

TRAJECTORY CONTROL FOR A QUADCOPTER

Konstantin Metodiev

Space Research and Technology Institute – Bulgarian Academy of Sciences
e-mail: komet@space.bas.bg

Keywords: Quadcopter, PID controller, GNU Octave

Abstract: This article presents a mathematical framework for modeling the spatial motion of a quadcopter. The model incorporates a proportional-derivative (PD) controller to maintain the quadcopter's attitude and make it follow an initially assigned path. A system of first-order ordinary differential equations (ODEs) with constant coefficients describes both linear and angular momenta conservation. To solve these equations, the explicit Dormand-Prince method of order 4 with adaptive time stepping, implemented via the ode45 function in GNU Octave, is employed. The algorithm is well-suited for non-stiff ODE systems. The study further analyzes the performance of the PD controller, presenting the simulation results graphically and discussing the effectiveness of the control strategy in stabilizing the quadcopter's motion.

ТРАЕКТОРНО УПРАВЛЕНИЕ НА КВАДРОКОПТЕР

Константин Методиев

Институт за космически изследвания и технологии – Българска академия на науките
e-mail: komet@space.bas.bg

Ключови думи: Квадрокоптер, ПИД регулатор, GNU Octave

Резюме: Тази статия представя математически модел на пространственото движение на квадрокоптер. Моделът включва пропорционално-диференциращ (PD) регулатор, който поддържа ориентацията на квадрокоптера в пространството и го заставя да следва първоначално зададен път. Система от обикновени диференциални уравнения (ОДУ) от първи ред с постоянни коефициенти описва съхранението на количеството на движение и кинетичния момент. За решаването на тези уравнения се използва явен метод на Дорманд-Принс от ред 4 с адаптивна стъпка на интегриране, реализиран чрез функцията ode45 в GNU Octave. Алгоритъмът е подходящ за не-твърди системи ОДУ. Изследването допълнително анализира производителността на регулатора, представяйки резултатите от симулацията графично и обсъждайки ефективността на стратегията за управление при стабилизиране на движението на квадрокоптера.

Introduction

Quadcopter trajectory control requires guiding the quadcopter along a desired path or trajectory with precision and stability. The control process includes various control strategies and algorithms designed to manage the quadcopter's position, velocity, and orientation over time. For example, PID (Proportional-Integral-Derivative) control is a widely used feedback control strategy that can be effectively applied to a quadcopter's trajectory tracking.

The quadcopter's flight involves controlling its position (x, y, z) and orientation (roll, pitch, yaw) in space incessantly. When following a predefined trajectory, the control system minimizes the error between desired and actual position and orientation of the quadcopter. In the current project, two proportional-derivative controllers are implemented: one controls the orientation, while the other controls the position of the aerial vehicle. A proportional-derivative controller for quadcopter attitude has already been theoretically developed by Luukkonen in the monograph [1]. Regrettably, there is no source code available for public access. The current project sole purpose is to develop a source code in GNU Octave to examine the controller capabilities and propose improvements if feasible.

Materials and Methods

The body frame of reference (fixed to the quadcopter) provides a way to describe position, orientation, and motion of points or features relative to the body itself, rather than a fixed external point. In the current study case, the reference frame origin is placed at the quadcopter mass center. The axes are aligned with the quadcopter structure depending on the flight configuration: either “x” or “+,” as shown in Fig. 1. In the current study, the flight configuration “x” is solely considered.



Fig. 1. Inertial and body reference frames, flight configuration “x”, a simplified design [2]

Based on the symbolic conventions typically adopted in quadcopter motion analysis (as illustrated in Fig. 1), the state vector in the body reference frame is usually expressed as

$$(1) \quad \|\mathbf{v} \quad \boldsymbol{\omega}\|^T = \|u \quad v \quad w \quad p \quad q \quad r\|^T$$

where $\mathbf{v} = [u, v, w]^T$ are linear velocities, $\boldsymbol{\omega} = [p, q, r]^T$ are angular rates. The first-order system of ordinary differential equations describing the quadcopter's motion in body reference frame reveals conservation of both linear and angular momentum of the mechanical system as follows:

