Libyan Open University Journal of Applied Sciences (LOUJAS)
Volume 2 - Issue 1 - 2026 - Pages 165-176

CONTROL OF QUADCOPTER ALTITUDE USING PID CONTROLLER

Khalfallh A. E. Husain*[1], Mahmoud A. Mahmoud[2], Moataz M. Essa[3], Mohammed A. Shtewi[4]
[1,2,3,4] Electrical and Electronics Department, Faculty of Engineering, Sirte University, Sirte, Libya.
***Email (for reference researcher)**: khalfhusain@su.edu.ly

التحكم في ارتفاع الطائرة رباعية المراوح باستخدام المتحكم التناسبي-التكاملي-التفاضلي PID controller

خلف الله عبد الله حسين*[1]، محمود أحمد محمود[2]، معتز محمد عيسى[3]، محمد عبد المنعم الشتيوي[4]
[1،2،3،4]قسم الهندسة الكهربائية والإلكترونية، كلية الهندسة، جامعة سرت، سرت، ليبيا.

Received: 10-01-2026; Accepted: 28-02-2026; Published: 27-03-2026

Abstract
A quadcopter is an unmanned aerial vehicle (UAV) capable of vertical take-off and landing. This study develops a nonlinear mathematical model describing the quadcopter's system dynamics and implements a PID controller for altitude regulation, assuming roll, pitch, and yaw remain stable. The model is derived using Newton's and Euler's equations in the appropriate coordinate frame and simulated using MATLAB-Simulink. Simulation results demonstrate that the drone reaches the desired altitude with minimal overshoot, while vertical velocity and acceleration converge to zero at steady state, indicating stable hovering. Moreover, the controller effectively tracks time-varying reference altitudes, showing accurate and robust performance. These results confirm that the proposed PID control strategy achieves reliable and precise altitude regulation for quadcopter applications.

Keywords : Altitude, Quadcopter, PID Controller, Step Response, Reference Tracking

الملخص:
تُعد الطائرة رباعية المراوح من المركبات الجوية غير المأهولة القادرة على الإقلاع والهبوط العمودي، وقد حظيت باهتمام واسع في التطبيقات الحديثة. تهدف هذه الدراسة إلى تطوير نموذج رياضي غير خطي لتمثيل ديناميكيات نظام الطائرة رباعية المراوح، إلى جانب تصميم متحكم تناسبي–تكاملي–تفاضلي (PID) لتنظيم الارتفاع، وذلك بافتراض ثبات زوايا الميل (Roll)والانحدار (Pitch) والانعراج.(Yaw)
تم اشتقاق النموذج الرياضي بالاعتماد على معادلات نيوتن–أويلر ضمن الإطار الإحداثي المناسب، كما تم تنفيذ المحاكاة باستخدام بيئةMATLAB/Simulink . وقد أظهرت نتائج المحاكاة أن النظام المقترح قادر على الوصول إلى الارتفاع المطلوب بزمن استجابة مناسب وتجاوز محدود، مع تقارب كل من السرعة الرأسية والتسارع إلى الصفر عند الحالة المستقرة، مما يدل على تحقيق حالة تحليق مستقرة.
علاوة على ذلك، أظهر المتحكم المقترح كفاءة عالية في تتبع الإشارات المرجعية المتغيرة زمنياً، مع تحقيق دقة واستقرار ملحوظين. وتؤكد هذه النتائج فاعلية استراتيجية التحكم المعتمدة في توفير أداء موثوق ودقيق لتنظيم ارتفاع الطائرة رباعية المراوح.
الكلمات المفتاحية: الارتفاع، الطائرة رباعية المراوح، متحكمPID ، استجابة الخطوة، تتبع الإشارة المرجعية.

Introduction
The quadcopter, a compact multi-rotor UAV, consists of four rotors arranged in a cross configuration with opposite rotations for torque balance, as illustrated in Figure 1. Its nonlinear dynamics involve six degrees of freedom and four rotor-speed inputs, with motion controlled by adjusting the rotor torques (T1–T4) (Usman, 2020; Ferry, 2017).

