transport the payloads while also exploiting the exponential coordinates on Special Orthogonal-3. In spite of under inertia perturbations, the attitude of the quadrotor can be regulated at the reference angles by using the PD controller. Nevertheless, their model was just oversimplified. The payload was modelled as cubic-shaped which attached perfectly in the centre of quadrotor.

Here, we propose an improvement from the previous work by Pratama. We present a singularity-free PD controller for quadrotor on Special Orthogonal-3, while also robust against inertia perturbations. The proposed PD controller serves its purpose for a quadrotor to transport the payload without suffering from the singularity. Furthermore, a rather complex model for the payload is proposed here. The payload is not attached perfectly on the centre, instead, the payload is positioned in different points.

The rests of this paper are as follows. Brief explanation to the rotational matrix on Special Orthogonal-3 is exposed in Section II. Later, in Section III we describe the model for quadrotor and the inertia perturbation. The attitude control tracking can be found in Section IV. Here, the parameter of PD controller is determined by using linearization on moving operating point in exponential coordinate, or simply the gain scheduling mechanism. It can be asserted that those are the main contribution of this paper, to achieve the stability of the closed-loop system for different operating points. The inverse kinematics on exponential coordinate is elaborated here. In addition, we also present the dynamics compensator to calculate the nominal torque. Numerical simulation and its analysis are presented in Section V. Last but not the least, the summary and conclusion of this paper can be found in Section VI.

II. Rotation on SO(3) Group

In this section, we are going to discuss the rotation matrix on the Special Orthogonal-3 group or $\mathrm{SO}(3)$ for more conveniently, and the attitude representation on exponential coordinates based on Murray [18].

First things first, we need to distinguish the definition of $\mathrm{SO}(3)$ and $\mathfrak{so}(3)$. The term $\mathrm{SO}(3)$ is a group of a 3-by-3 orthogonal matrix. It is called special since its determinant is always $+1$. Suppose $\mathbf{R}$ is the rotation matrix in 3-dimensional space, thus $\mathrm{SO}(3)$ can be denoted as

$$\mathrm{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3\times 3} | \mathbf{R}\mathbf{R}^{\mathsf{T}} = \mathbf{I}_{3\times 3}, \det = +1\}. \quad (1)$$

If we apply multiplication matrices $\mathbf{R}$ and its transpose, it yields a 3-by-3 identity matrix $\mathbf{I}_{3\times 3}$.

The next term is $\mathfrak{so}(3)$, which is the Lie algebra of $\mathrm{SO}(3)$. It is important to be noted that $\mathfrak{so}(3)$ is a skew-symmetric matrix. Let $\mathbf{S}$ is a skew-symmetric matrix, thus $\mathfrak{so}(3)$ can be defined mathematically as

$$\mathfrak{so}(3) = \{\mathbf{S} \in \mathbb{R}^{3\times 3} | -\mathbf{S} = \mathbf{S}^{\mathsf{T}}\}. \quad (2)$$

The exponential coordinates has capability to represent any 3-dimensional rotation of the rigid body with a rotation of a given axis by some amount [19]–[21]. This feature tells the different between the exponential coordinates and the Euler angles representation. The exponential coordinates uses single arbitrary axis. On the contrary, the Euler angles representation utilizes composition of three consecutive rotations.

Suppose $\mathbf{q} \in \mathbf{R}^3$ is a point on the rigid body, which rotates about a given axis by some amount. Meanwhile, $\boldsymbol{\omega} \in \mathbb{R}^3$ be a unit vector specifying the rotation axis. Therefore, the angular velocity of point $\mathbf{q}$ about $\boldsymbol{\omega}$ axis can be defined as

$$\dot{\mathbf{q}}(t) = \boldsymbol{\omega} \times \mathbf{q}(t) = \widehat{\boldsymbol{\omega}}\mathbf{q}(t). \quad (3)$$

Based on the screw theory, let the vector $\boldsymbol{\omega} = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^{\mathsf{T}}$, can be mapped into a skew-symmetric matrix $\widehat{\boldsymbol{\omega}} \in \mathbb{R}^{3\times 3}$, given as

$$\widehat{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \quad (4)$$

Let assume the rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times 3}$, belongs to vector space $\mathrm{SO}(3)$ which can be mapped from $\mathfrak{so}(3)$ by applying the exponential operator as

$$\mathbf{R} = \exp(\widehat{\boldsymbol{\omega}}\vartheta) = \mathbf{I} + \vartheta\widehat{\boldsymbol{\omega}} + \frac{1}{2!}(\vartheta\widehat{\boldsymbol{\omega}})^2 + \frac{1}{3!}(\vartheta\widehat{\boldsymbol{\omega}})^3 + \dots. \quad (5)$$

We can take help of Rodrigues' formula to make equation (5) into closed-form as

$$\mathbf{R} = \mathbf{I} + \frac{\widehat{\boldsymbol{\omega}}}{\|\boldsymbol{\omega}\|}\sin(\|\boldsymbol{\omega}\|\vartheta) + \frac{\widehat{\boldsymbol{\omega}}^2}{\|\boldsymbol{\omega}\|^2}(1 - \cos(\|\boldsymbol{\omega}\|\vartheta)). \quad (6)$$

The mapping works both sides, since any orthogonal matrix on $\mathrm{SO}(3)$ also can be mapped into $\mathfrak{so}(3)$ by applying the logarithmic operator as

$$\log(\mathbf{R}) = \frac{\varphi}{2\sin\varphi}\left(\mathbf{R} - \mathbf{R}^{\mathsf{T}}\right), \quad (7)$$

where φ by definition of Bullo is $\arccos\left(\frac{tr(\mathbf{R})-1}{2}\right)$ and $|\varphi| < \pi$ [22]. Interestingly, let $\mathbf{R} = \mathbf{I}$, therefore we can choose the ω arbitrarily.

The exponential coordinates also comes in handy to map the rotating point in a rigid body. Now, please look back at the equation (3). Let $\mathbf{q}(0)$ is the initial point, we can utilize the exponential map to obtain the current point at time t as

$$\mathbf{q}(t) = \exp^{\widehat{\boldsymbol{\omega}}t}\mathbf{q}(0). \quad (8)$$

The exponential function in equation (8) can be seen as a rotation matrix. This rotation matrix represents the orientation of body frame attached on the rigid body relative to inertial frame [18].

III. Modeling

In this part the necessary model of quadrotor and the inertia perturbations are explained. The model of quadrotor based on dynamics and kinematics equations on exponential coordinate of $\mathrm{SO}(3)$ is exposed in Subsection III-A. Meanwhile, the inertia perturbation matrix caused by the payloads is explained later in Subsection III-B.

A. Quadrotor Model

As stated before, any 3-dimensional rotation of quadrotor can be represented by single rotation of a given axis by some amount. The direction of this axis is formed by vector $\boldsymbol{\omega}$ as illustrated in Figure 1.

The rotation angle ϑ is represented by dotted arrow, while the angle vector on exponential coordinates can be denoted as

$$\boldsymbol{\zeta} = \begin{bmatrix} \zeta_1 & \zeta_2 & \zeta_3 \end{bmatrix}^{\mathsf{T}} = \boldsymbol{\omega}\vartheta. \quad (9)$$

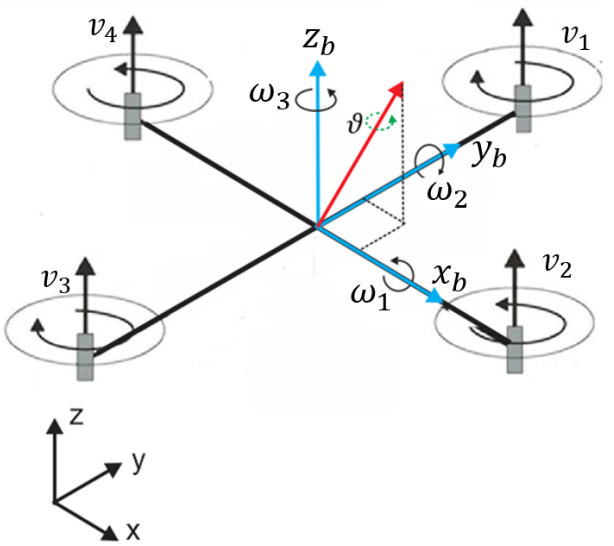

Fig. 1: Model of the quadrotor

Suppose $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathsf{T}}$ is a vector position of quadrotor, and its angular velocity relative to the body frame can denoted as vector $\mathbf{\Omega}_b = \begin{bmatrix} \Omega_1 & \Omega_2 & \Omega_3 \end{bmatrix}^{\mathsf{T}}$.

We take assumption that the quadrotor is symmetrical. Therefore, under this assumption, the initial inertia matrix of quadrotor can be defined as

$$\mathbf{J}_0 = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}. \tag{10}$$

After explaining the initial inertia matrix of quadrotor, we will explain its dynamics kinematics equations. Let $\mathbf{v} = \begin{bmatrix} v_1^2 & v_2^2 & v_3^2 & v_4^2 \end{bmatrix}^{\mathsf{T}}$ is the input vector of quadrotor, where its elements are the squared angular speed of its rotors. The torques along x, y, and $z-$axis can be generated by manipulating the angular speed of its rotors. The relationship between them can be described as

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} db\left(v_1^2 - v_3^2\right) \\ db\left(v_2^2 - v_4^2\right) \\ k\left(\left(v_1^2 + v_3^2\right) - \left(v_2^2 + v_4^2\right)\right) \end{bmatrix}, \tag{11}$$

where the constant b, d, and k consecutively are the thrust coefficient, the length of its arm, and the coefficient of propeller torque.

