

Self-Tuning PID Controller for Quadcopter using Fuzzy Logic

A'dilah Baharuddin ^{a,1}, Mohd Ariffanan Mohd Basri ^{a,2,*}

^a Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia

¹ adilahbaharuddin@gmail.com; ² ariffanan@fke.utm.my

* Corresponding Author

ARTICLE INFO

ABSTRACT

Article history

Received August 07, 2023

Revised September 22, 2023

Accepted October 03, 2023

Keywords

PID Controller;

Fuzzy Logic;

Quadcopter;

Self-Tuning

Tracking has become a necessary feature of a drone. This is due to the demand for drones, especially quadcopters, to be used for activities such as surveillance, monitoring, and filming. It is crucial to ensure the quadcopters perform the tracking with stable flight. Despite the advantages of having VTOL ability and great maneuverability, quadcopters require an effective controller to overcome their under-actuation and instability behavior. Even though a PID controller is commonly used and promising with its simple mechanism, it requires very proper tuning to ensure the stability of the system is not affected. In this paper, a simple Fuzzy algorithm is proposed to be incorporated into a PID controller to form a self-tuning Fuzzy PID controller. The Fuzzy logic controller works as the self-adjuster to the PID parameters. A mathematical model of the DJI Tello quadcopter is derived with position and attitude control loops that are designed to track a variety of trajectories with stable flight. The proposed method uses a simple architecture where the ranges of PID parameters are used as scaling factors for Fuzzy controller outputs. The results of the simulations show the tracking error performance metrics, which are IAE, ISE, and RMSE, are smaller compared to the values of the PID controller. Beyond its impact on quadcopter control, the proposed self-tuning approach holds promise for broader applications in nonlinear systems.

This is an open-access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



1. Introduction

An unmanned aerial vehicle, or UAV, is a drone that can fly and maintain altitude, allowing it to undertake more cost-effective operations and vital missions without requiring a human operator or jeopardizing human life [1]. UAVS are widely addressed as drones, which is an acronym for Dynamic Remotely Operated Navigation Equipment [2]. Drones are classified based on a few parameters, such as size, mass, endurance, flight altitude, function, payload, configuration, and flying principle [2][3]. In the configuration class, drones are divided according to the type of wings and rotors. Similarly, both types of drones rely on the aerodynamic action of their wings or rotors to lift upward [2]. The focus of this work is on quadcopter UAVs of the multi-rotor type.

Due to their enhanced stability and endurance in several activities, drones, including quadcopters, have piqued the interest of a wide range of military and civilian service industries, such as synthetic aperture radars (SAR), search and rescue, monitoring and tracking, evacuation, medical aids and package delivery, crop dusting, and even filming or photographing [1][4]. In addition, drones can operate effectively in areas of short path length, minimum turning angle, maximum roll, and fast speed

thanks to their great mobility, portability, scalability, and flexibility [1][4][5]. Quadcopters have become the most favorable drones for performing a variety of activities, as mentioned, thanks to their advantageous inherent dynamic nature, which provides them with maneuverability in addition to their simple mechanism and ability to perform VTOL and hovering [5].

Despite the appeal they offer, operating quadcopters is a hard task as under-actuation, non-linearity and instability characterize these aerial devices, which are aggravated by the difficulty in modelling due to multivariate, and hard to control due to the coupled variables [4][6]. Furthermore, tracking trajectory is one of the key functions of the quadcopter, especially when performing activities with limited time to implement measurements and decision-making procedures such as inspection of natural disaster areas and preliminary evaluation analysis. Thus, to efficiently stabilize quadcopters and accommodate a variety of operational conditions, innovative control systems are necessary.

The linear controllers, namely proportional-integral-derivative (PID), linear-quadratic-regulator (LQR), and H^∞ have been widely used and effectively implemented in quadcopters. However, these methods possess limitations, with improperly tuned PID prone to instability, the less robustness of LQR leading to an inability to deal with nonlinearities, and the highly complex H^∞ requiring crucial parameter adjustment [7]. Nonetheless, these methods have been commonly used from the early development of quadcopters until today due to their effectiveness and competence in obtaining stable flight conditions [7]. Hence, this work proposes a comprehensive approach by integrating fuzzy logic adaptability into the PID controller structure, yielding a self-tuning PID controller that improves stability and control precision.

In [8][9], a Fuzzy PID controller is proposed to control the attitude movements of a quadcopter and aims to have stable flight with fast response and a robust system. The method successfully improved the performance of attitude movement, but no guarantee of the stability of position movement is discussed in these papers. Therefore, the method is guaranteed to yield a good result for attitude tracking. In the case of position tracking [10], a robust Fuzzy PID controller is designed for the position loop and attitude loop based on Fuzzy adaptation scheme. The proposed method used the Karnik-Mendel algorithm to calculate the switch points of the system in addition to the requirement to decide the appropriate scaling factors that work as adjusters to the system. However, the proposed method could be time-consuming and tedious despite its effectiveness in improving performance. Thus, this work aims to design a self-tuning Fuzzy PID controller based on Fuzzy adaptation but using a simpler algorithm.

The core contribution of this work are as follows:

1. Development of a PID controller designed for attitude and position control loop to stabilize and track diverse trajectory shapes.
2. Implementation of an automated PID gain tuning process via Fuzzy logic to improve system performance.
3. Comprehensive performance comparison between the conventional PID approach and the proposed Fuzzy PID method involving analyses of transient response and tracking accuracy.

The subsequent sections of this paper are organized as follows: Section 2 presents the derivation of the quadcopters dynamic model in the 'x' cross configuration. Section 3 describes the design of both PID and Fuzzy PID controllers for the altitude, attitude, and position tracking. Section 4 provides an in-depth analysis of simulation results, comparing the performances of the PID and Fuzzy PID controllers. Finally, Section 5 concludes the paper, summarizing the key findings and contributions.

2. Quadcopter Modelling

A quadcopter is a rotary-wing UAV that is equipped with four rotors. The rotors are commonly arranged in a symmetrical cross (x) or plus (+) configuration [11]. As a quadcopter is a six-degree-of-freedom aircraft, there are six variables that represent the quadcopter's positions and orientations.

The mathematical model of the cross (x) configuration based on DJI Tello was derived in this work. This section covers the quadcopter's description, dynamic model derivation, and parameters.

2.1. Quadcopter's Description

The quadcopter employs a pair of blades rotating in the same direction, which are positioned on the x and y coordinates of the body frames to make it easier to control as each movement in either direction requires the speed of two blades responsible for the desired direction to be controlled [11]. The cross (x) configuration of the quadcopter with rotor 1 and rotor 2 rotating anticlockwise and rotor 3 and rotor 4 rotating clockwise is illustrated in Fig. 1, [12][13].

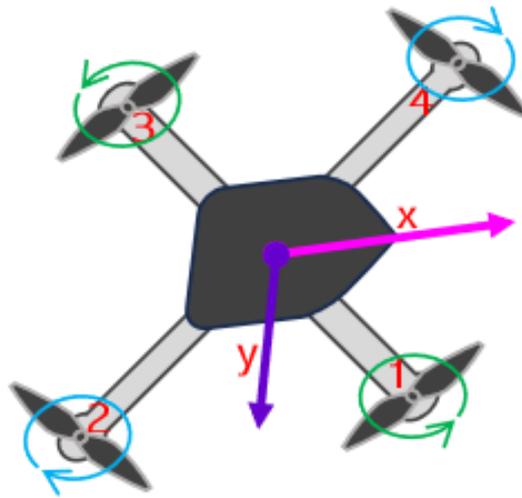


Fig. 1. Cross configuration quadcopter

The difference in lift force is used to generate motion by adjusting the rotor's velocity in pairs. The manipulation of rotors to create forward, backward, right-ward, left-ward, vertical, and hovering motions of the quadcopter is shown in Table 1, [3][11][14][15][16].

Table 1. Motion direction of quadcopter

Angle Orientation	Motion Direction	Rotor's Velocity
Pitching	Forward	Rotors 1 & 4 need to be slower than rotors 2 & 3
Pitching	Backward	Rotors 2 & 3 need to be slower than rotors 1 & 4
Rolling	Rightward	Rotors 3 & 4 need to be slower than rotors 1 & 2
Rolling	Leftward	Rotors 1 & 2 need to be slower than rotors 3 & 4
Yawing	Hovering	All rotors have the same velocity

The modeling techniques of quadcopter are based on the physics of the system with the system further broken into smaller subsystems for easier analysis, design, and modelling [11]. In order to simplify the mathematical equations in modeling, several different assumptions are made while still constructing a fairly realistic model [11]. The universal assumptions are herein presented [4][11][14][17][18]:

1. The quadcopter is perfectly symmetrical with respect to XY axes in structure with diagonal inertia matrix. Thus, cross (x) configuration can use plus (+) degree of freedom.
2. The quadcopter is equipped with four motors on its rigid frame physical structure.
3. The quadcopter has thrust and drag constants that are proportional to the square values of the motor's speed.
4. The quadcopter's center of mass and the origin of the body-fixed frame coincided.
5. The quadcopter has rigid propellers that work under the same conditions all the time with no blade flapping.

2.2. Quadcopter Dynamic Model

The body-fixed frame, $Q = \{XQ, YQ, ZQ\}$ and the earth-fixed frame, $E = \{XE, YE, ZE\}$ of quadcopter can be seen as in Fig. 2, in which their relationship is satisfied as $\{Q\}^T = R^T x \{E\}^T$ [6][11][13][15][18][19]. The quadcopter coordinates are generalized with six degrees of freedom, where absolute positions are described by (x, y, z) and the orientations are respectively described by roll (ϕ) , pitch (θ) and yaw (ψ) .

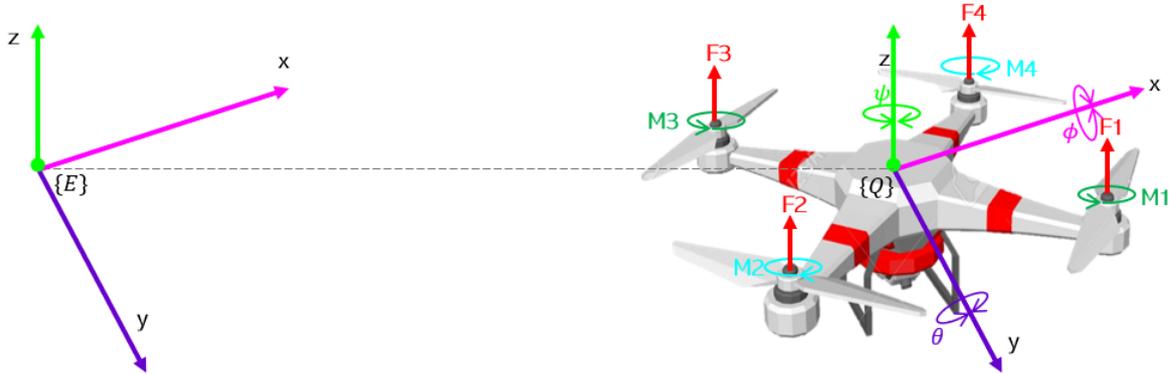


Fig. 2. Body-fixed frame and earth-fixed frame

Based on the frame in Fig. 2, the model is divided into a translational subsystem as in equation (1), where vector ξ in equation (2) represents the quadcopter position relative to inertial frame, and rotational subsystem, where vector η represents the quadcopter attitude [10][18][20][21][22][23].

