Modelling and Manual Tuning PID Control of Quadcopter

Chapter · July 2022

DOI: 10.1007/978-981-19-3923-5_30

2 authors:

Nur Hayati Binti Sahrir
Universiti Teknologi Malaysia
5 PUBLICATIONS **25** CITATIONS
SEE PROFILE

Ariffanan Basri
Universiti Teknologi Malaysia
84 PUBLICATIONS **1,229** CITATIONS
SEE PROFILE

Modelling and Manual Tuning PID Control of Quadcopter

Nur Hayati Sahrir and Mohd Ariffanan Mohd Basri(✉)

School of Electrical Engineering, Faculty of Engineering,
Universiti Teknologi Malaysia, UTM, 81310 Johor Bahru, Johor, Malaysia
ariffanan@fke.utm.my

Abstract. As the most common unmanned aerial vehicle either in the industry or public, quadcopter has gained a significant interest for future technological developments. There are vast applications of quadcopter such as aerial photography and videography, involved in search and rescue missions, spying and more. A quadcopter is under-actuated where there are six types of motion but only has four rotors to control the motions. In this paper, mathematical modelling of a quadcopter is formulated through the fundamental of Newton-Euler method. Varying the speed of the four rotors can produce thrust, roll, pitch and yaw torque which results in specific movements of the quadcopter. The Proportional-Integral-Derivative (PID) controller is employed in this study due to its simplicity and easy to design. The PID parameters are tuned using manual tuning technique. The quadcopter model is built and simulated with PID controllers using MATLAB Simulink. The simulation results demonstrated the effectiveness of the proposed well-tuned PID controller for altitude and attitude stabilization of the quadcopter.

Keywords: Modelling · PID controller · Quadcopter

1 Introduction

There are several types of Unmanned Aerial Vehicle (UAV)s across the world. A quadcopter or quadcopter, is one type of the multi-rotor drone's family. For drones, there are fixed-wing drone, multi-rotor drone, single rotor drone and hybrid drone. Where a quadcopter falls into the multi-rotor drone's category, there are also other type of multi-rotor drones which are tricopter, hexacopter and octocopter which each has three, six and eight propellers respectively. Quadcopter, on the other hand, has four propellers and is more common and widely commercialized in the market. A quadcopter has better stabilization mechanism compared to a tricopter and less complex with fewer parts and lower costs than hexacopter and octocopter. As the number of propellers increase, the drone can lift more payload. However, it also causes the drone to house a heavier battery as there are higher current draw for flight endurance.

As the name suggests, a quadcopter has four motors which each located equidistant from the center of body frame. Each motor actuates a rotor or propeller and they are

N. A. Wahab and Z. Mohamed (Eds.): *Control, Instrumentation and Mechatronics: Theory and Practice*, LNEE 921, pp. 346–357, 2022.
https://doi.org/10.1007/978-981-19-3923-5_30

being used for vertical take-off, maneuver the quadcopter in the air, and landing [1]. This small yet popular and useful has come in handy for several applications such as search and rescue after disasters, replace humans in sites which are harmful for humans [2], aerial surveillance [3] environmental protection, management of large infrastructures, carrying certain payloads for deliveries, recreational purposes [1] and aerial photography or videography [3, 4].

Several methods of controlling a quadcopter have been proposed from several decades before until recent years. Those are Linear Quadratic Regulator (LQR) Controller [5], Proportional-Integral-Derivative (PID) Controller [4], Fuzzy Logic Controller [6], Feedback Linearization Control, Sliding Mode Control [7] and Backstepping Control [8]. In this research, a PID controller will be used as it is widely used in practical applications. For high-performance devices like the quadcopter, PID controllers can only be utilized for plants with a short time delay [4]. Determination of the controller gains is the key for designing a PID controller.

This paper aims to design a quadcopter controller and simulate the design using MATLAB Simulink based on existing works. Since a quadcopter is an under-actuated model, this research will be going with independent outputs (z, ϕ, θ, ψ) until the end. The controller to be used is PID controller with manual tuning method.

