

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/373457893>

# Trajectory Tracking of a Quadcopter UAV using PID Controller

Article in ELEKTRIKA- Journal of Electrical Engineering · August 2023

DOI: 10.11113/elektrika.v22n2.440

---

CITATIONS

6

---

READS

1,092

2 authors, including:



Ariffanan Basri

Universiti Teknologi Malaysia

84 PUBLICATIONS 1,229 CITATIONS

SEE PROFILE

# Trajectory Tracking of a Quadcopter UAV using PID Controller

A'dilah Baharuddin and Mohd Ariffanan Mohd Basri\*

Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia.

\*Corresponding author: ariffanan@fke.utm.my

**Abstract:** UAVs or Drones are aircraft with no onboard pilot to control the flight. They are introduced in a few categories such as single-rotor, multi-rotors, fixed-wing, and hybrid VTOL. As for multirotor drones, quadcopters are the most well-known either commercially or in the research field. Due to its advantages, a quadcopter has been chosen to perform various tasks across various fields such as entertainment, military, meteorological reconnaissance, civil and emergency responses. As the demand for quadcopters has diverged, the required features of quadcopters have also diverged. One of the current features required by quadcopters is the ability to track trajectories. However, due to its nature of non-linearity, under-actuated and unstable, controlling quadcopter for an accurate and stable performance is quite a challenge. Despite the various proposed methods throughout the past decades, PID controller is still used as either the main controller or the base controller in most cases of industrial control, including quadcopter, mainly due to its simplicity and robustness. However, to design a proper PID controller for quadcopter system is a challenge as it defies the control inputs of four with its six degree-of-freedom form, in which six inputs are required to be controlled to ensure a stable and accurate flight. This paper derived a mathematically model of a quadcopter with Newton-Euler's equation. Some assumptions on the body and structure of the quadcopter are taken into account to make the modelling possible. Then, a manually tuned PID controller is designed to achieve the objective of controlling the operation and stability of the quadcopter during its flight. The designed controller is tested with five different trajectories which are circular, square, lemniscate, zigzag, and spiral. The results show the proposed controller successfully tracks the desired trajectories, which prove PID controller can be used to control a quadcopter.

**Keywords:** PID Controller, Quadcopter, Trajectory Tracking, UAV

© 2023 Penerbit UTM Press. All rights reserved

*Article History: received 12 March 2023; accepted 12 July 2023; published 28 August 2023.*

## 1. INTRODUCTION

Drones are another word used to address unmanned aerial vehicles (UAVs). Drone is an English term for the humming sound emitted by male honeybees while flying, and the term is adopted by UAVs, as it produces a similar sound during its flight [1]. Drone requires no human pilot on board, as it either can fly autonomously or can be controlled via radio wave [2]. The drone is categorised as single-rotor drone, multirotor drone, fixed-wing drone, and hybrid vertical take-off and landing (VTOL). Multi-rotors drone's constructions are divided according to its frame constructions of the arms and rotors, the basic element of a drone. There are bi-copters for two-engine drones, tri-copters for three-engine drones, quadcopters for four-engine drones, hexacopters for six-engine drones, and octocopters for eight-engine drones [2,3].

Among the types of multi-rotors, quadcopter is the most innovative and actively used across various field to perform various tasks. Quadcopter possesses advantages such as cost-friendly, ease to maintain, simple mechanical structure, can adapt to more complex flight environment, accessibility and highly manoeuvrable, performing VTOL and hover flight [4-6]. The quadcopter has been chosen to

perform various tasks such as surveying, mapping, photography, videography, daily patrolling, and rescue operations [2-9]. Most tasks may expose the quadcopter with limited time to implement measures or decision-making situations. Thus, tracking the trajectory has become one of the requirements during the completion of the task [5,8].

Despite the possessed advantages of quadcopter, it is naturally a device with non-linearity, unstable, and generally underactuated. These characteristics contribute to the challenges in designing a system to control the operation and stability of a quadcopter in different environments and situations [4,5,8,10]. Some controlling methods have been proposed in the past decades, including PID controller, LQR controller, H-infinity controller, and other metaheuristic methods such as sliding mode, backstepping and generic algorithm. Due to its simplicity and robustness, PID controller has been one of the mostly used controller in industrial control actions, including the quadcopter.

In articles [11-17], PID controller is used as the main controller for quadcopter system. In [11], PID controller is utilized to track the attitude and altitude under different

conditions. The parameters are tuned with Ziegler-Nichols' method. Meanwhile, in [10,12] PID parameters are tuned with manual tuning, where several trial-and-error are required to achieve the best performance. Both works concluded that the targets successfully achieved, yet the parameter setting still need re-adjustment every time the target values is changed. Other than that, PID controller is also utilised as the base controller for some metaheuristic methods. For example, in [5,7,18], the PID controller parameters are tuned with fuzzy logic method. In order to adopt the method, the system of quadcopter is built with PID controller beforehand. Thus, the proper values of gains of PID controller are required in the works.

However, controlling a quadcopter with PID to fly it with stability and accuracy for tracking tasks become a challenge as there are six variables involved to be controlled, correspond to the six degree-of-freedom of quadcopter in which contradict to only four control inputs [18].

With objective to control the operation and stability of the quadcopter during its flight in tracking trajectories, this paper aims to model the quadcopter and design a PID controller system with manual tuning method as the base controlling system of quadcopter. The contributions of this works are summarized as follows:

1. A mathematical model of the quadcopter is designed based on Newton-Euler's.
2. A PID controller is designed to achieve the stabilization and track the trajectories of circular, square, lemniscate, zigzag, and spiral.
3. A base control system of quadcopter is designed to be used as the primary model for intelligent and adaptive methods.