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

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

The elementary rotations about the x , y , and z axes are defined using Euler angles. The final orientation of the quadcopter with respect to the corresponding inertial axis, inertial trigonometric functions, and their matrix representations are shown in Fig. 3 [18][22]. Based on Fig. 3, rotation matrix of the body relative to the inertial frame, $R_T = R\psi R\theta R\phi$ in equation (3) is resulted from the rotations on the linear independent axis [9][12][13][19][21][24].

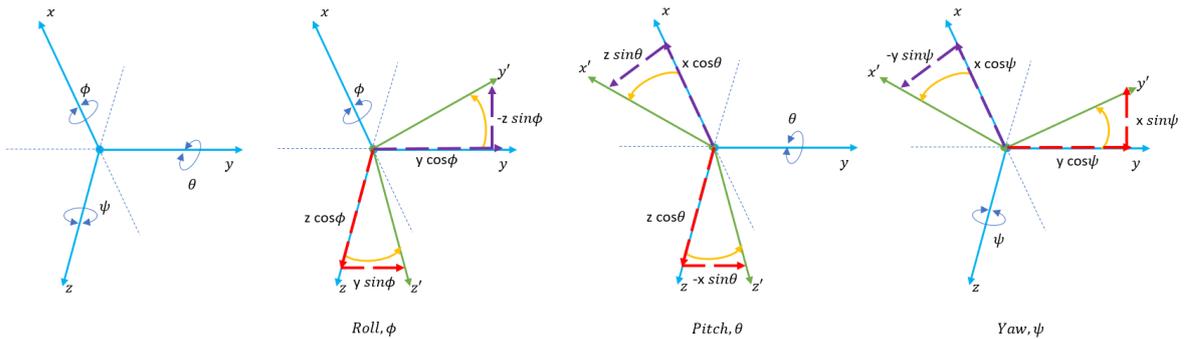


Fig. 3. Euler's angles of rotations

$$R_T = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\psi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \tag{3}$$

As quadcopter model is assumed to be a rigid body, hence Newton-Euler is used to describe the dynamic quadcopter where Newton second rule of motion is used to obtain the translation and the rotation is obtained from Euler's equation of rotation [24][25]. The equation (4) is dynamic quadcopter translational that made of gravitational coefficient, g , the z -axis vector matrix, $E_z = [0 \quad 0 \quad 1]^T$

where U_1 is the total thrust force generated by four rotors, $U_1 = \sum_{i=1}^4 F_i$ [24]. The m is the mass of quadcopter. The translational drag of airframe is $F_d = [k_1\dot{x} \quad k_2\dot{y} \quad k_3\dot{z}]^T$ where k_1, k_2 and k_3 are the air drag coefficients of translational [25]. The equation (4) is rewritten as equation (5) which represents the translation of dynamic quadcopter.

$$m\ddot{\xi} = -mgE_z + U_1R_T E_z - F_d \quad (4)$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{U_1}{m} \begin{bmatrix} c\phi s\theta c\psi + s\phi s\psi \\ c\phi s\theta s\psi - s\phi c\psi \\ c\phi c\theta \end{bmatrix} - \frac{1}{m} \begin{bmatrix} k_1\dot{x} \\ k_2\dot{y} \\ k_3\dot{z} \end{bmatrix} \quad (5)$$

The rotation of dynamic quadcopter is presented in equation (6). 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, $J_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$, the total rotor speed, Ω_d generated from rotors, working in two pairs. U_2, U_3 and U_4 are the total torque related to quadcopter [24][25]. k_4, k_5 and k_6 are the aerodynamic friction coefficients. The equation (6) is rewritten as equation (7) and equation (8) is the control input vector of quadcopter where l is the distance between the rotor axis and the center of quadcopter mass, b is thrust constant and d is the thrust factor.

$$I\dot{\eta} = -\eta \times I\dot{\eta} - J_r(\dot{\eta} \times E_z)\Omega_d + \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} + \begin{bmatrix} k_4\dot{\phi}^2 \\ k_5\dot{\theta}^2 \\ k_6\dot{\psi}^2 \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} I_{xx}\ddot{\phi} \\ I_{yy}\ddot{\theta} \\ I_{zz}\ddot{\psi} \end{bmatrix} = \begin{bmatrix} (I_{yy} - I_{zz})\dot{\psi}\dot{\theta} - (J_r\Omega_d)\dot{\theta} + U_2 \\ (I_{zz} - I_{xx})\dot{\psi}\dot{\phi} + (J_r\Omega_d)\dot{\phi} + U_3 \\ (I_{xx} - I_{yy})\dot{\theta}\dot{\phi} + U_4 \end{bmatrix} + \begin{bmatrix} k_4\dot{\phi}^2 \\ k_5\dot{\theta}^2 \\ k_6\dot{\psi}^2 \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ \frac{\sqrt{2}}{2}lb & -\frac{\sqrt{2}}{2}lb & -\frac{\sqrt{2}}{2}lb & \frac{\sqrt{2}}{2}lb \\ \frac{\sqrt{2}}{2}lb & \frac{\sqrt{2}}{2}lb & -\frac{\sqrt{2}}{2}lb & -\frac{\sqrt{2}}{2}lb \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (8)$$

The derived model simulated in MATLAB Simulink with parameters of DJI Tello as listed in Table 2. The listed parameters are adopted from existing studies using DJI Tello quadcopter [17][25].

Table 2. Adopted parameters of the DJI Tello quadcopter

Parameters	Unit	Value
Quadrotor mass, m	kg	0.080
Lateral moment arm, l	m	0.060
Thrust coefficient, b	kgm/rad ²	0.566e ⁻⁵
Drag coefficient, d	kgm ² /rad ²	0.762e ⁻⁷
Rolling moment of inertia, I_{xx}	kgm ²	0.679e ⁻²
Pitching moment of inertia, I_{yy}	kgm ²	0.679e ⁻²
Yawing moment of inertia, I_{zz}	kgm ²	1.313e ⁻²
Rotor Inertia, J_r	kgm ²	4.95e ⁻⁵
Translational drag coefficients, k_1, k_2, k_3	kg/s	3.365e ⁻²
Aerodynamic friction coefficients, k_4, k_5, k_6	kgm ² /rad ²	4.609e ⁻³

In this work, the trajectory tracking controller for the quadcopter is designed without considering external disturbances such as wind disturbances or sensor noise, as no hardware integration is conducted to evaluate the proposed method. Fig. 4, Fig. 5, Fig. 6 and Fig. 7 show the derived model of quadcopter, motor speed calculation, rotation and translation of dynamic quadcopter respectively.