Therefore, the dynamics of quadrotor based on the Euler equation is as follows

$$\dot{\mathbf{\Omega}}_b = -\mathbf{J_0}^{-1}(\mathbf{\Omega}_b \times \mathbf{J_0}\mathbf{\Omega}_b) + \mathbf{J_0}^{-1}\boldsymbol{\tau}. \tag{12}$$

On the other hand, its kinematics can be derived from the velocity of rotating body as

$$\dot{\mathbf{R}} = \mathbf{R}\widehat{\mathbf{\Omega}}_b. \tag{13}$$

The above equation that defines the kinematics of quadrotor can be rewritten on the exponential coordinates as

$$\dot{\zeta} = \left(\mathbf{I} + \frac{1}{2}\widehat{\zeta} + \left(1 - \alpha(\|\zeta\|)\frac{(\widehat{\zeta})^2}{\|\zeta\|^2}\right)\right)\mathbf{\Omega}_b, \tag{14}$$

where $\alpha\left(\|\zeta\|\right) = \left(\frac{\|\zeta\|}{2}\right)\cot\left(\frac{\|\zeta\|}{2}\right)$. Furthermore, based on the equation (12) and (14), the second-order equation of quadrotor on $\mathrm{SO}(3)$ can be denoted as

$$\begin{bmatrix} \dot{\zeta} \\ \dot{\mathbf{\Omega}}_b \end{bmatrix} = \begin{bmatrix} \left(\mathbf{I} + \frac{1}{2}\widehat{\zeta} + \left(1 - \alpha\left(\|\zeta\|\right)\frac{(\widehat{\zeta})^2}{\|\zeta\|^2}\right)\right)\mathbf{\Omega}_b \\ -\mathbf{J_0}^{-1}\left(\mathbf{\Omega}_b \times \mathbf{J_0}\mathbf{\Omega}_b\right) + \mathbf{J_0}^{-1}\boldsymbol{\tau} \end{bmatrix}. \tag{15}$$

B. Inertia Perturbation Matrix

There are three inertia matrices refered in this paper, namely $\mathbf{J}_0$, $\Delta\mathbf{J}$, and $\mathbf{J}$. As explained before in Subsection III-A, $\mathbf{J}_0$ is the inertia matrix of quadrotor in nominal condition, without carrying any payload.

Meanwhile, $\Delta\mathbf{J}$ is the inertia perturbation matrix due to the payloads. Last mentioned, $\mathbf{J}$ is the total amount of the inertia matrix of quadrotor while transporting the payloads which can be defined as

$$\mathbf{J} = \mathbf{J}_0 + \Delta\mathbf{J}. \tag{16}$$

Since the inertia matrix in nominal condition has been explained before, here we can specifically discuss both $\Delta\mathbf{J}$ and $\mathbf{J}$. Practically speaking, the magnitude and direction of $\Delta\mathbf{J}$ are unknown since the mass and laying position of payload are not decided before. Therefore, in order to simplify the problem, we chose to model the payload be a cubic-shaped as shown in Figure 2, with its dimensions are $0.2\,\mathrm{m}$ of length, $0.2\,\mathrm{m}$ of width, and $0.2\,\mathrm{m}$ of height. Meantime, its positions in regard to a, b, c, d, e, and f can be adjusted to vary the laying position of the payload.

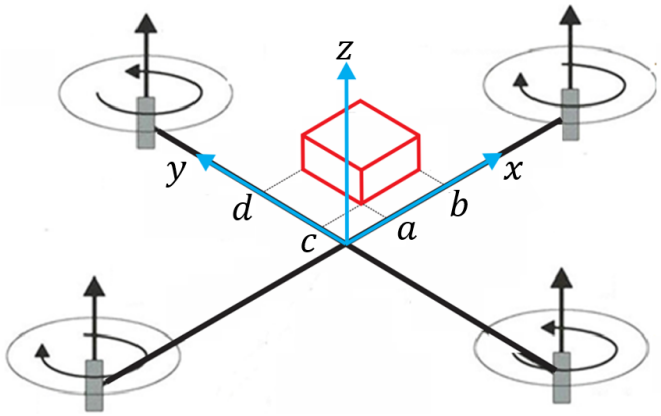

Fig. 2: Model of the payload

Based on those model, the inertia perturbation matrix $\Delta\mathbf{J}$ can be denoted as

$$\Delta\mathbf{J} = \begin{bmatrix} \Delta J_{xx} & \Delta J_{xy} & \Delta J_{xz} \\ \Delta J_{yx} & \Delta J_{yy} & \Delta J_{yz} \\ \Delta J_{zx} & \Delta J_{zy} & \Delta J_{zz} \end{bmatrix}. \tag{17}$$

We can use the inertia tensor to determine the elements of

$\Delta\mathbf{J}$ as

$$\Delta J_{xx} = \int_a^b \int_c^d \int_e^f \rho(y^2+z^2)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,, \quad (18)$$

$$\Delta J_{yy} = \int_a^b \int_c^d \int_e^f \rho(x^2+z^2)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,, \quad (19)$$

$$\Delta J_{zz} = \int_a^b \int_c^d \int_e^f \rho(x^2+y^2)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,, \quad (20)$$

$$\Delta J_{xy} = -\int_a^b \int_c^d \int_e^f \rho(xy)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,, \quad (21)$$

$$\Delta J_{xz} = -\int_a^b \int_c^d \int_e^f \rho(xz)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,, \quad (22)$$

$$\Delta J_{yz} = -\int_a^b \int_c^d \int_e^f \rho(yz)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x\,. \quad (23)$$

where the off-diagonals are equivalent to their respective, such that $\Delta J_{xy} \equiv \Delta J_{yx}$, $\Delta J_{xz} \equiv \Delta J_{zx}$, and $\Delta J_{yz} \equiv \Delta J_{zy}$. We write it this way to avoid any redundant equations.

Those equations can be rewritten into more convenient forms as

$$\Delta J_{xx} = \frac{1}{3}\rho(b-a)((d^3-c^3)(f-e) + (f^3-e^3)(d-c)), \quad (24)$$

$$\Delta J_{yy} = \frac{1}{3}\rho(d-c)((b^3-a^3)(f-e) + (f^3-e^3)(b-a)), \quad (25)$$

$$\Delta J_{zz} = \frac{1}{3}\rho(f-e)((b^3-a^3)(d-c) + (d^3-c^3)(b-a)), \quad (26)$$

$$\Delta J_{xy} = -\frac{1}{4}\rho(b^2-a^2)(d^2-c^2)(f-e), \quad (27)$$

$$\Delta J_{xz} = -\frac{1}{4}\rho(b^2-a^2)(f^2-e^2)(d-c), \quad (28)$$

$$\Delta J_{yz} = -\frac{1}{4}\rho(d^2-c^2)(f^2-e^2)(b-a), \quad (29)$$

where the density of the payload ρ is as follows

$$\rho = \frac{m_b}{|\,b-a\,|\,|\,d-c\,|\,|\,f-e\,|}, \quad (30)$$

and m_b is the mass of the payload. Nevertheless, the mass of payload and and its laying position are limited in certain ranges. The laying position of the payload can be anywhere between $-0.5\,\mathrm{m}$ and $0.5\,\mathrm{m}$ from the centre. Meanwhile, its mass should be less than $0.5\,\mathrm{kg}$.

IV. Design of Attitude Tracking Control on SO(3) Group

The performance of quadrotor is significantly affected by its control algorithm, specifically the attitude control. It is not exageration. Since technically, the control-loop of quadrotor can be devided into the inner and outer-loop [23]. The inner-loop is focus to regulate the altitude and the attitude of quadrotor in 3-dimensional space. On the other hand, the outer-loop takes responsibility to handle the translation along $x-$ and $y-$axis. Both of them can not serve their purpose without a proper attitude control [24]. Without employing a proper controller, any small noises and errors on its attitude may lead to unwinding behavior [13], [14].

The PD controller plus dynamics compensator are proposed here, which is illustrated in Figure 3.

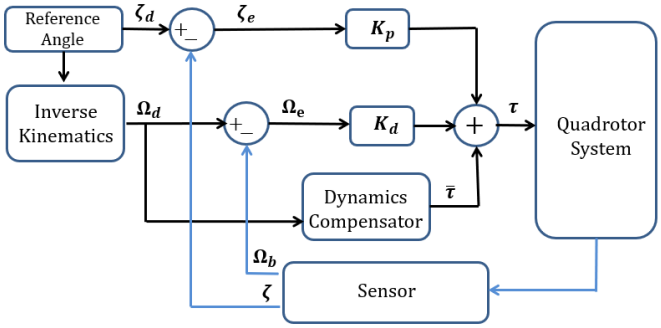

Fig. 3: Design of attitude tracking control

Instead the more sophisticated one, the PD controller is employed to handle the errors caused by parameter uncertainty and the inertia perturbations. It is quite simple yet robust enough to stabilize the attitude of quadrotor [25]–[28].

A. Inverse Kinematics

In the case of trajectory tracking, the vector angle $\boldsymbol{\zeta}$ is changing over time. It indicates that the angular velocity $\boldsymbol{\Omega}_b$ has its own dynamics. Based on Figure 3, $\boldsymbol{\Omega}_d$ is needed to design the PD controller. The equation (14), explains the forward kinematics where the angle rates is affected by the angular velocity. The angle rates in equation (14) can be rewritten as

$$\dot{\boldsymbol{\zeta}} = \sum_{n=0}^{\infty} \frac{(-1)^n \mathrm{B}_n}{n!} \mathrm{ad}_{\boldsymbol{\zeta}}^n (\boldsymbol{\Omega}_b)\,, \quad (31)$$

where B_n is the Bernoulli number.