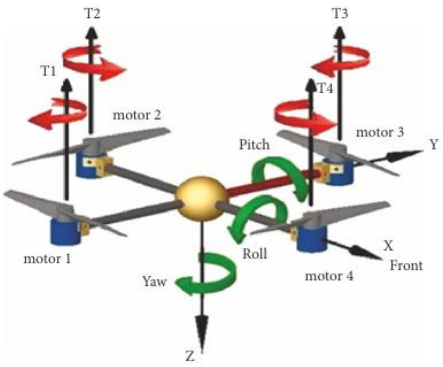

Fig. 1. Quadcopter schematic showing rotors (T1–T4) and rotational axes (roll, pitch, yaw) in the body frame.

To enhance stability, the quadcopter is equipped with multiple sensors, including the BMP085 (pressure and altitude) sensor and a gyroscope, enabling the microcontroller to maintain a predetermined altitude (Ferry, 2017; Tytler, 2017b). The Arduino microcontroller is widely used for its programming flexibility, while additional sensors such as cameras and ultrasonic devices help estimate distances relative to the ground and GPS-based positions (Usman, 2020; Portescap, 2002). The frame supports the motors, LiPo battery, propellers, and all other components.

2. Literature Survey

Several studies have addressed the mathematical modeling and control of quadcopters. Matrix methods and bond graphs are employed to describe their dynamic behavior (Usman, 2020; (Ferry, 2017) while foundational modeling approaches provide a basis for future research (Tytler, 2017b).

The system's differential equations are derived from Newton–Euler and Euler–Lagrange formulations. Various control algorithms have been proposed for attitude stabilization and trajectory tracking, including linear control techniques applied after linearizing the system around a stable point (Portescap, 2002).

Recent studies have explored various control and design approaches for quadcopters. A fuzzy self-tuning PID controller was proposed to regulate altitude, allowing comparison with a conventional linear PID controller (Surriani, Budiyanto, & Arrofiq, 2017).

Sliding mode control combined with Lyapunov theory has been used to enhance altitude tracking under ground effects and time-varying payloads, validated through simulations and experiments (Lee, Xuan-Mung, Nguyen, & Hong, 2023).

Additionally, the design of a general-purpose delivery drone has been investigated, with simulations in MATLAB, SolidWorks, Gazebo, and Proteus to optimize materials, ensure durability, and guarantee safety (Karim, 2020).

3. PID Controller

Controllers are essential for maintaining quadcopter stability. In this study, a PID controller Fig.2 is employed to regulate the system, as the vehicle becomes unstable without proper control. Due to the highly nonlinear dynamics of the quadcopter, the PID gains are tuned using a trial-and-error approach for simplicity (Usman, 2020; Abo-Khsheem & Husain, 2018; Husain & Khalifa, 2026).

Libyan Open University Journal of Applied Sciences (LOUJAS)
Volume 2 - Issue 1 - 2026 - Pages 165-176

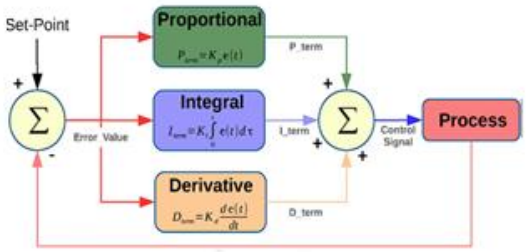

Fig. 2. PID controller configuration.

4. Mathematical Model of Quadcopter System

The basic motion of a quadcopter is achieved by controlling the relative speeds of its motors, allowing the vehicle to tilt in different directions as shown in Fig. 3. Tilting forward is accomplished by reducing the speed of the front motors, whereas tilting backward is achieved by reducing the speed of the rear motors.

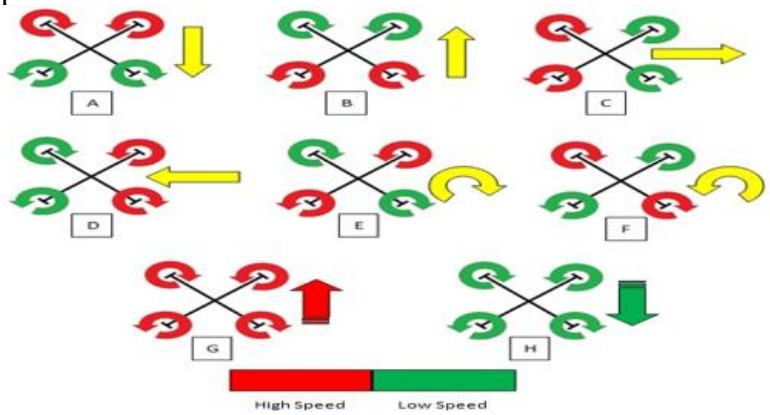

Fig.3. Basic flight movements of a quadcopter

To describe the quadcopter's motion accurately, the concept of six degrees of freedom (6-DOF) is used, including pitching, rolling, and yawing. The 6-DOF framework defines both the position and orientation of the vehicle in three-dimensional space. The vehicle's position is represented by the coordinates x,y,zx, y, zx,y,z in an inertial reference frame, as shown in Fig. 4, where the origin is fixed at the quadcopter body (Sarin, 2020).