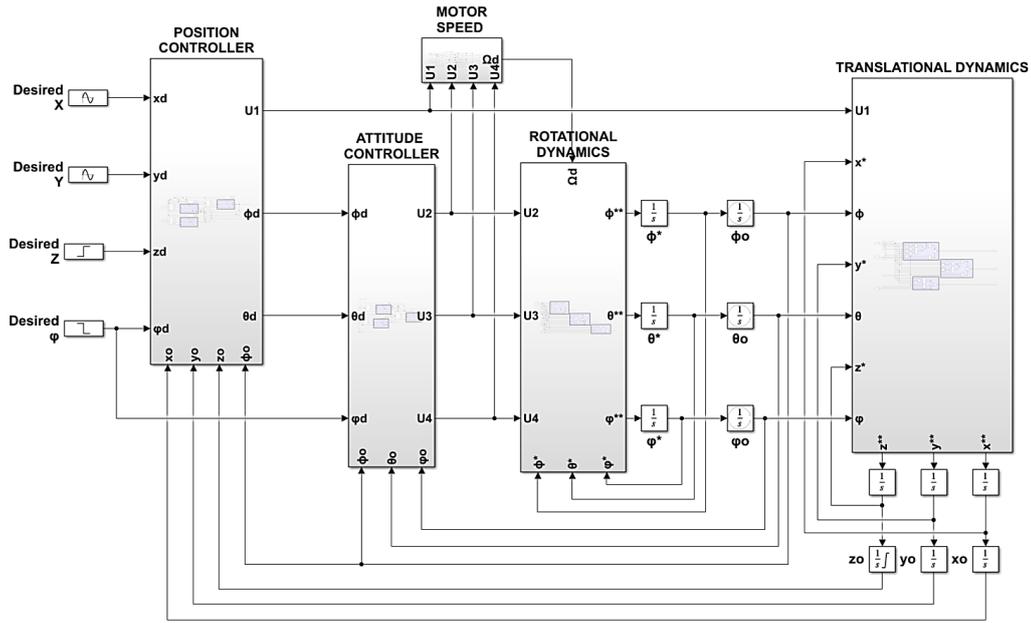


Fig. 4. Simulink blocks of quadcopter model

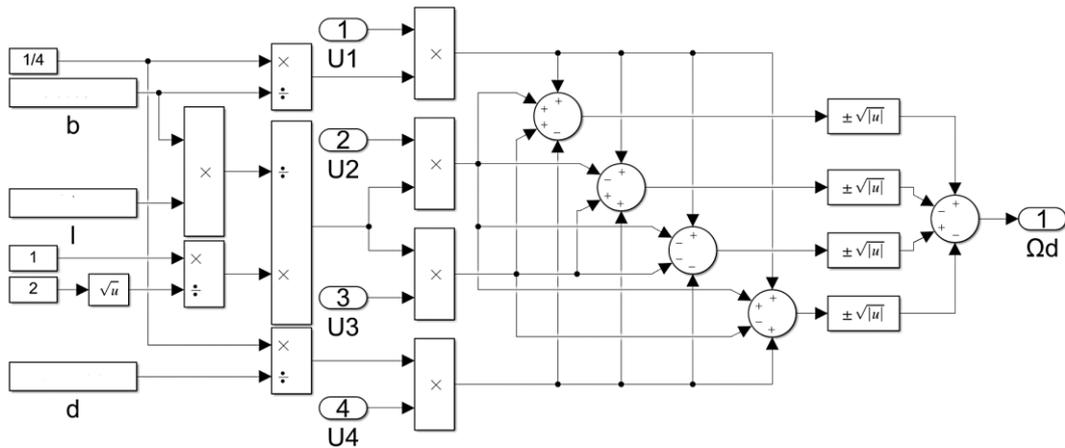


Fig. 5. Simulink block of motor speed of quadcopter

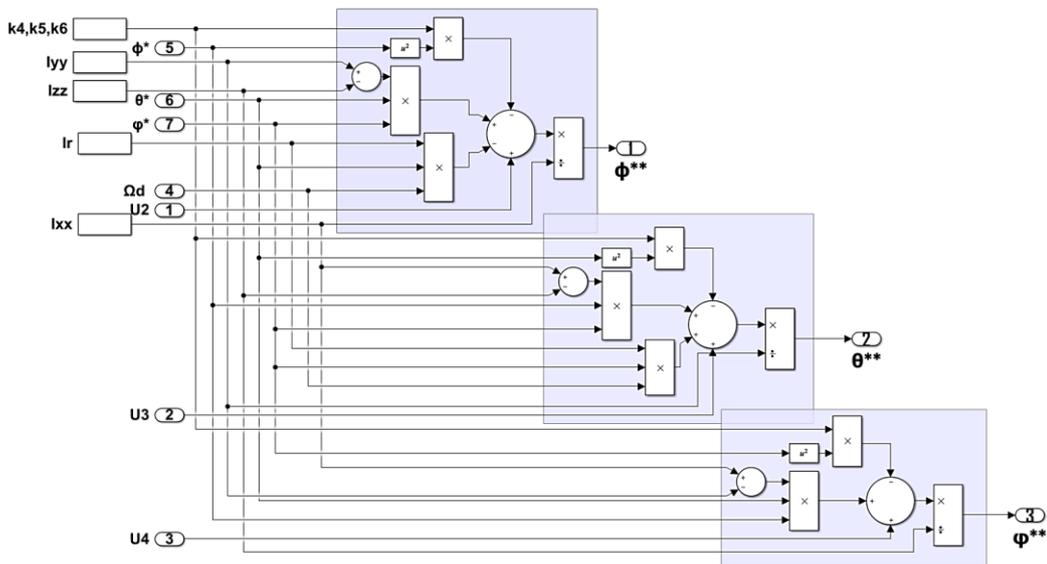


Fig. 6. Simulink blocks of rotation of dynamic quadcopter

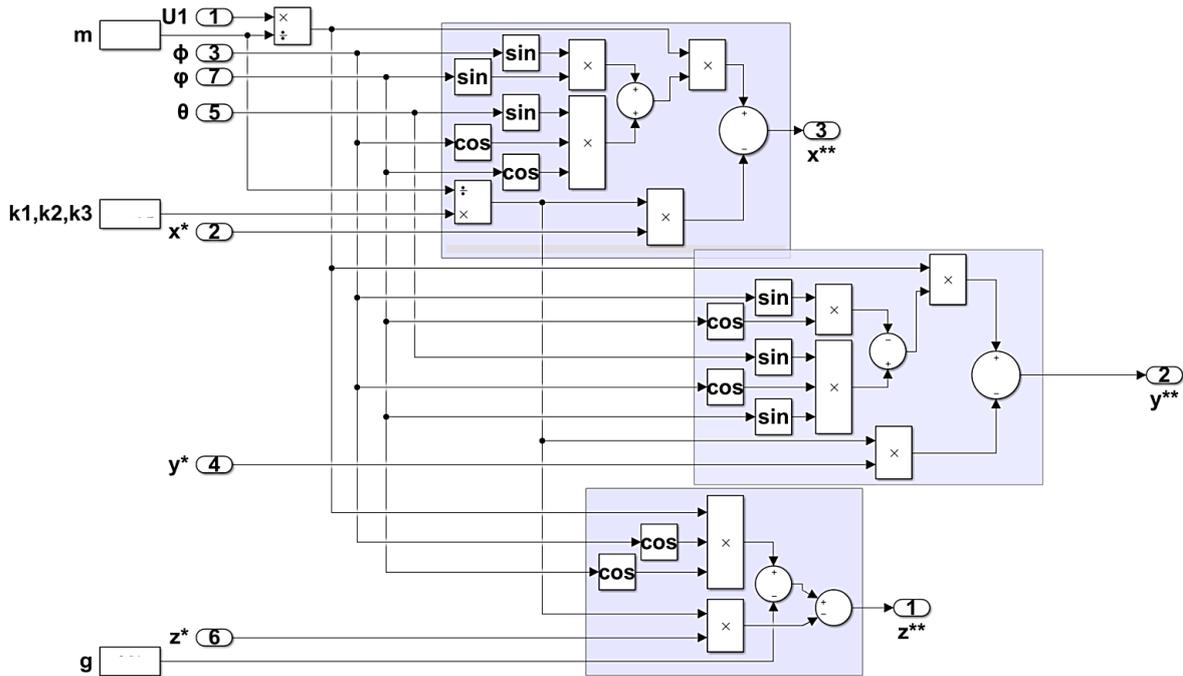


Fig. 7. Simulink block of translation of dynamic quadcopter

3. Quadcopter Controller Design

A quadcopter is unstable and require controllers to work properly. Many control strategies and algorithms can be used. This paper utilized a PID controller and applied Fuzzy logic techniques to form a self-tuning PID controller. The controller is applied to all variables ($x, y, z, \phi, \theta, \psi$) of dynamic model of quadcopter. The cascade control structure of the quadcopter is shown in Fig. 8.

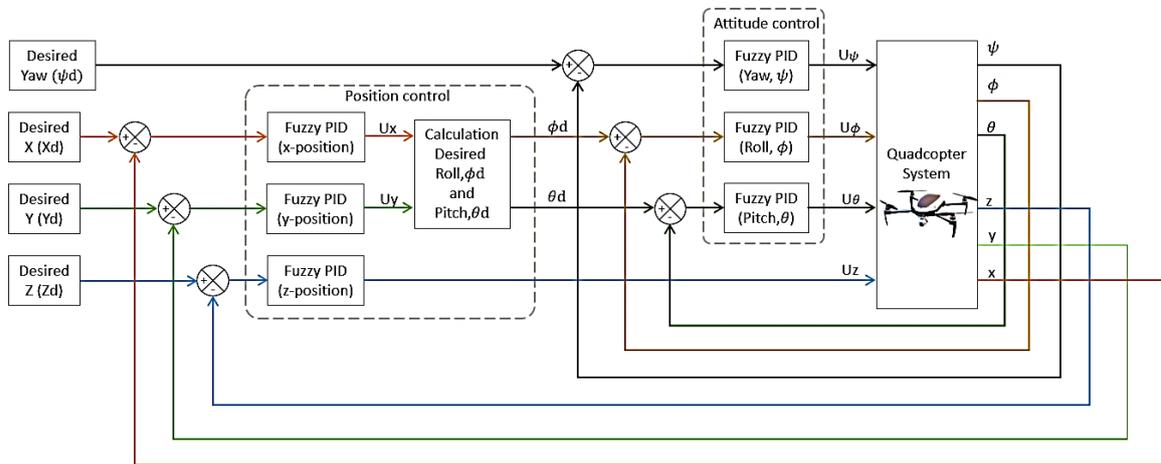


Fig. 8. Block diagram of proposed controller for quadcopter

In controller designing process some changes take place in the model derivation for position control. The derived translational of dynamic quadcopter in equation (5) are changed for x and y axes translation, where the \ddot{x} and \ddot{y} are redefined as u_x and u_y , respectively. The redefined model for x and y axes is written in (9) [23]. The desired roll, ϕ_d and pitch, θ_d are not defined in the simulation as this work aims to track the desired trajectory x_d, y_d, z_d , and ψ_d . Based on virtual control of position theorems, the generation desired roll, ϕ_d and pitch, θ_d and stabilizing the altitude of quadcopter is obtained by position control [23]. According to the theorems equation (11) is redefined as equation (12) and equation (13) where u_{1d} is the output of altitude, z control.

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \frac{m}{u_1} \begin{bmatrix} c\phi s\theta c\psi + s\phi s\psi \\ c\phi s\theta s\psi - s\phi c\psi \end{bmatrix} - \frac{1}{m} \begin{bmatrix} k_1 \dot{x} \\ k_2 \dot{y} \end{bmatrix} \quad (9)$$

$$\phi_d = \arcsin(u_x s\psi_d - u_y c\psi_d) \quad (10)$$

$$\theta_d = \arcsin \frac{(u_x c\psi_d + u_y s\psi_d)}{c\phi} \quad (11)$$

3.1. PID Controller

To achieve precise tracking and good performance, PID-type controllers are usually used in the cascade control system [25]. In the process, a PID controller works in close form to calculate the difference between a set point and the desired set point and works on minimizing it by tuning the control variables [24][25]. The relation between the error of the system and PID controller output is defined by equation (12), where equation (13) represents the difference between desired and generated output [19][21][22][25].

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

$$e(t) = \text{desired}(t) - \text{generated}(t) \quad (13)$$

K_p is proportional gain, K_i is integral gain and K_d is derivative gain. K_p is increased to reduce steady-state error, yet overly high values K_p can cause unstable oscillations. K_i is increased to remove the feedback system's steady-state error, while K_d is increased to lessen the overshoot and settling time of the feedback system output signal [12][15][16][26][27], Fig. 9 shows the block diagram of PID controller.

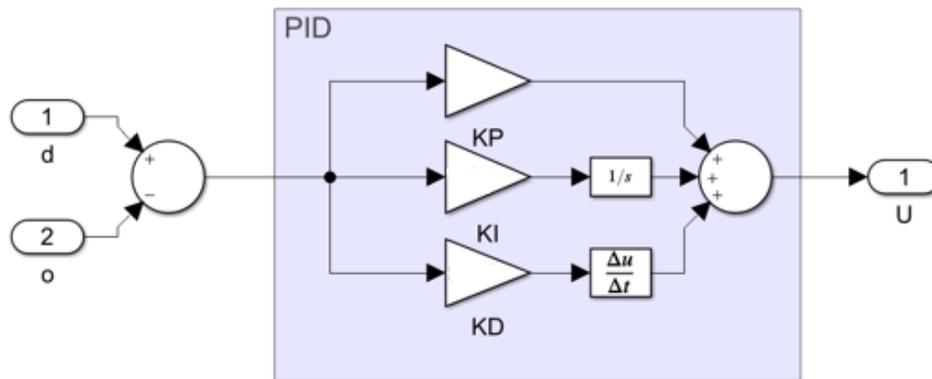


Fig. 9. Simulink blocks of PID controller

3.1.1. PID Tuning Method

PID controllers possess no knowledge of the process plant, which causes them to be unable to adjust and respond to changes. Thus, it requires proper tuning to generate satisfying performance. One of the most commonly used methods is Ziegler-Nichols, which is divided into step response rules and frequency response rules. Both rules started with setting integral and derivative gain to zero while raising proportional gain to a value that will oscillate the system [28][29].