Suppose we only get the angle rates, the angular velocity remains unknown. Here, by applying the inverse kinematics, we can calculate the angular velocity regard to the angle rates. Thus, the equation holds

$$\widehat{\boldsymbol{\Omega}}_d = \sum_{n=0}^{\infty} \frac{(-1)^n}{(n+1)!} \mathrm{ad}_{\widehat{\boldsymbol{\zeta}}_d}^n \left(\dot{\widehat{\boldsymbol{\zeta}}}_d\right). \quad (32)$$

Generally, adjoint $\mathrm{ad}_{\widehat{\boldsymbol{\zeta}}}\left(\dot{\widehat{\boldsymbol{\zeta}}}\right)$, which works on $\mathfrak{so}(3)$, can be elaborated as follows

$$\begin{aligned}\mathrm{ad}_{\widehat{\boldsymbol{\zeta}}}\left(\dot{\widehat{\boldsymbol{\zeta}}}\right) &= \widehat{\boldsymbol{\zeta}}\,\dot{\widehat{\boldsymbol{\zeta}}} - \dot{\widehat{\boldsymbol{\zeta}}}\,\widehat{\boldsymbol{\zeta}} \\ &= \begin{bmatrix} 0 & \dot{\zeta}_1\zeta_2 - \dot{\zeta}_2\zeta_1 & \dot{\zeta}_1\zeta_3 - \dot{\zeta}_3\zeta_1 \\ \dot{\zeta}_2\zeta_1 - \dot{\zeta}_1\zeta_2 & 0 & \dot{\zeta}_2\zeta_3 - \dot{\zeta}_3\zeta_2 \\ \dot{\zeta}_3\zeta_1 - \dot{\zeta}_1\zeta_3 & \dot{\zeta}_3\zeta_2 - \dot{\zeta}_2\zeta_3 & 0 \end{bmatrix},\end{aligned}$$

The matrix above is a skew-symetric matrix ($\mathfrak{so}(3)$). As explained in section 2, vee operator ($\vee$) recover skew-symmetric matrix into vector form $\left(\mathrm{ad}_{\widehat{\boldsymbol{\zeta}}}\left(\dot{\widehat{\boldsymbol{\zeta}}}\right)\right)^{\vee}$, as follows

$$\begin{aligned}\left(\mathrm{ad}_{\widehat{\boldsymbol{\zeta}}}\left(\dot{\widehat{\boldsymbol{\zeta}}}\right)\right)^{\vee} &= \begin{bmatrix} \zeta_2\dot{\zeta}_3 - \zeta_3\dot{\zeta}_2 \\ \zeta_3\dot{\zeta}_1 - \zeta_1\dot{\zeta}_3 \\ \zeta_1\dot{\zeta}_2 - \zeta_2\dot{\zeta}_1 \end{bmatrix}, \\ &= \begin{bmatrix} 0 & -\zeta_3 & \zeta_2 \\ \zeta_3 & 0 & -\zeta_1 \\ -\zeta_2 & \zeta_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\zeta}_1 \\ \dot{\zeta}_2 \\ \dot{\zeta}_3 \end{bmatrix}, \\ &= \widehat{\boldsymbol{\zeta}}\,\dot{\boldsymbol{\zeta}}. \end{aligned} \quad (33)$$

Therefore, it can be proved that

$$\left(\mathrm{ad}_{\widehat{\zeta}}^{n}\left(\dot{\widehat{\zeta}}\right)\right)^{\vee}=\widehat{\zeta}^{n}\dot{\zeta}. \tag{34}$$

Substitute equation (34) into (32), then we have the inverse kinematics in vector form as follows

$$\mathbf{\Omega}=\sum_{n=0}^{\infty}\frac{(-1)^{n}}{(n+1)!}\,\widehat{\zeta}^{n}\dot{\zeta}. \tag{35}$$

Lets assume

$$\begin{aligned} f(\zeta) &= \sum_{n=0}^{\infty}\frac{1}{(n+1)!}\widehat{\zeta}^{n}, \\ &= \mathbf{I}_{3\times 3}+\frac{1}{2!}\hat{\zeta}+\frac{1}{3!}\widehat{\zeta}^{2}+\frac{1}{4!}\widehat{\zeta}^{3}+\frac{1}{5!}\widehat{\zeta}^{4}+..., \\ &= \mathbf{I}_{3\times 3}+\sum_{n=1}^{\infty}\frac{1}{(2n)!}\widehat{\zeta}^{2n-1}, \\ &\quad +\sum_{n=1}^{\infty}\frac{1}{(2n+1)!}\widehat{\zeta}^{2n}. \end{aligned} \tag{36}$$

The infinitesimal $\sum_{n=1}^{\infty}\frac{1}{(2n)!}\widehat{\zeta}^{2n-1}$ can be elaborated further as

$$\begin{aligned} \sum_{n=1}^{\infty}\frac{1}{(2n)!}\widehat{\zeta}^{2n-1} &= \sum_{n=1}^{\infty}\frac{(-1)^{n-1}}{(2n)!}\|\zeta\|^{2(n-1)}\widehat{\zeta}, \\ &= -\sum_{n=1}^{\infty}\frac{(-1)^{n}}{(2n)!}\|\zeta\|^{2n}\frac{\widehat{\zeta}}{\|\zeta\|^{2}}, \\ &= -\left(\sum_{n=0}^{\infty}\frac{(-1)^{n}}{(2n)!}\|\zeta\|^{2n}\frac{\widehat{\zeta}}{\|\zeta\|^{2}}\right) \\ &\quad +\frac{\widehat{\zeta}}{\|\zeta\|^{2}}, \\ &= (-\cos\|\zeta\|+1)\frac{\widehat{\zeta}}{\|\zeta\|^{2}}. \end{aligned} \tag{37}$$

Meanwhile, infinitesimal $\sum_{n=1}^{\infty}\frac{1}{(2n+1)!}\widehat{\zeta}^{2n}$ can be deduced as

$$\begin{aligned} \sum_{n=1}^{\infty}\frac{1}{(2n+1)!}\widehat{\zeta}^{2n} &= \sum_{n=1}^{\infty}\frac{(-1)^{n-1}}{(2n+1)!}\|\zeta\|^{2(n-1)}\widehat{\zeta}^{2}, \\ &= -\sum_{n=1}^{\infty}\frac{(-1)^{n}}{(2n+1)!}\|\zeta\|^{2n+1}\frac{\widehat{\zeta}^{2}}{\|\zeta\|^{3}}, \\ &= -\left(\sum_{n=0}^{\infty}\frac{(-1)^{n}}{(2n+1)!}\|\zeta\|^{2n+1}\frac{\widehat{\zeta}^{2}}{\|\zeta\|^{3}}\right) \\ &\quad +\frac{\widehat{\zeta}^{2}}{\|\zeta\|^{2}}, \\ &= \left(-\frac{1}{\|\zeta\|}\sin\|\zeta\|+1\right)\frac{\widehat{\zeta}^{2}}{\|\zeta\|^{2}}. \end{aligned} \tag{38}$$

Substitute equation (37) and (38) into (36) then we have

$$\begin{aligned} f(\zeta) &= \mathbf{I}_{3\times 3}+(1-\cos\|\zeta\|)\frac{\widehat{\zeta}}{\|\zeta\|^{2}} \\ &\quad +\left(1-\frac{1}{\|\zeta\|}\sin\|\zeta\|\right)\frac{\widehat{\zeta}^{2}}{\|\zeta\|^{2}}. \end{aligned} \tag{39}$$

As we know that

$$\mathbf{\Omega}=f(-\zeta)\dot{\zeta}, \tag{40}$$

so that, inverse kinematics of quadrotor on exponential coordinate can be designed using equation (40) and (39), as follows

$$\begin{aligned} \mathbf{\Omega}_{d} &= \left(\mathbf{I}_{3\times 3}-(1-\cos(\|\zeta_{d}\|))\frac{\widehat{\zeta}_{d}}{\|\zeta_{d}\|^{2}}\right)\dot{\zeta}_{d} \\ &\quad +\left(\left(1-\frac{1}{\|\zeta_{d}\|}\sin(\|\zeta_{d}\|)\right)\frac{\widehat{\zeta}_{d}^{2}}{\|\zeta_{d}\|^{2}}\right)\dot{\zeta}_{d}. \end{aligned} \tag{41}$$

B. Dynamics Compensator

Dynamics compensator is necessary to calculate the torques, which can be derived from inverse dynamics of our system. The dynamics equation of quadrotor has been stated before in equation (12), and by inversing it we have the dynamics compensator as

$$\bar{\boldsymbol{\tau}}=\mathbf{J}_{0}\dot{\mathbf{\Omega}}_{b}+\widehat{\mathbf{\Omega}}_{b}\mathbf{J}_{0}\mathbf{\Omega}_{b}. \tag{42}$$

C. Proportional-Derivative Control with Linearization on Moving Operating Point

Generally, the PD controller uses the proportional and derivative error term. Hence, it is necessary to explain it first. We will explain it by an example, since this way is more convenient. Let say we need to regulate a variable δ to reach a condition δ_d, we can utilize the controller k as

$$k=-kp\;\delta_{e}-kd\;\dot{\delta}_{e}, \tag{43}$$

where $\delta_e=|\delta-\delta_d|$ is the error term and $\dot{\delta}_e$ is its derivative. Meanwhile, k_p and k_d are the proportional and derivative gain respectively.

Here, in this research, we have two references, ζ_d and $\mathbf{\Omega}_d$, and two error terms, ζ_e and $\mathbf{\Omega}_e$. All of them are vectors in 3-dimensional space.

Since the error term has been elaborated, therefore we can discuss the design of our controller. The proposed controller is a 3-by-6 matrix as

$$\mathbf{K}=\begin{bmatrix}\mathbf{K}_{p} & \mathbf{K}_{d}\end{bmatrix}. \tag{44}$$

It consists of two submatrices, $\mathbf{K}_p$ and $\mathbf{K}_d$. The proportional gain matrix is $\mathbf{K}_p$, and the derivative is $\mathbf{K}_d$. It is getting more complicated to tune its parameters, since both of proportional and derivative gain are 3-by-3 matrices, not scalars.

