Jurnal Kejuruteraan 38(4) 2026: 1757-1771
https://doi.org/10.17576/jkukm-2026-38(4)-17

# Design and Simulation of ANFIS-PID for Trajectory Tracking Control of a Quadcopter UAV

A'dilah Baharuddin[a], Mohd Ariffanan Mohd Basri[a*], Aminurrashid Noordin[b] & Siti Marhainis Othman[c]

[a]*Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia.*

[b]*Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, Durian Tunggal, Melaka 76100, Malacca, Malaysia.*

[c]*Faculty of Electrical Engineering & Technology, Universiti Malaysia Perlis, Pauh Putra Campus, 02600 Arau, Perlis, Malaysia.*

[*]*Corresponding author: ariffanan@fke.utm.my*

*Received 11 July 2025, Received in revised form 14 March 2026*
*Accepted 14 April 2026, Available online 30 July 2026*

ABSTRACT

*The application of quadcopters continues to grow across various industries to execute duties requiring competent flight control and tracking capabilities. Due to the under-actuation, non-linearity, and instability of the system, which are the nature of the quadcopter, challenges are encountered in designing a proper controller. One of the most used controllers is the PID controller. A PID controller demands precise tuning for accurate and steady operation. However, certain modeling techniques of the quadcopter cause Ziegler-Nichols, the commonly used tuning method, to become inapplicable. Consequently, a manual tuning approach, which consumes a long adjustment time and is prone to contribute desirable responses, is used. To overcome the disadvantages that resulted from the manual tuning process, an attempt to optimize the performance of the system by designing an auto-tuning tool for the PID parameters using an adaptive neuro-fuzzy inference system (ANFIS). Based on the simulations of the manually tuned PID controller, the training data for the ANFIS system is collected. The trained data are utilized as parameters to generate the auto-tuning process in the designed ANFIS-PID. The ANFIS-PID controller is simulated with five different trajectories, and the results are analyzed by comparison with the PID controller. The analysis found that several small improvements are achieved with the ANFIS-PID. Hence, further adjustments and fine-tuning are suggested to be applied during the training sessions. In the future, the disturbances and obstacle factors should be taken into consideration in designing and modeling the quadcopter and its system to make it more applicable in real time.*

*Keywords: ANFIS; neuro-fuzzy; PID controller; quadcopter; trajectory tracking; UAV*

## INTRODUCTION

A quadcopter is a four-engine unmanned aerial vehicle (UAV), commonly addressed as a drone. Besides the quadcopter, the types of drones include the two-engine drone (bicopter), the three-engine drone (tricopter), the six-engine drone (hexacopter), and the eight-engine drone (octocopter). The number of engines also indicates the number of arms and rotors, which are the fundamental components of drones and help determine the frame types (Custers 2016). These drones are categorized as multicopters, a category of drones with more than one rotor. Other categories of drones include the single-rotor, fixed-wing, and hybrid vertical take-off and landing (VTOL) drones.

Originally, drones were primarily used in the military domain, especially for warfare in remote zone places, where the first type was used for almost 150 years as bombs made of balloons filled with explosives before a primitive automatic plane was made (Custers 2016; Cauchard et al. 2015). However, nowadays, the personal or civilian drones, which are smaller in size and motor-driven, are increasing with the availability and affordability (Idrissi et al. 2022; Custers 2016; Cauchard et al. 2015). Due to the rapid increase of interest from the research community in drones, continuous development has been applied to drones, making them more applicable for widespread use by both civilians and the military.

The application of drones has spread over various fields and sectors in which they are used publicly for monitoring, surveying, mapping, observing, filming, and inspection activities such as crime prevention and scene reconstruction, environmental filming, geological surveys, disaster scene inspection, search and rescue operations, and other operations demanding long distances, long ranges, and high altitudes (Idrissi et al. 2022; Cauchard et al. 2015). Due to their simple mechanical structure, high maneuverability, affordability, and ability to hover and perform VTOL, multicopters, especially quadcopters, are gaining significant attention compared to other types of drones (Burggräf et al. 2019; Rao et al. 2022).

In performing those activities, trajectory tracking with stable flight becomes a critical task for a quadcopter, especially in the lack of sufficient time for the quadcopter to implement measures and decision-making procedures (Azar et al. 2021). In order to equip a quadcopter with the ability, a properly designed controller is required. However, due to the nature of the system, where the quadcopter is an under-actuated, non-linear, unstable system that operates with six degrees of freedom, designing a controller is challenging, especially since the control inputs of the quadcopter are six, in contrast to controller systems with four inputs only (Idrissi et al. 2022; Noordin et al. 2020; Bayisa 2019).

Consequently, to find a proper controller for a quadcopter, the research community has shown much interest in various approaches and algorithms, including Proportional-Integral-Derivative (PID) (Kowalik et al. 2021; Nakamura et al. 2019; Satla et al. 2018), Linear-Quadratic-Regulator (LQR), H-infinity, Backstepping, Sliding Mode, and others. With advantages such as simple structure, robustness, accuracy, stability, and a smaller number of tuning parameters, the PID controller is widely used in industry, including in quadcopters (Mahroona et al. 2021; Zhang 2020; Kose & Oktay 2019; Chopra et al. 2014).

However, PID controllers require proper tuning, and some methods like Ziegler-Nichols' may not work for certain systems even though they are simple and practical because the method is not very precise (Cheng 2017). When Ziegler-Nichols' method fails to achieve sustained oscillation, it becomes inapplicable for tuning the system (Yun, L. et al. 2006). Meanwhile, the manual tuning method led to a long adjustment time and produced a response with a large steady-state error (Han et al. 2018).

In order to design a controller system for a quadcopter, this paper derived the model of the quadcopter and simulated the model with a PID controller. With the aim of auto-tuning the PID controller for better performance, the Adaptive Neuro-Fuzzy Inference System (ANFIS) is implemented with the PID structure. The model is simulated with MATLAB/Simulink and put to a comparison to analyze the performance with both controllers. The contributions of this paper are listed as follows:

1. Designing ANFIS-PID controllers using a derived mathematical model of a quadcopter based on PID controller's data.
2. Evaluating, comparing, and analyzing the performance of PID and ANFIS-PID controllers in tracking different trajectories.

This paper is organized in five sections. Section 2 derived and explained the dynamic model in × configuration, while section 3 explained the PID and ANFIS-PID controller designs for quadcopter trajectory tracking. Section 4 presented the results and discussed the analysis conducted for the controllers' performance. Finally, section 5 concluded the work and all sections.

## METHODOLOGY

A quadcopter's rotors are commonly assembled in either a + configuration or × configuration. Six variables are used to represent the positions and orientations of a quadcopter according to the six-degree-of-freedom feature of an aircraft. An × configuration quadcopter model is derived mathematically, and this section describes its derivation and parameters.

For × configuration quadcopters, the rotors are running in pairs with different directions to eliminate anti-torque effects and produce movements of translation and rotation, which are generated by lift forces produced by the rotating propellers (Noordin et al. 2020; Bayisa, A., 2019). The movement of a quadcopter can be generated with changes and manipulation to the speed of rotating propellers. Based on Figure 1, quadcopter movements are summarized in Table 1 (Noordin, A., et al. 2023).

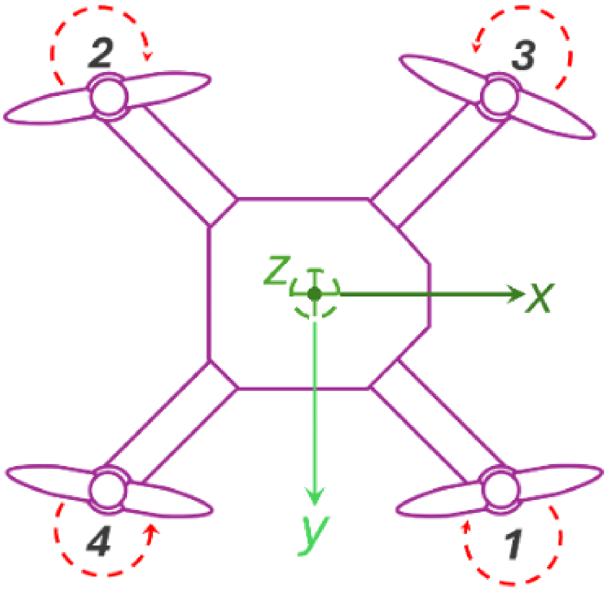


FIGURE 1. The illustration of the × configuration quadcopter.