However, a system with no sustained oscillation is unable to be tuned with this method. In this work, the derived model used a manual tuning approach to tune the PID parameters, as in attempts to utilize Ziegler-Nichols, resulting in no sustained oscillation. Increasing each PID gain affects the transient response of the system, and the effects are summarized in Table 3. Based on the rules listed, the PID parameters obtained for PID controller are listed in Table 4.

The rules for manual tuning of PID parameters are listed [30][31]:

1. K_p , K_i and K_d are set to zero.
2. K_p is raised until oscillations are fully sustained or nearly sustained, if fully sustained is impossible to reach. The simulation time is advised to be set at a long simulation stop time.
3. K_d is raised slightly to reduce the oscillation to 1 period. The simulation can be run with shorter simulation stop time.
4. K_i is increased until the steady state is near the setpoint, and the single oscillation oscillates around the setpoint.
5. The rules are repeated until the desired output is achieved.

Table 3. The effect of increasing the PID gains value

Parameters	K_p	K_i	K_d
Overshoot (%)	Increase	Increase	Decrease
Rise time (s)	Decrease	Decrease	Minor change
Settling time (s)	Minor change	Increase	Decrease
Steady-state error	Decrease	Eliminate	No effect (Theoretically)
Stability	Degrade	Degrade	Improved (Small K_d)

Table 4. The gains of PID parameters for PID controller

Variable	K_p	K_i	K_d
Position (x , y), Altitude (z)	0.50	0.10	1.00
Attitude (ϕ , θ , ψ)	1.00	0.01	0.10

3.2. Fuzzy Controller

The manual tuning method is an easy way for PID controllers to obtain a reasonable result because it has an intuitive approach and little process knowledge is required. However, it may take a long time to obtain the gains that generate a reasonable result, and it is difficult to determine if the final settings are optimal. In addition, there is no guarantee that the system will attain a durable and stable solution, which could put the entire plant at risk. Due to their little knowledge of the process plant, PID controllers are unable to adjust themselves automatically when the system faces a certain change [32]. Hence, the metaheuristic, specifically the Fuzzy logic technique, is integrated into the PID controller to ensure the output is obtained as desired and the parameters are tuned automatically when changes are applied to the system.

Fuzzy logic is a robust command that does not require precise knowledge of a mathematical model [7][33]. By exploiting a human understanding of the plant in the control design process and decision-making, Fuzzy control works based on Fuzzy set theory, linguistic variables, and Fuzzy inference [34]. A Fuzzy control system is composed of Fuzzification, Fuzzy rule bases, Fuzzy inference, and Defuzzification [35], Fig. 10 shows the general block diagram of the Fuzzy inference system [36].

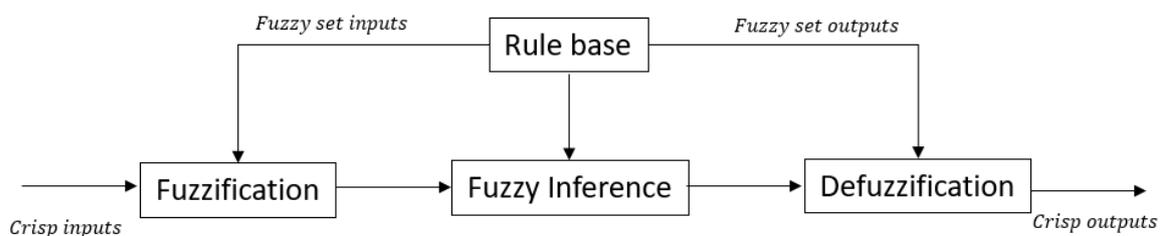


Fig. 10. Fuzzy inference system

For position, altitude, and attitude of quadcopter control, the inputs are the error, $e(t)$ which represents the difference between the desired and generated trajectory, and the change rate of the error,

$de(t)$. Based on Fig. 10, these inputs, which are in the form of crisp values, are fed to the Fuzzy inference system, and transformed into language variables via Fuzzification process. Prior knowledge to determine the range of input variables are required in Fuzzification process to map the input variables in a certain range to discrete intervals [9].

The Fuzzy logic controller requires a rule base and Fuzzy set outputs to adjust the PID parameters. The type of rule base chosen for this work is Mamdani-type, where it's a system of multiple input and multiple output (MIMO), as it takes 2 inputs and generates 3 outputs, K_{pf} , K_{if} and K_{df} . The Fuzzy set inputs are set to $e(t) \in [-10, 10]$ and $de(t) \in [-15, 15]$ for position and altitude control, while $e(t) \in [-0.5, 0.5]$ and $de(t) \in [-50, 50]$ for attitude control. These domains are shown in Fig. 11 position and altitude and Fig. 12 for attitude.

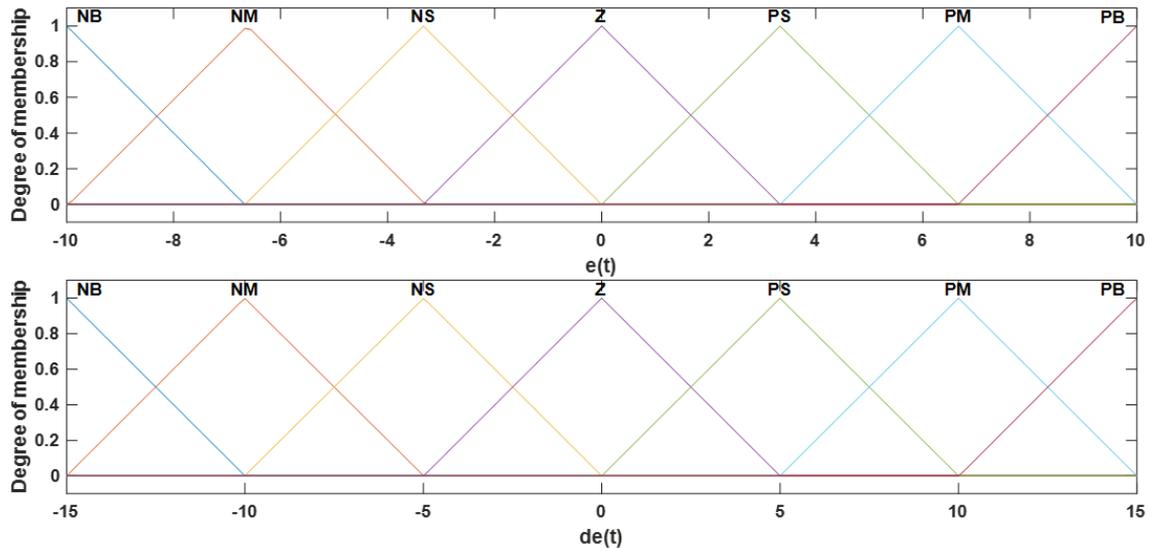


Fig. 11. The inputs membership functions of position and altitude

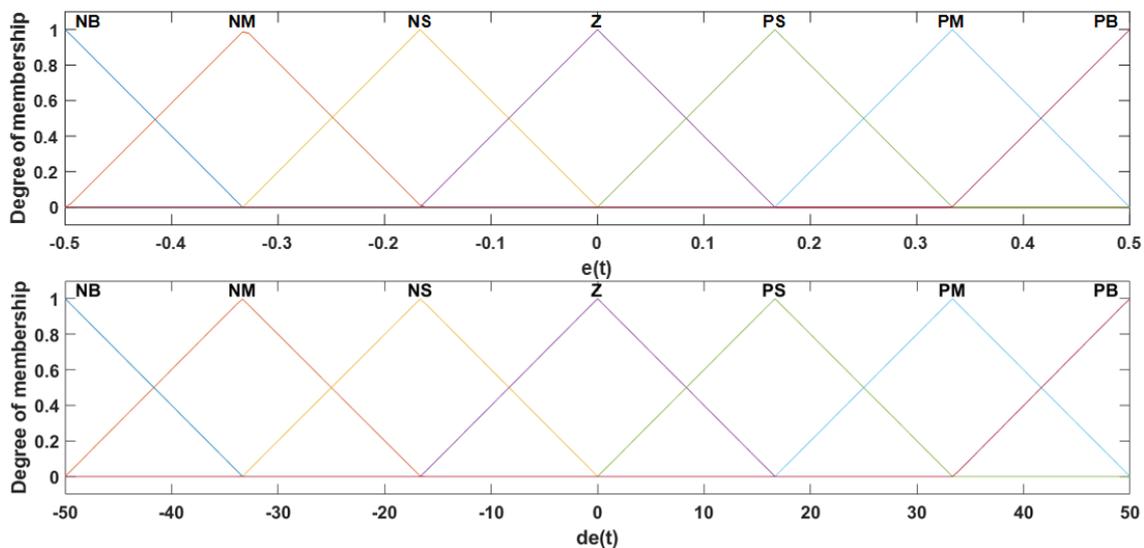


Fig. 12. The inputs membership functions of attitude

Meanwhile, the Fuzzy set outputs are set according to $K_p \in [K_{pmin}, K_{pmax}]$, $K_i \in [K_{imin}, K_{imax}]$, and $K_d \in [K_{dmin}, K_{dmax}]$. The Fuzzy set outputs for position and altitude are set to $K_p \in [0.5, 5.0]$, $K_i \in [0.1, 1.0]$, and $K_d \in [1.0, 10]$ as shown in Fig. 13, and for attitude are set to $K_p \in [1.0, 10]$, $K_i \in [0.01, 0.10]$, and $K_d \in [0.1, 1.0]$ as shown in Fig. 14. These Fuzzy sets are calibrated over intervals $[0, 1]$. The degree of membership function is defined by triangular membership functions.