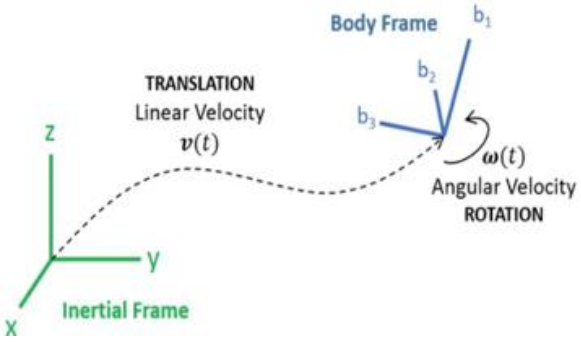

Fig.4. Inertial and body frames of reference

For orientation, Euler angles (ϕ,θ,ψ) are employed due to their simplicity. The final orientation of the quadcopter with respect to the body frame is obtained using a transformation from the inertial frame to the body frame. Fig. 5 illustrates the rotation of the body axes b_1, b_2, b_3 relative to the inertial axes.

Libyan Open University Journal of Applied Sciences (LOUJAS)
Volume 2 - Issue 1 - 2026 - Pages 165-176

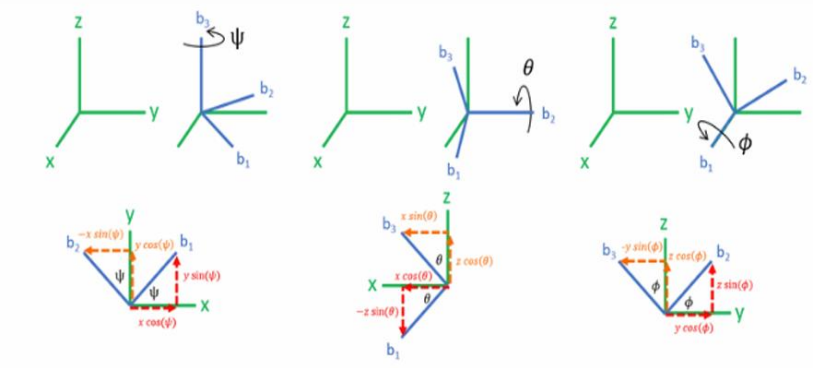

Fig.5. Euler angles representation

The transformation from the inertial frame to the body frame is expressed as:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (1)$$

By multiplying the three rotation matrices, a single transformation matrix C_n^b is obtained:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \cos\psi\sin\theta\sin\phi & \cos\psi\cos\phi + \sin\theta\sin\phi\sin\psi & \cos\theta\sin\phi \\ \sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2)$$

The transformation from the body frame to the inertial frame can be obtained using the inverse of the transformation matrix C_b^n:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta \\ \cos\theta\sin\psi & \cos\psi\cos\phi + \sin\theta\sin\phi\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (3)$$

4.1. Newton's Second Law of Motion

The translational and rotational dynamics presented in Equation (4) are derived from Newton's second law of motion. This law states that the applied force is equal to the time rate of change of linear momentum(Serway, Jewett, & Peroomian, 2000). For a body with constant mass, the relationship can be written as:

$$F = \frac{dP}{dt} = m\frac{dV}{dt} = ma \quad (4)$$

The angular momentum associated with the motion is expressed in Equation (5) as the cross product of the position vector and the linear momentum rate of change:

$$H = \vec{r} \times F = \vec{r} \times m\frac{dV}{dt} \quad (5)$$

The moment (M) acting on the body can therefore be written as the time derivative of the angular momentum, which leads to the expression shown in Equation (6):

$$M = \frac{dH}{dt} = I\frac{d\omega}{dt} = I\dot{\omega} \quad (6)$$

Where **F** represents the applied force, **P** denotes the linear momentum, **m** is the mass of the vehicle, **a** is the linear acceleration, **V** is the velocity, $\vec{\mathbf{r}}$ is the position vector, **H** represents angular momentum, **I** is the moment of inertia, and $\dot{\boldsymbol{\omega}}$ denotes the angular acceleration.