2 Modelling of the Quadcopter

A quadcopter is an under-actuated UAV where it has only four rotors to control six degrees-of-freedom (DOF) movement. Figure 1 shows a detailed configuration of a quadcopter. It can be seen that two opposite rotors (F1 and F3) are rotating clockwise while the other two (F2 and F4) rotates counterclockwise. This is based on Newton's Third Law which states that 'For every action, there is an equal and opposite direction'. This is to ensure a balance state of hovering of the quadcopter where the speed of all rotors is the same but each two has opposite rotating direction, cancelling out the rotating torque. Due to the torque acting in opposite direction of each propeller, if all rotors happen to rotate clockwise, the body of the quadcopter will tend to rotate counterclockwise and out of balance state of hovering. Several assumptions are being made to ease the quadcopter modelling [9]:

1. The whole structure of quadcopter is assumed rigid and symmetrical with all four rotors.
2. The quadcopter body frame origin and center of mass coincides.
3. Aerodynamic effects and surrounding wind disturbances are negligible at low speed.
4. A relatively fast rotor dynamics but rotors are considered rigid, so blade flapping is neglected.

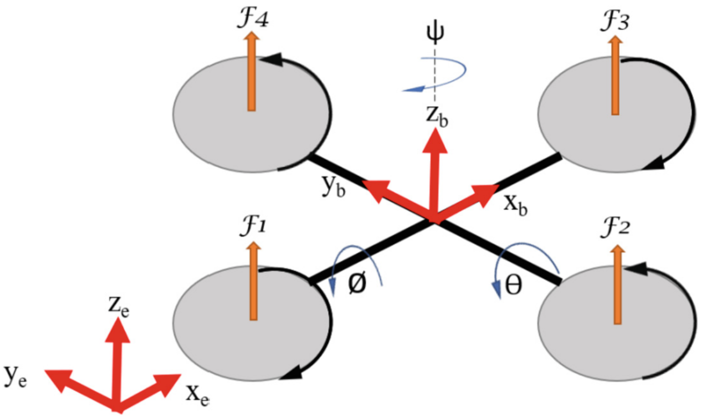

Fig. 1. The quadcopter configuration

There are two frame axes taking part in this system, the earth frame and the body frame. The earth frame is fixed regardless the location of the quadcopter while the body frame always located at the center of quadcopter with respect to the earth frame. The position of the quadcopter is controlled with respect to the earth frame and the orientation is controlled with respect to the body frame [3]. The forces (F1, F2, F4, F4) shown in the figure are the lift forces (or thrust forces if the quadcopter moves upward). Conventionally, East, North, Up (ENU) coordinate system is used where x-axis points East, y-axis points North and z-axis points upwards.

In the quadcopter, the four rotors which are fixed-located equidistant from the center of quadcopter, are the keys for maneuvering the quadcopter itself. A total sum of the thrust force of all rotors is the input, u_1. There are translational (x, y, z) and rotational motion of the quadcopter (ϕ, θ, ψ). The roll (ϕ) movement happens when the speed of rotor 2 increase (decreases) while the speed of rotor 4 decreases (increases), produces an angular rotation along x-axis. The pitch (θ) movement occurs when the speed of rotor 1 increase (decreases) while the rotor 3 decreases (increases), produces an angular rotation along y-axis. The yaw (ψ) movement happened when the speed of rotor 1 and rotor 3 increase (decreases) while the speed of rotor 2 and rotor 4 decreases (increases), produces rotation along z-axis and causes an imbalance in torque. While keeping the total thrust at constant, the quadcopter should move around at the same altitude (z) with changing attitude (ϕ, θ, ψ).

For roll and pitch orientation, even though the speed of one rotor increases while the speed of the opposite same direction rotor decreases, it will not affect yaw instead, it will move to another position depending on the angular rate of roll and pitch. A clearer visualization of the quadcopter movement based on changing altitude and attitude is shown in [10].