There are several methods to determine those parameters. Here, we decide to utilize the pole placement. It is simple yet reliable to tune the parameter of PD controller. Unfortunately, it only works in linear system, whereas the quadrotor system is a nonlinear system. It is a good idea to linearize it first. We can consider to linearize it along the references angle.

The second-order equation of quadrotor has been stated before in equation (15), and by taking help of Jacobian method, the linearized one is given as

$$\dot{\mathbf{x}}=\mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{u}, \tag{45}$$

where $\mathbf{x}=\begin{bmatrix}\zeta_{e1} & \zeta_{e2} & \zeta_{e3} & \Omega_{e1} & \Omega_{e2} & \Omega_{e3}\end{bmatrix}^{\mathsf{T}}$ is the state vector and $\mathbf{u}=\mathbf{K}\mathbf{x}$ is the control input.

Meanwhile, $\mathbf{A}$ and $\mathbf{B}$ are 6-by-6 state matrix and 6-by-3 input matrix respectively. We define both of them as

$$\mathbf{A} = \begin{bmatrix} 0_{3\times3} & \mathbf{F}_{3\times3} \\ 0_{3\times3} & \mathbf{G}_{3\times3} \end{bmatrix}, \quad (46)$$

$$\mathbf{B} = \begin{bmatrix} 0_{3\times3} \\ \mathbf{H}_{3\times3} \end{bmatrix}. \quad (47)$$

where $0_{3\times3}$ is 3-by-3 zero matrix, meanwhile their submatrices $\mathbf{F}_{3\times3}$, $\mathbf{G}_{3\times3}$, and $\mathbf{H}_{3\times3}$ can be written as

$$\mathbf{F}_{3\times3} = \mathbf{I}_{3\times3} + \frac{\widehat{\boldsymbol{\zeta}}_d}{2} + \left(1 - \frac{\|\boldsymbol{\zeta}_d\|}{2}\cot\left(\frac{\|\boldsymbol{\zeta}_d\|}{2}\right)\right)\frac{\widehat{\boldsymbol{\zeta}}_d^2}{\|\boldsymbol{\zeta}_d\|^2}, \quad (48)$$

$$\mathbf{G}_{3\times3} = -\mathbf{J}_0^{-1}\widehat{\boldsymbol{\Omega}}_d\mathbf{J}_0, \quad (49)$$

$$\mathbf{H}_{3\times3} = \mathbf{J}_0^{-1}. \quad (50)$$

Apparently, the inertia matrix $\mathbf{J}_0$ is used here, instead of $\mathbf{J}$. It is due to the fact that the controller needs to be designed in nominal condition. Even though, the inertia perturbations may occur later, which is a different case here.

Therefore, the closed-loop system can be written as

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\,\mathbf{x}. \quad (51)$$

Based on equation (51), its characteristics can be denoted as

$$s^2\mathbf{I}_{3\times3} + \left(\mathbf{J}^{-1}\mathbf{K}_d - \mathbf{G}_{3\times3}\right)s\mathbf{I}_{3\times3} + \mathbf{J}^{-1}\mathbf{K}_p\mathbf{F}_{3\times3} = 0. \quad (52)$$

Whereas, the desired characteristics can be defined as

$$s^2\mathbf{I}_{3\times3} + 2\xi\eta s\mathbf{I}_{3\times3} + \eta^2\mathbf{I}_{3\times3} = 0, \quad (53)$$

where ξ and η are the damping ratio and natural frequency of desired closed-loop system consecutively. The desired closed-loop system can be achieved by following controller

$$\mathbf{K}_p = \mathbf{J}_0\eta^2\mathbf{F}_{3\times3}^{-1}, \quad (54)$$

$$\mathbf{K}_d = 2\xi\eta\mathbf{J}_0 + \mathbf{J}_0\mathbf{G}_{3\times3}. \quad (55)$$

Since the damping ratio and the natural frequency are greater than 0, therefore we can choose any positive definite matrices for $\mathbf{K}_p$ and $\mathbf{K}_d$. Eventually, the proposed attitude control system is given as

$$\boldsymbol{\tau} = \mathbf{J}_0\dot{\boldsymbol{\Omega}}_b + \widehat{\boldsymbol{\Omega}}_b\mathbf{J}_0\boldsymbol{\Omega}_b - \mathbf{K}_p\boldsymbol{\zeta}_e - \mathbf{K}_d\boldsymbol{\Omega}_e, \quad (56)$$

where $\mathbf{K}_p, \mathbf{K}_d$ and $\mathbf{J}_0$ are positive definite matrices. Simply, the whole process can be represented briefly as seen in Algorithm 1.

Algorithm 1 Design of attitude tracking controller.

begin
Define the inverse kinematics and the dynamics compensator based as the equation (41) and (42);
do:
Linearize the quadrotor system, see equation (45);
Decide the damping ratio ξ and natural frequency η that satisfy equation (53);
Calculate the gains $\mathbf{K}_p$ and $\mathbf{K}_d$;
end

D. Stability of Closed-Loop System

After elaborating the control scheme in equation (56), in this subsection, we will explain the stability of closed-loop system for attitude control of quadrotor in the sense of Lyapunov. Let us substitute equation (56) into equation (12), therefore we have

$$\mathbf{J}_0\dot{\boldsymbol{\Omega}}_b = -(\boldsymbol{\Omega}_b \times \mathbf{J}_0\boldsymbol{\Omega}_b) + \mathbf{J}_0\dot{\boldsymbol{\Omega}}_d + \widehat{\boldsymbol{\Omega}}_b\mathbf{J}_0\boldsymbol{\Omega}_b - \mathbf{K}_p\boldsymbol{\zeta}_e - \mathbf{K}_d\boldsymbol{\Omega}_e. \quad (57)$$

The closed-loop system satisfies

$$\dot{\boldsymbol{\zeta}}_e = \sum_{n=0}^{\infty}\frac{(-1)^n\mathrm{B}_n}{n!}\left(\mathrm{ad}_{\widehat{\boldsymbol{\zeta}}_e}^n\left(\widehat{\boldsymbol{\Omega}}_e\right)\right)^{\vee} = \boldsymbol{\beta}_{\boldsymbol{\zeta}_e}\boldsymbol{\Omega}_e,$$

$$\mathbf{J}_0\dot{\boldsymbol{\Omega}}_e = -\mathbf{K}_p\boldsymbol{\zeta}_e - \mathbf{K}_d\boldsymbol{\Omega}_e, \quad (58)$$

where $\boldsymbol{\beta}_{\boldsymbol{\zeta}_e}$ can be defined as

$$\boldsymbol{\beta}_{\boldsymbol{\zeta}_e} = \sum_{n=0}^{\infty}\frac{(-1)^n\mathrm{B}_n}{n!}\mathrm{ad}_{\widehat{\boldsymbol{\zeta}}_e^n}. \quad (59)$$

Further explanation can be seen at Bullo [22]

Next, the stability of a nonlinear system can be elaborated in the sense of Lyapunov. Let V be the Lyapunov function and $\dot{V}$ be its derivative. The system can be asserted to be stable, as long as V is positive definite, meanwhile its derivative $\dot{V}$ is negative definite. First, let us define the Lyapunov function as

$$V = \frac{1}{2}\left\langle\boldsymbol{\zeta}_e, \boldsymbol{\zeta}_e\right\rangle + \frac{1}{2}\left\langle\boldsymbol{\Omega}_e, \mathbf{K}_p^{-1}\mathbf{J}_0\boldsymbol{\Omega}_e\right\rangle + c\left\langle\boldsymbol{\zeta}_e, \boldsymbol{\Omega}_e\right\rangle. \quad (60)$$

The equation (60) can be rewriten into quadratic form. In order to do so, let the matrix $\mathbf{P}$ such as

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_{3\times3} & c\mathbf{I}_{3\times3} \\ c\mathbf{I}_{3\times3} & \mathbf{K}_p^{-1}\mathbf{J}_0 \end{bmatrix}, \quad (61)$$

and the vector $\boldsymbol{\gamma}_e$ be

$$\boldsymbol{\gamma}_e = \begin{bmatrix} \boldsymbol{\zeta}_e \\ \boldsymbol{\Omega}_e \end{bmatrix}. \quad (62)$$

Hence, the Lyapunov function can be rewritten into quadratic form as

$$V = \frac{1}{2}\boldsymbol{\gamma}_e^T\mathbf{P}\boldsymbol{\gamma}_e. \quad (63)$$