This paper is organized into five sections. Section 2 deliberately explains the structure of the quadcopter and the derivation of the dynamics model. Section 3 shows the design of the PID controller for altitude, attitude, and position tracking. Section 4 shows the results and a discussion of the trajectories tracking simulations. Section 5 conclude the works and all section of this paper.

## 2. QUADCOPTER MODELLING

The quadcopter is equipped with four arms and rotors. The most common arrangements of quadcopter are symmetrical cross (x) or plus (+). The positions and rotations of the quadcopter are represented by six variables of the six degrees of freedom. A cross (x) quadcopter mathematical model is derived in this paper. This section includes the description of the structure, the dynamic model derivation, and the parameters of quadcopter.

### 2.1 Quadcopter Structure

Figure 1 shows an illustration of the cross (x) quadcopter. The rotors are running in different pairs' directions to eliminate the anti-torque effects and produce translational and rotational movement; thus rotor 1 and rotor 2 are running in counter-clockwise direction, while rotor 3 and rotor 4 are running in clockwise direction [7,10].

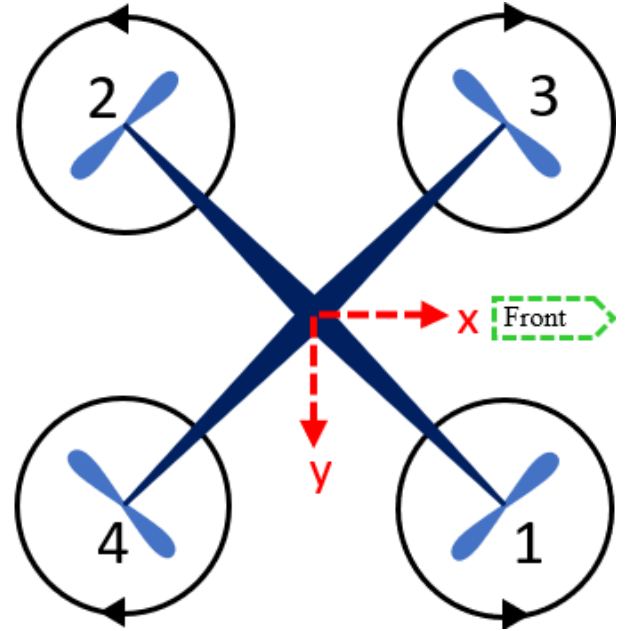


Figure 1: The structure of cross (x) quadcopter.

The motions and movements of the quadcopter are produced by the lifting force generated by the rotating rotors [10]. In order to simplify the mathematical model of the quadcopter, several assumptions are made as follows [7]:

1. The quadcopter is assumed to be a rigid body.
2. The structure of assumed to be asymmetrical with respect to  $x, y$  axis. Therefore, the degree-of-freedom of plus (+) structure can be used for cross (x) structure.
3. The center of the mass and the origin of the body fixed frame coincide.
4. The rotors are considered as rigid, no blade flapping occurs.
5. The rotors work under the same conditions at any time, in which the thrust coefficient and reaction torque are the same.

A motion is initiated by producing difference in the lift by manipulating the rotor velocity. The motions of the quadcopter are summarised in Table 1 [10].

Table 1: The motion of the cross (x) quadcopter.

Motion	Rotors Velocity Manipulation
Hovering	Same speed for all rotors
Rolling	Rotors 1&4, or Rotors 2&3
Pitching	Rotors 1&3, or Rotors 2&4
Yawing	Rotors 1&2, or Rotors 3&4

### 2.2 Quadcopter's Dynamic

Figure 2 shows the body-fixed frame and earth-fixed frame of the quadcopter, where the body fixed frame,  $Q = \{X_Q, Y_Q, Z_Q\}$  and the earth-fixed frame,  $E = \{X_E, Y_E, Z_E\}$  and their relationship is satisfied as  $\{Q\}^T = R^T \times \{E\}^T$  [7].

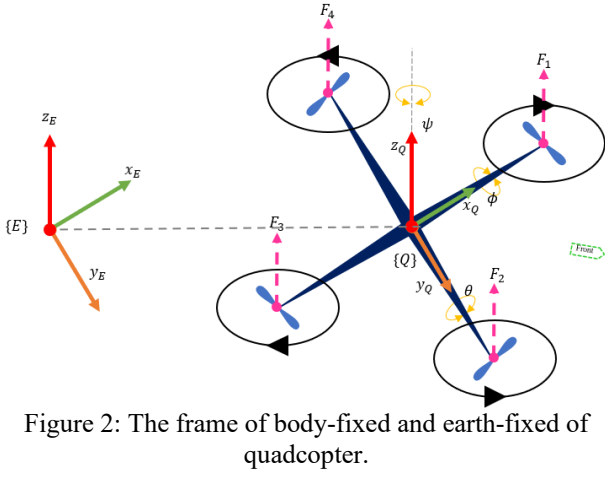


Figure 2: The frame of body-fixed and earth-fixed of quadcopter.

The absolute positions of the quadcopter are represented by  $(x, y, z)$  and the orientations are, respectively, described by roll  $(\phi)$ , pitch  $(\theta)$  and yaw  $(\psi)$ . Based on Figure 2, the quadcopter model is divided into position relative to inertial frame,  $\xi$  (1) and quadcopter attitude,  $\eta$  (2).

$$\xi = [x \ y \ z]^T \in \mathbb{R}^3 \quad (1)$$

$$\eta = [\phi \ \theta \ \psi]^T \in \mathbb{R}^3 \quad (2)$$

The elementary rotations about the  $x$ ,  $y$ , and  $z$  axes are defined using Euler angles. The final orientation with respect to the corresponding inertial axis, inertial trigonometric functions, and their representations is shown in Figure 3 [12].  $R_\phi$  (3) is a single rotation of roll,  $\phi$  radius around  $x$  axis,  $R_\theta$  (4) is a single rotation of pitch,  $\theta$  radius around  $y$  axis and  $R_\psi$  (5) is a single rotation of yaw,  $\psi$  around  $z$  axis.