$$(2) \quad \begin{bmatrix} m\mathbf{I}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega} \times (m\mathbf{v}) \\ \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{bmatrix}$$

where $\mathbf{I}_{3,3}$ is a 3 by 3 identity matrix, $\mathbf{0}_{3,3}$ is a 3 by 3 zero matrix, $m = \text{const}$ stands for the quadcopter mass, $\mathbf{f}$ and $\boldsymbol{\tau}$ denote externally applied forces and torques respectively. It is admissible to simplify the inertia tensor $\mathbf{I}$ further due to symmetry

$$(3) \quad \mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

The inverse of a diagonal matrix might be found after replacing the main diagonal elements with their reciprocals. As regards vector $[\boldsymbol{\omega} \times (m\mathbf{v})]; [\boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega})]^T$, it might be alternatively computed by multiplying Coriolis centripetal matrix and the state vector as follows:

$$(4) \quad \begin{bmatrix} \boldsymbol{\omega} \times (m\mathbf{v}) \\ \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) \end{bmatrix} = \begin{bmatrix} [m\boldsymbol{\omega}]_{\times,3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & [-\mathbf{I}\boldsymbol{\omega}]_{\times,3,3} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix}$$

where symbol $[\cdot]_{\times,3,3}$ denotes a skew symmetric matrix, for example

$$(5) \quad [m\boldsymbol{\omega}]_{\times,3,3} = m \cdot \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}; \quad [-\mathbf{I}\boldsymbol{\omega}]_{\times,3,3} = -1 \cdot \begin{bmatrix} 0 & -I_{zz}r & I_{yy}q \\ I_{zz}r & 0 & -I_{xx}p \\ -I_{yy}q & I_{xx}p & 0 \end{bmatrix}$$

Finally, the system first order ordinary differential equations with constant coefficients describing the quadcopter motion in body frame of reference assumes the conclusive form

$$(6) \quad \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} m\mathbf{I}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{I} \end{bmatrix}^{-1} \cdot \left(\begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{bmatrix} - \begin{bmatrix} [m\boldsymbol{\omega}]_{\times,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & [-\mathbf{I}\boldsymbol{\omega}]_{\times,3} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \right)$$

The vectorized form above is suitable for GNU Octave to perform calculations quickly.

Benić at al., [3] suggest using a hybrid coordinate system to solve the aforementioned governing equations. Specifically, the conservation of linear momentum is described in inertial frame, while the conservation of angular momentum is defined in body frame of reference. In this setup, the centripetal force $\boldsymbol{\omega} \times (m\mathbf{v})$ becomes zero, which also makes the top-left 3x3 part of the Coriolis matrix zero. External forces and torques $[\mathbf{f}, \boldsymbol{\tau}]$ acting on the quadcopter need to be transformed accordingly. In order to do this, the following rotation matrix might be used (assuming a ZYX rotation sequence) to convert between the body and inertial frames.

$$(7) \quad \mathbf{R} = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix}$$

where $Sx = \sin(x)$ and $Cx = \cos(x)$. The transformation matrix $\mathbf{R}$ is orthogonal, hence $\mathbf{R}^{-1} = \mathbf{R}^T$.

Alternatively, the rotation matrix could be derived using quaternions. Given a normalized axis $\mathbf{a}$ and a rotation angle θ , the corresponding quaternion components are

$$(8) \quad q_w = \cos(\theta/2) \quad q_x = a_x \sin(\theta/2) \quad q_y = a_y \sin(\theta/2) \quad q_z = a_z \sin(\theta/2)$$

The formula above is successively applied to Cartesian components of the rotation axis, namely $[a_x, 0, 0]$, $[0, a_y, 0]$, $[0, 0, a_z]$, and corresponding Euler angles $[\phi, 0, 0]$, $[0, \theta, 0]$, $[0, 0, \psi]$, following an initially assigned sequence, for example XYZ. In this way, three quaternions are obtained, i.e. q_1 , q_2 , and q_3 which are subsequently multiplied.