Based on equation (60), the derivative of Lyapunov function can be denoted as

$$\begin{aligned}
\dot{V} &= \left\langle\boldsymbol{\zeta}_e, \dot{\boldsymbol{\zeta}}_e\right\rangle + \left\langle\boldsymbol{\Omega}_e, \mathbf{K}_p^{-1}\mathbf{J}_0\dot{\boldsymbol{\Omega}}_e\right\rangle \\
&\quad + c\left\langle\boldsymbol{\beta}_{\boldsymbol{\zeta}_e}, \boldsymbol{\Omega}_e\right\rangle + c\left\langle\boldsymbol{\zeta}_e, \dot{\boldsymbol{\Omega}}_e\right\rangle, \\
&\leq \left\langle\boldsymbol{\zeta}_e, \boldsymbol{\Omega}_e\right\rangle + \left\langle\boldsymbol{\Omega}_e, \mathbf{K}_p^{-1}\mathbf{J}_0\dot{\boldsymbol{\Omega}}_e\right\rangle \\
&\quad + c\left\langle\boldsymbol{\Omega}_e, \boldsymbol{\Omega}_e\right\rangle + c\left\langle\boldsymbol{\zeta}_e, \dot{\boldsymbol{\Omega}}_e\right\rangle, \\
&\leq \left\langle\boldsymbol{\zeta}_e, \boldsymbol{\Omega}_e\right\rangle + \left\langle\boldsymbol{\Omega}_e, \mathbf{K}_p^{-1}\left(-\mathbf{K}_p\boldsymbol{\zeta}_e - \mathbf{K}_d\boldsymbol{\Omega}_e\right)\right\rangle \\
&\quad + c\left\langle\boldsymbol{\Omega}_e, \boldsymbol{\Omega}_e\right\rangle + c\left\langle\boldsymbol{\zeta}_e, -\mathbf{J}_0^{-1}\mathbf{K}_p\boldsymbol{\zeta}_e - \mathbf{J}_0^{-1}\mathbf{K}_d\boldsymbol{\Omega}_e\right\rangle, \\
&\leq -\left\langle\boldsymbol{\Omega}_e, \mathbf{K}_p^{-1}\mathbf{K}_d\boldsymbol{\Omega}_e\right\rangle + c\left\langle\boldsymbol{\Omega}_e, \boldsymbol{\Omega}_e\right\rangle \\
&\quad - c\left\langle\boldsymbol{\zeta}_e, \mathbf{J}_0^{-1}\mathbf{K}_p\boldsymbol{\zeta}_e\right\rangle - c\left\langle\boldsymbol{\zeta}_e, \mathbf{J}_0^{-1}\mathbf{K}_d\boldsymbol{\Omega}_e\right\rangle. \quad (64)
\end{aligned}$$

It should be noted that the first term of equation (64) is simplified by assumption, such that

$$\left\langle \zeta_e, \dot{\zeta}_e \right\rangle = \langle \zeta_e, \mathbf{\Omega}_e \rangle , \tag{65}$$

with the upper bound be

$$\left\langle \boldsymbol{\beta}_{\zeta_e}, \mathbf{\Omega}_e \right\rangle = \langle \mathbf{\Omega}_e, \mathbf{\Omega}_e \rangle , \tag{66}$$

those are well described by Bullo [22]. Now, let us define the matrix $\mathbf{Q}$ such as

$$\mathbf{Q} = \begin{bmatrix} c\mathbf{J}_0^{-1}\mathbf{K}_p & \frac{c}{2}\mathbf{J}_0\mathbf{K}_d \\ \frac{c}{2}\mathbf{J}_0\mathbf{K}_d & \mathbf{K}_p^{-1}\mathbf{K}_d - c\mathbf{I}_{3\times 3} \end{bmatrix}. \tag{67}$$

Hence, the derivative of Lyapunov function, ($\dot{V}$), can be simplified as

$$\dot{V} \leq -\frac{1}{2}\boldsymbol{\gamma}_e^T \mathbf{Q} \boldsymbol{\gamma}_e. \tag{68}$$

As we stated before, that the system is considered stable in the sense of Lyapunov as long as V be positive definite and $\dot{V}$ be negative definite. In order to satisfy the condition, therefore $\mathbf{P}$ and $\mathbf{Q}$, both of them should be made positive definite matrices. It appears that the constant c is significant to determine either $\mathbf{P}$ and $\mathbf{Q}$ be positive or negative definite matrices. Both of them can be made positive definite matrices by assigning a very small positive constant. Let say $c = 0.0001$, hence it comes to a valid Lyapunov function. Further details can be seen in Sandiwan [10].

V. Simulation and Analysis

Before stepping any further, we have to determine some parameters. The inertia matrix of the quadrotor is the first in our list, by taking reference from Pounds [29], we have

$$\mathbf{J}_0 = \begin{bmatrix} 0.082 & 0 & 0 \\ 0 & 0.0845 & 0 \\ 0 & 0 & 0.1377 \end{bmatrix}.$$

Next, we have $\xi = 1$ and $\eta = 8\,\mathrm{rad/second}$. They are needed to determine the desired characteristics equation of our system. Meanwhile, the gains $\mathbf{K}_p$ and $\mathbf{K}_d$ are changing over time with respect to its dynamics.

A. Simulation 1: Attitude Tracking on Nominal Condition

The first simulation is conducted on nominal condition. It means, there is inertia perturbation due to the payload applied on the quadrotor. Here, our proposed control need to track the angular velocity and in the same time regulate the attitude to its references. The initial conditions for $\boldsymbol{\zeta}_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}$ and $\mathbf{\Omega}_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}$. We can see the results in Figure 4-7.

Figure 4 shows that the angular velocity can follow up to the references well. The deviation can only be noticed easily during short periode of time, which is when the simulation started. Implicitly, if the angular velocity can be tracked, therefore it goes the same way for the angle vector. The angles ζ_1, ζ_2, and ζ_3 in Figure 5 represents the attitude of quadrotor in 3-dimensional space, respectively in $x-$, $y-$, and $z-$axis. It takes less than 1 second for them to converge. Furthermore, even though it is relatively small, the control torque in Figure 6 is capable to makes the system work properly.

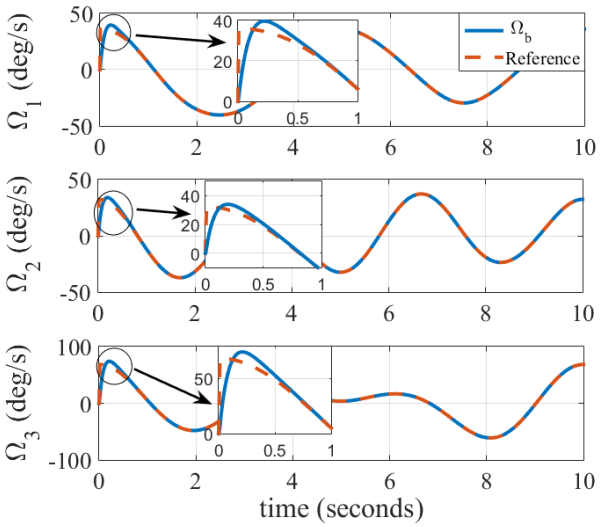

Fig. 4: The angular velocities Ω_1, Ω_2, and Ω_3 track their references.

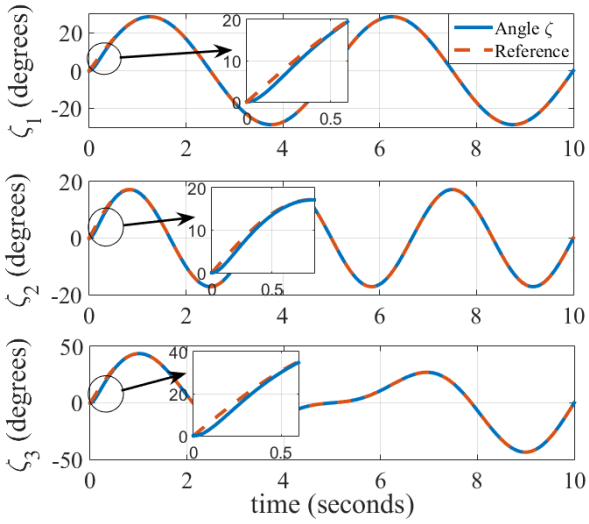

Fig. 5: The angles ζ_1, ζ_2, and ζ_3 track their references.

Since the representation of exponential coordinate is not so direct, we need to define the error function to evaluate the performance of attitude tracking. Error function is defined as follows [30]

$$\mathsf{e}(\mathbf{R_d}, \mathbf{R}) = 2 - \sqrt{1 + \mathrm{Tr}(\mathbf{R_d}^{\mathsf{T}}\mathbf{R})} \tag{69}$$

where $\mathbf{R_d}$ and $\mathbf{R}$ are desired rotation matrix and current rotation matrix repectively. The error rotation matrix can be denoted as $\mathbf{R_e} = \mathbf{R_d}^{\mathsf{T}}\mathbf{R}$, while $\mathrm{Tr}(\mathbf{R_e})$ is its trace. The trace of error rotation matrix is bounded by $-1 \leq \mathrm{Tr}(\mathbf{R_e}) \leq 3$. The error function can be minimized, $\mathsf{e}(\mathbf{R_d}, \mathbf{R}) = 0$, if and only if the conditions for $\mathbf{R_d} = \mathbf{R}$ are met. It should be noted that if the error of attitude tracking is increasing, it will lead to a larger error function $\mathsf{e}(\mathbf{R_d}, \mathbf{R})$ [30].

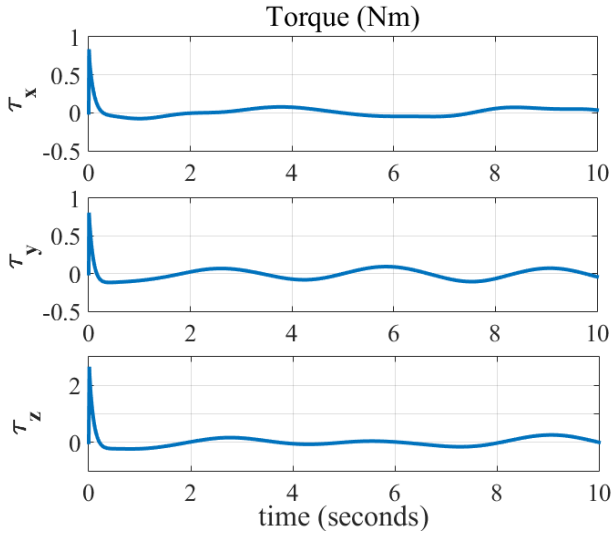

Fig. 6: Torque control generated by our proposed controller.