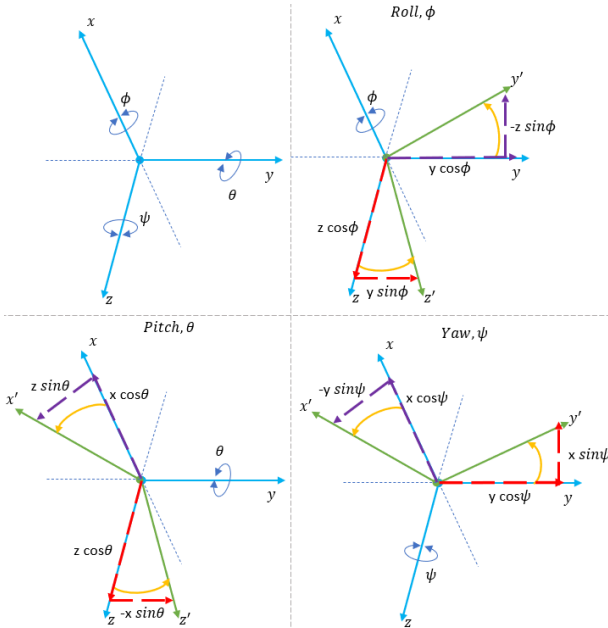


Figure 3: Elementary rotation of a quadcopter.

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \quad (3)$$

$$R_\theta = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \quad (4)$$

$$R_\psi = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

From (3-5), the rotation matrix of the body frame relative to the inertial frame is obtained (6).

$$R_T = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\psi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (6)$$

Newton-Euler's is used to derive the dynamic of quadcopter as the model is assumed to be a rigid body. The translation of the dynamic of the quadcopter is described in (7).

$$m\ddot{\xi} = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -mgE_z + U_1 R_T E_z \quad (7)$$

The gravitational coefficient,  $g$  is defined in negative state as the direction of  $z$ -axis is upward. The  $z$ -axis vector matrix,  $E_z = [0 \ 0 \ 1]^T$  whereas  $U_1$  is the total thrust force generated by four rotors and satisfied as  $U_1 = \sum_{i=1}^4 F_i$  [8]. The final equation of (7) is then formulated as (8).

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{U_1}{m} \begin{bmatrix} c\phi s\theta c\psi + s\phi s\psi \\ c\phi s\theta s\psi - s\phi c\psi \\ c\phi c\theta \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (8)$$

Figure 4 shows the simulation blocks of translational dynamics of quadcopter, based on equation (8).

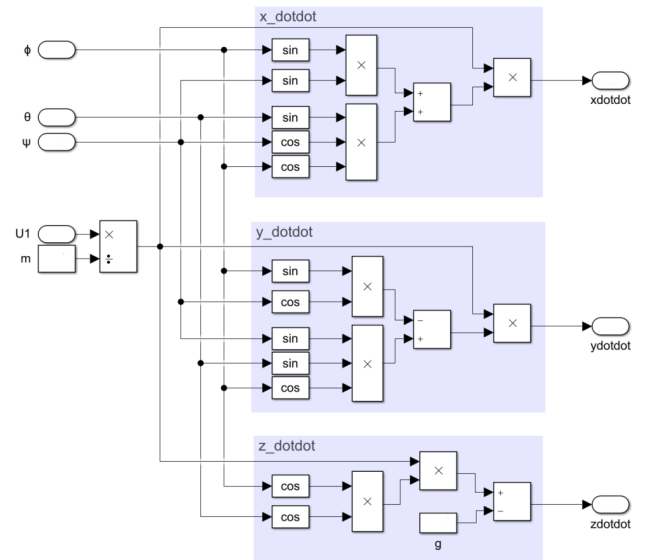


Figure 4: The Simulink blocks of translational dynamics.

The rotational subsystem of the quadcopter dynamic is described in (10). The moment of inertia,  $I$  is defined as  $I = \text{diagonal} [I_{xx} I_{yy} I_{zz}]^T$  as it is a 3-by-3 diagonal matrix, the rotor inertia,  $\mathcal{J}_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$ , the

total rotor speed,  $\Omega_d$  generated from rotors, working in two pairs.  $U_2, U_3$  and  $U_4$  are the total torque related to quadcopter [7]. Equation (9) is finalised as in (10) and the control inputs are renamed as in (11).

$$I\ddot{\eta} = -\dot{\eta} \times I\dot{\eta} - J_r(\dot{\eta} \times E_z)\Omega_d + [U_2 \ U_3 \ U_4]^T \quad (9)$$

$$\begin{bmatrix} I_{xx}\ddot{\phi} \\ I_{yy}\ddot{\theta} \\ I_{zz}\ddot{\psi} \end{bmatrix} = \begin{bmatrix} (I_{yy} - I_{zz})\dot{\psi}\dot{\theta} - (J_r\Omega_d)\dot{\theta} + lU_2 \\ (I_{zz} - I_{xx})\dot{\psi}\dot{\phi} + (J_r\Omega_d)\dot{\phi} + lU_3 \\ (I_{xx} - I_{yy})\dot{\theta}\dot{\phi} + U_4 \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ -b & b & b & -b \\ -b & b & -b & b \\ d & d & -d & -d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (11)$$

Based on equation (10), the model of rotational dynamics is modelled in Simulink as shown in Figure 5, and the rotor speed calculation block is modelled as in Figure 6 based on equation (11).