Several assumptions about the quadcopter are made in the mathematical model to simplify the model's derivation. The assumptions made are as follows (Noordin et al. 2020):

1. The quadcopter is assumed to be a rigid body with rigid propellers in which no blade flapping occurs.
2. The quadcopter structure is assumed to be symmetric with respect to the $x$, $y$-axes to use + configuration degree of freedom for × configuration.
3. The body-fixed frame origin and the mass center are coinciding.
4. The propellers work under similar conditions at any time to ensure the coefficient and reaction torque coefficient are the same.

TABLE 1. The movement generations of the × configuration quadcopter

| Movement | Speed Manipulation |
|---|---|
| Forward | Rotors 1 & 3 slower than rotors 2 & 4 |
| Backward | Rotors 2 & 4 slower than rotors 1 & 3 |
| Rightward | Rotors 2 & 3 slower than rotors 1 & 4 |
| Leftward | Rotors 1 & 4 slower than rotors 2 & 3 |
| Upward | Increase either clockwise-rotating rotors or anti-clockwise-rotating rotors |
| Downward | Decrease either clockwise-rotating rotors or anti-clockwise-rotating rotors |
| Hovering | Equalize the speed of all rotors |

## MATHEMATICAL MODEL

Figure 2 shows the orientation and axes of the body-fixed frame, $Q = \{x_Q, y_Q, z_Q\}$, of a quadcopter along with the earth-fixed frame, $E = \{x_E, y_E, z_E\}$, that satisfies $\{Q\}_T = R_T \, x \, \{E\}_T$, the generalized coordinates of the quadcopter (Eltayeb, A., et al. 2021; Noordin et. al, 2020). The earth-fixed frame refers to the fixed coordinate system depicted by $x$, $y$, and $z$ coordinates with the origin at the center of the Earth.

Meanwhile, the body-fixed frame represents a mobile coordinate system with respect to the gravity body's center, depicted by $\phi$, $\theta$, and $\psi$ angles. Figure 2 also shows the orientations and positions of the quadcopter, which are generalized with six degrees of freedom.

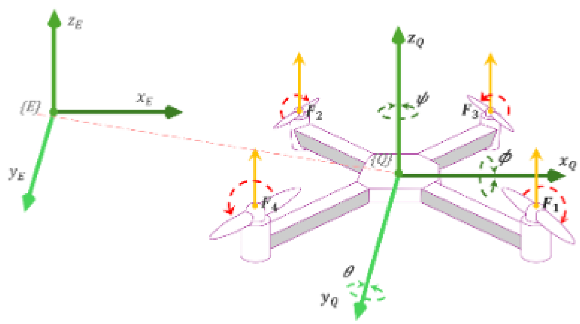


FIGURE 2. The body-fixed and earth-fixed frames of a quadcopter.

The relative positions of the quadcopter are classified as a translational subsystem, as in Equation (1), and the orientations of the quadcopter, which are also called its attitudes, are classified as a rotational subsystem, as in Equation (2). Equation (1) represents the position of the drone, and Equation (2) represents its attitude.

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

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

By using Euler angles, the elementary rotations of the quadcopter about the x, y, and z axes are defined, where the quadcopter orientation with respect to the inertial axis produced the single rotation of roll ($\phi$), pitch ($\theta$), and yaw ($\psi$), where roll is the radius around the $x$-axis, pitch is the radius around the $y$-axis, and yaw is the radius around the z-axis (Abdulkareem, A., et al. 2022; Usman, 2020).

The rotations are represented by $R_\phi$ in Equation (3), $R_\theta$ in Equation (4), and $R_\psi$ in Equation (5) and used to produce Equation (7), the representation of the body frame relative to the inertial frame rotation matrix. Equation (3) to Equation (5) represent the basic Euler angles, while Equation (6) derives the rigid-body orientation based on the combination of Equation (3), Equation (4), and Equation (5).

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

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

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

$$R_T = R_\psi \times R_\theta \times R_\phi \quad (6)$$

The $c$ and $s$ in the matrices symbolize cosine and sine, respectively. The matrix $R_T$ in Equation (6) depicts the relationship between the earth frame, which is treated as the inertial frame, and the body frame.

As the quadcopter is assumed to be a rigid body, the dynamics of a quadcopter are describable using Newton-Euler, in which the translational dynamics are presented by Equation (7), while the rotational dynamics are presented by Equation (8).

$$m\ddot{\xi} = -mgE_z + U_1 R_T E_z \quad (7)$$

$$I\ddot{\eta} = -\dot{\eta} \times I\dot{\eta} - \mathcal{J}_r \dot{\eta} E_z \Omega_d + \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (8)$$

The direction of the z-axis is pointing upward, opposing the gravitational force, as shown in Figure 2. Thus, $g$ in Equation (7) is defined in a negative state. While in Equation (9), the moment of inertia, $l$, is defined as a diagonal matrix of $[I_{xx} I_{yy} I_{zz}]^T$ as it is a 3-by-3 diagonal matrix. The $E_z$ is the $z$-axis vector matrix with the value of $[0 \;\; 0 \;\; 1]^T$, and $U_1$ is the total thrust force generated by four rotors, satisfied as $U_1 = \sum_{I=4}^{4} F_i$ (Abdulkareem, A., et al. 2022; Noordin et. al, 2020). These respective values are inserted and simplified to form the final equation of translational dynamics, which is presented in Equation (9).

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{u_1}{m} \begin{bmatrix} c\phi s\theta c\psi + s\phi s\psi \\ c\phi s\theta s\psi - s\phi c\psi \\ c\phi c\theta \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (9)$$

Meanwhile, the inertia of the rotor that works as the measure of the resistance of the motor attached to the rotor to change its speed is represented by $J_r$ in Equation (8), and it is defined by $J_r = \Omega_1 + \Omega_2 - \Omega_3 - \Omega_4$. It is calculated with the speed of each rotor, according to the rotating directions, where the negative sign represents the anticlockwise direction and the positive sign is the representation of the clockwise direction.

The $\Omega_d$ is the total speed generated by the rotors, where the rotors work in a pair according to the rotating directions, and $U_2$, $U_3$ and $U_4$ are the total torque produced by the rotors (Noordin et. al, 2020). These parameters of Equation (8) are further expanded into Equation (10).

The $U_1$, $U_2$, $U_3$ and $U_4$ are utilized as the control inputs of the model. Through rearrangement of Equation (10), the control inputs are extracted into Equation (11). The arrangement is made according to the torque applied to the drone along an axis, which is the difference between each propeller's applied torques on the other axes (Kose, O. & T. Oktay 2019).

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

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

The derived model is built in MATLAB/Simulink using the parameters listed in Table 2 (Noordin, A., et al. 2023). The Simulink blocks of the translational dynamics based on Equation (10) are shown in Figure 3, while Figure 4 shows the blocks of the rotational dynamics, built based on Equation (9).

Figure 5 demonstrates the rotor speed calculation where the control inputs in Equation (11) are utilized. Equation (11) is the rearrangement of Equation (9) and

Equation (10) with the substitution of $U_n = \sum_{n=1\to4}^{4} b\,\Omega_n^2$, where $b$ is the thrust coefficient and $d$ is the drag coefficient that works in place of the thrust coefficient for $U_4$, while $\Omega_n$ is the angular speed of the rotor.