4.2. Thrust Equations

The thrust equation of each of the four motors can described as in equation (7) to equation (10)

-Vertical thrust.

$$U1 = b(w_1^2 + w_2^2 + w_3^2 + w_4^2) \quad (7)$$

-Roll thrust.

$$U2 = bsin(\frac{pi}{4})(w_1^2 - w_2^2 - w_3^2 + w_4^2) \quad (8)$$

-Pitch thrust.

$$U3 = bsin(\frac{pi}{4})(-w_1^2 - w_2^2 + w_3^2 + w_4^2) \quad (9)$$

-Yaw thrust.

$$U4 = d(w_1^2 - w_2^2 + w_3^2 - w_4^2) \quad (10)$$

The overall angular velocity of all the propellers combined obtained as in equation (11).

$$\omega_5 = \omega_1 - \omega_2 + \omega_3 - \omega_4 \quad (11)$$

4.3. Moments of Inertia

The moment of inertia represents the torque required to initiate or stop rotational motion. Equations (12)–(14) describe the moments about the three axes of the symmetrical vehicle.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (12)$$

$I_{xx}, I_{yy}\ and\ I_{zz}$ denote the moments of inertia about the principal axes of the body frame. For quadcopters, the operating altitude is typically limited; therefore, air density can be assumed constant (Li & Liu, 2020).

$$I_{zz} = \left(\frac{1}{3} \times \frac{M}{4} \times l^2\right) \times 4 \quad (13)$$

$$I_{zz} = (\frac{1}{3} \times \frac{0.944}{4} \times 0.230^2) \times 4$$

$$I_{zz} = 0.0166\text{kg}m^2$$

$$I_{xx} = I_{yy} = 0.0071\text{kg}m^2 \quad (14)$$

Where: $m = mass\ of\ vhical$=0.944kg and l=length of vehicle=0.230m.

To estimate the moment of inertia of the propeller, one half of the propeller should be divided into three parts as shown in fig.6.

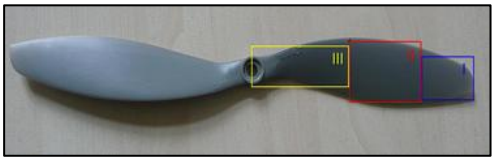

Fig.6.Propeller divided into three parts.

This way, it is able to calculate their moments of inertia using equation (15) and equation (16), which contribute to the overall moment of inertia of the whole propeller (Serway, Jewett, & Peroomian, 2000; Mukarram, Fiaz, & Khan, 2015).

Table 1: Propeller's parts properties

part	mass (g)	width (cm)	height (cm)	r (cm) (distance from the shaft)
I	1.7	3	2.5	9
II	2.6	4.4	2.8	5
III	1.2	5	1.2	0

$$I = m((\frac{h^2}{12}) + (\frac{w^2}{12})) \qquad (15)$$

$$I_p = 2(I_1 + I_2 + I_3) \qquad (16)$$

$$I_p = 4.439 \times 10^{-5} kgm^2$$

4.4. Thrust Coefficient

When the quadcopter operates in a hovering condition, the total thrust generated by the four rotors balances the gravitational force acting on the vehicle. Consequently, each rotor produces an equal share of the total thrust required to maintain equilibrium. Therefore, the thrust generated by a single rotor can be expressed as one quarter of the quadcopter's total weight (Mukarram, Fiaz, & Khan, 2015)

The thrust produced by a rotor is proportional to the thrust coefficient and the rotational speed of the motor, and it can be expressed as:

Thrust = (thrust coefficient) × (motor rotational speed)

$$b = \frac{mg}{4\times\omega^2} \qquad (17)$$

$$b = \frac{0.944\times9.8}{4\times1250^2} = 1.480 \times 10^{-6} kgm^2$$

4.5. Rotor Inertia

The rotor inertia is represented in equation (18), so

$$\mathrm{J_r} = \frac{1}{2} \times m_r \mathrm{R}^2 \qquad (18)$$

$$\mathrm{J_r} = \frac{1}{2} \times (0.047 \times 0.215^2) = 1.086 \times 10^{-3} kgm^2$$