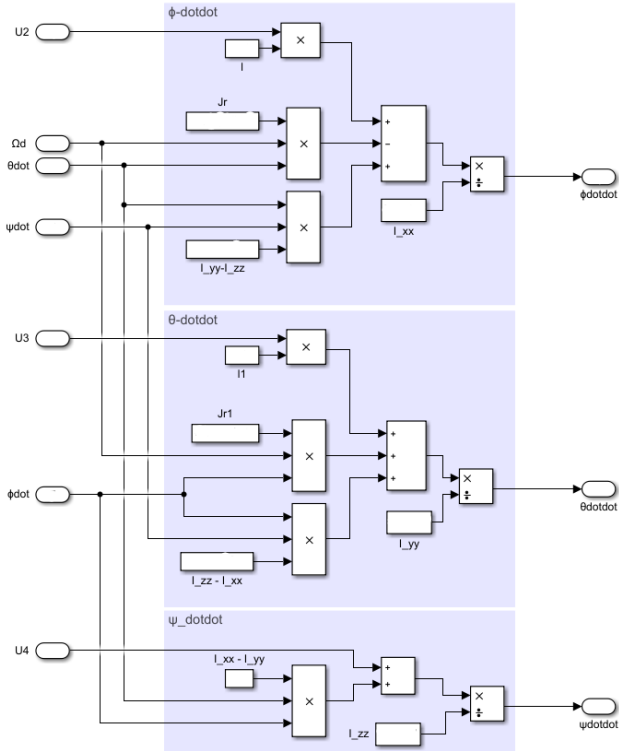


Figure 5: The Simulink blocks of rotational dynamic of quadcopter.

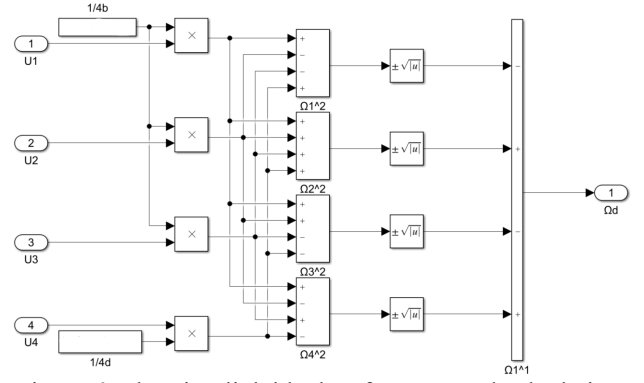


Figure 6: The Simulink blocks of rotor speed calculation.

### 2.3 Quadcopter Parameters

The parameters used for the quadcopter model are listed in Table 2.

Table 2: Quadcopter parameters.

Parameter	Unit	Value
Quadrotor mass, $m$	$kg$	0.800
Lateral moment arm, $l$	$m$	0.300
Thrust coefficient, $b$	$Ns^2$	$1.9232 \times 10^{-5}$
Drag coefficient, $d$	$Nms^2$	$4.003 \times 10^{-7}$
Rolling moment of inertia, $I_{xx}$	$kgm^2$	0.01567
Pitching moment of inertia, $I_{yy}$	$kgm^2$	0.01567
Yawing moment of inertia, $I_{zz}$	$kgm^2$	0.02834
Gravity, $g$	$g/s^2$	9.81
Rotor Inertia, $J_r$	$kgm^2$	$6.01 \times 10^{-5}$

### 3. PID CONTROLLER DESIGN

In order to ensure the quadcopter can work and conduct assigned tasks properly, a controller is required to be implemented to the system. This paper design and simulate a PID controller to control all dynamic variables of the quadcopter, which are  $x, y, z, \phi, \theta$ , and  $\psi$ . Figure 7 shows the general block diagram of the PID control system.

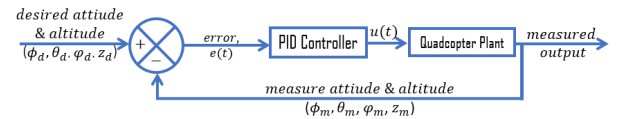


Figure 7: Block diagram of PID control.

The PID controller is applied mainly in process control and industry, where it can be claimed to be one of the most favourable controllers, with a usage rate of more than 95% [13]. The advantage of PID is that it is easy to implement and tune. For the quadcopter, the PID controller is used to adjust the speed of all the rotors to achieve the desired orientation of the quadcopter. The PID controller worked as the corrector of the difference between the desired set-point and the measured output.

#### 3.1 Controller Design

The error of the system,  $e(t)$  is defined in (12) and  $u_t$  is the control input, while PID controller output relation,  $x_d(t)$  is defined in (13), and  $x(t)$  is the present state or measured value [7 – 21]. Figure 8 shows the simulation blocks of PID controller.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (12)$$

$$e(t) = x_d(t) - x(t) \quad (13)$$

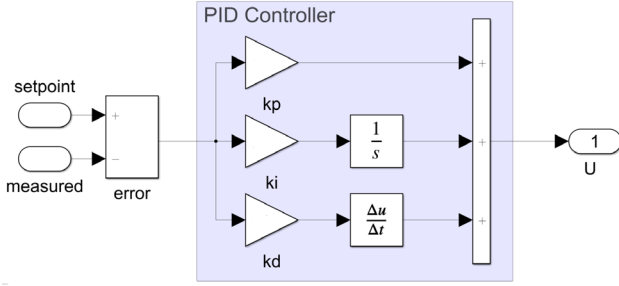


Figure 8: The Simulink blocks of PID Controller.