The translational dynamic equation of this quadcopter is obtained from Newton's Second Law and Euler's Rotational Equation of Motion and is defined as below [9]:

$$\mathrm{m}\ddot{\Gamma} = u_T R e_z - m g e_z \tag{1}$$

where, m is the quadcopter mass, R is the rotation matrix, g is the gravitational acceleration, $e_z = (0\ 0\ 1)^T$ is the unit vector and u_T is the total thrust of all four rotors.

The rotation matrix of the body frame with respect to the earth frame is given by [11]:

$$R_b^e = \begin{bmatrix} C\theta C\psi & S\phi S\theta C\psi - S\psi C\phi & S\theta C\phi C\psi + S\phi S\psi \\ S\psi C\theta & S\phi S\theta S\psi + C\phi C\psi & S\theta S\psi C\phi - S\phi C\psi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{bmatrix} \tag{2}$$

where, C(angle) represents cosine while S(angle) represents sine. Three Euler angles namely roll angle (ϕ), pitch angle (θ) and yaw angle (ψ) form the orientation of the quadcopter.

The rotational dynamic equation of this quadcopter is given by [9]:

$$I\dot{\omega} = -\omega \times I\omega - J_r(\omega \times e_z)\Omega_r + \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{3}$$

where, it represents all airframe body torque, gyroscopic torque and Coriolis torque. I is the quadcopter inertia matrix, $\dot{\omega}$ is the angular acceleration vector, ω is the angular velocity vector, J_r is the rotor's inertia, Ω_r is the relative speed of all rotors and $\tau_\phi, \tau_\theta, \tau_\psi$ represent roll, pitch and yaw torque respectively.

Based on Eq. (1) and (3), the produced translational and rotational equation are as shown below:

$$\begin{aligned} \ddot{x} &= \tfrac{1}{m}(S\theta C\phi C\psi + S\phi S\psi)U_1 \\ \ddot{y} &= \tfrac{1}{m}(S\theta C\psi C\phi + S\phi S\psi)U_1 \\ \ddot{z} &= \tfrac{1}{m}(C\phi C\theta)U_1 - g \\ \ddot{\phi} &= \tfrac{1}{I_{xx}}\left[\left(I_{yy} - I_{zz}\right)\dot{\theta}\dot{\psi} - J_r\dot{\theta}\Omega_r + lU_2\right] \\ \ddot{\theta} &= \tfrac{1}{I_{yy}}\left[(I_{zz} - I_{xx})\dot{\phi}\dot{\psi} + J_r\dot{\phi}\Omega_r + lU_3\right] \\ \ddot{\psi} &= \tfrac{1}{I_{zz}}\left[\left(I_{xx} - I_{yy}\right)\dot{\theta}\dot{\phi} + U_4\right] \end{aligned} \tag{4}$$

Note: $\Omega_r = (\omega_1 - \omega_2 + \omega_3 - \omega_4)$.

The speed of each rotor is obtained from Eq. (5) [1]. As can be seen in Fig. 2, the speeds produced are used to generate a relative speed, Ω_r, which is required for quadcopter dynamics as shown in Eq. (4).

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} K_T & K_T & K_T & K_T \\ 0 & -\updownarrow K_T & 0 & \updownarrow K_T \\ \updownarrow K_T & 0 & -\updownarrow K_T & 0 \\ K_d & -K_d & K_d & -K_d \end{bmatrix}^{-1} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{5}$$

