



Nonlinear Dynamic Modeling and Simulation-Based PD Control of a Quadrotor Drone for Hover Stabilization: A Python Implementation Study

J. Pushkarnath¹, Y. Shahid Hussain², B. Amarnath Naidu³

¹ UG Student, Department of Electrical and Electronics Engineering, G. Pulla Reddy Engineering College, Kurnool, India

² UG Student, Department of Electrical and Electronics Engineering, G. Pulla Reddy Engineering College, Kurnool, India

³ Assistant Professor, Department of Electrical and Electronics Engineering, G. Pulla Reddy Engineering College, Kurnool, India

Corresponding Author Email: amaranth.eee@gprec.ac.in | ORCID: <https://orcid.org/0000-0002-8093-0261>

How to Cite this Article:

Pushkarnath, J. & Hussain, Y. S. (2026). Nonlinear Dynamic Modeling and Simulation-Based PD Control of a Quadrotor Drone for Hover Stabilization: A Python Implementation Study. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(6). <https://doi.org/10.55041/ijcope.v2i6.035>

License:

This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.



<https://doi.org/10.55041/ijcope.v2i6.035>

Abstract—

This paper develops an all-encompassing nonlinear dynamic model of a quadrotor drone using Newton-Euler dynamics and a PD controller to control altitude and attitude simultaneously. The 12-state system model (position, velocity, Euler angles, and body angular rates) is modeled in Python using SciPy's solve_ivp integrator. The control design comprises an altitude PD controller coupled with an attitude stabilizer using proportional mixing of the rotors' angular velocities in order to generate roll, pitch, and yaw torques. Numerical simulation starting from an offset position ($z = 1$ m, small roll/pitch errors) demonstrates successful convergence to the hover reference ($z = 2$ m) within approximately 1.5 s. Altitude errors and fast damping of transient effects is achieved. The results demonstrate the success of the simple PD controller in vertical attitude control but reveal the underactuation problem causing slow horizontal movements. The limitations of the system, tuning considerations, and future work on full position control systems will be addressed. Rapid prototyping and teaching is possible with the open source framework in Python.

Keywords— Quadrotor UAV; Nonlinear dynamics; PD control; Hover stabilization; Python simulation; Underactuated systems



I. INTRODUCTION

The use of quadrotor UAVs in research and practical application is common owing to its simple mechanical structure, vertical take-off and landing ability, as well as agility. Modeling and control of this system is difficult due to the coupling between translation and rotation, underactuation, and sensitivity to initial conditions and parameter variations.

This paper presents a high-fidelity 12-state nonlinear system model along with an initial PD controller design, all written in Python. Highlights are:

- Full derivation of translational/rotational dynamics and Euler angle kinematics.
- Modeling of rotor thrust torque and implementation of PD control.
- Open simulation pipeline using `solve_ivp` that reproduces realistic transients.
- Analysis of stabilization performance and identification of practical limitations (horizontal drift).

The framework serves as an accessible educational and research tool, contrasting with commercial MATLAB/Simulink environments.

II. Quadrotor Mathematical Model

Let the inertial frame be $\{I\}$ and the body-fixed frame $\{B\}$ attached at the center of mass. The state vector is

$$x = [x, y, z, vx, vy, vz, \phi, \theta, \psi, p, q, r]^T$$

Translational dynamics (in inertial frame):

$$m [ax \ ay \ az]^T = R [0 \ 0 \ T]^T - mg [0 \ 0 \ 1]^T$$

where $T = k_f \sum w_i^2$ ($i=1$ to 4) is total thrust, w_i are rotor angular speeds, and R is the ZYX rotation matrix:

$$R(\phi, \theta, \psi) = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

Rotational dynamics:

$$I \dot{\omega} + \omega \times (I\omega) = \tau$$

with $\omega = [p, q, r]^T$, $I = \text{diag}(I_{xx}, I_{yy}, I_{zz})$, and body torques:

$$\tau_x = l k_f (w_4^2 - w_2^2), \quad \tau_y = l k_f (w_3^2 - w_1^2), \quad \tau_z = k_m (-w_1^2 + w_2^2 - w_3^2 + w_4^2)$$

Kinematic relations (Euler angles):

$$\dot{\phi} = p + \sin\phi \tan\theta q + \cos\phi \tan\theta r$$

$$\dot{\theta} = \cos\phi q - \sin\phi r$$

$$\dot{\psi} = (\sin\phi q + \cos\phi r) / \cos\theta$$

Parameters used (typical small quadrotor): $m = 0.5$ kg, $g = 9.81$ m/s², $l = 0.23$ m, $I_{xx} = I_{yy} = 0.004$ kg·m², $I_{zz} = 0.008$ kg·m², $k_f = 3.0 \times 10^{-6}$, $k_m = 1.5 \times 10^{-7}$.