Where: m_r = mass of the motor = 0.047Kg and R= the dimension of the motor on the quadacopter axis=0.215m

4.6. Angular Acceleration

The equations (19) to (21) represent the angular acceleration of the drone system depending on Coriolis Theorem.

$$\ddot{\Phi} = \frac{\dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + J_r\dot{\theta}w + I(U2)}{I_{xx}} \qquad (19)$$

$$\ddot{\theta} = \frac{\Phi\dot{\psi}(I_{zz} - I_{xx}) - J_r\Phi w + I(U3)}{I_{yy}} \qquad (20)$$

$$\ddot{\psi} = \frac{\dot{\theta}\dot{\Phi}(I_{xx} - I_{yy}) + I(U4)}{I_{zz}} \qquad (21)$$

4.7. Linear Acceleration

The linear acceleration can now be calculated by the double integrates of the angular acceleration in equation (22) to(24) according to Coriolis Theorem.

$$\ddot{X} = \frac{(\sin\Phi\sin\psi) + (\cos\Phi\cos\psi\sin\Theta)\mathrm{U1} - A_x\dot{x}}{m} \qquad (22)$$

$$\ddot{Y} = \frac{-(\sin\Phi\cos\psi) + (\cos\Phi\sin\psi\sin\Theta)\mathrm{U1} - A_y\dot{y}}{m} \qquad (23)$$

$$\ddot{Z} = \frac{(\cos\Theta\cos\Phi)\mathrm{U1} - A_z\dot{z} - (\mathrm{mg})}{m} \qquad (24)$$

5. Simulation and Evaluation of The control Model of The Quadcopter System

The altitude model of the vehicle was developed using the MATLAB/Simulink library based on the nonlinear dynamic equations of the system. Fig.7 illustrates the plant modeling subsystem, which was constructed to represent the dynamic behavior of the quadcopter. In this subsystem, the state variables (x, y, z, θ, Φ, Ψ) are considered as system outputs, while the motor inputs are represented by M1, M2, M3, and M4.

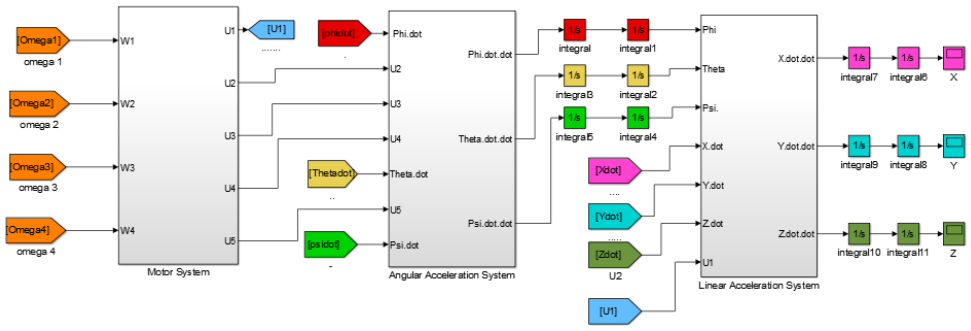

Fig.7: block model of quadcopter system.

The model expresses the angular acceleration of eq.19 to eq.21. Fig.8, fig.9 and fig.10 represent the rolling, pitching and yawing acceleration respectively.

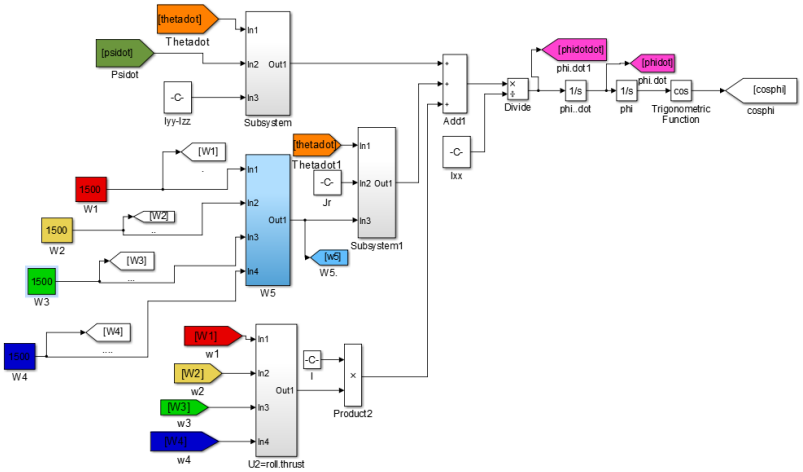

Fig.8.block model of rolling acceleration.