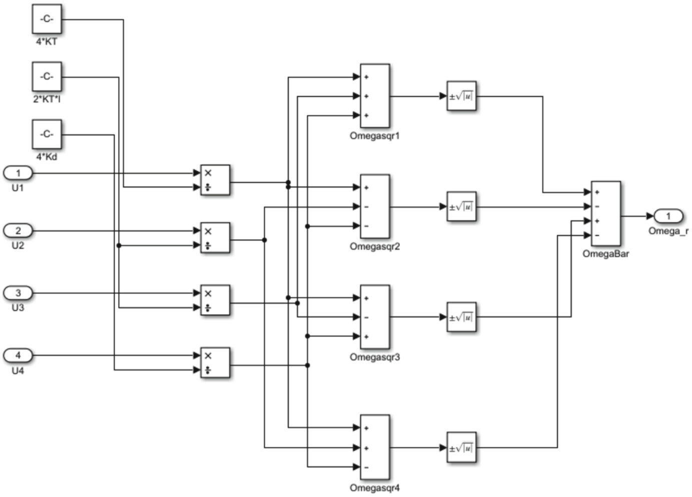

Fig. 2. Simulation to produce relative speed, Ω_r

3 PID Controller

In this paper, PID controller is used to attenuate the error produced from the difference of actual state and desired state [13]. This control system is also known as closed-loop feedback system where it helps to maintain the output as nearly the same as the desired state. This shows the robustness of the system. Both altitude and attitude controller use PID controller to give out control inputs U1, U2, U3 and U4. All of these control inputs are the fundamental in producing the outputs z, ϕ, θ, ψ.

Since this system has four PID controllers, the tuning process will start with altitude controller first, while disconnecting the other three control inputs. The following equations shows the PID controller equations to obtain all control inputs U1, U2, U3, U4 which then used to produce the relative speed, Ω_r. Control inputs U2, U3 and U4 are produced by utilizing the angular error while U1 utilizes the altitude error [14].

$$\begin{gathered} U_1(t) = K_P e_z(t) + K_I \int_0^t e_z(t)dt + K_D \frac{e_z(t) - e_z(t-1)}{dt} \\ U_2(t) = K_P e_\phi(t) + K_I \int_0^t e_\phi(t)dt + K_D \frac{e_\phi(t) - e_\phi(t-1)}{dt} \\ U_3(t) = K_P e_\theta(t) + K_I \int_0^t e_\theta(t)dt + K_D \frac{e_\theta(t) - e_\theta(t-1)}{dt} \\ U_4(t) = K_P e_\psi(t) + K_I \int_0^t e_\psi(t)dt + K_D \frac{e_\psi(t) - e_\psi(t-1)}{dt} \end{gathered} \tag{6}$$

The steps for PID gains manual tuning starting with altitude controller are as described below:

(1) Set the gain of K_P, K_I, and K_D to be zero.

(2) Increase K_P until the oscillations are fully sustained or nearly sustained (if fully sustained is impossible to reach) as shown in Fig. 3. This method is recommended to use a high simulation stop time (in this case, it is 100 s).

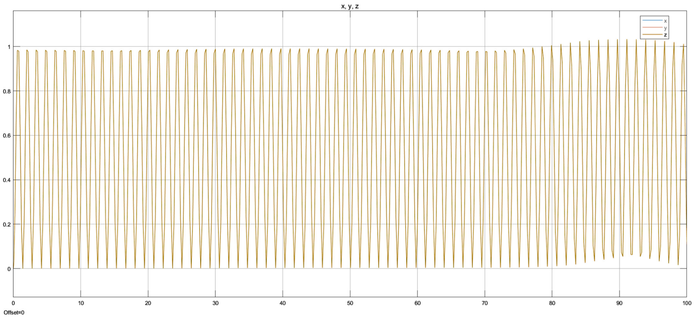

Fig. 3. Increase K_P value

(3) Increase K_D slightly to make the oscillations reduced to only 1 period as shown in Fig. 4. Run the simulation at lower simulation stop time (in this case, it is 10 s). This is to make tuning process easier and clearer.

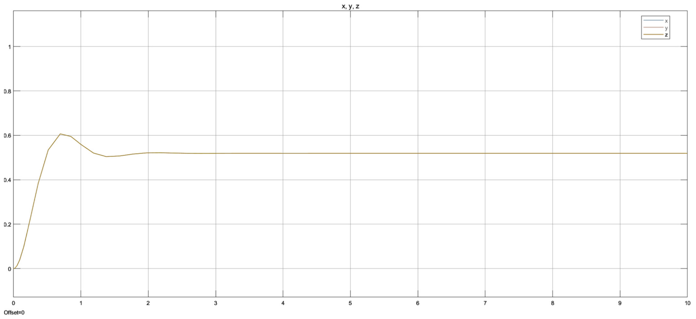

Fig. 4. Increase K_D value

(4) Increase K_I until the steady state is just around the setpoint and the single oscillation to oscillate around the setpoint too (in this case, the setpoint is 1) as can be seen in Fig. 5.