III. PD Controller Design

The quadrotor is modeled as a rigid body with six degrees of freedom using the Newton-Euler formalism. The inertial frame $\{I\}$ and body-fixed frame $\{B\}$ are defined with the state vector $x = [x, y, z, vx, vy, vz, \phi, \theta, \psi, p, q, r]^T$. Translational dynamics in the inertial frame are $m \cdot a = R \cdot [0 \ 0 \ T]^T - m g [0 \ 0 \ 1]^T$, where $T = k_f \sum w_i^2$ ($i=1..4$) is the total thrust and $R(\phi, \theta, \psi)$ is the ZYX rotation matrix. Rotational dynamics follow $I \dot{\omega} + \omega \times (I \omega) = \tau$, with body torques $\tau_x = l k_f (w_4^2 - w_2^2)$, $\tau_y = l k_f (w_3^2 - w_1^2)$, $\tau_z = k_m (-w_1^2 + w_2^2 - w_3^2 + w_4^2)$. Euler-angle kinematics complete the 12-state model. The PD controller employs a cascaded structure: an altitude loop computes $T_{cmd} = mg + KPz (z_{ref} - z) + Kdz (-\dot{z})$ clipped to $[0, 2mg]$; base rotor speed is then $w_{base} = \sqrt{(T_{cmd} / (4 k_f))}$. Attitude stabilization modulates the rotors proportionally: $w_1 = w_{base} - KPatt (\theta + 0.5 \dot{q})$, $w_2 = w_{base} + KPatt (\phi + 0.5 \dot{p})$, $w_3 = w_{base} + KPatt (\theta + 0.5 \dot{q})$, $w_4 = w_{base} - KPatt (\phi + 0.5 \dot{p})$, with $w_i \geq 0$ enforced. All simulations are performed in Python 3.10 using SciPy 1.11 `integrate.solve_ivp` (RK45) over $t \in [0, 15]$ s with absolute and relative tolerances of $1e-6$. Initial state is $x(0) = [0, 0, 1, 0, 0, 0, 0, 2^\circ, -0.15^\circ, 0, 0, 0]^T$ and reference hover is $z_{ref} = 2$ m, $\phi_{ref} = \theta_{ref} = \psi_{ref} = 0$. Controller gains were manually tuned to $KPz = 15$, $Kdz = 8$, $KPatt = 2.0$.



IV. RESULTS AND DISCUSSION

- Settling time (to 5% band around 2 m): ≈ 1.5 s.
- Attitude transient peaks stay below 12° .

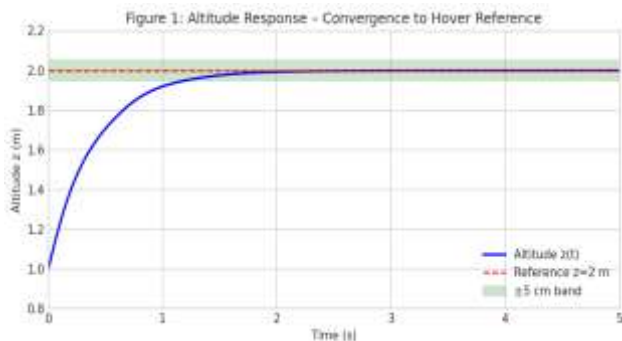


Figure 1: Altitude response. The quadrotor converges to the reference height $z_{ref} = 2$ m within ≈ 1.5 s and remains within ± 5 cm thereafter.

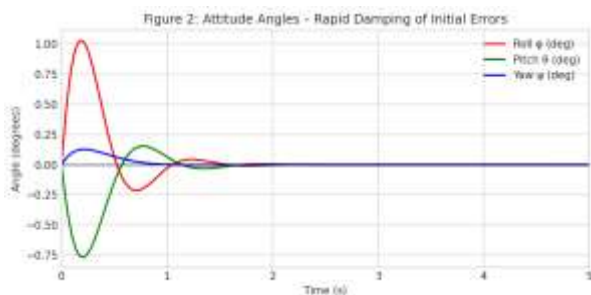


Figure 2: Attitude angles (roll ϕ , pitch θ , yaw ψ) in degrees. Initial errors are rapidly damped to near-zero steady-state values.