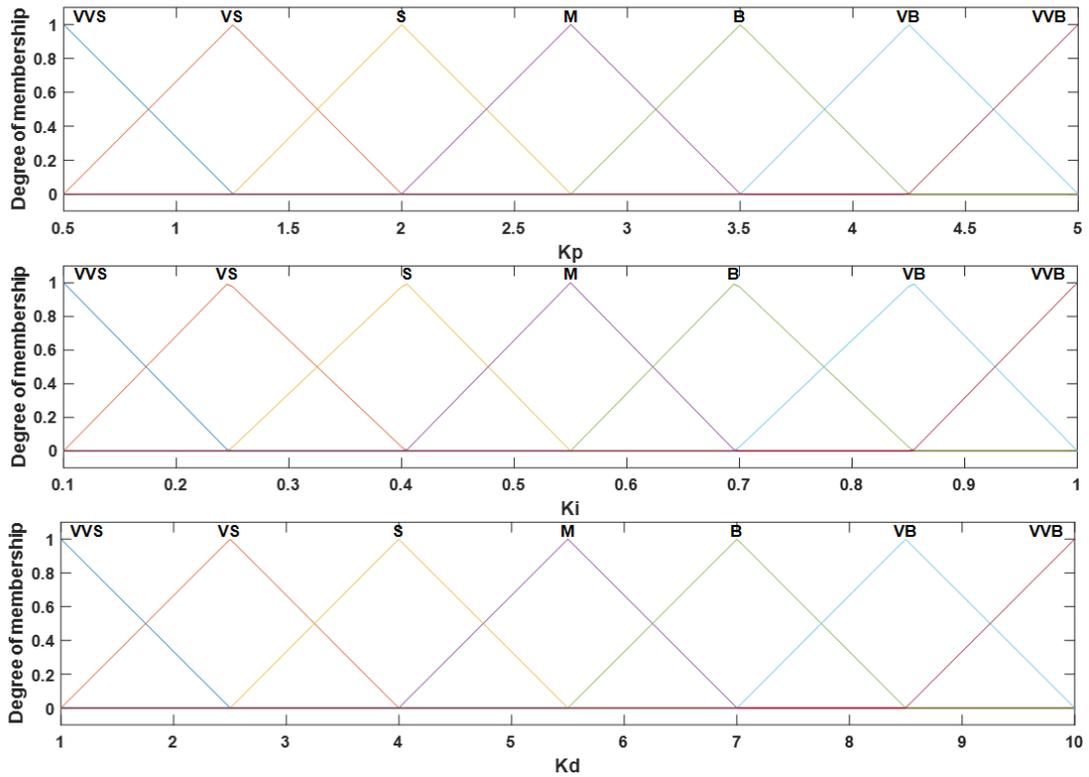


Fig. 13. The output membership functions of position and altitude

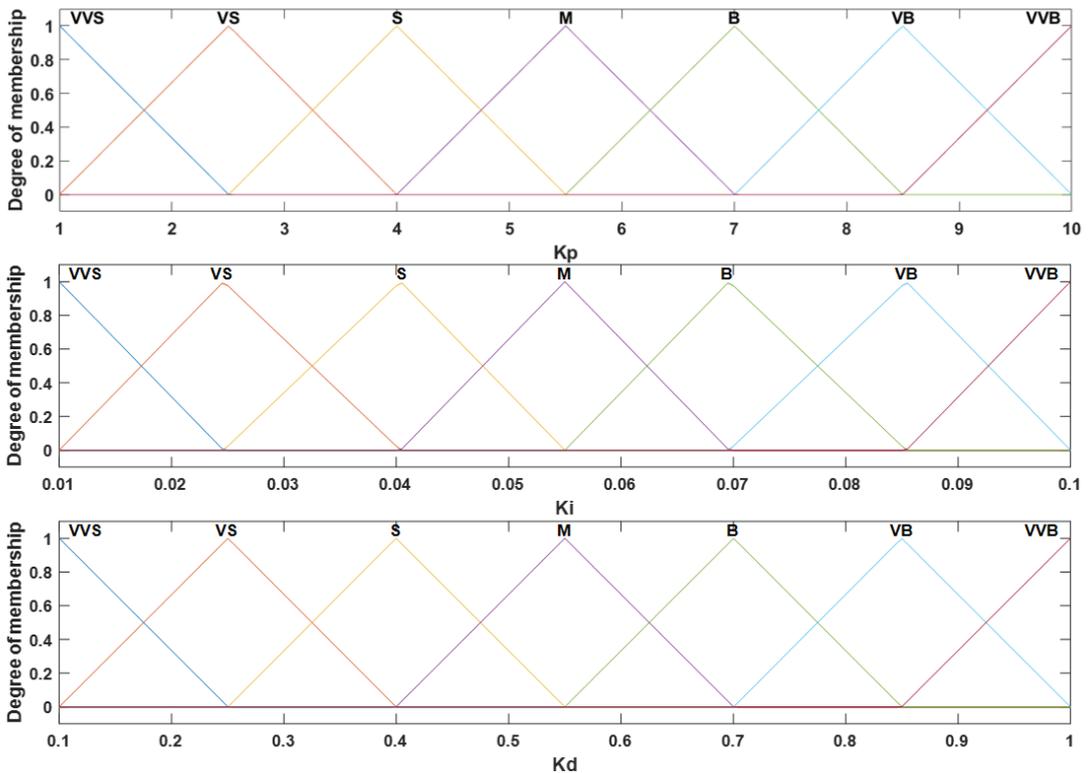


Fig. 14. The output membership functions of attitude

Mamdani-type Fuzzy controllers use IF-THEN rules to form the linguistic form of Fuzzy logic called the membership function. Based on seven linguistic variables, 49 rules of membership functions are formed for each output. The rules of membership functions for the quadcopter are listed in Table 5 and Table 6 [37].

The last process of the Fuzzy controller is defuzzification where the outputs obtained from the Fuzzy inference are in linguistic form. In order to be fed into the PID controller, these outputs have to be converted into crisp or numerical numbers. For Mamdani-type Fuzzy controllers, the most common defuzzification method is the gravity center method, and the specific calculation is defined in equation (14).

$$Output = \frac{\sum_{i=1}^q \mu_j(w_i) \cdot w_i}{\sum_{i=1}^q \mu_j(w_i)} \quad (14)$$

Table 5. Fuzzy rules for K_p and K_i

Error changes, δe	Error, e						
	NB	NM	NS	Z	PS	PM	PB
NB	M	S	VS	VVS	VS	S	M
NM	B	M	S	VS	S	M	B
NS	VB	B	M	S	M	B	VB
Z	VVB	VB	B	M	B	VB	VVB
PS	VB	B	M	S	M	B	VB
PM	B	M	S	VS	S	M	B
PB	M	S	VS	VVS	VS	S	M

Negative-Big (NB), Negative-Medium (NM), Negative-Small (NS), Zero (Z), Positive-Small (PS), Positive Medium (PM), Positive-Big (PB), Very Very-Small (VVS), Very-Small (VS), Small (S), Medium (M), Big (B), Very-Big (VB), Very-Very-Big (VVB)

Table 6. Fuzzy rules for K_d

Error changes, δe	Error, e						
	NB	NM	NS	Z	PS	PM	PB
NB	M	B	VB	VVB	VB	B	M
NM	S	M	B	VB	B	M	S
NS	VS	S	M	B	M	S	VS
Z	VVS	VS	S	M	S	VS	VVS
PS	VS	S	M	B	M	S	VS
PM	S	M	B	VB	B	M	S
PB	M	B	VB	VVB	VB	B	M

Negative-Big (NB), Negative-Medium (NM), Negative-Small (NS), Zero (Z), Positive-Small (PS), Positive Medium (PM), Positive-Big (PB), Very Very-Small (VVS), Very-Small (VS), Small (S), Medium (M), Big (B), Very-Big (VB), Very-Very-Big (VVB)

3.3. Fuzzy PID Controller

The generated outputs of Fuzzy logic controller, K_{pf} , K_{if} , and K_{df} are fed to the PID controller to form Fuzzy PID controller. Fig. 15 shows the generalized block diagram of the Fuzzy PID controller and equation (15) defines the Fuzzy PID controller [32][37][38][39].

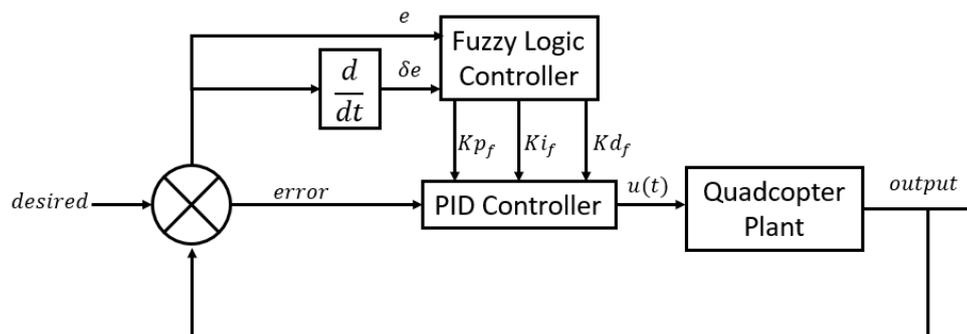


Fig. 15. The block diagram of Fuzzy PID controller

$$u(t)_{FPID} = \left((K_p + K_{pf})e(t) + (K_i + K_{if}) \int_0^t e(t)dt + (K_d + K_{df}) \frac{de(t)}{dt} \right) \quad (15)$$

The minimum and maximum variable ranges of the PID controller gains are obtained from the trial-and-error simulation using PID controller. The same values are used into equation (16), equation (17) and equation (18), and the Fuzzy PID controller is simulated as shown in Fig. 16, [32][38][40].

$$K_{pf} = \frac{K_p - K_{pmin}}{K_{pmax} - K_{pmin}} \quad (16)$$

$$K_{if} = \frac{K_i - K_{imin}}{K_{imax} - K_{imin}} \quad (17)$$

$$K_{df} = \frac{K_d - K_{dmin}}{K_{dmax} - K_{dmin}} \quad (18)$$

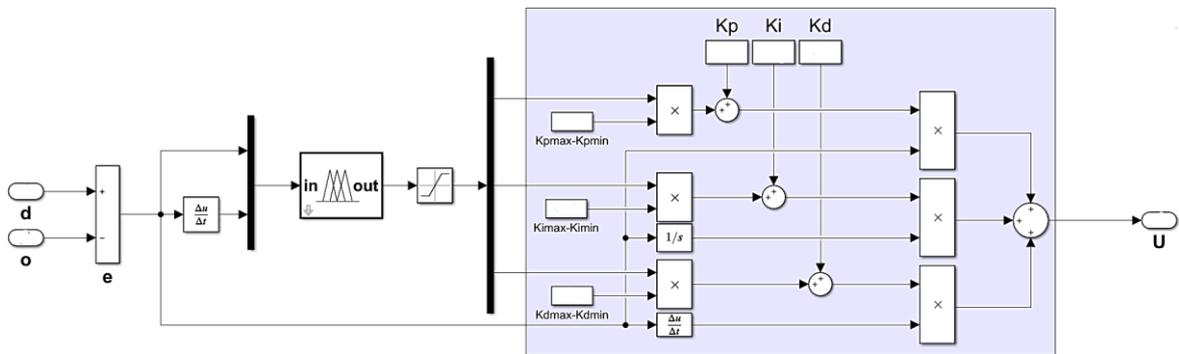


Fig. 16. The Simulink blocks of self-tuning Fuzzy PID controller

4. Results and Discussion

The simulations using MATLAB Simulink for the proposed controller are conducted on a few different trajectories. The simulation time step is set to 0.01s. Each trajectory is simulated with both the PID controller and the proposed Fuzzy PID controller. The results obtained are compared and analyzed with Integral-Absolute-Error (IAE) in equation (19), and Integral-Time-Absolute-Error (ISE) in equation (20) for position of the quadcopter [10][39]. The total root mean square error (RMSE) is also calculated for position using equation (21), [40][41].