Quaternions are frequently split into a scalar term s and a vector term $\mathbf{v}$ as suggested for example in paper [4]:

$$(9) \quad q = [s \quad \mathbf{v}] \quad s \in \mathbb{R} \quad \mathbf{v} = a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$$

Using the notation above, the quaternion product is computed using the formula

$$(10) \quad q_1 q_2 = (s_1 s_2 - \mathbf{v}_1 \mathbf{v}_2) + (s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2)$$

Then, based on Kuipers, [5], the transformation matrix (from body to inertial frame of reference) for any rotation sequence is derived from the product $[q_w, q_x, q_y, q_z] = (q_1 q_2) q_3$ as follows

$$(11) \quad \mathbf{R} = \begin{bmatrix} 2(q_w^2 + q_x^2) - 1 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & 2(q_w^2 + q_y^2) - 1 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & 2(q_w^2 + q_z^2) - 1 \end{bmatrix}$$

Externally applied force due to motor spinning written in body frame of reference is, [1]

$$(12) \quad \mathbf{T} = \begin{bmatrix} 0 \\ 0 \\ k \sum_{i=1}^4 \omega_i^2 \end{bmatrix}$$

Externally applied force due to quadcopter weight written in body frame of reference is, [1]

$$(13) \quad \mathbf{g} = \mathbf{R}^{-1} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$

where $g = 9.81 \text{ m/s}^2$ is the Earth gravitational acceleration. The total external force vector applied to the quadcopter in body frame of reference is following sum

$$(14) \quad \mathbf{f} = \mathbf{T} + \mathbf{g}$$

Externally applied torque due to motor spinning written in body frame of reference depends upon the quadcopter flight configuration. Two distinct forms are following:

- Flight configuration “+”

$$(15) \quad \mathbf{E} = \begin{bmatrix} lk(-\omega_2^2 + \omega_4^2) \\ lk(-\omega_1^2 + \omega_3^2) \\ b \sum_{i=1}^4 (-1)^i \omega_i^2 \end{bmatrix}$$

- Flight configuration “x”

$$(16) \quad \mathbf{E} = \begin{bmatrix} lk \sin\left(\frac{\pi}{4}\right)(\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) \\ lk \sin\left(\frac{\pi}{4}\right)(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2) \\ b \sum_{i=1}^4 (-1)^i \omega_i^2 \end{bmatrix}$$

where k is lift constant, b is drag constant, l is distance between quadcopter mass center and motor axis. Externally applied torque due to gyroscopic effect written in body frame of reference is, [3]

$$(17) \quad \mathbf{o} = - \left(\mathbf{\omega} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) I_r \sum_{i=1}^4 (-1)^i \omega_i$$

where I_r is the propeller moment of inertia. The total external torque vector applied to the quadcopter in body frame of reference is following sum

$$(18) \quad \boldsymbol{\tau} = \mathbf{E} + \mathbf{o}$$

Euler angles time rate of change (inertial frame of reference) is related to body frame angular rates according to following formulae, [6]:

$$(19) \quad \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S_\varphi \\ 0 & C_\varphi & C_\theta S_\varphi \\ 0 & -S_\varphi & C_\theta C_\varphi \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

$$(20) \quad \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & S_\varphi T_\theta & C_\varphi T_\theta \\ 0 & C_\varphi & -S_\varphi \\ 0 & S_\varphi / C_\theta & C_\varphi / C_\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

where $S_x = \sin(x)$, $C_x = \cos(x)$, and $T_x = \tan(x)$. Note singularity if $\theta = \pm\pi/2$ rad.

Linear acceleration in direction of z axis is affected by thrust T only (altitude controller) whilst angular accelerations of Euler angles roll φ , pitch θ , and yaw ψ are affected by torques τ_φ , τ_θ , and τ_ψ respectively (attitude controller). Luukkonen, [1] suggests following proportional – derivative controller governing thrust and torques in hybrid reference frame:

$$(21) \quad \begin{aligned} T &= \left[g + K_{z,d} (\dot{z}_d - \dot{z}) + K_{z,p} (z_d - z) \right] \frac{m}{C_\varphi C_\theta} \\ \tau_\varphi &= \left[K_{\varphi,d} (\dot{\varphi}_d - \dot{\varphi}) + K_{\varphi,p} (\varphi_d - \varphi) \right] I_{xx} \\ \tau_\theta &= \left[K_{\theta,d} (\dot{\theta}_d - \dot{\theta}) + K_{\theta,p} (\theta_d - \theta) \right] I_{yy} \\ \tau_\psi &= \left[K_{\psi,d} (\dot{\psi}_d - \dot{\psi}) + K_{\psi,p} (\psi_d - \psi) \right] I_{zz} \end{aligned}$$