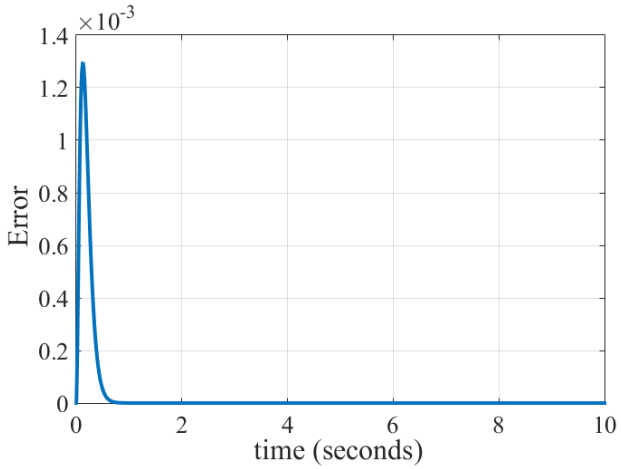

Fig. 7: Error function for attitude tracking in simulation 1.

Figure 7 shows the value of error function when the proposed controller is implemented. The error is about 0.00013 at the begining of simulation then it decreases quickly. It can be seen that the error reaches 0 just few seconds, right after the start of simulation. Those results verify that the proposed PD controller plus dynamics compensator can be used to track the attitude and angular velocity in the nominal condition.

B. Simulation 2: Eminence of SO(3)-based Controller

In order to keep the objectivity, here, we compare our controller and RPY-based controller by Mahmoud [31]. Both controllers need to track the references while under singularity. The angle reference in the $y-$axis oscillates about $90°$ for both of them. We can see the results for the RPY-based controller and its references in Figure 8 and 9.

If we look onto Figure 8, we only see the references as straight lines. Actually, it is not correct, the references are oscillating as illustrated in Figure 9. Moreover, it goes as expected, the RPY-based controller fails to track the attitude. The RPY-based controller suffers from singularity when the pitch angle θ turns around $90°$.

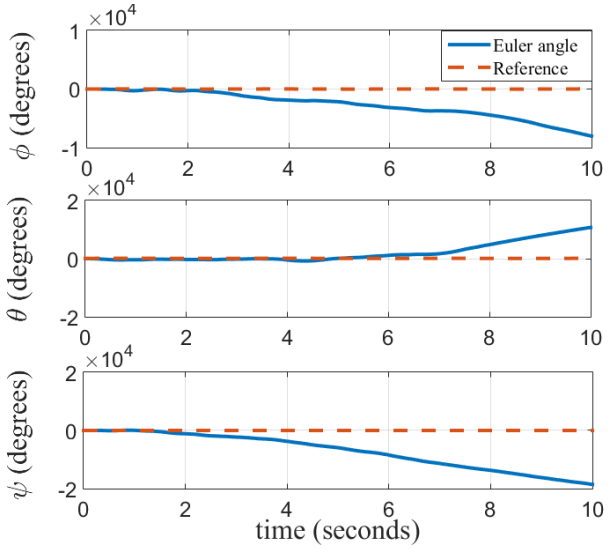

Fig. 8: Attitude tracking on Euler angles representation.

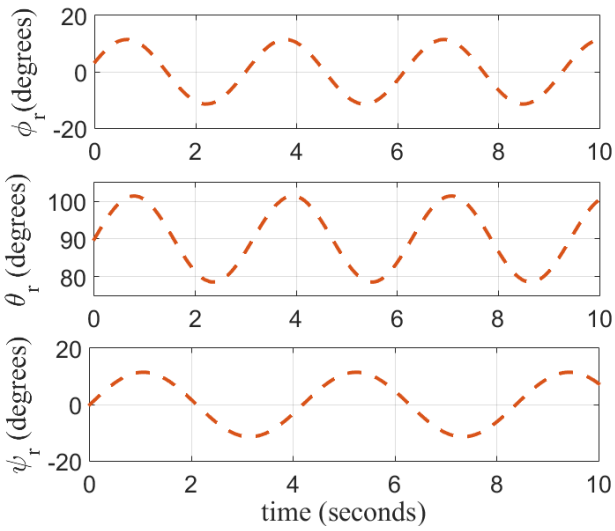

Fig. 9: The reference angles for RPY-based controller.

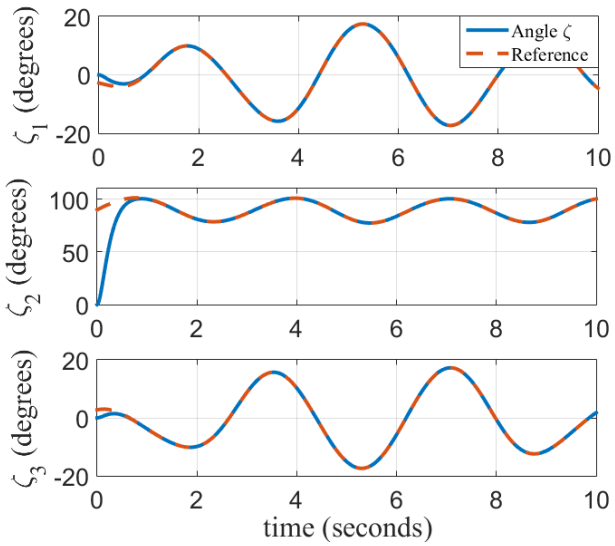

Fig. 10: Attitude tracking on $\mathrm{SO}(3)$.

The phenomenon may lead to the instability of the quadrotor, that makes the angles deviate very far from their references. On the contrary, the $\mathrm{SO}(3)$-based controller prevails

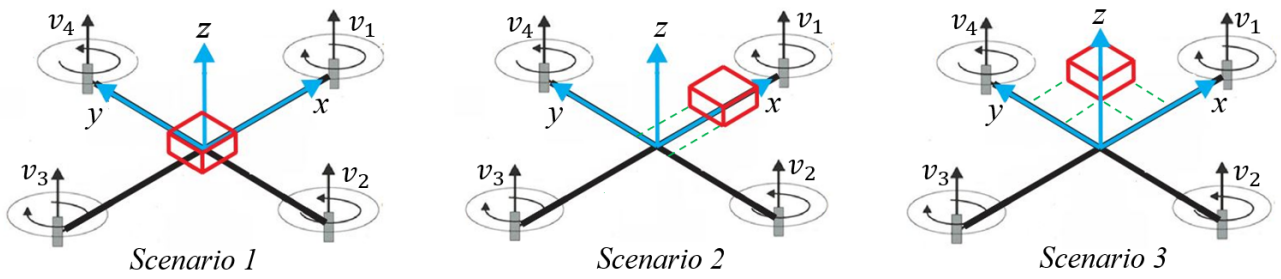

Fig. 11: Laying positions of the payload

to track the references, as exposed in Figure 10. It shows that the SO(3)-based controller does not suffer from any singularity despite after ζ_2 reaching 90°. Moreover, it can be seen that the errors are relatively small. Based on those results, we can assert that the SO(3)-based controller is better than the RPY-based controller in terms of reliability against singularity.

C. Simulation 3: Attitude Tracking under Inertia Perturbation

After evaluating the SO(3)-based PD controller plus dynamics compensator in the nominal condition, here the inertia perturbation will be applied. This way, we can verify the robustness of our proposed PD controller against inertia perturbation. It has been mentioned before that the quadrotor is under inertia perturbation due to the payload. The thing that has not been explained before is its laying position on the quadrotor.

Supposedly, different laying positions yield different inertia perturbations to the quadrotor. Hence, we take several scenarios based on the laying position of the payload. There are three scenarios, where each scenario has different laying position of the payload. Those laying positions can be seen in Figure 11.

In Scenario 1, we have the payloads attached in the centre of gravity (CoG) of the quadrotor. Meanwhile, in Scenario 2, the payload is laying on near the rotor number 1. It is located on the $x-$axis of the body frame of the quadrotor. Next, Scenario 3, the payload is attached in the body of quadrotor, which is between rotor number 1 and 4. The laying positions on Table I are in meters, while its mass is $0.5\,\mathrm{kg}$.

TABLE I: Laying Position of the Payload

Scenario 1					
a	b	c	d	e	f
-0.05	0.05	-0.1	0.1	0	0.2
Scenario 2					
a	b	c	d	e	f
0.4	0.5	-0.1	0.1	0	0.2
Scenario 3					
a	b	c	d	e	f
-0.05	0.05	0.3	0.5	0	0.2

We can calculate the inertia perturbations due to the payload by substituting the data on Table I into equations in Subsection III-B, it yields

$$\Delta\mathbf{J}_1 = \begin{bmatrix} 0.0083 & 0 & 0 \\ 0 & 0.0071 & 0 \\ 0 & 0 & 0.0021 \end{bmatrix},$$

$$\Delta\mathbf{J}_2 = \begin{bmatrix} 0.0083 & 0 & -0.0225 \\ 0 & 0.1083 & 0 \\ -0.0225 & 0 & 0.1033 \end{bmatrix},$$

$$\Delta\mathbf{J}_3 = \begin{bmatrix} 0.0883 & -0.0900 & -0.0225 \\ -0.0900 & 0.1083 & -0.0200 \\ -0.0225 & -0.0200 & 0.1833 \end{bmatrix},$$

where respectively, $\Delta\mathbf{J}_1$, $\Delta\mathbf{J}_2$, and $\Delta\mathbf{J}_3$ are the inertia perturbations in Scenario 1, 2, and 3. The results are exposed in Figure 12, 13, and 14.