$$IAE_{position} = \int_0^t (|e_x| + |e_y| + |e_z|) dt \quad (19)$$

$$ISE_{position} = \int_0^t (e_x^2 + e_y^2 + e_z^2) dt \quad (20)$$

$$RMSE_{position} = \sqrt{\frac{1}{n_i} \sum_{i=1}^{n_i} ((e_x^i)^2 + (e_y^i)^2 + (e_z^i)^2)} \quad (21)$$

In trajectory tracking, the desired roll and pitch are no longer defined, as the desired roll and pitch are the outputs of position control. Thus, the desired position, x and y are defined instead. The desired altitude, z is defined as in altitude and attitude tracking, in which the initial value is 0 and the desired

value is 1. The desired yaw, ψ is set from 0.5 to 0. These inputs are predefined input values for circular, square, lemniscate, and zigzag trajectories. The simulation stop time for every trajectory is set to 180 s.

4.1. Circular Trajectory

The desired positions for circular trajectory are set as $x_d = 10\sin(\frac{\pi}{10}t)$ and $y_d = 10\cos(\frac{\pi}{10}t)$. These pre-defined desired outputs are used for both PID and Fuzzy PID controllers. Fig. 17 shows the circular trajectory tracking output of the quadcopter with the PID controller, and Fuzzy PID controller outputs.

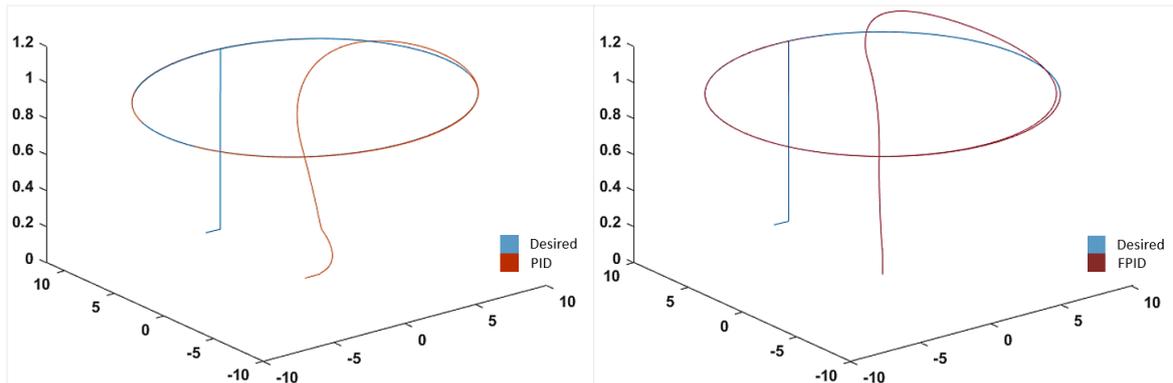


Fig. 17. The outputs of circular tracking

Based on Fig. 17, both controllers successfully track the desired trajectory with obvious improvement in altitude tracking. Table 7 listed the information of the tracking by both controllers. From the listed data, both controllers show almost significant records. However, altitude shows improvement in the rise time record, where the value decreased by 31.18% with the Fuzzy-PID controller. A significant improvement is observed in position y as well, where the rise time is reduced by 21.10% with the Fuzzy PID compared to the PID controller. The overshoot percentage of position y also reduced by 34.32%. On the other hand, position x shows a small improvement in rise time, where the value is decreased by 0.21% with the Fuzzy PID controller.

The performance of both controllers is evaluated by tracking errors. the values of the tracking errors for circular tracking are listed in Table 8. Even though the improvement only occurred to position y and altitude, the values of IAE, ISE and RMSE reduced significantly with the Fuzzy PID controller.

Table 7. The step info of circular tracking

Controller	Variable	Rise Time	Settling Time	Overshoot
PID	x	6.6818	179.4440	33.0230
	y	3.5764	179.5654	78.3446
	z	6.2036	22.2857	3.0709
FPID	x	6.6676	179.4481	33.1440
	y	2.8216	179.5660	51.4564
	z	4.2693	18.2188	3.8344

Table 8. The tracking errors of circular trajectory

Error Calculation	PID	FPID	Reduction (%)
IAE	119.6	72.24	39.60
ISE	584.1	319.2	45.35
RMSE	1.802	1.397	22.48

4.2. Square Trajectory

The desired position inputs for square trajectory tracking are set as $x_d = [10, 10, -10, -10, 10]$ and $y_d = [10, -10, -10, 10, 10]$. for $t_{x,y} = [0, 30, 60, 90, 120]$. Fig. 18 shows the output of the PID controller and the Fuzzy PID controllers.

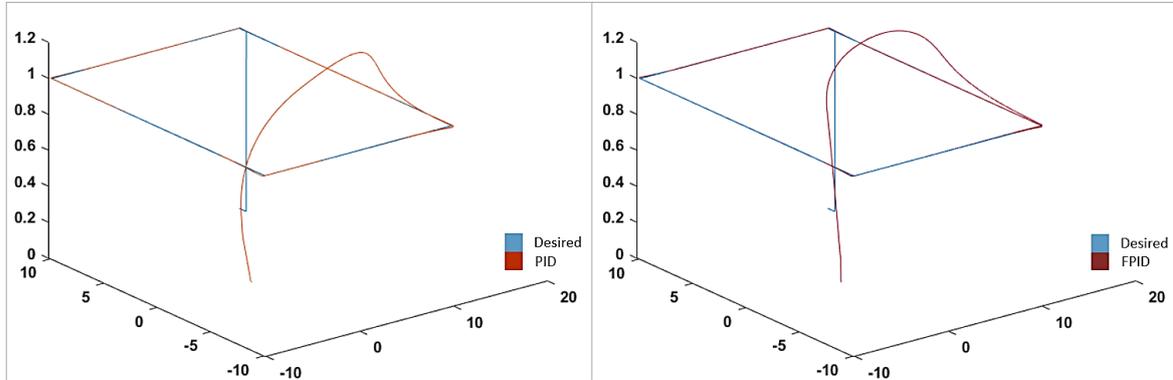


Fig. 18. The outputs for square tracking

The desired trajectory is successfully tracked by both controllers. From Fig. 18, no significant difference is observed except the higher overshoot of altitude tracking is generated by the Fuzzy PID controller. By comparing the results from Fig.20 and the data listed in Table 9, the higher overshoot percentage of altitude is verified as the value record by the Fuzzy PID controller is increased by 29.17% compared to the PID controller.

Table 9. The step info of square tracking

Controller	Variable	Rise time	Settling time	Overshoot
PID	x	12.0037	179.1962	1.7152
	y	10.8449	149.3955	1.9268
	z	6.15460	21.9733	2.9799
FPID	x	12.0172	179.2757	1.8706
	y	10.8971	149.3911	1.2711
	z	4.2659	18.2298	3.8491

The tracking errors values of square tracking in Table 10 shows the integration of Fuzzy techniques to PID controller improves the outputs of the system, as all errors measured from the Fuzzy PID are smaller than the PID controller.

Table 10. The tracking errors of square tracking

Error Calculation	PID	FPID	Reduction (%)
IAE	189.9	142	25.22
ISE	1072	675.8	36.96
RMSE	2.441	1.939	20.57

4.3. Lemniscate Trajectory

The desired position inputs for the lemniscate trajectory are set as $x_d = 10\sin\left(\frac{\pi}{10}t\right)$ and $y_d = 10\cos\left(\frac{\pi}{20}t\right)$. The result of PID controller and Fuzzy PID controller is shown in Fig. 19.

The outputs shown in Fig. 19 illustrate almost identical tracking of position x and y by both controllers, but different tracking outputs of altitude are observed. The tracking of altitude by the Fuzzy PID controller presents a trajectory with a bigger overshoot of altitude compared to the PID controller. This is verified by the data in Table 11, where the overshoot produced by the Fuzzy PID controller is higher in value by 26.05% than the PID controller. However, the rise time and the

settling time of altitude are reduced by 30% and 17.62%, respectively. A slight improvement in positions x and y is also observed, where the rise time of position x is reduced by 0.22% and 0.91% for position y . The output of the tracking by the Fuzzy PID is proven to be better than that of the PID controller and verified by the data in Table 12, where the errors are reduced when Fuzzy PID controller is used in the simulations.

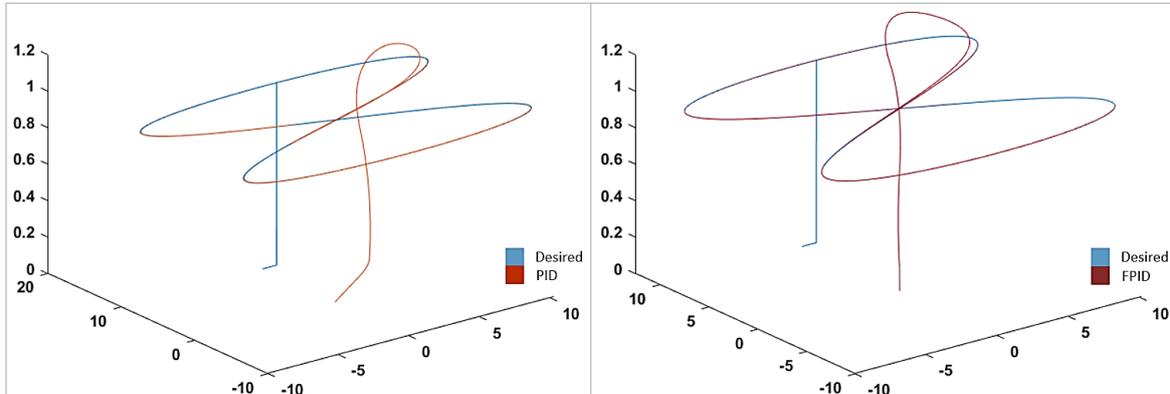


Fig. 19. The outputs of lemniscate tracking

Table 11. The step info of lemniscate tracking

Controller	Variable	Rise time	Settling time	Overshoot
PID	x	6.68190	179.4440	33.0230
	y	17.4847	177.8954	9.76110
	z	6.16090	22.14430	3.06170
FPID	x	6.66750	179.4479	33.1400
	y	17.3255	178.0987	9.75350
	z	4.26890	18.24340	3.85920

Table 12. The tracking errors of lemniscate tracking

Error Calculation	PID	FPID	Reduction (%)
IAE	123.9	80.66	34.90
ISE	642.4	378.1	41.14
RMSE	1.890	1.450	23.28

4.4. Zigzag Trajectory

The desired position inputs for the zigzag trajectory are set as $x_d = [4, 4, 10, 10, 4, 4, -2, -2, -8, -8, -2, -2, 4]$ and $y_d = [4, 12, 8, 0, 4, -4, 0, -8, -4, 4, 0, 8, 4]$ for $t_{x,y} = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]$. The output of both controllers is shown in Fig. 20.