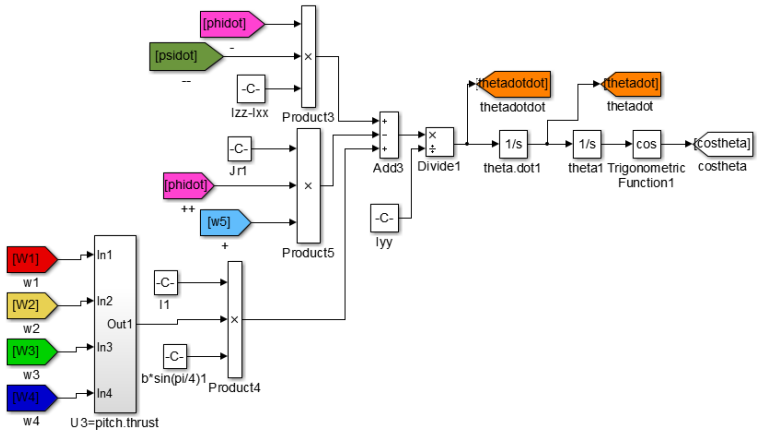

Fig.9.block model of pitching acceleration.

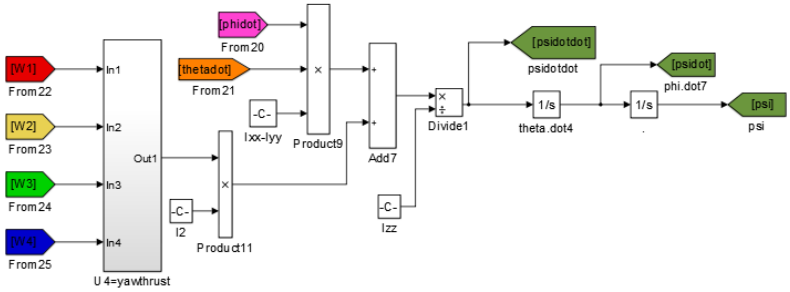

Fig.10.block model of yawing acceleration.

Figure 11 represent the altitude model of quadcopter's system as mentioned in the equation (24).

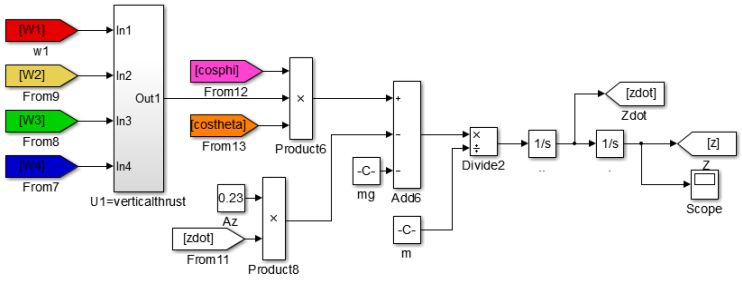

Fig.11.modeling of altitude quadrotor system.

The simulation of the open loop quadcopter system is shown in fig. 12. The quadcopter start hovering at motor speed of 1500 rpm.

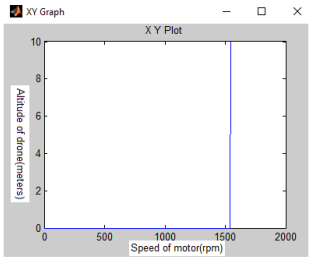

Fig.12.altitude with varying speed of motors.

Fig.13 shows the altitude response of the drone to a step input together with the corresponding error signal. The drone rises from the initial position and gradually approaches the desired

altitude as the controller reduces the tracking error. Over time, the error decreases and converges to zero, indicating that the drone reaches the desired altitude and maintains a stable hovering condition.

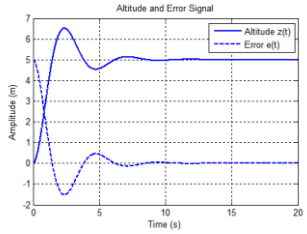

Fig. 13. Drone altitude and tracking error for a step input, showing rise to the desired height.