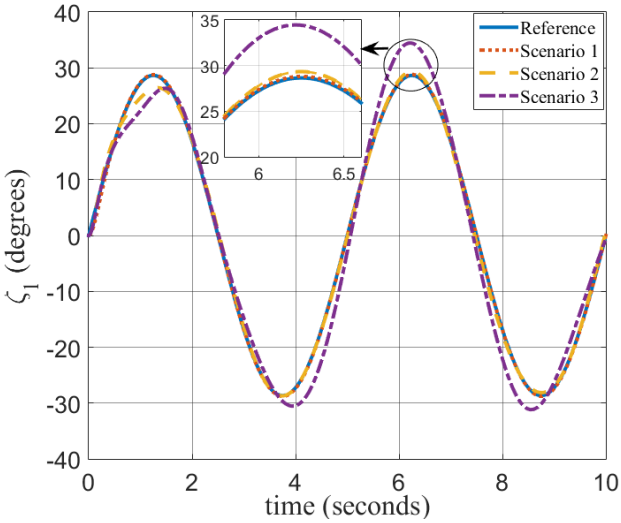

Fig. 12: Attitude tracking on ζ_1 under various inertia perturbations

It can be verified from Figure 12, 13, and 14, that the errors are quite small. Despite the inertia perturbations, the references on ζ_1, ζ_2, and ζ_3 can be tracked almost perfectly. Moreover, just how good the attitude tracking works in each scenario can be evaluated by utilizing the integral of absolute error or simply IAE as

$$\mathrm{IAE} = \int |\mathrm{e}|dt, \tag{70}$$

where $|\mathrm{e}|$ is absolute value of the error term.

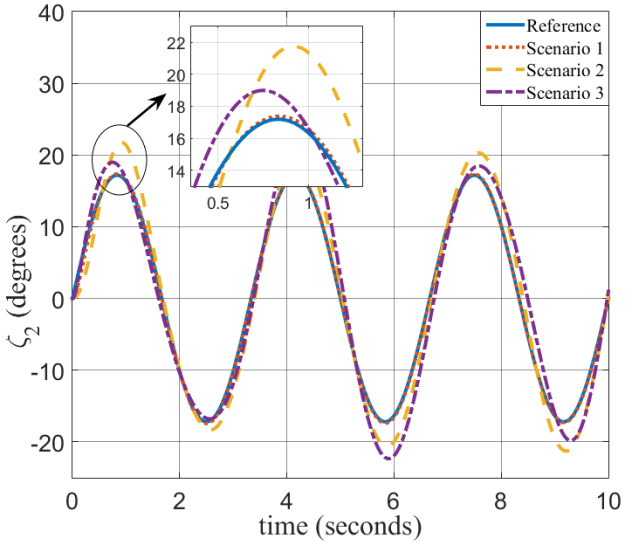

Fig. 13: Attitude tracking on ζ_2 under various inertia perturbations

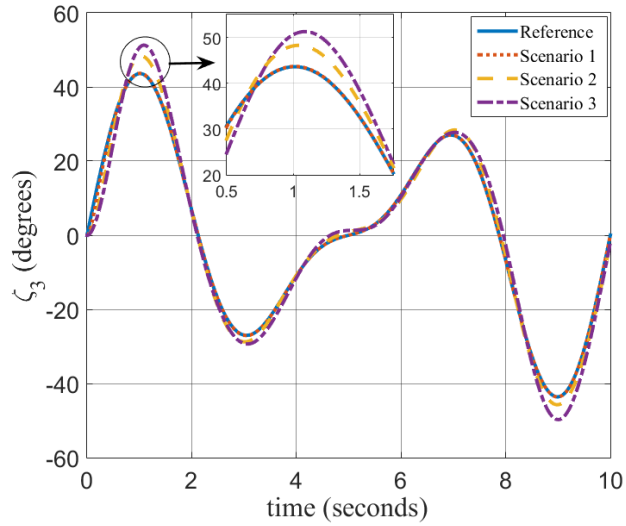

Fig. 14: Attitude tracking on ζ_3 under various inertia perturbations

The comparison of errors in each scenario can be seen in Figure 15. Meanwhile, the information about IAE in all scenarios can be seen in Table II.

TABLE II: Integral of Absolute Error of Attitude Tracking in Simulation 3

Integral of Absolute Error		
Scenario 1	**Scenario 2**	**Scenario 3**
3.24×10^{-3}	15.44×10^{-3}	32.19×10^{-3}

Based on Table II, we can see that the Scenario 3 has the largest error. As stated before that any disturbace or parameter uncertainty will affect the system, the smaller the disturbance, the smaller the error tracking, and vice versa.

Since the Scenario 3 has the largest inertia perturbation, then it is not a surprse that it has the largest error compared to the other scenarios. The largest error is not more than 0.003 degrees as can be seen in Figure 15. Another thing that can be addressed here is the fact that the changes of

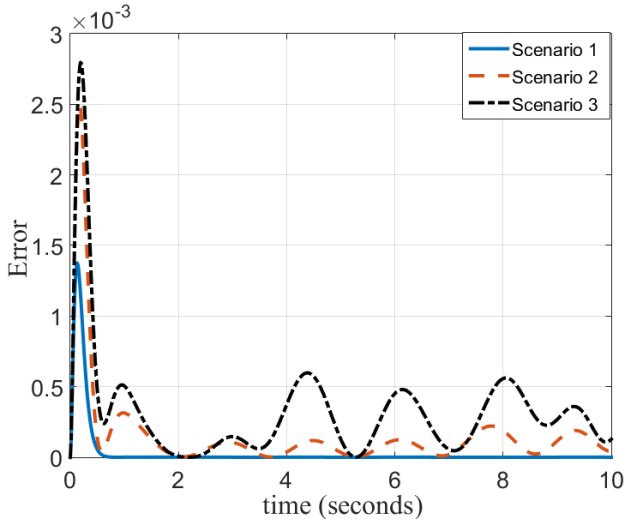

Fig. 15: Attitude tracking error in simulation 3

directions, where the angles are rising then turning down or vice versa, will lead to the slightly larger error. Those results verify that our controller perform well in dealing with inertia perturbations due to those payloads.

D. Simulation 4: Attitude Tracking under Changes of Centre of Gravity and Modelling Error

Simulation 4 is conducted to verify the robustness of the proposed PD controller againts parameter uncertainty both externally and internally. Here the term external parameter uncertainty refers to the addition of payload, which in this simulation is assumed to be liquid. On the other hand, the term internal parameter uncertainty refers to the errors that arise due to the inaccuracies in the system modelling.

Furthermore, a modelling errors takes place in this simulation. Modelling errors are expressed by giving

$$\mathrm{P}_{\mathrm{control}} = 0.8\ \mathrm{P}_{\mathrm{quadrotor}}, \tag{71}$$

where the term $\mathrm{P}_{\mathrm{control}}$ refers to the parameters used in the proposed control, meanwhile $\mathrm{P}_{\mathrm{quadrotor}}$ refers to parameters used in the actual plant. It indicates that the parameters on the controller are slightly different than the actual one.

In reality, the payload may come in various forms. As an instance it may be liquid. The problem itself is quite interesting, since any little movements may affect the CoG of the payload. Here, we consider it into the simulation. Physically, the container is attached right in the centre of quadrotor. It containts liquid which its CoG changes over time as shown in Figure 16.

We can see from Figure 16, where the $x-$ and $y-$axis are representing the coordinate of the CoG, meanwhile, the $z-$axis gives us information of the time. At the start of the simulation, the CoG is located right in the center, which are at $x = 0$ and $y = 0$. Then, it changes over time for 10 seconds. Even more, as time goes by, it deviates further.

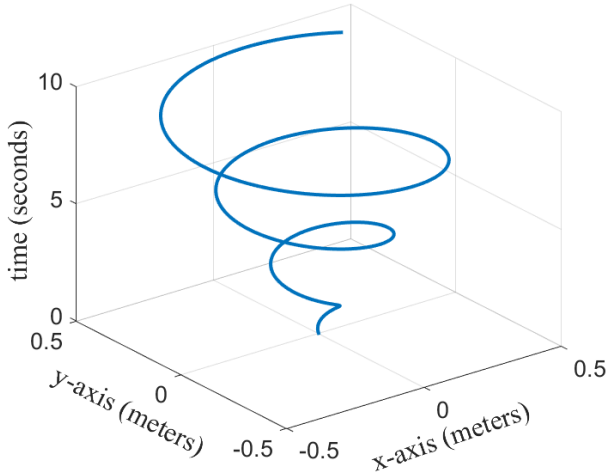

Fig. 16: Changes of centre of gravity

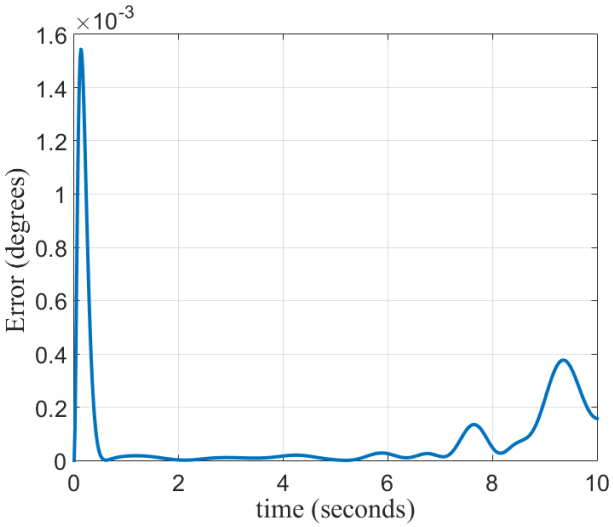

Fig. 18: Error of attitude tracking in simulation 4

Despite the CoG changes over time and modeling error takes place, the angle ζ_1, ζ_2, and ζ_3 track almost perfectly to their references as shown in Figure 17. It is worth to mention that when the CoG deviates further, it may leads to greater errors on tracking the references.