TABLE 2. The parameters of the quadcopter

| Parameter | Value |
|---|---|
| Mass of quadcopter (*m*) | 1.250 kg |
| Lateral moment arm (*l*) | 0.200 m |
| Thrust coefficient (*b*) | $2.920\text{x}10^{-6}$ Ns$^2$ |
| Drag coefficient (*d*) | $0.112\text{x}10^{-6}$ Nms$^2$ |
| Rolling moment of inertia (*lxx*) | $0.679\text{x}10^{-2}$ kgm$^2$ |
| Pitching moment of inertia (*lyy*) | $0.679\text{x}10^{-2}$ kgm$^2$ |
| Yawing moment of inertia (*lzz*) | $1.313\text{x}10^{-2}$ kgm$^2$ |
| Rotor inertia ($J_r$) | $4.95\text{x}10^{-5}$ kgm$^2$ |
| Gravitational force (*g*) | 9.81 gs$^2$ |

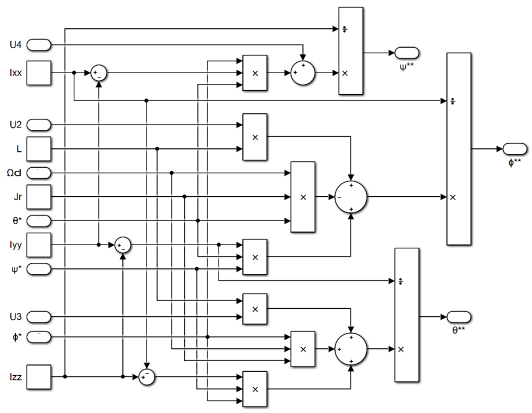


FIGURE 3. The simulation blocks of the translational subsystem

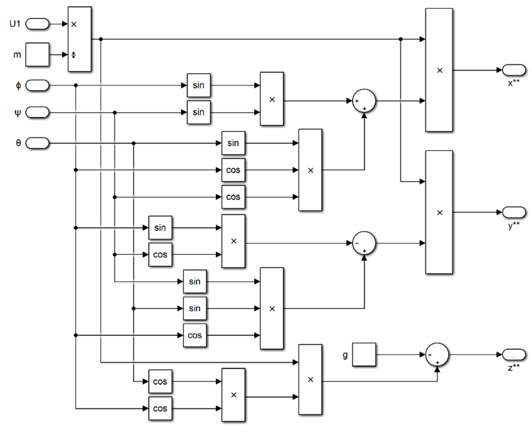


FIGURE 4. The simulation blocks of the rotational subsystem

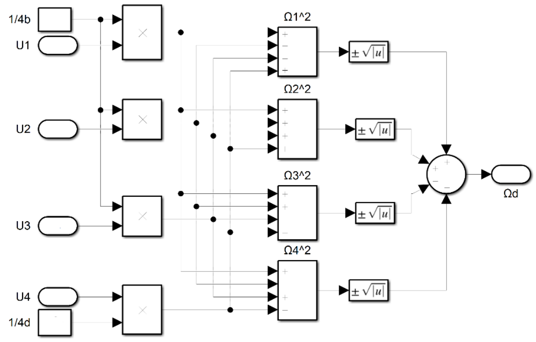


FIGURE 5. The simulation blocks of the rotor speed calculation.

## CONTROLLER FOR QUADCOPTER

Various methods can be used as a controller of the quadcopter to overcome its instability during the flight. This paper applied a PID controller as the base controller and adopted the ANFIS method to the base to automatically tune and improve the performance of quadcopter trajectory tracking. The trajectory to be used will be defined with the desired values of $x_d$, $y_d$, $z_d$ and $\psi_d$.

These values are utilized as the inputs of the controller as shown in Figure 6, the illustration of the derived model of the quadcopter, while Figure 7 shows the Simulink blocks of the quadcopter model with controllers. Both illustrations and block diagram are built upon the derivation of equations in the previous section.

The controllers block consists of position, altitude, and attitude controllers, while the translational and rotational subsystems are built of the blocks in Figure 3 and Figure 4 from the previous section, respectively. According to the previously derived Equation (11), the rotational subsystem acquires input values from the controller subsystem. Hence, a block of rotor speed calculations constructed as shown in Figure 5 is a requirement to complete the model as derived.

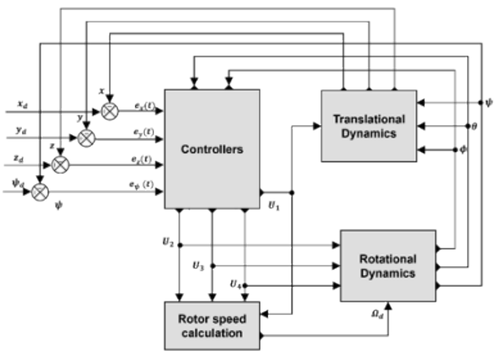


FIGURE 6. The block diagram of the quadcopter model with controller.

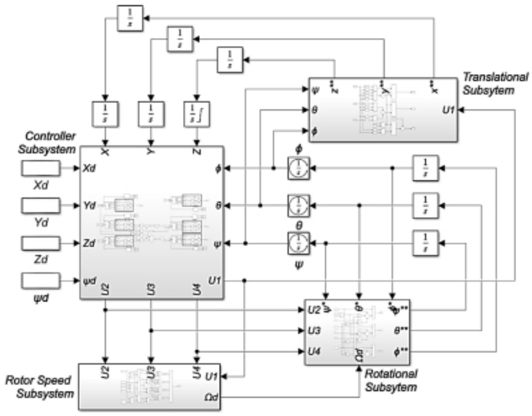


FIGURE 7. The simulation blocks of the quadcopter model with controller.

## PID CONTROLLER

The PID controller corrects the error of differences between the desired responses and the output responses by using its three parameters, which are proportional gain ($k_P$), integral gain ($k_I$), and derivative gain ($k_D$). These parameters are the gains of the controller, which are tuned manually to achieve a better performance.

In a quadcopter system, a PID controller adjusts the speed of motors to achieve the desired orientation. Equation (12) defines the errors that are fed to the controller as inputs, $u_t$ and the relation of inputs and outputs of the controller, where $e(t)$ is the difference between the desired and measured values.

In the manual tuning method, the $k_P$, and $k_I$ are tuned to reduce and remove the steady-state error, yet an overly high value will lead to unstable oscillations. On the other hand, the $k_D$ is increased to reduce the overshoot and settling time of the output responses.

$$u(t) = k_P e(t) + k_I \int_0^t e(t)dt + k_D \frac{de(t)}{dt} \quad (12)$$

Based on Equation (12), the general block diagram of the PID controller is shown in Figure 8. Including Equation (12), the blocks in Figure 8 are utilized for position, altitude, and attitude controllers.

$U_2$, $U_3$ and $U_4$ in Equation (9) are used as the inputs of the attitude controller. Thus, Equation (11) is redefined as in Equation (13) for attitude control, where $n=(\phi,\theta,\psi)$. The same case applied to altitude control, where Equation (11) is reformed into Equation (14), as altitude movement is generated with the total thrust generated by all rotors. Thus, $U_1$ is fed to the controller of the quadcopter's altitude.

However, a different equation is applied to position control. For trajectory tracking, positions x and y of the quadcopter cannot simply be controlled with $U_1$ (Noordin et. al, 2020). Therefore, the inputs of position control are manipulated with roll, $\phi$ and pitch, $\theta$ angles, and small-angle assumptions ($s\phi_d \cong \phi_d$, $s\theta_d \cong \theta_d$, $c\phi_d = c\theta_d = 1$) are applied to Equation (9) to form Equation (15) as the quadcopter is operating in hover positions.

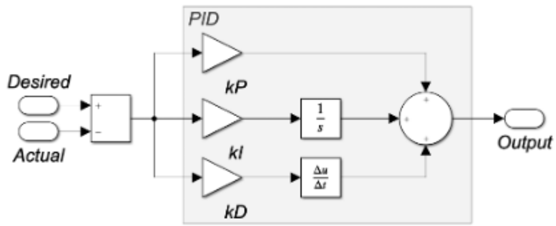


FIGURE 8. The simulation blocks of the PID controller

$$U_{2,3,4}(t) = k_{P_n} e(t)_n + k_{I_n} \int e(t)_n dt + k_{D_n} \frac{de(t)_n}{dt} \quad (13)$$

$$U_1(t) = k_{P_z} e(t)_z + k_{I_z} \int e(t)_z dt + k_{D_z} \frac{de(t)_z}{dt} \quad (14)$$

The $\phi_d$ is the desired roll, and $\phi_d$ is the desired pitch whereas $u_x$ and $u_y$ in Equation (16) are inputs of the control signal, which are then used according to Equation (12). Figure 9 shows the Simulink blocks for position control.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \frac{U_1}{m} \begin{bmatrix} s\psi & c\psi \\ -c\psi & s\psi \end{bmatrix} \begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} \quad (15)$$

$$\begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} = \begin{bmatrix} u_x s\psi_d - u_y c\psi_d \\ u_x c\psi_d + u_y s\psi_d \end{bmatrix} \quad (16)$$

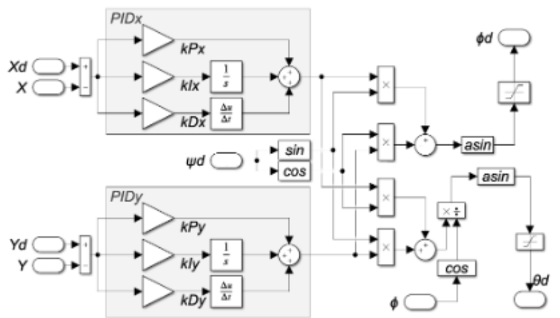


FIGURE 9. The simulation blocks of the position controller

ANFIS-PID CONTROLLER

ANFIS is a homogeneous architecture or a hybrid network, in which it can be addressed as a special neural network with fuzzy parameters (Karim et al. 2018; Jali et al. 2015; Jang, 1993). This architecture takes two inputs, which are the error of differences between the desired responses and the output and derivation of the errors from the PID system, to produce one output. For the quadcopter system, the produced output is fed to the PID controller.