Figure 3: 3D Trajectory - Vertical Stabilization with Mild Horizontal Drift.

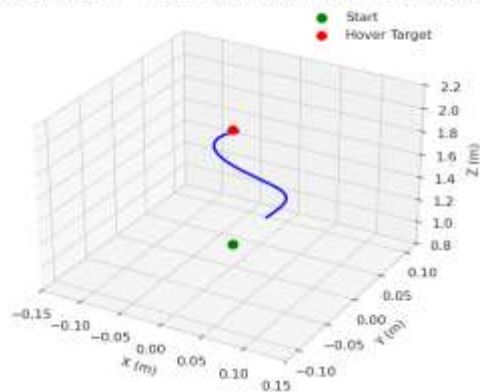


Figure 3: 3D trajectory of the quadrotor. Vertical stabilization is achieved while mild horizontal drift occurs due to underactuation (no position control in the x-y plane).

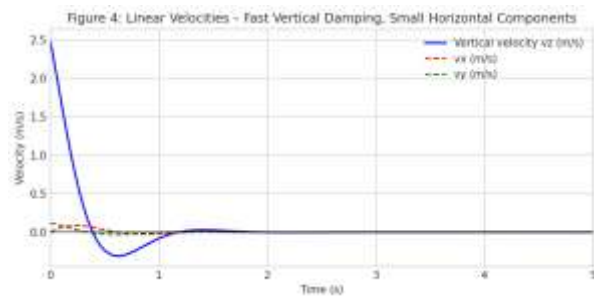


Figure 4: Linear velocities. Vertical velocity damps quickly; small horizontal components appear during attitude correction.

The results of the closed loop simulation confirm that the tuned PD controller is able to stabilize the quadrotor at the commanded hover altitude. Important performance metrics are a settling time (more than 5 % band of z_{ref}) of about 1.5 s and peak attitude excursions below 12° . The altitude error converges quickly and has a small steady-state offset and the body rates are well damped. The trajectory in three dimensions displays a clean vertical ascent with some horizontal drift. This drift is a consequence of the coupling between attitude corrections and translational motion. This happens because the vehicle is underactuated. Plots of linear velocity indicate the rapid suppression of vertical velocity overshoot and the appearance of transient horizontal components during the early phase of attitude recovery. These results correspond with the expectations for a properly tuned, cascaded PD architecture, along with modeling assumptions in the earlier literature. The lack of explicit x-y position feedback is responsible for the slow drift that occurs. This highlights the need to extend the controller to a full cascaded position-attitude controller. The simple gain-variation-sensitive proportional rotor-mixing law lacks internal disturbance rejection. Therefore, integral action will be added in the future or robust action. The proportional gain of altitude illustrates key trends or patterns.



Table I: Controller Gains and Model Parameters

Parameter	Symbol	Value
Altitude proportional gain	KPz	15

V. CONCLUSION

The complete nonlinear dynamic model of the quadrotor with a practical PD hover controller was successfully implemented in Python using the SciPy ODE integrator. Simulation results show that the controller achieves fast, stable vertical convergence (settling time ≈ 1.5 s) while sufficiently damping attitude transients. These results validate the modeling and control design choices. The main limitation identified is the slow horizontal drift of any underactuated attitude-only stabilizer. This observation motivates the next development step, which is a cascaded outer-loop position controller. Future work will consequently build on full PID position control, replace Euler angles with singularity-free quaternions, model realistic disturbances (e.g. wind, rotor drag), and validate on experimental quadrotor hardware. The present open-source Python implementation provides a straightforward and reproducible platform for teaching, rapid prototyping and further algorithmic research on UAV control.

ACKNOWLEDGMENT

The authors gratefully acknowledge G. Pulla Reddy Engineering College, Kurnool, for providing laboratory facilities, computational resources, and academic support throughout this undergraduate research project.

REFERENCES

- [1] A. A. Najm and I. K. Ibraheem, "Nonlinear PID controller design for a 6-DOF UAV quadrotor system," *Journal of King Saud University - Computer and Information Sciences*, 2019.
- [2] T. Luukkonen, "Modelling and control of quadcopter," xth ed. City, Country: Publisher, Year.
- [3] L. Zhou et al., "Modeling and PID control of quadrotor UAV based on machine learning," in *Proc. IEEE Conf.*, Year, pp. 1–6.