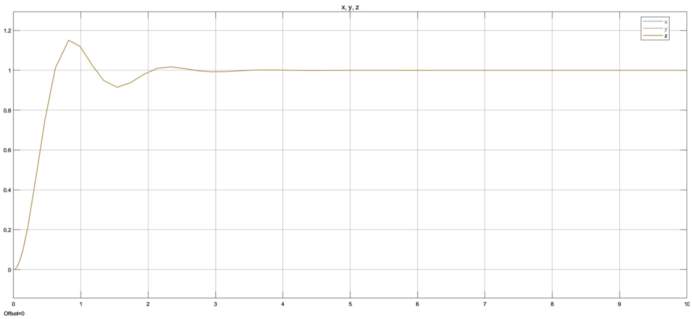

Fig. 5. Increase K_I value

(5) Table 1 is referred for further tuning. Specifically, increasing K_P to make the steady-state error goes zero. Increasing K_I to eliminate error and settles to the setpoint. Increasing K_D to reduce the overshoots until a smooth output is achieved as can be seen in Fig. 6.

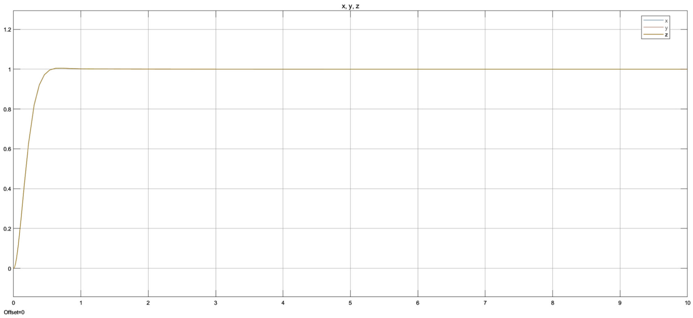

Fig. 6. Fine-tuned altitude controller with settling time at 2.282 s and zero e_{ss}

Table 1. Effects of increasing a parameter independently

Parameter	Rise time	Settling time	%OS	ESS	Stability
K_P	Decrease	Small change	Increase	Decrease	Degrade
K_I	Decrease	Increase	Increase	Eliminate	Degrade
K_D	Minor change	Decrease	Decrease	No effect in theory	Improve if K_D is small

(6) Repeat steps 1–5 for attitude controllers starting with roll (phi) controller, pitch (theta) and lastly yaw (psi).

The stability of this system is assured when the output can follow the input reference with zero steady-state error, or at least, the output stays around the given input reference without going further away from the setpoint by time. This has been partially proved in Fig. 6 and will be fully proved in the next section. Figure 7 and 8 show the Simulink blocks for controlling altitude and attitude using PID approach.