From equation (12), the  $K_p$  is proportional gain,  $K_i$  is integral gain and  $K_d$  is derivative gain. To determine the value of each gain, manual adjustment, which is a trial-and-error method, is applied according to the steps listed below [12]:

1. The gains of  $k_p$ ,  $k_i$  and  $k_d$  are initially set to zero.
2. The value of  $k_p$  is increased until the oscillation is fully or nearly sustained. It is recommended to use a high simulation stop time.
3. The value of  $k_d$  is increased slightly to reduce the oscillation to only 1 period. To simplify the tuning process, it is recommended to run the simulation at a lower stop time.
4. The value of  $k_i$  is increased until the steady state settled around the setpoint with only one oscillation left around the setpoint.

Each increment made to the values of gains affect the output performance. Table 3 summarized the effects of manual tuning applied to the gains value [16].

Table 3: Effects of increasing values of PID gains.

Parameter	$k_p$	$k_i$	$k_d$
Overshoot (%)	Increase	Increase	Reduce
Rise time (s)	Reduce	Reduce	Slight Changes
Settling time (s)	Slight Changes	Increase	Reduce
Steady-state error	Reduce	Eliminate	Theoreticaly No Effect
Stability	Degrade	Degrade	Improved (small $k_d$ )

### 3.1.1 Altitude Controller

The altitude is represented by  $z$  and is described in (8). To initiate a movement of quadcopter along  $z$ -axis, thrust force must be generated by all four rotors.  $U_1$  is the control input for altitude, and the PID controller for altitude is derived as in (14).

$$U_{1PID} = k_p^z e(t)_z + k_i^z \int e(t)_z dt + k_d^z \dot{e}(t)_z \quad (14)$$

$$e(t) = z_d(t) - z(t) \quad (15)$$

### 3.1.2 Attitude Controller

The derivation of the attitude PID controller is formulated as in (16).  $U_2$  is the control input of roll,  $\phi$ ,  $U_3$  is the control input of pitch,  $\theta$  and  $U_4$  is the control input of yaw,  $\psi$ .  $n$  is  $(\phi, \theta, \psi)$ .

$$U_{2,3,4PID} = k_p^n e(t)_n + k_i^n \int e(t)_n dt + k_d^n \dot{e}(t)_n \quad (16)$$

$$e(t)_n = n_d(t) - n(t) \quad (17)$$

### 3.1.3 Position Controller

To realize trajectory tracking of quadcopter, a manipulation of  $x$  and  $y$  positions need to be made with the roll,  $\phi$  and pitch,  $\theta$  angles as  $x$  and  $y$  positions cannot be controlled with  $U_1$  [17]. Quadcopter operates at hovering position which make the angle of roll and pitch are small-angle values. Therefore, the dynamics equations of the  $x$  and  $y$  positions are simplified based on the small angle assumptions ( $s\phi_d \cong \phi_d$ ,  $s\theta_d \cong \theta_d$ ,  $c\phi_d = c\theta_d = 1$ ). Thus, equations (18) and (19) are manipulated to derive equation of PID controller for positions (20)

$$\ddot{x} = \frac{U_1(s\psi\phi_d + c\psi\theta_d)}{m} \quad (18)$$

$$\ddot{y} = \frac{U_1(s\psi\theta_d - c\psi\phi_d)}{m} \quad (19)$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \frac{U_1}{m} \begin{bmatrix} s\psi & c\psi \\ -c\psi & s\psi \end{bmatrix} \begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} \quad (20)$$

$$\phi_d = (u_x s\psi_d - u_y c\psi_d) \quad (21)$$

$$\theta_d = (u_x c\psi_d + u_y s\psi_d) \quad (22)$$

$\phi_d$  is the desired roll, and  $\theta_d$  is the desired pitch.  $u_x$  and  $u_y$  are inputs control signal which both are then designed to be used in PID controller. Equation (23) is the final PID equation for position control.  $v$  is  $(x, y)$ . Figure 9 shows the blocks of simulation for position PID control.

$$u_{x,yPID} = k_p^v e(t)_v + k_i^v \int e(t)_v dt + k_d^v \dot{e}(t)_v \quad (23)$$

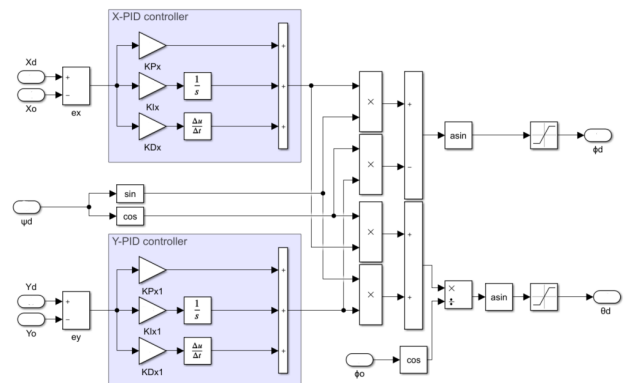


Figure 9: The Simulink block for the position PID controller.

Based on the illustration of PID controller of quadcopter in Figure 10, the complete model of quadcopter using PID controller is simulated as in Figure 11.