These layers are the compositions of an interconnected neuron set that is connected by direct connections, and each neuron models a parameterized function (Selma et al. 2022). Figure 10 shows the ANFIS architecture of two inputs and one output, which is made of five layers (Hussain et al. 2014). The architecture is based on Takagi-Sugeno's first-order reasoning model and two fuzzy if-then rules. The rules are listed as follows:

1. if $x$ is $A_1$ and $y$ is $B_1$, then $p_1\, xq_1\, y{+}r_1$.
2. if $x$ is $A_2$ and $y$ is $B_2$, then $f_2{=}p_2\, x{+}q_2\, y{+}r_2$.

The x and y are the inputs, while $A_n$ and $B_n$ are the fuzzy sets. The $f_n$ is the outputs within the region of the fuzzy, while $p_{n,}\, q_n$ and $r_n$ on the other hand are the training process determining parameters, with $n = 1, 2$. Each layer of architecture shown in Figure 10 is defined as follows:

1. Layer 1 is a fuzzification process, where it generates the grades of membership through fuzzification with its adaptive nodes.
2. Layer 2 generates the firing strengths. Using the fixed nodes, the firing strengths are generated by multiplying the incoming signals and producing the t-norm operator. The м symbolizes multipliers, while the outputs, $\omega_n$, are the fuzzy strengths of each rule.
3. Layer 3 is a normalization process (И) that generates firing strengths, which is normalized through fixed nodes. The computational normalization factors are chosen by the sum of the weight functions.
4. Layer 4 is a defuzzification process that calculates the Takagi-Sugeno's first-order rules for each fuzzy rule based on the consequent parameters using adaptive nodes.
5. Layer 5 is an output layer that computes the weighted global output, $f$ of the system by summing the incoming signals through a single fixed node.

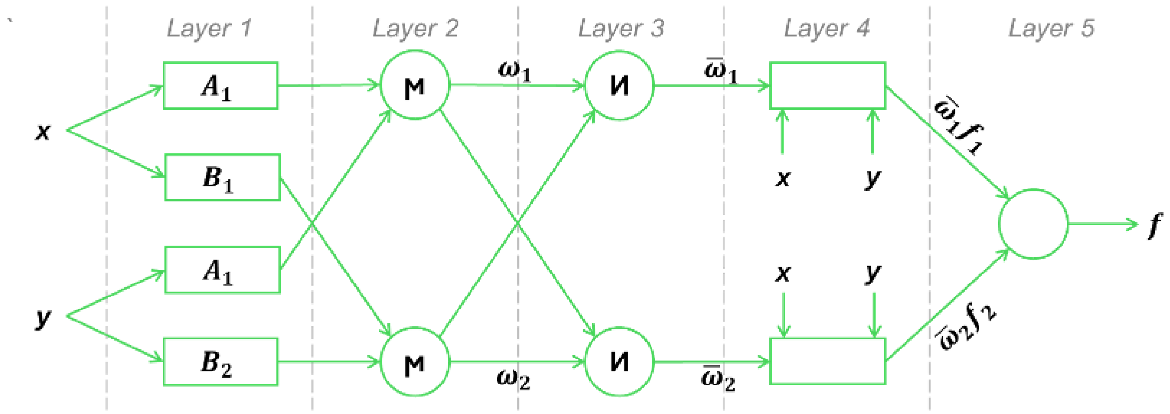


FIGURE 10. The general architecture of ANFIS technique

The data obtained from the simulations of the PID controller are uploaded into the ANFIS Toolbox to go through the training phase. In this paper, the data used for the training phase are the differences between desired and measured values, $e\,(t)$, and the derivative of the differences, $de\,(t)$. These data are recorded by data block along with the output of PID tuned values, as constructed in Figure 11.

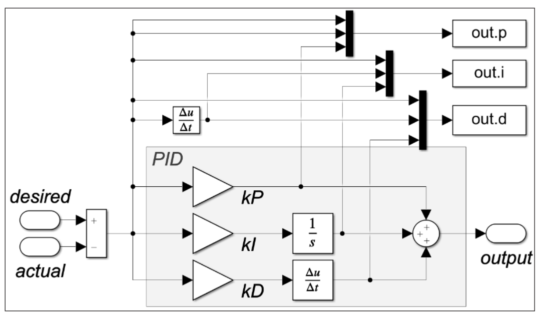


FIGURE 11. The blocks of data recording process for ANFIS training

The training process is conducted using the ANFIS toolbox, where the data are uploaded to be trained with sufficient epochs. In the ANFIS info section shown in Figure 12, the details of the data are listed together with the membership function's number. The Gaussian membership function of 7x7 matrices is used in the fuzzification process, or layer one. The Gaussian function is said to be a smooth and concise notation (Selma et al. 2022).

Figure 13 shows the uploaded data to be trained, and the converged data obtained upon the completion of a training session. The fully trained data is identified as converged when a stable state is reached and the data is no longer decreasing, as a minimum level of acceptable error is reached.

Employing the trained data, the ANFIS-PID controller is built as in Figure 14 for each variable of position, altitude, and attitude. The ANFIS block utilized identical input as the PID controller, with the output from the fuzzy inference system clock of the ANFIS is fed as an adjuster to the PID gains accordingly.

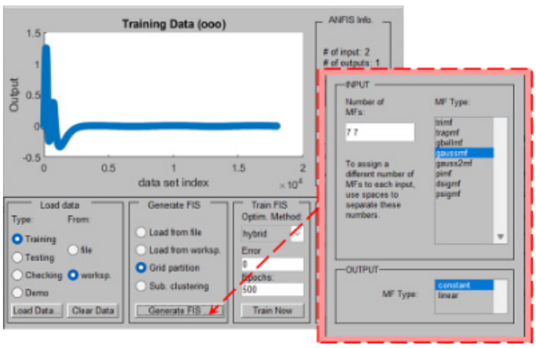


FIGURE 12. The sample training toolbox setting of the ANFIS

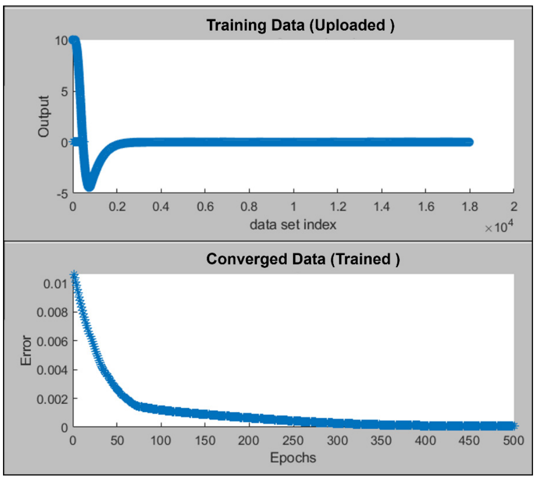


FIGURE 13. The sample of the training data uploaded and the converged data from the training session

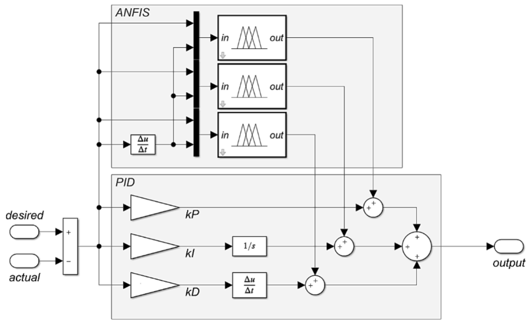


FIGURE 14. The simulation blocks for the ANFIS-PID controller

## PERFORMANCE VALUATION INDEX

In evaluating the performances of controllers for a quadcopter's trajectory tracking, the step responses of the position and altitude are recorded and analyzed for each controller in every simulation conducted. The basic parameters of step responses recorded include the rising time, settling time, and overshoot.