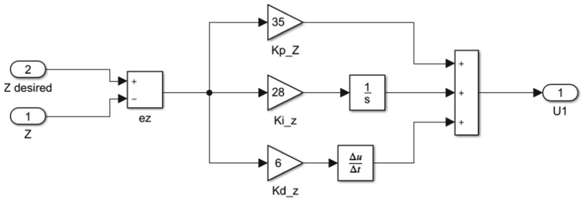

Fig. 7. Simulation blocks to produce altitude control input, U1

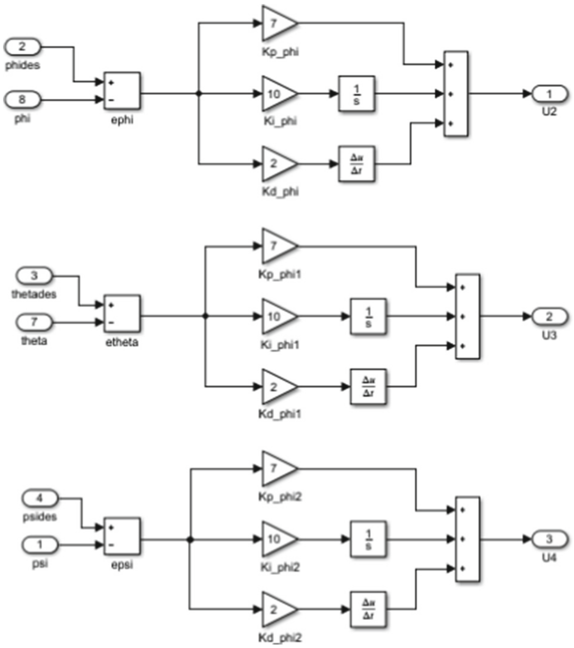

Fig. 8. Simulation blocks to produce attitude control inputs, U2, U3 and U4

4 Simulation Results

The final top level of quadcopter simulation using MATLAB Simulink is as shown in Fig. 9 The physical parameters are depicted in Table 2.

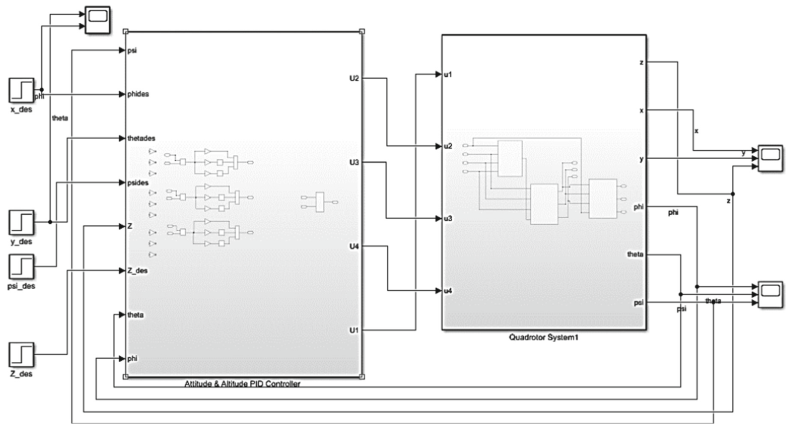

Fig. 9. Top-level quadcopter simulation blocks using Simulink

Table 2. Parameters of the quadcopter model [12]

Parameter	Value	Unit
g	9.81	$\mathrm{m} \cdot \mathrm{s}^2$
m	0.5	kg
↕	0.2	m
$J_x = J_y$	4.85×10^{-3}	$\mathrm{kg} \cdot \mathrm{m}^2$
J_z	8.81×10^{-3}	$\mathrm{kg} \cdot \mathrm{m}^2$
J_r	3.36×10^{-5}	$\mathrm{kg} \cdot \mathrm{m}^2$
K_T	2.92×10^{-6}	$\mathrm{kg} \cdot \mathrm{m}$
K_d	1.12×10^{-7}	$\mathrm{kg} \cdot \mathrm{m}^2$

The objectives of tuning the PID controller are to minimize the rise time, maximum overshoot and the settling time in order to achieve a better stability of the system [15]. By using manual PID tuning method and following all fundamental equations of a quadcopter with specific parameters, the results are obtained as shown in Table 3 below.