The subscript d denotes “desired” state and $C_x = \cos(x)$. Motor angular velocities are computed from eq. (12), eq. (15), and eq. (16) according to adopted flight configuration. The “Symbolic” package available in GNU Octave was used to compute the required angular velocities with regard to thrust and torque vectors. Controller coefficients K and K_p ($i = \varphi, \theta, \psi$) are borrowed from monograph [1], so are quadcopter overall dimensions and mass characteristics. Obtained results are following:

- Flight configuration “+”

$$(22) \quad \begin{aligned} \omega_1^2 &= \frac{Tbl - 2b\tau_\theta - kl\tau_\psi}{4bkl} & \omega_2^2 &= \frac{Tbl - 2b\tau_\theta + kl\tau_\psi}{4bkl} \\ \omega_3^2 &= \frac{Tbl + 2b\tau_\theta - kl\tau_\psi}{4bkl} & \omega_4^2 &= \frac{Tbl + 2b\tau_\theta + kl\tau_\psi}{4bkl} \end{aligned}$$

- Flight configuration “x”

$$(23) \quad \begin{aligned} \omega_1^2 &= \frac{Tbl \sin\left(\frac{\pi}{4}\right) + b\tau_\phi - b\tau_\theta - kl \sin\left(\frac{\pi}{4}\right) \tau_\psi}{4bkl \sin\left(\frac{\pi}{4}\right)} & \omega_2^2 &= \frac{Tbl \sin\left(\frac{\pi}{4}\right) - b\tau_\phi - b\tau_\theta + kl \sin\left(\frac{\pi}{4}\right) \tau_\psi}{4bkl \sin\left(\frac{\pi}{4}\right)} \\ \omega_3^2 &= \frac{Tbl \sin\left(\frac{\pi}{4}\right) - b\tau_\phi + b\tau_\theta - kl \sin\left(\frac{\pi}{4}\right) \tau_\psi}{4bkl \sin\left(\frac{\pi}{4}\right)} & \omega_4^2 &= \frac{Tbl \sin\left(\frac{\pi}{4}\right) + b\tau_\phi + b\tau_\theta + kl \sin\left(\frac{\pi}{4}\right) \tau_\psi}{4bkl \sin\left(\frac{\pi}{4}\right)} \end{aligned}$$

PD controller coefficients for positioning within XY plane are derived through heuristics.

Results

Figures below present the quadcopter moving along initially assigned square and staggered paths (upper left), motor revolutions (upper right), roll angle (lower left), and pitch angle (lower right).