Fig.14 presents the vertical velocity of the drone during the step response. The velocity increases during the initial motion and gradually decreases to zero as the drone reaches the desired altitude.

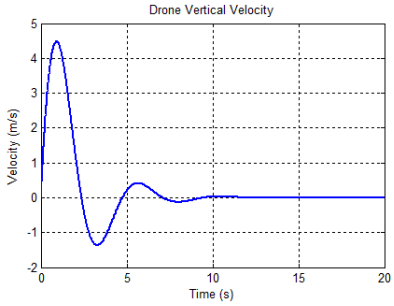

Fig. 14. Drone vertical velocity response to a step input, indicating acceleration and approach to hover.

Fig.15 illustrates the acceleration response of the drone. A positive acceleration appears at the beginning to initiate the motion, then it decreases and converges to zero when the thrust balances the gravitational force at steady state.

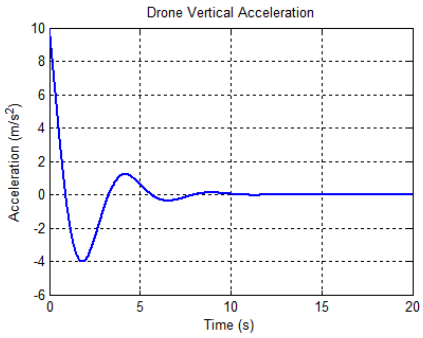

Fig. 15. Drone vertical acceleration response to a step input, showing initial thrust and convergence to zero at steady state.

The subsequent section outlines the MATLAB implementation of the quadcopter system, addressing both three-dimensional (3D) and two-dimensional (2D) motion representations.

Libyan Open University Journal of Applied Sciences (LOUJAS)
Volume 2 - Issue 1 - 2026 - Pages 165-176

```
% 3D and 2D Drone Altitude Control Simulation Code

clc; clear; close all;

%% Simulation parameters

dt = 0.01;          % time step (s)

T = 100;            % total simulation time (s)

N = T/dt;           % number of steps

g = 9.81;           % gravity (m/s^2)

m = 0.944;          % drone mass (kg)

time = 0:dt:T-dt;       %% Initialize state variables

pos = zeros(3, N);      % position [x; y; z]

vel = zeros(3, N);      % velocity [vx; vy; vz]

%% PID Controller gains

Kp = 6.0;

Kd = 4.0;

%% Create altitude reference trajectory (matches graph)

z_ref = zeros(1, N);

for k = 1:N t = time(k);

if t < 10;

z_ref(k) = 0;

elseif t < 30           % Ramp up to 4

z_ref(k) = (4/20)*(t-10);

elseif t < 40           % Hold at 4

z_ref(k) = 4;

elseif t < 55           % Ramp down to 2

z_ref(k) = 4 - (2/15)*(t-40);

elseif t < 70 % Hold at 2

z_ref(k) = 2;

elseif t < 90           % Ramp up to 4

z_ref(k) = 2 + (2/20)*(t-70);

else                    % Hold at 4

z_ref(k) = 4;
```

```
%% Simulation loop

for k = 1:N-1       % Only altitude control (z direction)

error_z = z_ref(k) - pos(3,k);

d_error_z = -vel(3,k);

% PD control with gravity compensation

u_z = Kp*error_z + Kd*d_error_z + m*g;      % Acceleration

acc_z = u_z/m - g;          % Update states

vel(3,k+1) = vel(3,k) + acc_z*dt;

pos(3,k+1) = pos(3,k) + vel(3,k+1)*dt;

end

%% Simple 3D visualization (vertical motion only)

figure;

for k = 1:100:N clf;

plot3(0,0,pos(3,k),'ro','MarkerSize',10,'MarkerFaceColor','r');

hold on;

plot3(0,0,z_ref(k),'go','MarkerSize',12,'LineWidth',2);

grid on;

xlim([-1 1]); ylim([-1 1]); zlim([0 5]); xlabel('X');

ylabel('Y'); zlabel('Z');

title(sprintf('Drone Altitude (Time = %.1f s)', time(k))); drawnow;

end
```

Fig.16 shows the 2-D simulation graph, and it is clear that the quadcopter system can track the changeable set-point with zero displacement in x and y direction.