In order to evaluate the improvement made by the controller through the application of the ANFIS, the integral of time-weighted absolute error (ITAE), is measured as the evaluation index of the system in every simulation. As the index emphasizes minimizing the error that occurs in the response, it becomes particularly suitable to be used in evaluating responses with oscillations, like *sine* and *cosine* signals, other than being an ideal evaluating parameter for transient and steady-state responses (Maghfiroh, H., et al. 2021; Almabrok, A., et al. 2018; Kumar, R., et al. 2015).

Equation (17) is the derivation of the ITAE index, where $t_{ss}$ is the steady-state time. The ITAE values for position and altitude are extracted upon the completion of system simulations.

$$ITAE = \int_0^{t_{ss}} t|e(t)|dt \qquad (17)$$

## RESULT AND DISCUSSION

The simulations of the derived model of the quadcopter are conducted for five trajectories, which include a circular, a square, a helical, a lemniscate, and a zigzag. The inputs of the model are the desired positions, altitude, and yaw, while roll and pitch are obtained throughout the simulation.

For each trajectory, the desired yaw is set to $\psi = sin\,(\frac{\pi}{10}t)$, while the altitude of all trajectories except for helical is set as a step response of 1 with an initial value of 0. Exceptionally, the desired altitude of the helical trajectory is set to be a ramp with a 0.5 slope. The desired positions of variables $x$ and $y$ for each trajectory are different, as the positions determine the desired trajectory.

The simulation's step size for every simulation is set to 0.01 s. As ANFIS-PID required data for training, each trajectory is simulated with both PID and ANFIS-PID controllers. The results for both controllers are recorded and compared in order to evaluate the performance of the proposed system. The values of rising time (TR), settling time (TS), overshoot percentage (OP), and ITAE are used in evaluating the performance of systems. These characteristics are measured for the altitude of the quadcopter.

However, due to the applied input signals of the position $x$ and $y$ are not a step response but a continuous signal, no steady state to be achieved in every simulation. Hence, the settling time for the position $x$ and $y$ are not measured.

This paper used the same values of PID parameter gain for the PID controller and the ANFIS-PID controller. The values are listed as in Table 3. The analysis for each trajectory is elaborated into separate subsections accordingly and further discussed in the discussion's subsections at the end of this section.

TABLE 3. The values of the gains of the PID

| Parameters | $x, y$ | $z$ | ϕ,θ,ψ |
|---|---|---|---|
| $k_P$ | 1.00 | 15.0 | 1.00 |
| $k_I$ | 0.15 | 5.00 | 0.50 |
| $k_D$ | 1.80 | 10.0 | 0.10 |

## CIRCULAR TRAJECTORY TRACKING

The desired circular trajectory is defined with desired x, $x_d = 10sin\,(\frac{\pi}{10}t)$, and desired y, $y_d = 10cos\,(\frac{\pi}{10}t)$. The simulation time of circular trajectory tracking is set to 150 s. Figure 15 shows the outputs of circular trajectory tracking with PID and ANFIS-PID controllers in a 3D view and top view. The performances of both PID and ANFIS-PID controllers are evaluated and recorded as listed in Table 4.

Based on the observation, the top view of the responses shows that the ANFIS-PID controller has a noticeable improvement in position tracking, where the response displays a less overshoot curve compared to the PID controller. The listed values in Table 4 also reflected the improvement in overshoot percentage, where the percentage for overshoot of x improved by approximately ≈3.65% with the ANFIS-PID controller at approximately ≈59.78%, while the PID controller was at approximately ≈56.61%.

On the contrary, the percentage of overshoots shows different cases when the percentage increases from ≈0.19% by the PID controller to ≈7.59% in the ANFIS-PID controller.

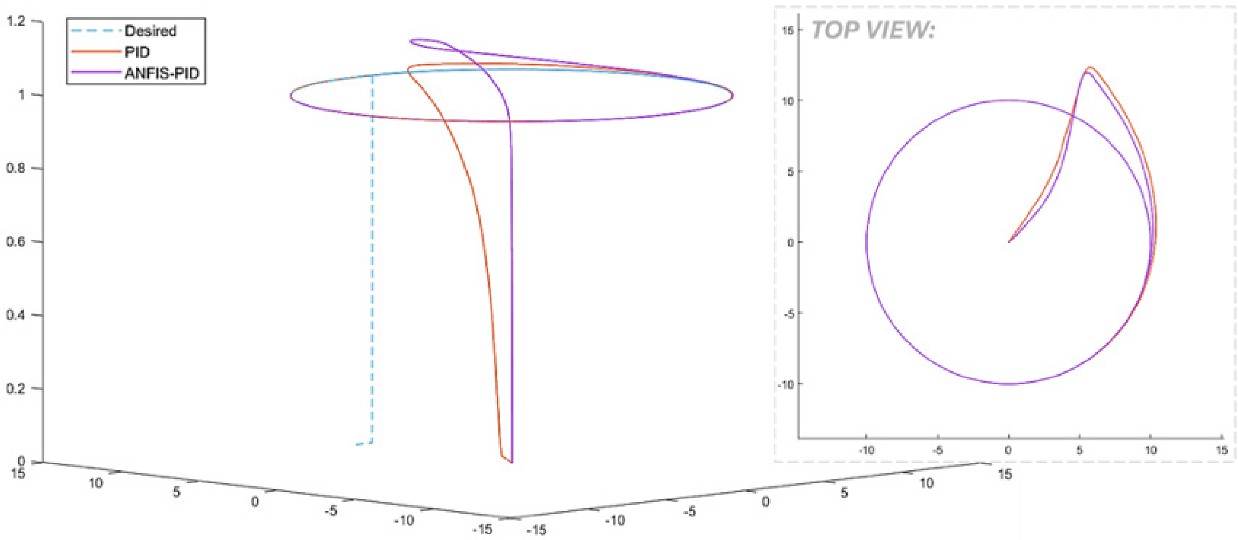


FIGURE 15. The output responses of circular trajectory tracking.

In addition, the rising time became longer with the ANFIS-PID controller. The cases are reflected in Figure 15, as the output response of the ANFIS-PID shows the altitude experienced a larger overshooting curve and a longer time taken to settle at the desired response.

Other than that, the ITAE values measured in the *x*, *y*, and *z* of the ANFIS-PID controller are smaller than those of the PID controller.

TABLE 4. The evaluation of circular trajectory tracking performance

| Control System | Metric | x | y | z |
|---|---|---|---|---|
| PID | TR, *s* | 5.1365 | 6.4021 | 1.9577 |
| | TS, *s* | - | - | 6.3898 |
| | OP, % | 59.777 | 31.578 | 0.191 |
| | ITAE | 1377 | 9282 | 370.6 |
| ANFIS-PID | TR, *s* | 6.1160 | 6.4071 | 0.9297 |
| | TS, *s* | - | - | 10.420 |
| | OP, % | 56.613 | 31.598 | 7.593 |
| | ITAE | 957 | 8574 | 124.4 |

## SQUARE TRAJECTORY TRACKING

The desired inputs of the square trajectory are defined as $x_d$ = [0, 10, 0, -10, 0] and $y_d$ = [10, 0, -10, 0, 10] for $t_{(x, y)}$ = [0, 30, 60, 90, 120]. Figure 16 shows the outputs of square trajectory tracking from 3D and top views.

Although the top view observation found the output responses of both controllers almost overlapped perfectly with the desired response, the changes due to the application of the ANFIS are reflected by the pattern of the responses from the 3D view. Table 5 lists the measured indexes to further analyse the controllers' performance.

