Islamic University Journal of Applied Sciences (IUJAS)
https://journals.iu.edu.sa/jesc
Special Issue, February 2026, Pages 111-126
https://igee.univ-boumerdes.dz/ncaeee25/

Enhancing Quadrotor UAV Trajectory Tracking Using Adaptive Fuzzy PID Control

Hocine LOUBAR [1*], Mohammed Idris ARIF [1], Reda Zakaria BAFFOU[1], Razika Zamoum BOUSHAKI [1]

[1] M'hamed Bougara University (UMBB), Boumerdes, Algeria, h.loubar@univ-boumerdes.dz; dm.arif@ensta.edu.dz; redazakariabaffou@gmail.com; r.boushaki@univ-boumerdes.dz

*Corresponding author: (Hocine LOUBAR), *Email Address*: h.loubar@univ-boumerdes.dz

Abstract

Unmanned aerial vehicles (UAVs) require precise and robust control strategies to ensure safe and efficient flight. This work focuses on the Adaptive Fuzzy PID (AFPID) controller as the main method, which integrates classical Proportional-Integral-Derivative (PID) control with fuzzy logic principles to achieve real-time parameter adaptation. The PID and fuzzy PID controllers are considered as baseline methods for performance comparison with the adaptive fuzzy PID. Simulation was done using MATLAB to evaluate the controllers in terms of trajectory tracking, settling time, Root Mean Square error and robustness under disturbance. Since PID and FPID exhibited similar response shapes to AFPID, only their performance metrics were reported for comparison. The results demonstrate that all controllers successfully stabilize the UAV without steady-state error, while AFPID provides slightly improved settling times and disturbance rejection compared to PID and FPID. These findings confirm the effectiveness of adaptive fuzzy control as a reliable solution for UAV path-tracking tasks.

Keywords: Unmanned Aerial Vehicle (UAV); Trajectory tracking; Quadrotor control; Adaptive Fuzzy PID.

https://doi.org/10.63070/jesc.2026.010
Received 20 November 2025; Revised 16 January 2026; Accepted 25 January 2026.
Available online 31 January 2026.
Published by Islamic University of Madinah on behalf of *Islamic University Journal of Applied Sciences*.
This is an open access article published under the Creative Commons Attribution 4.0 International license (CC BY 4.0): https://creativecommons.org/licenses/by/4.0/

1. Introduction

Unmanned aerial vehicles (UAVs) are increasingly employed in diverse applications such as surveillance [1], delivery [2], and Agriculture [3], where precise and robust control strategies are essential to ensure stable flight and accurate trajectory tracking.

In recent quadrotor UAV research, advanced nonlinear based controllers (e.g., backstepping, sliding-mode, and model predictive control) have been widely studied to handle the vehicle's nonlinear dynamics [4]. While these approaches can provide high precision, they often involve complex modeling or heavy computation, making classical proportional–integral–derivative (PID) control attractive for its simplicity and ease of implementation. Quadrotor UAV control has been extensively studied, and PID-based controllers are commonly used but often enhanced with fuzzy logic for greater robustness [5]. For example, Melo et al. (2022) applied a fuzzy gain-scheduling mechanism to adjust PID gains for quadrotor altitude and position control, reporting better trajectory tracking and resilience than a conventional PID [5]. Ye et al. (2024) developed an adaptive fuzzy-PID controller for a quadrotor, finding much lower overshoot and higher disturbance rejection compared to a standard PID [6]. These studies demonstrate that combining a classical PID structure with real-time fuzzy adaptation (AFPID) can significantly enhance UAV flight stability and tracking performance.

In this work, the Adaptive Fuzzy PID (AFPID) controller is presented as the main control approach for trajectory tracking. The AFPID combines the principles of classical proportional–integral–derivative (PID) and fuzzy PID (FPID) control, while introducing adaptive mechanisms that dynamically adjust the control gains in real time to enhance robustness and flexibility. For comparison, PID and FPID are considered as baseline methods. Since their response plots were nearly identical to AFPID, only their performance metrics—such as settling time and root mean square error, and robustness under disturbance —are reported in the results section.