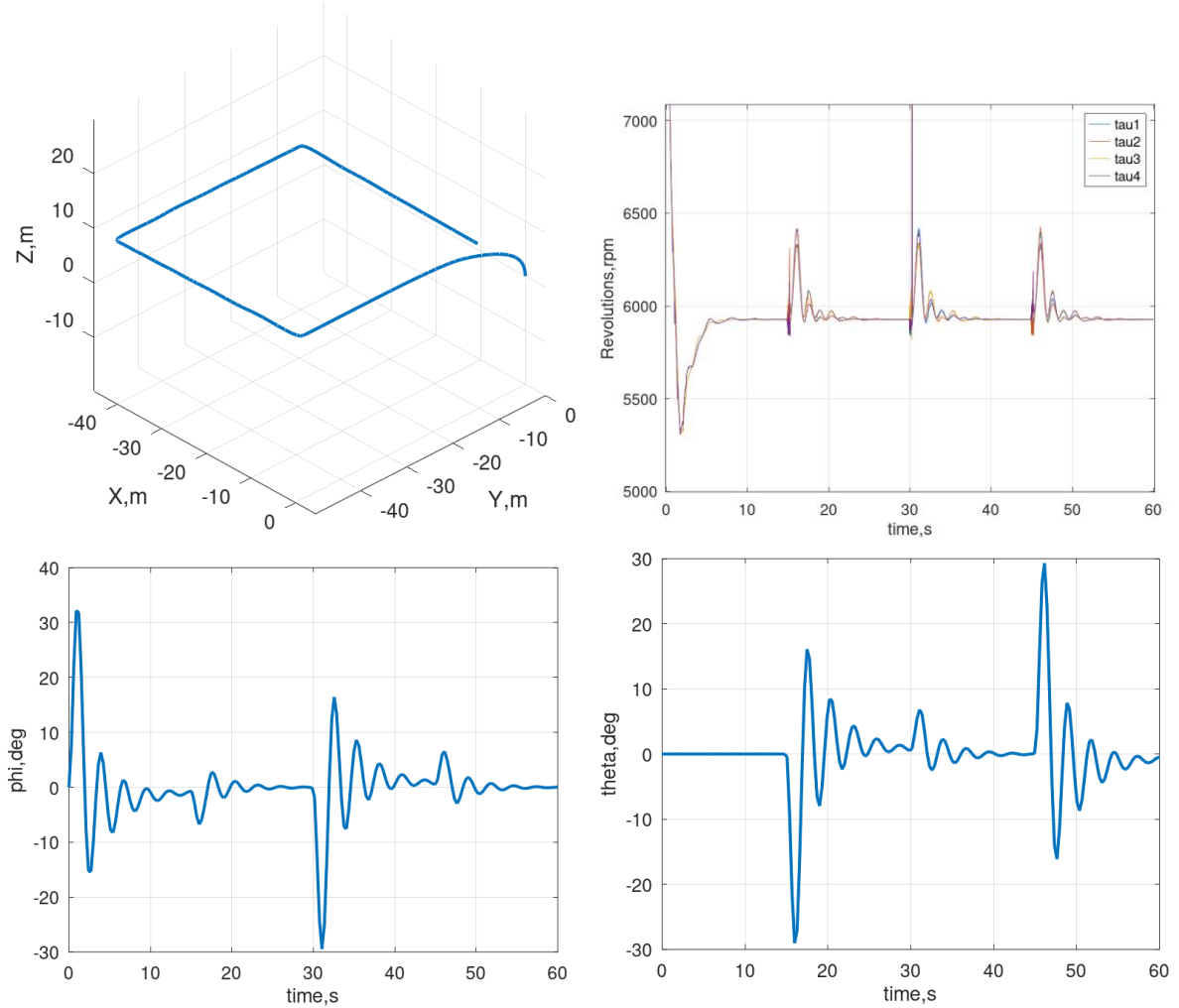


Fig. 2. The quadcopter following a square trajectory at altitude $H = 10$ m, yaw angle $\psi = 0$