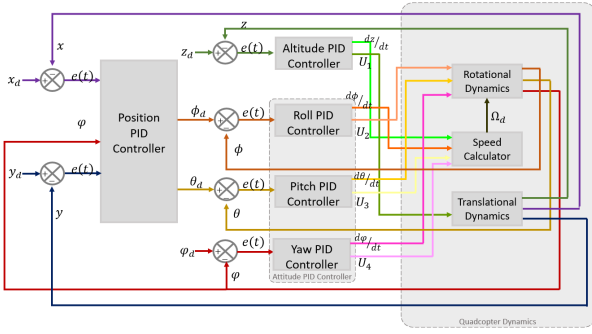


Figure 10: The illustration of quadcopter plant with PID controller.

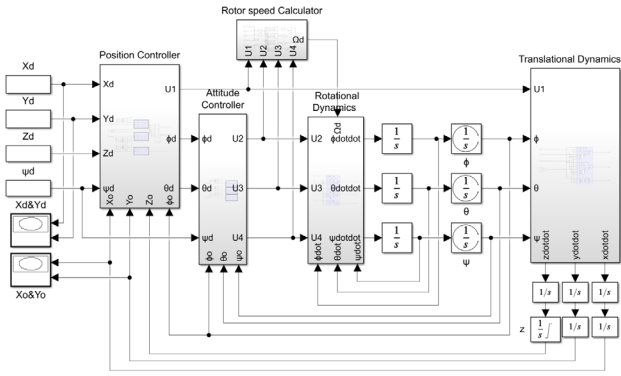


Figure 11: The Simulink blocks of PID control quadcopter.

#### 4. RESULT AND DISCUSSION

Based on the derived model and the designed PID controller, simulations for quadcopter tracking are developed in MATLAB/Simulink environment. The time step for each simulation is set to 0.01s. Simulations of circular, square, lemniscate, zigzag, and spiral trajectories are carried out to test the system.

##### 4.1 Trajectory Tracking

For trajectory tracking, the designed model and controller are tested with five types of trajectories. In trajectories tracking, the desired roll and desired pitch are no longer defined as in altitude and attitude tracking. The only defined desired values are altitude and yaw. For the five types of trajectories, the altitude is set from the initial value, 0 to achieve the final value, 1, while the yaw value is set with an initial value of 0.

The desired positions,  $x$ , and  $y$  are set individually for each type of trajectory. Table 4 listed the gains values used for trajectories tracking.

Table 4: PID gains values for tracking trajectories.

Gains	Position ( $x, y$ )	Altitude ( $z$ )	Attitude ( $\phi, \theta, \psi$ )
$K_p$	2.0	35.0	13.0
$K_i$	0.3	14.5	0.5
$K_d$	4.0	9.0	1.5

The values of gains listed in Table 4 are obtained through trial-and-error, according to the steps listed in

Section 3.1. The process of trial-and-error is also called as the manual tuning process. This process consumed several attempts, and some time until satisfying results of trajectories tracking are obtained. The manual tuning method is chosen for this paper as the trials on utilizing Ziegler-Nichols' method resulted in failure, in which indicate the method is not applicable with the proposed model.

For circular trajectory, the positions are set as  $x_d = 10\sin(\frac{\pi}{10}t)$  and  $y_d = 10\cos(\frac{\pi}{10}t)$ . Figure 12 shows the output response of circular trajectory tracking. The simulation time is set to 120s.

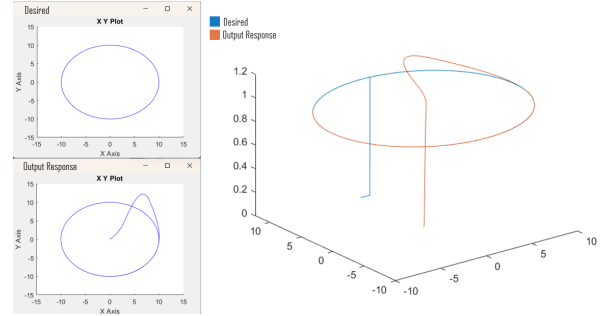


Figure 12: The output response of circular trajectory.

For the square trajectory, the desired inputs of position are set as  $x_d = [0 \ 0 \ 6 \ 6 \ 0 \ 0]$  for  $t_x = [0 \ 30 \ 50 \ 85 \ 105 \ 120]$  and  $y_d = [0 \ 0 \ 6 \ 6 \ 0 \ 0]$  for  $t_y = [0 \ 5 \ 25 \ 55 \ 80 \ 120]$ . The output response for square trajectory is shown in Figure 13. The simulation time is set to 120s.

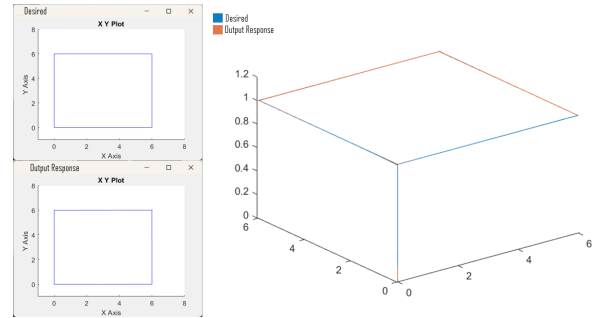


Figure 13: The output response of square trajectory.

For the lemniscate trajectory, the desired position inputs are set to  $x_d = 10\sin(\frac{\pi}{10}t)$  and  $y_d = 10\cos(\frac{\pi}{20}t)$ . Figure 14 shows the output response of lemniscate trajectory tracking. The simulation time is set to 150s.