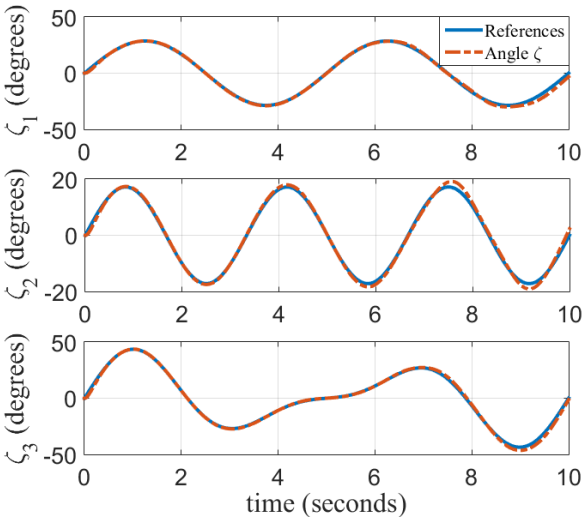

Fig. 17: Attitude tracking under dynamics inertia perturbation and modelling error

We can analyze it further by plotting how the errors evolve over the time as can be seen in Figure 18.

Based on Figure 18, it shows that the attitude tracking error does not exceed 0.0016. Eventually, at the end of the simulation, the error is less than 0.0005. Eventhough the CoG deviates considerably, the references still can be tracked. It implies that those errors are not significant to harm the system. Overall, the controller is robust enough to overcome modeling errors and changes of CoG due to the payload.

VI. Conclusion

The attitude tracking control of transporting quadrotor on $\mathrm{SO}(3)$ group with linearization on moving operating points yields satisfying results. We also compare it with the RPY-based controller. It offers better reliability than the RPY-based controller against singularity. Furthermore, despite under inertia perturbations due to the payload, the controller prevails to track the references. Lastly, the controller is temped by introducing the time-varying CoG and modeling error. Even though the CoG of the payload is changing over time and modeling error takes place, the controller can keep it up to regulate the attitude of the quadrotor.

References

[1] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 20–32, Sept 2012.

[2] H. Lim, J. Park, D. Lee, and H. J. Kim, "Build your own quadrotor: Open-source projects on unmanned aerial vehicles," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 33–45, Sept 2012.

[3] K. M. Lee, "Design of a smart phone application controlling agricultural watering system with a drone," in *Lecture Notes in Engineering and Computer Science: Proceedings of the World Congress on Engineering and Computer Science 2018*, vol. I, 23-25 October 2018, pp. 30–32.

[4] S. Kawabata, K. Nohara, J. H. Lee, H. Suzuki, T. Takiguchi, O. S. Park, and S. Okamoto, "Development of drone system embedded with multiple distance sensors for inspection task of social infrastructure," in *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2018*, vol. II, 14-16 March 2018, pp. 804–808.

[5] K. Nohara, S. Kawabata, J. H. Lee, S. Okamoto, H. Suzuki, T. Takiguchi, and O. S. Park, "Development of drone system embedded with multiple distance sensors for inspection task of social infrastructure," in *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2018*, vol. II, 14-16 March 2018, pp. 542–546.

[6] M. Kan, S. Okamoto, and J. H. Lee, "Development of drone capable of autonomous flight using gps," in *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2018*, vol. II, 14-16 March 2018, pp. 665–669.

[7] D. A. Umarhadi, P. Danoedoro, P. Wicaksono, P. Widayani, W. Nurbandi, and A. Juniansah, "The comparison of canopy density measurement using uav and hemispherical photography for remote sensing based mapping," in *2018 4th International Conference on Science and Technology (ICST)*, 2018, pp. 1–5.

[8] C. A. Rokhmana and M. U. Nuha, "Modification of pocket camera as a sensor for uav-based plant health detection system," in *2019 5th International Conference on Science and Technology (ICST)*, 2019, pp. 1–6.

[9] R. Panuntun, O. Wahyunggoro, S. Herdjunanto, A. Rafsanzani, and N. Setiawan, "Networked control system in quadrotor altitude control with time delay compensation," *Journal of Physics: Conference Series*, vol. 1577, p. 012031, July 2020.

[10] A. P. Sandiwan, A. Cahyadi, and S. Herdjunanto, "Robust proportional-derivative control on so(3) with disturbance compensation for quadrotor uav," *International Journal of Control, Automation and Systems*, vol. 15, no. 5, pp. 2329–2342, Oct 2017.
[11] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch, "Rigid-body attitude control," *IEEE Control Systems*, vol. 31, no. 3, pp. 30–51, June 2011.
[12] M. T. Hussein and M. N. Nemah, "Modeling and control of quadrotor systems," in *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*, Oct 2015, pp. 725–730.
[13] T. Lee, "Exponential stability of an attitude tracking control system on so(3) for large-angle rotational maneuvers," *Systems & Control Letters*, vol. 61, no. 1, pp. 231 – 237, 2012.
[14] T.-H. Wu and T. Lee, "Angular velocity observer for attitude tracking on so(3) with the separation property," *International Journal of Control, Automation and Systems*, vol. 14, no. 5, pp. 1289–1298, Oct 2016.
[15] B. C. Min, J. H. Hong, and E. T. Matson, "Adaptive robust control (arc) for an altitude control of a quadrotor type uav carrying an unknown payloads," in *2011 11th International Conference on Control, Automation and Systems*, Oct 2011, pp. 1147–1151.
[16] Z. Wang, S. Liang, T. Liu, and L. Zhang, "Robust adaptive attitude control of quadrotors with load uncertainties," in *2016 Chinese Control and Decision Conference (CCDC)*, May 2016, pp. 5962–5967.
[17] G. N. P. Pratama, A. I. Cahyadi, and S. Herdjunanto, "Robust proportional-derivative control on so(3) for transporting quadrotor with load uncertainties," *IAENG International Journal of Computer Science*, vol. 45, no. 2, pp. 275–284, 2018.
[18] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1994.
[19] N. Setiawan, S. Herdjunanto, and A. I. Cahyadi, "Proportional-derivative control for quadrotor stabilization under inertia perturbation," in *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)*, 2019, pp. 402–406.
[20] A. H. Ginting, O. Wahyunggoro, A. I. Cahyadi, and M. Faris, "Stabilization of quadrotor on so(3) with robust proportional-derivative to compensate cog," in *2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2019, pp. 1–6.
[21] A. H. Ginting, O. Wahyunggoro, and A. I. Cahyadi, "Robust proportional-derivative control on so(3) to compensate the unknown center of gravity of quadrotor," *Engineering Letters*, vol. 28, no. 2, pp. 359–370, 2020.
[22] F. Bullo and R. M. Murray, "Proportional derivative (pd) control on the euclidean group," in *3rd European Control Conference*, 1995, pp. 1091–1097.
[23] N. Cao and A. F. Lynch, "Inner-outer loop control for quadrotor uavs with input and state constraints," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1797–1804, Sep. 2016.
[24] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5420–5425.
[25] A. Tayebi and S. McGilvray, "Attitude stabilization of a vtol quadrotor aircraft," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 562–571, May 2006.
[26] I. C. Dikmen, A. Arisoy, and H. Temeltas, "Attitude control of a quadrotor," in *2009 4th International Conference on Recent Advances in Space Technologies*, June 2009, pp. 722–727.
[27] H. Liu, Y. Bai, G. Lu, and Y. Zhong, "Brief paper - robust attitude control of uncertain quadrotors," *IET Control Theory Applications*, vol. 7, no. 11, pp. 1583–1589, July 2013.
[28] H. Tnunay, M. Q. Abdurrohman, Y. Nugroho, R. Inovan, A. Cahyadi, and Y. Yamamoto, "Auto-tuning quadcopter using loop shaping," in *2013 International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, 2013, pp. 111–115.
[29] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a large quadrotor robot," *Control Engineering Practice*, vol. 18, no. 7, pp. 692–699, July 2010.
[30] T. Lee, "Geometric tracking control of the attitude dynamics of a rigid body on so(3)," in *Proceedings of the 2011 American Control Conference*, 2011, pp. 1200–1205.
[31] O. E. Mahmoud, M. R. Roman, and J. F. Nasry, "Linear and nonlinear stabilizing control of quadrotor uav," in *2014 International Conference on Engineering and Technology (ICET)*, April 2014, pp. 1–8.

Nurman Setiawan obtained his master degree from Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Indonesia. He is curently lecturer in Department of Electrical Engineering Education, Universitas Negeri Yogyakarta, Indonesia. His research interest involves control system and robotics.

Gilang Nugraha Putu Pratama is an engineer in Beehive Drones. obtained his degree from Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Indonesia. His research interest involves control system and robotics. He is also an IAENG Member since 2018.

Hendra Ari Winarno obtained his master in Electrical Engineering from Universitas Gadjah Mada Indonesia. He is curently lecturer in Department of Electrical Engineering, Universitas Muhammadiyah Gresik, Indonesia. His research areas involves robotics and biomedical engineering.

Samiadji Herdjunanto obtained his bachelor degree from Department of Electrical Engineering, Faculty of Engineering, Universitas Gadjah Mada. Later he got his master degree from Ohio State University, USA, and doctoral degree from Universitas Gadjah Mada. He is curently lecturer in Department of Electrical Engineering, Universitas Gadjah Mada, Indonesia. His research areas involves control, signal processing, fault detection, isolation and reconstruction.

Adha Imam Cahyadi is an IAENG Member. He obtained his bachelor degree from Department of Electrical Engineering, Faculty of Engineering, Universitas Gadjah Mada in 2002. Later he got his master in Control Engineering from KMITL in 2005, Thailand, and Doctor of Engineering from Tokai University Japan in 2008. He is curently lecturer in Department of Electrical Engineering, Universitas Gadjah Mada, Indonesia. His research areas involves mechanical control systems, telemanipulation systems, and Unmanned Aerial Vehicles.