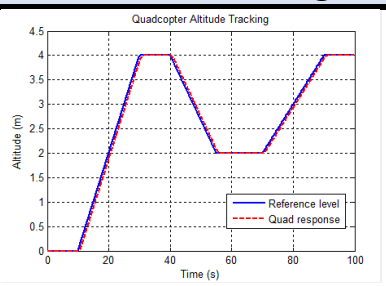

Fig.16.quadcopter tracking step and ramp commands of 2-D.

The 3-D simulation of the quadcopter system shown in the fig.17, which exhibits high maneuverability with substantial stability.

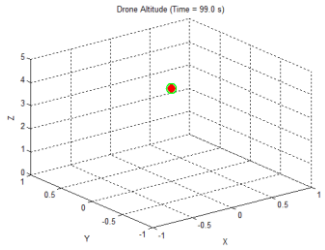

Fig.17: quadcopter tracking step and ramp commands of 3-D.

The angular acceleration of pitching, rolling and yawing are zero as in the fig.8.

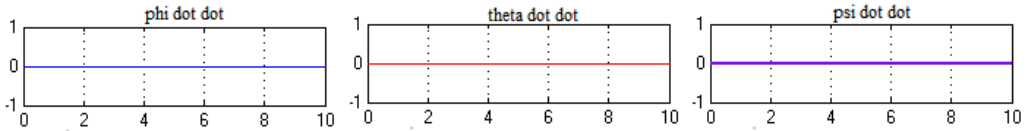

Fig.18: angular acceleration of quadcopter system.

6. Conclusions

The nonlinear mathematical and simulation models of the quadcopter control system were developed based on the vehicle parameters. The PID controller used for altitude control demonstrates satisfactory performance, providing stable vertical thrust and smooth step response. Simulation results show that the drone quickly rises to the desired altitude with minimal overshoot, and the vertical velocity and acceleration converge to zero at steady state, indicating stable hovering. Moreover, the controller effectively tracks a time-varying reference altitude, maintaining accurate response to changes in the set point. Overall, the simulation results confirm that the controller achieves reliable and precise altitude regulation.

Reference

1. Usman, M. (2020). Quadcopter modelling and control with matlab/simulink implementation.
2. Ferry, N. (2017). Quadcopter plant model and control system development with matlab/simulink implementation. Kate Gleason College of Engineering Rochester Institute of Technology Rochester, New York.
3. Tytler, C. (2017). Modeling Vehicle Dynamics–Quadcopter Equations of Motion. Autonomy in motion.
4. Portescap, A. (2002). Rotor inertia. Accessed in June 2019..
5. Surriani, A., Budiyanto, M. U., & Arrofiq, M. (2017, August). Altitude control of quadrotor using fuzzy self-tuning PID controller. In 2017 5th International conference on Instrumentation, Control, and Automation (ICA) (pp. 67-72). IEEE. https://doi.org/10.1109/ICA.2017.8382210

6. Lee, J. W., Xuan-Mung, N., Nguyen, N. P., & Hong, S. K. (2023). Adaptive altitude flight control of quadcopter under ground effect and time-varying load: Theory and experiments. Journal of Vibration and Control, 29(3-4), 571–581. https://doi.org/10.1177/10775463221123456
7. Karim, C. B. (2020). The Design and Development of a General-Purpose Drone (Doctoral dissertation, School of Science and Engineering, Al Akhawayn University).
8. Abo-Khsheem, K. A., & Husain, K. (2018, January). Active control of surge compressor system.
9. Husain, K., & Khalifa, G. H. (2026). Liquid level control using a self-tuning PID controller based on fuzzy logic. International Science and Technology Journal, 38(1), 1–17. https://doi.org/10.62341/istj-vol38-2-66
10. Sarin, P. (2020). Quadcopter dynamics digital lab course project Ep315 guide. Department of Physics, Indian Institute of Technology, Bombay..
11. Serway, R. A., Jewett, J. W., & Peroomian, V. (2000). Physics for scientists and engineers (Vol. 2). Philadelphia: Saunders college publishing.
12. Li, J., & Liu, W. (2020). Lifeline engineering systems. Springer Nature Singapore, Singapore.
13. Mukarram, A., Fiaz, U. A., & Khan, U. I. (2015). Altitude control of a quadcopter. Department of Electrical Engineering, Pakistan Institute of Engineering and Applied Sciences, Islamabad, Pakistan.