Table 3. Results of simulation

Test	Attributes & respective PID gains	Rise time, T_r (s)	Settling time, T_s (s)	Percent overshoot (%OS)
P-D	Altitude(z): Kp = 35, Ki = 0, Kd = 6	0.248	1.962	5.851
	Roll (ϕ): Kp = 7, Ki = 0, Kd = 2	0.611	2.621	0.501
	Pitch (θ): Kp = 7, Ki = 0, Kd = 2	0.485	2.214	0.490
	Yaw (ψ): Kp = 7, Ki = 0, Kd = 2	0.612	2.970	0.497
P-I-D	Altitude(z): Kp = 100, Ki = 190, Kd = 15	0.187	2.573	15.698
	Roll (ϕ): Kp = 100, Ki = 39, Kd = 27	0.476	16.642	6.989
	Pitch (θ): Kp = 100, Ki = 39, Kd = 27	0.467	14.511	6.989
	Yaw (ψ): Kp = 100, Ki = 39, Kd = 27	0.476	13.009	6.989
P-I-D	Altitude(z): Kp = 35, Ki = 28, Kd = 6	0.279	3.677	3.646
	Roll (ϕ): Kp = 7, Ki = 10, Kd = 2	0.359	4.772	19.880
	Pitch (θ): Kp = 7, Ki = 10, Kd = 2	0.295	4.443	17.059
	Yaw (ψ): Kp = 7, Ki = 10, Kd = 2	0.361	4.907	18.452

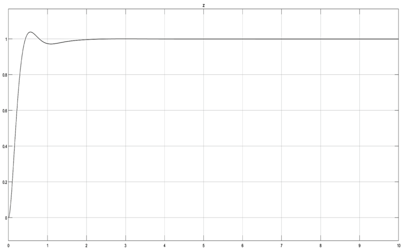

Fig. 10. Z-position

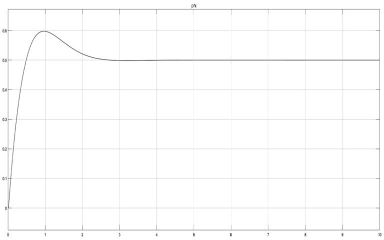

Fig. 11. Roll position

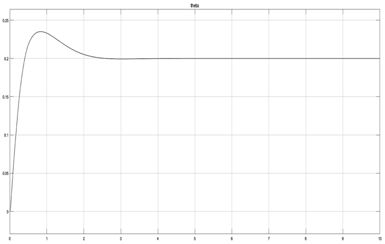

Fig. 12. Pitch position

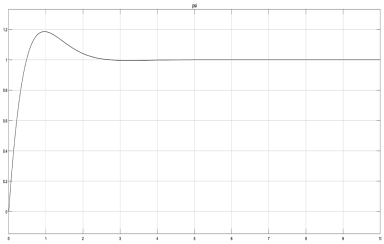

Fig. 13. Yaw position

From the results, it can be seen that by utilizing only P-D controller, the percent overshoot and settling time improve but altitude controller cannot achieve the desired step input. By including integral gains into the system, it eliminates errors and settle to the setpoint with a steady state error and the system accurately follows desired setpoint [16]. In this project, the final test in Table 3 is the most ideal PID tuning which the settling time is quite fast with zero steady state error and high stability over time as proved in Fig. 10, 11, 12 and 13.

5 Conclusions and Future Work

This paper has clearly explained the basic modelling of a quadcopter from the fundamental physics and robotics until the final building the simulation blocks. The quadcopter model is designed to only give out the actual altitude and attitude of the quadcopter which are the z position, roll, pitch and yaw. This is because those are independent to each other and controlling them perfectly is one step before continuing to control the x and y positions. In the future work, a position controller will be designed and included in this model to produce x and y actual output based on desired position. Furthermore, a trajectory tracking also will be implemented along with improved intelligent PID controller and stabilization in nonlinear environment to refurbish this model into an autonomous quadcopter. In order to further improve the trajectory tracking and autonomous feature of this quadcopter, machine vision will be considerably implemented.

Acknowledgments. The authors would like to thank Ministry of Education (MOE) through Fundamental Research Grant Scheme, FRGS (R.J130000.7851.5F439) for supporting this research.