TABLE 5. The evaluation of the square trajectory tracking performance

| Control System | Metric | *x* | *y* | *z* |
|---|---|---|---|---|
| PID | TR, *s* | 24.715 | $4.113e^{-4}$ | 1.9552 |
| | TS, *s* | - | - | 6.4225 |
| | OP, % | 0.6400 | $5.846e^{6}$ | 0.1470 |
| | ITAE | 1083 | 9069 | 376.6 |
| ANFIS-PID | TR, *s* | 24.779 | 0.0062 | 0.9289 |
| | TS, *s* | - | - | 9.2132 |
| | OP, % | 0.5697 | $3.955e^{5}$ | 4.3915 |
| | ITAE | 721 | 8403 | 97.03 |

Based on the values listed in Table 5, the ITAEs evaluated of the ANFIS-PID are significantly smaller, where the ITAE of the *x* is reduced by ≈33.43%, the *y* is reduced by ≈7.34%, and the *z* is reduced by ≈74.2%. Other than that, the percentage of overshoots and the rise time measured also reflect the effect of ANFIS implementation. The decrements are measured in the overshoot percentage of the *x* and *y*. Yet the rise time of the variables recorded increments.

On the contrary, the *z* demonstrates the improvement in the rise time by ≈1.03 s. However, the response experienced a larger overshoot and took ≈2.79 s longer than the PID controller to achieve a steady state.

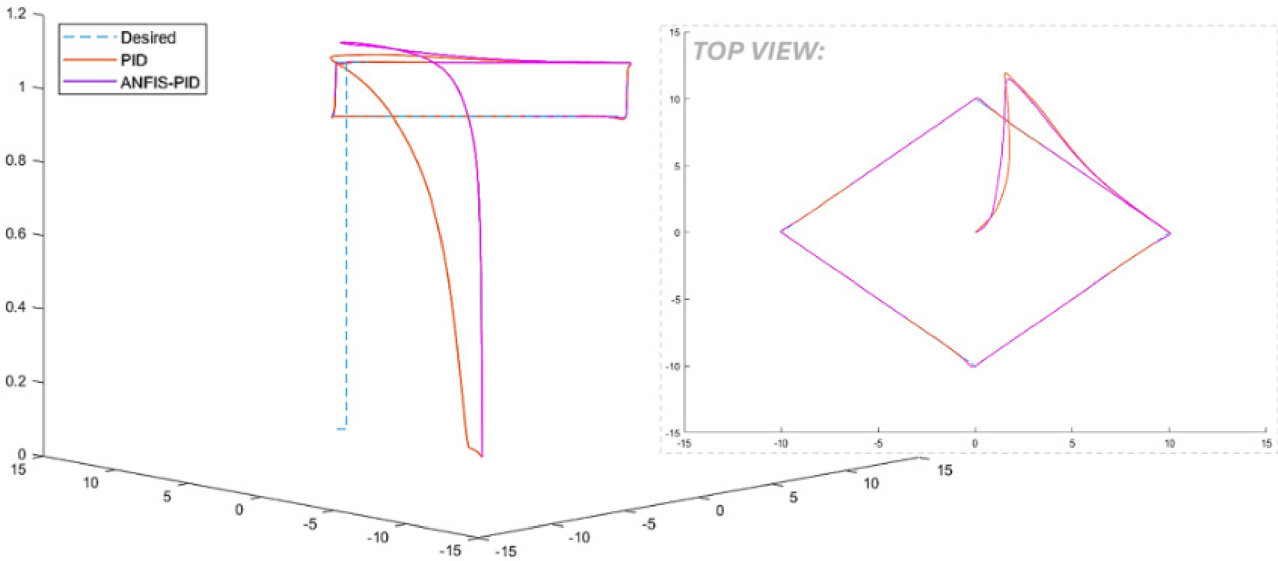


FIGURE 16. The output responses of the square trajectory tracking.

## HELICAL TRACKING TRAJECTORY

Helical desired positions for tracking are defined as if in the circular trajectory. The only change is applied to the desired altitude, as mentioned earlier in this section. The results of the simulation are portrayed in Figure 17. According to the observation, both controllers demonstrated identical tracking behaviors. Thus, the thorough evaluation is conducted through the measured performance metrics listed in Table 6.

The differences between the measured ITAEs justify the behaviors demonstrated by the output responses in Figure 17, as the *x* and *y* of the ANFIS-PID recorded less than a 2% reduction compared to the PID. Meanwhile, the ITAE of the *z* is reduced by ≈47.13% with the application of ANFIS, as the value dropped from 367.9 to 194.5. In return, the percentage of overshoots of the *z* is increased from 0% to ≈7.59%.

TABLE 6. The evaluation of the helical trajectory tracking performance

| Control System | Metrics | *x* | *y* | *z* |
|---|---|---|---|---|
| PID | TR, *s* | 5.7500 | 6.4029 | 120 |
| | OP, % | 57.456 | 31.558 | 0 |
| | ITAE | 996.6 | 8673 | 367.9 |
| ANFIS-PID | TR, *s* | 6.1160 | 6.4071 | 0.9273 |
| | OP, % | 56.613 | 31.598 | 7.593 |
| | ITAE | 988.1 | 8517 | 194.5 |

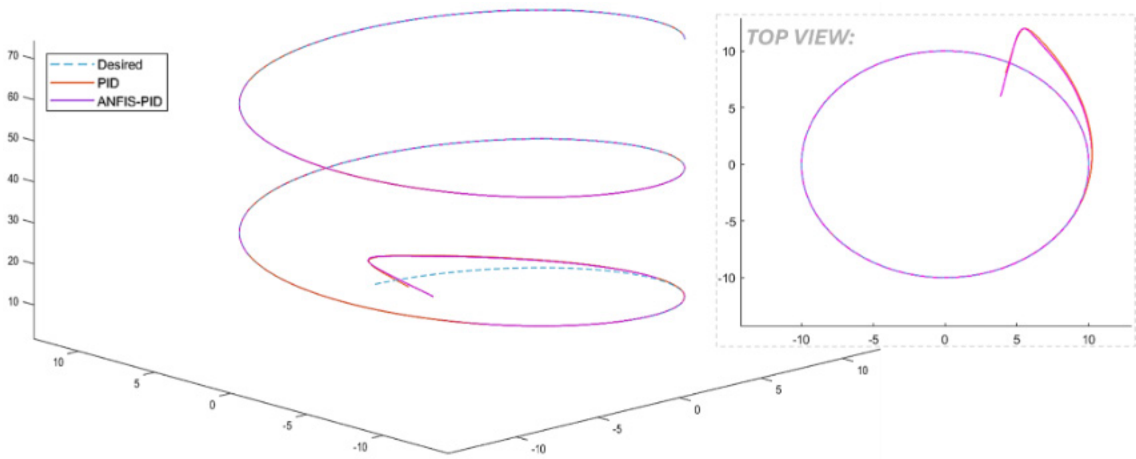


FIGURE 17. The output responses of the helical trajectory tracking.

## LEMNISCATE TRAJECTORY TRACKING

The desired inputs for the lemniscate trajectory racking simulation are defined as $x_d = 10\sin(\frac{\pi}{10}t)$ and $y_d = 10\cos(\frac{\pi}{20}t)$. The observation conducted on the output responses in Figure 18 found that the result of the ANFIS-PID shows a worse performance compared to the PID.

The response is observed to be off-track despite the trajectory being successfully tracked by both controllers. Even so, the *z* of the ANFIS-PID demonstrates better tracking performance. Table 7 records the values of the measured performance evaluation metrics.

Based on the values tabulated in Table 7, the *x* shows significant improvement with the application of ANFIS as the percentage of overshoots recorded is reduced by ≈0.67%. Although the rise time measured shows an almost identical record, a significant reduction is calculated from the value of ITAE, as the value decreased by ≈46.31%.

On the contrary, the effect on the *y* is reflected by the changes in the rise time measured, where the rise time taken is reduced by ≈0.5 s. In return, the percentage of overshoots of the *y* recorded increment by ≈6.2%, and the ITAE of the *y* increased significantly by ≈90.58%.