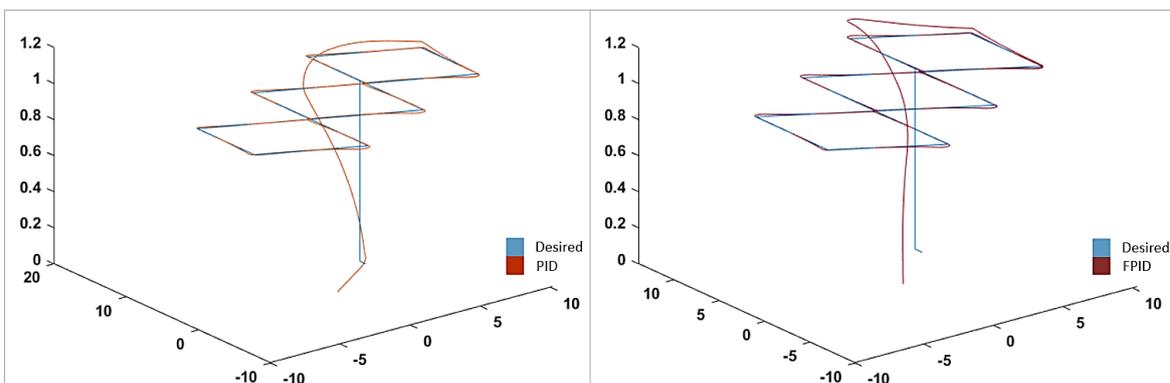


Fig. 20. The outputs of zigzag tracking

Based on Fig. 20, the desired trajectory is successfully tracked by both controllers. No significant difference is observed in the tracking of positions x and y . However, the tracking of altitude showed different behaviors between the controllers, where the overshoot is bigger in tracking with the Fuzzy PID controller than the PID controller. This behavior is also proven with the data from Table 13, where the overshoot percentage of altitude is 26% bigger when the Fuzzy PID controller is used. Yet, the rise time and settling time have improved by 31.24% and 15.36%, respectively.

Based on Table 13, there is a small difference in the rise time of position x the overshoot of overshoot of position y . The rise time of position x is reduced by 0.51%, while position y overshoot is reduced by 4.71%. The rise time of position y makes a big difference between the controllers, as the recorded value dropped by 88.54% with the use of the Fuzzy PID controller. The values of tracking errors recorded by the Fuzzy PID controller lower than the PID controller. These values are listed in Table 14.

Table 13. The step info of zigzag tracking

Controller	Variable	Rise time	Settling time	Overshoot
PID	x	2.6803	179.5938	304.4758
	y	0.1169	179.1857	1.8462e+04
	z	6.2077	21.5779	3.0287
FPID	x	2.6671	179.5889	307.6506
	y	0.0134	179.3120	1.7592e+04
	z	4.2684	18.2628	3.8187

Table 14. The tracking errors of zigzag tracking

Error Calculation	PID	FPID	Reduction (%)
IAE	127.7	91.22	28.57
ISE	290.5	138.8	52.22
RMSE	1.270	0.879	30.79

4.5. Spiral Trajectory

For spiral trajectory tracking, the desired positions are set as $x_d = 10\sin(\frac{\pi}{10}t)$ and $y_d = 10\cos(\frac{\pi}{10}t)$, similarly to circular trajectory tracking except for the altitude, z need to be set as an increasing value. Thus, a ramp with a slope of 0.5 is set as the desired value of z . The spiral tracking output is shown in Fig. 21.

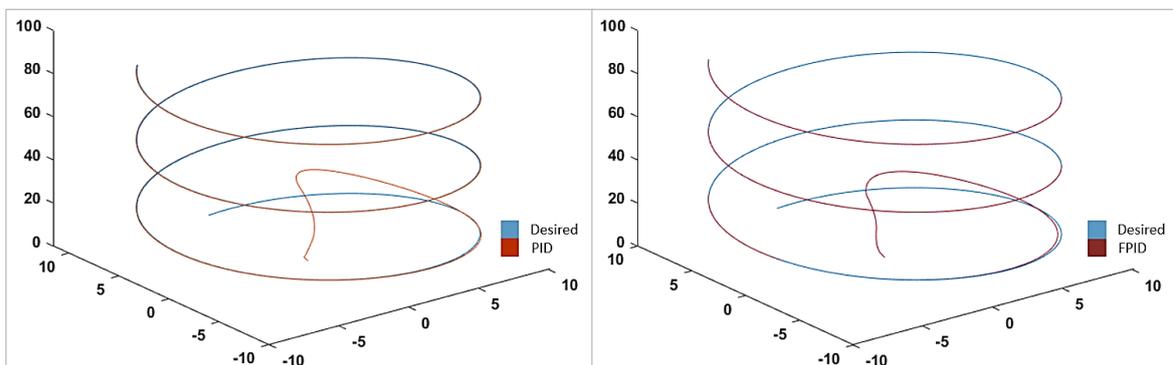


Fig. 21. The outputs of spiral trajectory tracking

From Fig. 21, both controllers show the ability to achieve desired trajectory. The output of the Fuzzy PID controller seems to have smaller overshoot in the positions. The value of the overshoot percentage recorded in Table 15 shows that the overshoot of Fuzzy PID controller is 0.35% bigger in position x and 30.54% lower in position y . The other values show no big difference between the controllers. However, the values of tracking errors in Table 16 shows that Fuzzy PID controller

produced better tracking results as the values of errors for the Fuzzy PID controller are smaller than the PID controller.

Table 15. The step info of spiral tracking

Controller	Variable	Rise time	Settling time	Overshoot
PID	x	6.6795	179.4489	33.0258
	y	3.4215	179.5658	74.0705
	z	144.0484	176.4008	0
FPID	x	6.6668	179.4476	33.1441
	y	3.0754	179.5658	51.4518
	z	143.9642	176.4002	0

Table 16. The tracking errors of spiral trajectory

Error Calculation	PID	FPID	Difference (%)
IAE	98.78	72.24	26.67
ISE	439.1	319.2	27.31
RMSE	1.562	1.333	14.66

4.6. Discussion

In general, the desired trajectory tracking is successfully obtained by the simulations of both the PID controller and the Fuzzy PID controllers. The integration of Fuzzy controller into the PID controller shows improvements in system performance without the need to tune the parameters manually. The system is adjusted by the Fuzzy inference in the Fuzzy controller, which automatically tunes the parameters to produce better results compared to the PID controller. Even though the recorded rise time, settling time and overshoot percentage have no big difference between the controller, the measured tracking errors of the Fuzzy controller are smaller than the PID controller.

The system is simulated with five different trajectories to test its effectiveness to adjust and produce a good result without further tuning the parameters. In comparison of spiral trajectory tracking in this paper and the existing study [10], the value is not as big, where in this work only 27.31% of reduction was recorded compared to 87.62% in the existing work. However, such a difference is expected as the existing work applied additional algorithm to generate the scaling factor for the Fuzzy controller, while this paper proposes a simpler algorithm for self-tuning Fuzzy PID controller. Thus, limitations should be expected along with the results.

The proposed method is proven to be effective theoretically. However, the application of the system to real-time may require further adjustment and additional algorithm as the system in this paper is modelled without considering the factors such as power consumption, external disturbances, and noises.

5. Conclusion

This work aimed to obtain a mathematical model of a quadcopter and design a controller to control position and attitude for trajectory tracking. A model of quadcopter based on DJI Tello drone is modelled and two control techniques are designed. The designed control systems are a conventional PID controller and a self-tuning Fuzzy PID controller. The proposed method used a simple Fuzzy controller algorithm. The simulations of five different trajectories shows the results improved and better with the integration of Fuzzy technique, where the values of tracking errors reduced with the application of Fuzzy controller. However, in order to apply the system to real-world application, further adjustments might be required, especially to overcome the external disturbances and noises. In the coming future works, the proposed method will be further evaluated and tested in both simulation and real-time applications with the presences of disturbances, uncertainties, and noises.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research was funded by the name of Ministry of Higher Education (MOHE), grant number (FRGS/1/2021/TK0/UTM/02/56).

Acknowledgment: The authors would like to thank Ministry of Higher Education (MOHE) through Fundamental Research Grant Scheme (FRGS/1/2021/TK0/UTM/02/56) and Universiti Teknologi Malaysia under UTMFR (Q.J130000.3823.22H67) for supporting this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