References

1. Iyer, A., Bansal, H.O.: Modelling, simulation, and implementation of PID controller on quadcopters. In: International Conference on Computer Communication Informatics, ICCCI (2021). https://doi.org/10.1109/ICCCI50826.2021.9402301
2. Chovancová, A., Fico, T., Chovanec, E., Hubinský, P.: Mathematical modelling and parameter identification of quadcopter (a survey). Procedia Eng. **96**, 172–181 (2014). https://doi.org/10.1016/j.proeng.2014.12.139

3. Fernando, H.C.T.E., De Silva, A.T.A., De Zoysa, M.D.C., Dilshan, K.A.D.C., Munasinghe, S.R.: Modelling, simulation and implementation of a quadcopter UAV. In: IEEE 8th International Conference on Industrial and Information Systems, ICIIS 2013, Conference Proceedings, pp. 207–212 (2013). https://doi.org/10.1109/ICIInfS.2013.6731982
4. Salih, A.L., Moghavvemi, M., Mohamed, H.A.F., Gaeid, K.S.: Modelling and PID controller design for a quadcopter unmanned air vehicle. In: IEEE International Conference on Automation, Quality and Testing, Robotics, AQTR 2010 - Proceeding, vol. 1, pp. 74–78 (2010). https://doi.org/10.1109/AQTR.2010.5520914
5. Jafar, H., Zareh, M., Roshanian, J., Nikkhah, A.: An optimal guidance law applied to quadcopter using LQR method. Trans. Jpn. Soc. Aeronaut. Space Sci. **53**(179), 32–39 (2010). https://doi.org/10.2322/tjsass.53.32
6. Santos, M., López, V., Morata, F.: Intelligent fuzzy controller of a quadcopter. In: Proceedings of the IEEE International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 141–146 (2010). https://doi.org/10.1109/ISKE.2010.5680812
7. Xu, R., Özgüner, Ü.: Sliding mode control of a quadcopter helicopter. In: Proceedings of the IEEE Conference on Decision Control, pp. 4957–4962 (2006). https://doi.org/10.1109/cdc.2006.377588
8. Regula, G., Lantos, B.: Backstepping based control design with state estimation and path tracking, to an indoor quadcopter helicopter. Period. Polytech. Electr. Eng. **53**(3–4), 151–161 (2009). https://doi.org/10.3311/pp.ee.2009-3-4.07
9. Mohd Basri, M.A., Husain, A.R., Danapalasingam, K.A.: Enhanced backstepping controller design with application to autonomous quadrotor unmanned aerial vehicle. J. Intell. Rob. Syst. **79**(2), 295–321 (2014). https://doi.org/10.1007/s10846-014-0072-3
10. Basri, M.A.M., Husain, A., Danapalasingam, K.: Stabilization and trajectory tracking control for underactuated quadrotor helicopter subject to wind-gust disturbance. Sadhana **40**(5), 1531–1553 (2015). https://doi.org/10.1007/s12046-015-0384-4
11. Bouabdallah, T.: Design and control of quadcopters with application to autonomous flying Samir Bouabdallah Thèse No 3727, vol. 3727 (2007). https://infoscience.epfl.ch/record/95939/files/EPFL_TH3727.pdf
12. Voos, H.: Nonlinear control of a quadcopter micro-UAV using. In: IEEE International Conference on Mechatronics, no. April, pp. 4–9 (2009)
13. Bao, N., Ran, X., Wu, Z., Xue, Y., Wang, K.: Research on attitude controller of quadcopter based on cascade PID control algorithm. In: Proceedings of the IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2017, vol. 2018-Janua, pp. 1493–1497 (2018). https://doi.org/10.1109/ITNEC.2017.8285044
14. Wang, P., Man, Z., Cao, Z., Zheng, J., Zhao, Y.: Dynamics modelling and linear control of quadcopter. In: International Conference on Advanced Mechatronic Systems, ICAMechS, pp. 498–503 (2016). https://doi.org/10.1109/ICAMechS.2016.7813499
15. Kanojiya, R.G.: Tuning of PID controller using Ziegler-Nichols method for speed control of DC motor. In: International Conference on Advances in Engineering, Science and Management, pp. 117–122 (2012)
16. Idrissi, M., Salami, M., Annaz, F.: A review of quadrotor unmanned aerial vehicles: applications, architectural design and control algorithms. J. Intell. Rob. Syst. **104**(2), 1–33 (2022). https://doi.org/10.1007/s10846-021-01527-7