TABLE 7. The evaluation of the lemniscate trajectory tracking performance

| Control System | Metric | *x* | *y* | *z* |
|---|---|---|---|---|
| PID | TR, *s* | 6.662 | 17.395 | 1.95702 |
| | TS, *s* | - | - | 6.4047 |
| | OP, % | 33.215 | 9.750 | 0.171 |
| | ITAE | 1078 | 9522 | 370.3 |
| ANFIS-PID | TR, *s* | 6.663 | 16.898 | 1.9914 |
| | TS, *s* | - | - | 10.107 |
| | OP, % | 33.191 | 9.685 | 4.3915 |
| | ITAE | 578.8 | $10.11e^4$ | 239.4 |

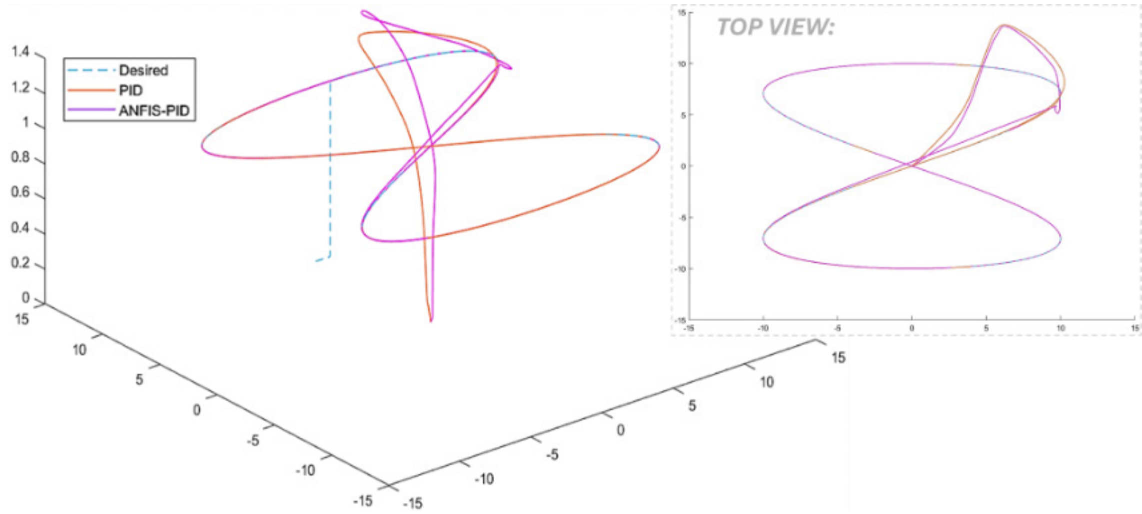


FIGURE 18. The output responses of the lemniscate trajectory tracking

ZIGZAG TRAJECTORY TRACKING

The input signals for the desired zigzag trajectory are defined as xd = [0, 12, 0, -12, 0, 12, 0, -12, 0, 12, 0, -12, 0, 12, 0, -12, 0] and yd = [0, 3, 6, 9, 12, 9, 6, 3, 0, -3, -6, -9, -12, -9, -6, -3, 0] for the time series of t(x, y) = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160]. The simulations for zigzag trajectory tracking are conducted for 160s per session. Figure 19 demonstrates the output responses from the 3D and top views perspectives.

The effect of the ANFIS application can be observed from the behavior of the output response in Figure 19. Although the tracking performance of the *x* and *y* are observed to be almost identical between the controllers, the comparison between the behavior of the *z*'s of the controllers found that some changes have taken place in the system upon the application of ANFIS. Thus, the performance metrics of the zigzag tracking are measured and listed as in Table 8 for further analysis.

From the values in Table 8, the ITAE of the *x* and *y* experienced noticeable reduction, where the *x* recorded ≈1.86% reduction, while the *y* reduced by ≈4.59%. The reduction in these values is justified by the records of the rise time and the percentage of overshoots. The percentage of overshoots of the *x* decreased by ≈0.12% upon the application of ANFIS, but in return, the rise time increased by ≈0.115 s.

On the other hand, despite a significant decrement in the ITAE of the *y*, the rise time and percentage of overshoots are slightly higher when the ANFIS is applied. The rise time increased from 0.3742s to 0.3822s, while the percentage of overshoots increased by ≈1.07%.

Meanwhile, the ITAE of the *z* reduced significantly by ≈71.62%, where the reduction is justified by the reduction in the rise time. The rise time recorded by the *z* is measured to decrease by ≈1s. However, the settling time and the percentage of overshoots increased by ≈2.58 s and ≈4.13%, respectively.

TABLE 8. The evaluation of the zigzag trajectory tracking performance

| Control System | Metric | *x* | *y* | *z* |
|---|---|---|---|---|
| PID | TR, *s* | 0.5495 | 0.3742 | 1.9309 |
| | TS, *s* | - | - | 6.3842 |
| | OP, % | $2.232e^3$ | $8.153e^3$ | 0.255 |
| | ITAE | $16.12e^3$ | 3279 | 430.6 |
| ANFIS-PID | TR, *s* | 0.6646 | 0.3822 | 0.9374 |
| | TS, *s* | - | - | 8.9395 |
| | OP, % | $2.110e^3$ | $9.225e^3$ | 4.3915 |
| | ITAE | $15.82e^3$ | 3135 | 122.2 |

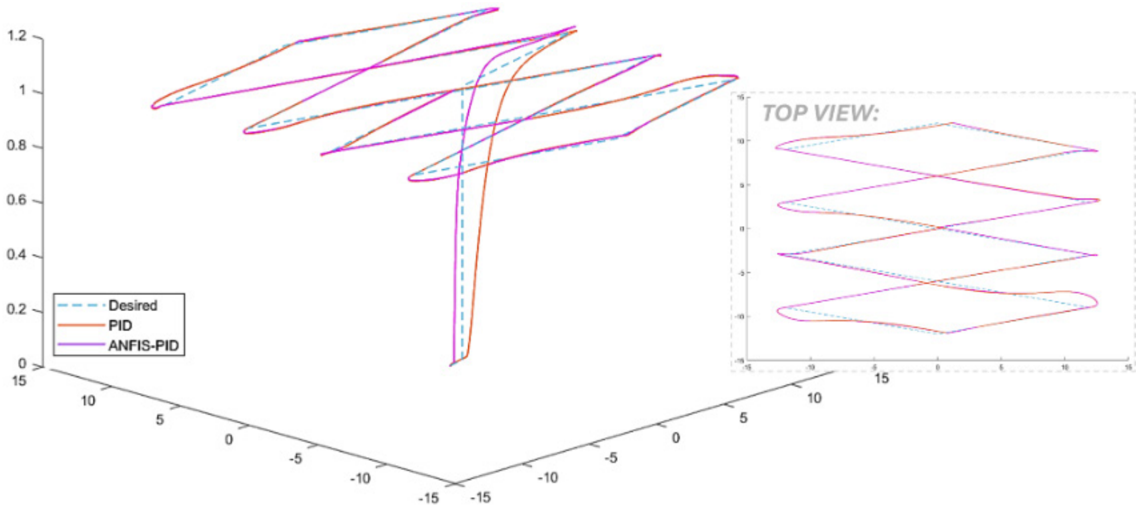


FIGURE 19. The output responses of zigzag trajectory tracking.

## DISCUSSION

In general, the results obtained in every trajectory tracking show almost identical outcomes when almost every ITAE value of ANFIS-PID measured recorded a decrement. This is an indicator of the improvement initiated by the ANFIS to the control system. Based on the results, the position controllers reflect the improvement through the decrements in the percentage of overshoots of either one or both of the *x* and *y*.

However, the reduction in the percentage of overshoots leads to an increment in the rise time. These behaviors justify that the trained ANFIS urged the system to achieve better stability while the desired position was tracked. Meanwhile, the performance of the *z* in every simulation conducted exhibits lower records of rise time and higher settling time and percentage of overshoots. The values indicate that the ANFIS application has urged the system to quickly track and approach the desired *z*.

The pattern of the behaviors of measured metrics has resulted from the inverse relation of the rise time with the settling time and the percentage of overshoot. In relation to the real-time scenario, the output responses reflect that the application of ANFIS-PID has increased the speed of the rotors of the quadcopter. Therefore, the quadcopter took off and flew rapidly in an attempt to approach the desired altitude and position in a shorter time. However, due to the lift and drag forces that by nature act on the quadcopter, oscillation and overshoot are the results of the inertia of the quadcopter. As a result, a longer time will be taken for the quadcopter to reach a steady state and settle at the desired value.

Thus, the output responses of the simulation conducted reflected that the designed ANFIS-PID control has only slightly improved the performance of position tracking, as most of the output responses of position variables show a reduction in overshoot percentage while slightly prolonging the rising time. Moreover, the application of the designed ANFIS-PID has worsened the performance of the altitude in general. Therefore, more adjustment should be conducted to the ANFIS-PID.