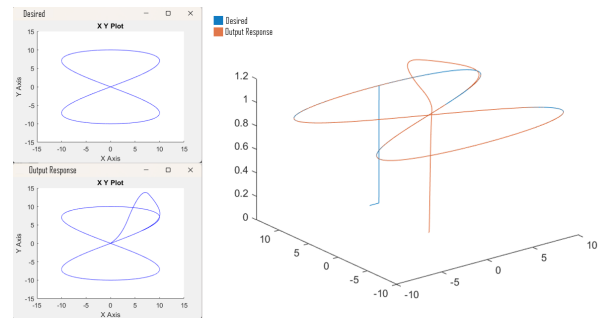


Figure 14: The output response of lemniscate tracking.



The desired positions output for zigzag trajectory are set as  $x_d = [0 \ 5 \ 0 \ 5 \ 0 \ 5 \ 0 \ 5 \ 0 \ 5 \ 0]$  for  $t_x = [0 \ 10 \ 20 \ 30 \ 40 \ 50 \ 60 \ 70 \ 80 \ 90 \ 100 \ 110 \ 120]$  and  $y_d = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0]$  for  $t_y = [0 \ 10 \ 20 \ 30 \ 40 \ 50 \ 60 \ 70 \ 80 \ 90 \ 100 \ 110 \ 120]$ . The output response is shown in Figure 15. The simulation time is set to 120s.

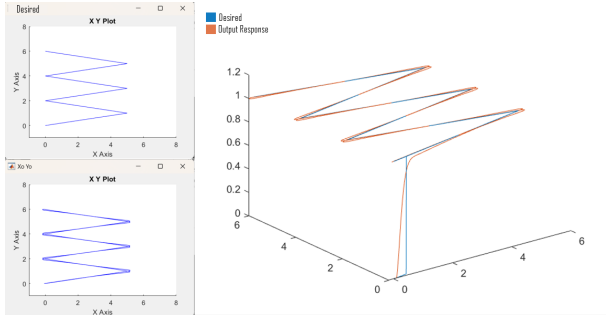


Figure 15: The output response for zigzag trajectory.

For the spiral trajectory, the altitude value is changed to a ramp with slope of 0.5 and the desired positions desired values are set to  $x_d = 10\sin(\frac{\pi}{10}t)$  and  $y_d = 10\cos(\frac{\pi}{10}t)$ , which are similar to the circular trajectory. The spiral trajectory output is shown in Figure 16. The simulation time is set to 120s.

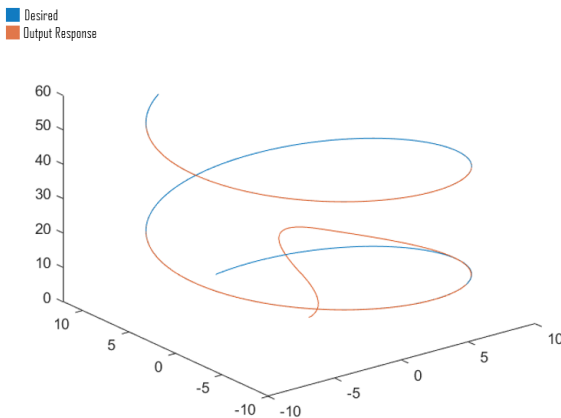


Figure 16: The output response of the spiral trajectory.

Table 5 listed the values of characteristic of the trajectories tracking responses. These values are analyzed used as the performance indicator of the system.

Table 5: Characteristic values of trajectories tracking.

Trajectory		Overshoot (%)	Rise time (s)	Settling Time (s)
Circular	$x$	86.4453	4.5006	119.6297
	$y$	44.9440	2.4610	119.3452
	$z$	0.0157	3.0721	0.4979
Square	$x$	$2.6901e^5$	28.9501	104.5876
	$y$	$4.3178e^8$	$3.2744e^{-4}$	79.4962
	$z$	0.0215	0.4989	3.0748
Lemniscate	$x$	54.9596	4.5138	149.5572
	$y$	298.5310	1.3504	149.4226
	$z$	$1.1027e^{-5}$	0.4979	3.0724
Zigzag	$x$	142.6944	0.1163	119.7871
	$y$	0.0000	$2.2933e^{-5}$	118.7988
	$z$	0.0580	0.4971	3.0071

Spiral	$x$	86.4453	4.5008	119.6311
	$y$	42.8905	2.6534	119.3453
	$z$	0.000	95.9987	117.60

## 4.2 Discussion

Based on the results obtained by simulations, the desired setpoints for all tested trajectories are successfully achieved by the system. Most responses recorded high settling time for positions  $x$  and  $y$ , as it indicates that the settling time for positions is the simulation time as the trajectories does not settle at a certain position of  $x$  and  $y$  unless the simulations is done. The settling time recorded by  $x$  and  $y$  of all trajectories are showing a very close value of the simulation time. For example, circular trajectory tracking is simulated with 120s of simulation time, and the settling time for position  $x$  and  $y$  of circular trajectory is 120s approximately. The same case can be observed in all trajectories. Therefore, for performance evaluations, the overshoot percentage and rise time can be used as the indicator for performance evaluations.

On top of that, except for spiral trajectories, the overshoot percentage, rise time and settling time of altitude,  $z$  can also be used as indicator as each simulation start with altitude of 0 and eventually reached and completed the simulation in hovering at altitude of 1.

However, the settling time of the altitude should not be taken into account for spiral trajectories, as the simulation does not have a settling point. This is proven by the settling time of altitude,  $z$  recorded in spiral trajectory has a value with no big difference to the simulation time.

In general, even though the system is proven to be able to achieve the desired setpoints, each response recorded overshoot percentage which indicate there is overshoot occurred in the response of the system. In addition, the trial-and-error method requires a long time to determine the best gains values. Therefore, a proper tuning or intelligence techniques should be considered to improve the performance of the responses.