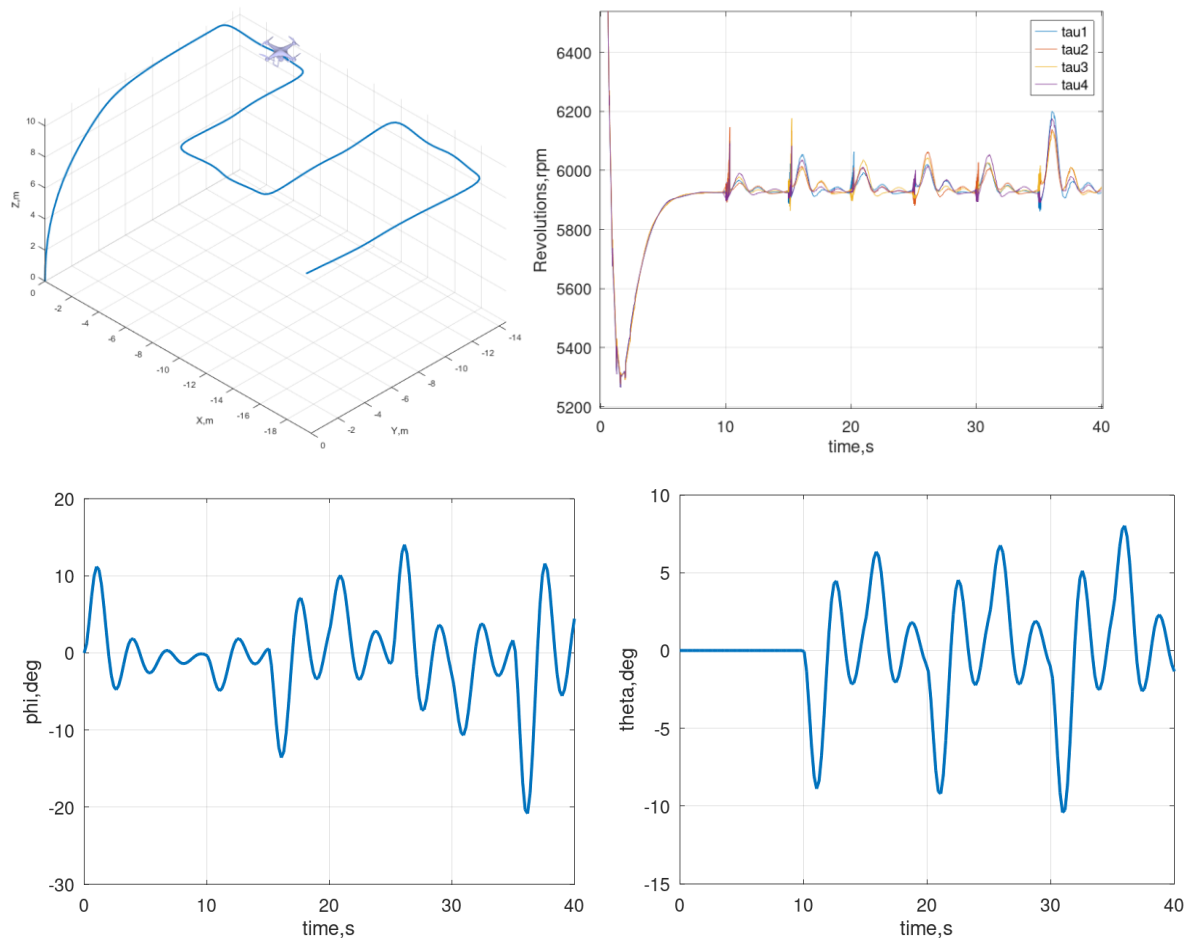


Fig. 3. The quadcopter following a staggered trajectory at altitude $H = 10$ m, yaw angle $\psi = 0$

Discussion

Integral additive fixes the controller output by adding up all the past errors over time. It helps eliminate steady-state error caused by constant biases or disturbances. If the actuator reaches its maximum limit, it can lead to integral windup, which causes overshoot and slows down the response. To prevent this, methods like conditional integration, back-calculation, and limiting the integral term are commonly used.

The Ziegler-Nichols method, [7] is a widely used approach for tuning PID controllers. It is straightforward and quick, but it can sometimes result in aggressive control, overshoot, or instability. Nevertheless, it is a good starting point for most applications including present research.

Software used to develop current research is GNU Octave, [8] with free license. The project source code might be downloaded from github.com [9].

References:

1. Luukkonen, T., Modelling and Control of a Quadcopter, research project, Aalto University, 2011 https://sal.aalto.fi/publications/pdf-files/eluu11_public.pdf
2. <https://grabcad.com/library/wildlife-drone-1>
3. Benić, Z., P. Piljek, D. Kotarski, Mathematical Modelling of Unmanned Aerial Vehicles with Four Rotors, Interdisciplinary Description of Complex Systems 14(1), 88–100, 2016
4. Wyss-Gallifent, J., Lecture notes Math431, Mathematics and Geometry for Computer Graphics, chapter Quaternions, Department of Mathematics, University of Maryland, 2021
5. Kuipers, Jack, Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality, Princeton University Press, 1999, ISBN 0-691-10298-8, p. 126
6. Alderete, T., Simulator Aero Model Implementation, NASA Ames Research Center, Moffett Field, California
7. https://en.wikipedia.org/wiki/Ziegler-Nichols_method
8. <https://octave.org/>
9. <https://github.com/samolet4e/Quadcopter/>