In addition, some of the aspects and criteria should be revised in order to fine-tune the proposed controller. One of the suggestions is to properly train the data by normalizing the input data within a finite range (Ulus, Ş. and İ. Eski 2019). In this paper, no input data normalizations are performed in the ANFIS training phase. In order to ensure all data are fully converged, a total of 500 epochs were chosen by trial and error to train the data. However, some data require a smaller number of epochs to be fully converged, and thus some data may experience overfitting.

Other than that, the excessive number of membership functions (MFs) used is also a contributing factor to overfitting occurrence, where the model becomes too complex to properly train on the data (Onu, C. E., et al. 2021; Surajudeen-Bakinde, N., et al. 2021). The applied number of MFs in this paper is 7, which is randomly chosen and may actually be an excessive number for the input data. The bigger number of MFs utilizes more rules and layers, which makes the structure become more complex (Aliferis, C., and G. Simon 2024; Sumarlinda, S.R.I., et al. 2024). Hence, a more precise number should be properly chosen for the data training.

## CONCLUSION

With the aim of tuning the PID controller for the quadcopter system, an ANFIS-PID controller is designed and trained using the data obtained from the PID controller. The ANFIS algorithm is trained with 500 epochs in the fuzzy inference rules made of 7x7 Gaussian membership functions. The designed algorithm is simulated together with a PID controller to track five types of trajectories.

The output responses are observed, and the performance parameters, including rising time, settling time, overshoot percentage, and integral error (ITAE), are collected and evaluated. The comparison between PID controllers and ANFIS-PID shows the ANFIS-PID recorded lower integral error. However, the performance parameters show only small improvements resulted from the application of ANFIS-PID in comparison to PID performances. Furthermore, ANFIS-PID generally produced worsened performance of altitude responses.

The outputs are discussed and believed to be affected by the training phase of the ANFIS algorithm, where the training may have produced an overfitting system. Thus, it is suggested to readjust and tune the algorithm by performing data normalization, using a sufficient epoch size, and choosing a suitable number of membership functions.

In the future, performance-impacting factors such as obstacle existence, wind disturbances, and load additions should be taken into consideration to produce designs and models that are applicable to the physical quadcopter.

## ACKNOWLEDGEMENT

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

## DECLARATION OF COMPETING INTEREST

None.

## REFERENCES

Abdulkareem, A. et al. 2022. Modeling and nonlinear control of a quadcopter for stabilization and trajectory tracking. *Journal of Engineering* 2022(1): 2449901.

Aliferis, C. & Simon, G. 2024. Overfitting, underfitting and general model overconfidence and under-performance pitfalls and best practices in machine learning and AI. In *Artificial Intelligence and Machine Learning in Health Care and Medical Sciences: Best Practices and Pitfalls*, edited by Simon, G. J. & Aliferis, C., 477–524. Cham: Springer International Publishing.

Almabrok, A. et al. 2018. Fast tuning of the PID controller in an HVAC system using the Big Bang–Big Crunch algorithm and FPGA technology. *Algorithms* 11: 146.

Azar, A. T., Koubaa, A., Ali Mohamed, N., Ibrahim, H. A., Ibrahim, Z. F., Kazim, M., Ammar, A., Benjdira, B., Khamis, A. M., Hameed, I. A. & Casalino, G. 2021. Drone deep reinforcement learning: A review. *Electronics* 10(9): 999.

Bayisa, A. 2019. Controlling quadcopter altitude using PID-control system. *International Journal of Engineering Research and* V8.

Burggräf, P., Pérez Martínez, A. R., Roth, H. & Wagner, J. 2019. Quadrotors in factory applications: Design and implementation of the quadrotor's P-PID cascade control system. *SN Applied Sciences* 1(7): 722.

Cauchard, J. R., E, J. L., Zhai, K. Y. & Landay, J. A. 2015. Drone & me: An exploration into natural human-drone interaction. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. Osaka, Japan.

Cheng, H. & Yang, Y. 2017. Model predictive control and PID for path following of an unmanned quadrotor helicopter. In *2017 12th IEEE Conference on*

*Industrial Electronics and Applications (ICIEA)*, 18–20.

Chopra, V., Singla, S. & Dewan, L. 2014. Comparative analysis of tuning a PID controller using intelligent methods. *Acta Polytechnica Hungarica* 11: 235–249.

Custers, B. 2016. Drones here, there and everywhere: Introduction and overview. In Vol. 27, 3–20.

Eltayeb Taha, A. et al. 2021. Quadcopter UAV dynamic modeling and PID trajectory tracking control design: 55–77.

Han, B., Zhou, Y., Deveerasetty, K. K. & Hu, C. 2018. A review of control algorithms for quadrotor. In *2018 IEEE International Conference on Information and Automation (ICIA)*, 951–956.

Hussain, R., Massoud, R. & Al-Mawaldi, M. 2014. ANFIS-PID control FES-supported sit-to-stand in paraplegics: Simulation study. *Journal of Biomedical Science and Engineering* 2014: 208–217.

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): 22.

Jali, M., Mustafa, N., Izzuddin, T., Ghazali, R., Jaafar, H. I. & My. 2015. ANFIS-PID controller for arm rehabilitation device. *International Journal of Engineering and Technology (IJET)* 7.

Jang, J. S. R. 1993. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics* 23(3): 665–685.

Karim, M. R., Dilwar, F. & Siddique, R. A. 2018. Predictive modeling of surface roughness in MQL assisted turning of SiC-Al alloy composites using artificial neural network and adaptive neuro fuzzy inference system.

Kose, O. & Oktay, T. 2019. Dynamic modeling and simulation of quadrotor for different flight conditions.

Kowalik, R., Łusiak, T. & Novak, A. 2021. A mathematical model for controlling a quadrotor UAV. *Transactions on Aerospace Research* 2021: 58–70.

Kumar, R. et al. 2015. Comparison among some well known control schemes with different tuning methods. *Journal of Applied Research and Technology* 13(3): 409–415.

Maghfiroh, H. et al. 2021. Performance evaluation of different objective function in PID tuned by PSO in DC-motor speed control. *IOP Conference Series: Materials Science and Engineering* 1096: 012061.

Mahroona, J., Arbab, N. K. & Farida, M. 2021. A comparative study of PID control for VTOL based UAV system. *Engineering Science and Technology International Research Journal* 5(1): 1–5.

Nakamura, M., Takaya, K., Ohta, H., Shibayama, K. & Kroumov, V. 2019. Quadrotor modeling and simulation for industrial application. In *2019 23rd International Conference on System Theory, Control and Computing (ICSTCC)*, 9–11.

Noordin, A., Mohd Basri, M. A. & Mohamed, Z. 2023. Real-time implementation of an adaptive PID controller for the quadrotor MAV embedded flight control system. *Aerospace* 10(1): 59.

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.

Onu, C. E. et al. 2021. Comparative analysis of RSM, ANN and ANFIS and the mechanistic modeling in eriochrome black-T dye adsorption using modified clay. *South African Journal of Chemical Engineering* 36: 24–42.

Rao, J., Li, B., Zhang, Z., Chen, D. & Giernacki, W. 2022. Position control of quadrotor UAV based on cascade fuzzy neural network. *Energies* 15(5): 1763.

Satla, Z., Elajrami, M. & Bendine, K. 2018. Easy tracking of UAV using PID controller. *Periodica Polytechnica Transportation Engineering* 47.

Selma, B., Chouraqui, S. & Selma, B. 2022. A genetic algorithm-based neuro-fuzzy controller for unmanned aerial vehicle control. *International Journal of Applied Metaheuristic Computing* 13: 1–23.

Sumarlinda, S. R. I. et al. 2024. The improvement prediction model using ANFIS for medical dataset. *Journal of Theoretical and Applied Information Technology* 102: 1663–1672.

Surajudeen-Bakinde, N. et al. 2021. Effect of membership functions and data size on the performance of ANFIS-based model for predicting path losses in the VHF and UHF bands. *Journal of Engineering Research*.

Ulus, Ş. & Eski, İ. 2019. ANFIS based longitudinal controller design of a fixed wing UAV.

Usman, M. 2020. Quadcopter modelling and control with MATLAB/Simulink implementation.

Yun, L., Kiam Heong, A. & Chong, G. C. Y. 2006. PID control system analysis and design. *IEEE Control Systems Magazine* 26(1): 32–41.

Zhang, Z. 2020. Application of PID simulation control mode in quadrotor aircraft. In *2020 International Conference on Computer Engineering and Application (ICCEA)*, 826–829.