## 5. CONCLUSION

This work aimed to derive a mathematical model of a quadcopter and design a controller that controls the operation and stability of a quadcopter during its flight. As quadcopters have been used extensively in various fields, it is important to ensure that the quadcopter has the ability to track trajectories. PID controller is used to control the modelled quadcopter. The responses of different trajectories tracking simulations show the quadcopter able to track the desired trajectories successfully with PID controller. However, each simulation experiences an overshoot. Each response recorded either a nonzero value or a quite high value of overshoot percentage. Therefore, an improvement should be made to improve the performance of the system.

## ACKNOWLEDGEMENT

The authors would like to thank Universiti Teknologi Malaysia under UTMFR (Q.J130000.3823.22H67) for supporting this research.



## REFERENCES

- [1] Uddin, M. (2020). Drone 101: A Must-Have Guide for Any Drone Enthusiast
- [2] Kardasz, P., Doskocz, J., Hejduk, M., Wiejkut, P., & Zarzycki, H. (2016). Drones and the possibilities of their using. *J. Civ. Environ. Eng.*, 6(3), 1-7.
- [3] Custers, B. (2016). Drones Here, There and Everywhere Introduction and Overview. In (Vol. 27, pp. 3-20).
- [4] Idrissi, M., Salami, M., & Annaz, F. (2022). A Review of Quadrotor Unmanned Aerial Vehicles: Applications, Architectural Design and Control Algorithms. *Journal of Intelligent & Robotic Systems*, 104(2), 1-33
- [5] Rabah, Mohamed & Rohan, Ali & Han, Yun-Jong & Kim, Seunghae. (2018). Design of Fuzzy-PID Controller for Quadcopter Trajectory-Tracking. *INTERNATIONAL JOURNAL of FUZZY LOGIC and INTELLIGENT SYSTEMS*. 18. 204-213. 10.5391/IJFIS.2018.18.3.204.
- [6] Han, B., Zhou, Y., Deveerasetty, K. K., & Hu, C. (2018). A review of control algorithms for quadrotor. Paper presented at the 2018 IEEE International Conference on Information and Automation (ICIA).
- [7] Noordin, A., Mohd Basri, M. A., Mohamed, Z., & Mat Lazim, I. (2021). Adaptive PID controller using sliding mode control approaches for quadrotor UAV attitude and position stabilization. *Arabian Journal for Science and Engineering*, 46(2), 963-981.
- [8] Azar, A. T., Koubaa, A., Ali Mohamed, N., Ibrahim, H. A., Ibrahim, Z. F., Kazim, M., . . . Hameed, I. A. (2021). Drone deep reinforcement learning: A review. *Electronics*, 10(9), 999.
- [9] Ghazbi, S. N., Aghli, Y., Alimohammadi, M., & Akbari, A. A. (2016). Quadrotors unmanned aerial vehicles: A review. *International journal on smart sensing and Intelligent Systems*, 9(1).
- [10] Bayisa, A. T., & Li-Hui, G. (2019). Controlling Quadcopter Altitude using PID-Control System. *International Journal of Engineering Research & Technology (IJERT)*, 8, 195-199
- [11] Zhang, Z. (2020, 18-20 March 2020). Application of PID Simulation Control Mode in Quadrotor Aircraft. Paper presented at the 2020 International Conference on Computer Engineering and Application (ICCEA).
- [12] Sahrir, N. H., & Mohd Basri, M. A. (2022). Modelling and Manual Tuning PID Control of Quadcopter. In *Control, Instrumentation and Mechatronics: Theory and Practice* (pp. 346-357): Springer.
- [13] Paiva, E., Soto, J., Salinas, J., & Ipanaqué, W. (2016, 19-21 Oct. 2016). Modeling, simulation and implementation of a modified PID controller for stabilizing a quadcopter. Paper presented at the 2016 IEEE International Conference on Automatica (ICA-ACCA).
- [14] Xi, C., & Wang, L. (2016, 3-4 Nov. 2016). Quadrotor cascade PID controller automatic tuning. Paper presented at the 2016 Australian Control Conference (AuCC).
- [15] Cedro, L., & Wiczorkowski, K. (2019). Optimizing PID controller gains to model the performance of a quadcopter. *Transportation Research Procedia*, 40, 156-169.
- [16] Cheng, H., & Yang, Y. (2017, 18-20 June 2017). Model predictive control and PID for path following of an unmanned quadrotor helicopter. Paper presented at the 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA).
- [17] Satla, Z., Elajrami, M., & Bendine, K. (2018). Easy Tracking of UAV Using PID Controller. *Periodica Polytechnica Transportation Engineering*, 47.
- [18] Surriani, A., & Arrofiq, M. (2017). Altitude control of quadrotor using fuzzy self-tuning PID controller. Paper presented at the 2017 5th International Conference on Instrumentation, Control, and Automation (ICA).
- [19] Usman, M. (2020). Quadcopter Modelling and Control With MATLAB/Simulink Implementation.
- [20] Akbari-Hasanjani, R., Javadi, S., & Sabbaghi-Nadooshan, R. (2015). DC motor speed control by self-tuning fuzzy PID algorithm. *Transactions of the Institute of Measurement and Control*, 37(2), 164-176.
- [21] Sattar, M., & Ismail, A. (2017). PID Control of a Quadrotor UAV. *International research Journal of Engineering and Technology*, 4(8), 1490-1493.
- [22] Ang, K. H., Chong, G., & Li, Y. (2005). PID control system analysis, design, and technology. *IEEE transactions on control systems technology*, 13(4), 559-576.