- [1] S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, and M. H. Alsharif, "Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review," *Drones*, vol. 6, no. 6, p. 147, 2022, <https://doi.org/10.3390/drones6060147>.
- [2] F. Ahmed, J. C. Mohanta, A. Keshari, and P. S. Yadav, "Recent Advances in Unmanned Aerial Vehicles: A Review," *Arabian Journal for Science and Engineering*, vol. 47, 2022, <https://doi.org/10.1007/s13369-022-06738-0>.
- [3] Rabab Benotsmane and József Vásárhelyi, "Towards Optimization of Energy Consumption of Tello Quad-Rotor with Mpc Model Implementation," *Energies*, vol. 15, no. 23, pp. 9207–9207, 2022, <https://doi.org/10.3390/en15239207>.
- [4] S. A. Kouritem, M. Mahmoud, N. Nahas, M. Abouheaf, and A. M. Saleh, "A self-adjusting multi-objective control approach for quadrotors," *Alexandria Engineering Journal*, vol. 76, pp. 543–556, 2023, <https://doi.org/10.1016/j.aej.2023.06.050>.
- [5] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Computer Communications*, vol. 149, pp. 270–299, 2020, <https://doi.org/10.1016/j.comcom.2019.10.014>.
- [6] I. S. Leal, C. Abeykoon, and Y. S. Perera, "Design, Simulation, Analysis and Optimization of PID and Fuzzy Based Control Systems for a Quadcopter," *Electronics*, vol. 10, no. 18, p. 2218, 2021, <https://doi.org/10.3390/electronics10182218>.
- [7] M. N. Shauqee, P. Rajendran, and N. Mohd Suhadis, "Quadrotor Controller Design Techniques and Applications Review," *INCAS Buletin*, vol. 13, no. 3, pp. 179–194, 2021, <https://doi.org/10.13111/2066-8201.2021.13.3.15>.
- [8] D. Li, "Fuzzy PID Controller to control the attitude of Quadrotor UAV," *Gyancity Journal of Engineering and Technology*, vol. 6, no. 1, pp. 1–11, 2020, <https://doi.org/10.21058/gjet.2020.61004>.
- [9] H. Zhang and Q. Feng, "Design and Implementation of Attitude Control for Quadrotor UAV Based on Adaptive Fuzzy PID," *2022 41st Chinese Control Conference (CCC)*, pp. 3533–3538, 2022, <https://doi.org/10.23919/ccc55666.2022.9901689>.
- [10] A. Eltayeb, M. F. Rahmat, M. A. M. Eltoum, M. H. S. Ibrahim, and M. A. M. Basri, "Trajectory Tracking for the Quadcopter UAV utilizing Fuzzy PID Control Approach," *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEE)*, pp. 1–6, 2021, <https://doi.org/10.1109/icceee49695.2021.9429636>.
- [11] S. Musa, "Techniques for Quadcopter modeling and Design: A Review," *Journal of unmanned system technology*, vol. 5, no. 3, pp. 66–75, 2018, <http://ojs.unsysdigital.com/index.php/just/article/view/10.21535%252Fjust.v5i3.981/0>.
- [12] A. K. Mohsin and J. A. Abdulhady, "Comparing dynamic model and flight control of plus and cross quadcopter configurations," *FME Transactions*, vol. 50, no. 4, pp. 683–692, 2022, <https://doi.org/10.5937/fme2204683M>.
- [13] M. Nakamura, K. Takaya, H. Ohta, K. Shibayama, and V. Kroumov, "Quadrotor Modeling and Simulation for Industrial Application," *2019 23rd International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 37–42, 2019, <https://doi.org/10.1109/icstcc.2019.8885708>.

-
- [14] T. Oktay and O. Köse, "Dynamic Modeling and Simulation of Quadcopter for Several Flight Conditions," *European journal of science and technology*, no. 15, pp. 132–142, 2019, <https://doi.org/10.31590/ejosat.507222>.
- [15] B. Elbasueny, M. I. A. El-Sebah, and S. Shaaban, "Dynamic Modelling and Control of quadrotor's attitude around its hover position," *2019 6th International Conference on Advanced Control Circuits and Systems (ACCS) & 2019 5th International Conference on New Paradigms in Electronics & information Technology (PEIT)*, pp. 247-252, 2019, <https://doi.org/10.1109/accs-peit48329.2019.9062846>.
- [16] L. Zhou, A. Pljonkin, and P. K. Singh, "Modeling and PID control of quadrotor UAV based on machine learning," *Journal of Intelligent Systems*, vol. 31, no. 1, pp. 1112–1122, 2022, <https://doi.org/10.1515/jisys-2021-0213>.
- [17] J. Rao, B. Li, Z. Zhang, D. Chen, and W. Giernacki, "Position Control of Quadrotor UAV Based on Cascade Fuzzy Neural Network," *Energies*, vol. 15, no. 5, p. 1763, 2022, <https://doi.org/10.3390/en15051763>.
- [18] O. Bouaiss, R. Mechougou and R. Ajgou, "Modeling, Control and Simulation of Quadrotor UAV," *2020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP)*, pp. 340-345, 2020, <https://doi.org/10.1109/ccssp49278.2020.9151687>.
- [19] R. Kowalik, T. Łusiak, and A. Novak, "A Mathematical Model for Controlling a Quadrotor UAV," *Transactions on Aerospace Research*, vol. 2021, no. 3, pp. 58–70, 2021, <https://doi.org/10.2478/tar-2021-0017>.
- [20] M. R. Kaplan, A. Eraslan, A. Beke, and T. Kumbasar, "Altitude and Position Control of Parrot Mambo Minidrone with PID and Fuzzy PID Controllers," *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, pp. 785-789, 2019, <https://doi.org/10.23919/ELECO47770.2019.8990445>.
- [21] W. Giernacki, J. Rao, S. Sladic, A. Bondyra, M. Retinger and T. Espinoza-Fraire, "DJI Tello Quadrotor as a Platform for Research and Education in Mobile Robotics and Control Engineering," *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 735-744, 2022, <https://doi.org/10.1109/icuas54217.2022.9836168>.
- [22] S. Zouaoui, E. Mohamed, and B. Kouider, "Easy Tracking of UAV Using PID Controller," *Periodica Polytechnica Transportation Engineering*, vol. 47, no. 3, pp. 171–177, 2018, <https://doi.org/10.3311/pptr.10838>.
- [23] N. S. Zuñiga-Peña, N. Hernández-Romero, J. C. Seck-Tuoh-Mora, J. Medina-Marin, and I. Barragan-Vite, "Improving 3D Path Tracking of Unmanned Aerial Vehicles through Optimization of Compensated PD and PID Controllers," *Applied Sciences*, vol. 12, no. 1, p. 99, 2021, <https://doi.org/10.3390/app12010099>.
- [24] A. Iyer and H. O. Bansal, "Modelling, Simulation, and Implementation of PID Controller on Quadrotors," *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1-7, 2021, <https://doi.org/10.1109/iccci50826.2021.9402301>.
- [25] A. Noordin, M. A. Mohd Basri, Z. Mohamed, and I. Mat Lazim, "Adaptive PID Controller Using Sliding Mode Control Approaches for Quadrotor UAV Attitude and Position Stabilization," *Arabian Journal for Science and Engineering*, vol. 46, no. 2, pp. 963–981, 2020, <https://doi.org/10.1007/s13369-020-04742-w>.
- [26] A. T. Bayisa and G. Li-Hui, "Controlling quadcopter altitude using PID-control system," *International Journal of Engineering Research & Technology (IJERT)*, vol. 8, pp. 195-199, 2019, <https://doi.org/10.17577/ijertv8is120118>.
- [27] M. Karahan and C. Kasnakoglu, "Modeling and Simulation of Quadrotor UAV Using PID Controller," *2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1-4, 2019, <https://doi.org/10.1109/ECAI46879.2019.9042043>.
- [28] V. V. Patel, "Ziegler-Nichols Tuning Method," *Resonance*, vol. 25, no. 10, pp. 1385–1397, 2020, <https://doi.org/10.1007/s12045-020-1058-z>.
-

-
- [29] S. M. Othman *et al.*, "Position Tracking Performance with Fine Tune Ziegler-Nichols PID Controller for Electro-Hydraulic Actuator in Aerospace Vehicle Model," *Journal of physics*, vol. 2107, no. 1, pp. 012064–012064, 2021, <https://doi.org/10.1088/1742-6596/2107/1/012064>.
- [30] N. H. Sahrir and M. A. Mohd Basri, "Modelling and Manual Tuning PID Control of Quadcopter," in *Control, instrumentation and mechatronics: Theory and practice*, pp. 346–357, 2022, https://doi.org/10.1007/978-981-19-3923-5_30.
- [31] M. Nassim and A. Abdelkader, "Speed Control of DC Motor Using Fuzzy PID Controller," *arXiv preprint arXiv:2108.05450*, 2021, <https://doi.org/10.48550/arXiv.2108.05450>.
- [32] H. H. Manap, A. K. Abdul Wahab, and F. Mohamed Zuki, "Control for Carbon Dioxide Exchange Process in a Membrane Oxygenator Using Online Self-Tuning Fuzzy-PID Controller," *Biomedical Signal Processing and Control*, vol. 64, p. 102300, 2021, <https://doi.org/10.1016/j.bspc.2020.102300>.
- [33] L. Farah, A. Haddouche, and A. Haddouche, "Comparison between proposed fuzzy logic and ANFIS for MPPT control for photovoltaic system," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 11, no. 2, p. 1065, 2020, <https://doi.org/10.11591/ijpeds.v11.i2.pp1065-1073>.
- [34] Y. Di and Z. Chen, "Discussion on tuning method of quantization factor of fuzzy PID controller," *Journal of physics*, vol. 2232, no. 1, pp. 012007–012007, 2022, <https://doi.org/10.1088/1742-6596/2232/1/012007>.
- [35] S. S. Kumar and G. Anitha, "A Novel Self-Tuning Fuzzy Logic-Based PID Controllers for Two-Axis Gimbal Stabilization in a Missile Seeker," *International Journal of Aerospace Engineering*, vol. 2021, pp. 1–12, 2021, <https://doi.org/10.1155/2021/8897556>.
- [36] R. Baz, K. E. Majdoub, F. Giri, and A. Taouni, "Self-tuning fuzzy PID speed controller for quarter electric vehicle driven by In-wheel BLDC motor and Pacejka's tire model," *IFAC-PapersOnLine*, vol. 55, no. 12, pp. 598–603, 2022, <https://doi.org/10.1016/j.ifacol.2022.07.377>.
- [37] E. Güçlü, İ. Aydın, and E. Akin, "Fuzzy PID Based Autonomous UAV Design for Railway Tracking," *2021 International Conference on Information Technology (ICIT)*, pp. 456-461, 2021, <https://doi.org/10.1109/icit52682.2021.9491687>.
- [38] D. Nimawat and N. Nagar, "Self-Tuning Fuzzy Pid Controllers Application on Industrial Hydraulic Actuator Using Unique System Identification Approach," *Journal of Communications Technology, Electronics and Computer Science*, vol. 22, pp. 1–6, 2019, <https://doi.org/10.22385/jctecs.v22i0.283>.
- [39] M. Rabah, A. Rohan, Y.-J. Han, and S.-H. Kim, "Design of Fuzzy-PID Controller for Quadcopter Trajectory-Tracking," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 3, pp. 204–213, 2018, <https://doi.org/10.5391/ijfis.2018.18.3.204>.
- [40] R. Kristiyono and W. Wiyono, "Autotuning Fuzzy PID Controller for Speed Control of BLDC Motor," *Journal of Robotics and Control (JRC)*, vol. 2, no. 5, 2021, <https://doi.org/10.18196/jrc.25114>.
- [41] K. Ilija, V. Čongradac, and F. Kulic, "A novel fuzzy logic scheme for PID controller auto-tuning," *Automatika*, vol. 63, no. 2, pp. 365–377, 2022, <https://doi.org/10.1080/00051144.2022.2043988>.