The paper structured as follows. Section II presents the modelling of the quadrotor. Section III details the control methodology, including the baseline controllers and the adaptive fuzzy PID. Section IV describes the simulation setup, while Section V presents and discusses the results. Finally, Section VI provides the conclusion.

2. Dynamic Model of Quadrotor

This section explores the dynamic modelling, which includes forces and torques affecting the quadrotor movement.

As shown in figure 1, two reference frames, also called coordinate system, are usually used to describe the absolute special position and orientation of the quadrotor [7].

The motion of the quadrotor can be classified into two subsystems; rotational subsystem which contains the angles: roll ϕ, pitch θ and yaw ψ. Translational subsystem contains altitude z, and x and y position.

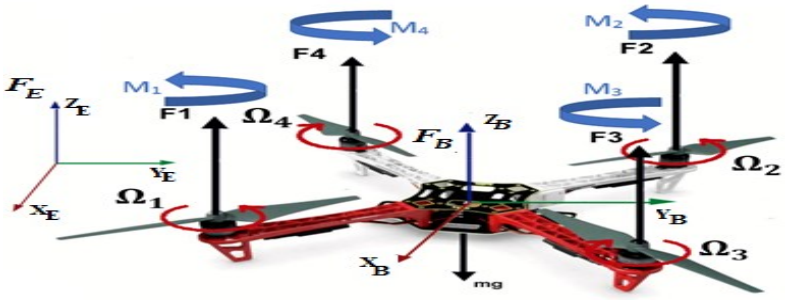


Figure 1. Quadrotor reference frames

2.1 Rotational Subsystem

Using the Newton-Euler method, to deduce the rotational equations of motion which are derived in the body frame [7]:

$$I\dot{\omega} + \omega \times I\omega = M_B \tag{1}$$

Where I is the inertia matrix of the quadrotor and it is a diagonal matrix, ω represents the angular body rate, M_B refers to the moments acting on the quadrotor in the body frame, and $\dot{\omega}$ is the rate of change of angular momentum in the body frame. So, by defining exactly the Inertia matrix and the moment of quadrotor:

- Inertia matrix

The inertia matrix is a diagonal matrix denoted as:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{2}$$

The off-diagonal elements, resulting from inertia, are null because of the quadrotor's symmetry. Where I_{xx}, $I_{yy}$and I_{zz} are the area moments of inertia about the principal axes in the body frame.

- Moments of the Quadrotor (MB):

The rotor generates a moment M_i, which has an opposite direction to the directions of the corresponding rotor. It divides into three moments in the body frame's axis, which are: M_x, M_y and M_z.

The moment of quadrotor will be defined as $M_B = [\, M_x \;\; M_y \;\; M_z]$

$$M_B = \begin{bmatrix} lK_f(-\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 - \Omega_4{}^2) \\ lK_f(\Omega_1{}^2 - \Omega_2{}^2 + \Omega_3{}^2 - \Omega_4{}^2) \\ K_M(\Omega_1{}^2 + \Omega_2{}^2 - \Omega_3{}^2 - \Omega_4{}^2) \end{bmatrix} \quad (3)$$

Where:

l: is the distance between the center of mass of the quadrotor and the axis of rotation of each rotor, $\boldsymbol{K_f}$ and $\boldsymbol{K_M}$ are the aerodynamic force and moment constants, respectively. And $\boldsymbol{\Omega_i}$ is the angular velocity of rotor i.

In addition, by replacing (3) in (1) and around the hover position, small angle assumption is made the equation will be:

$$I\begin{bmatrix}\ddot{\phi}\\ \ddot{\theta}\\ \ddot{\psi}\end{bmatrix} = \begin{bmatrix} lK_f(-\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 - \Omega_4{}^2) \\ lK_f(\Omega_1{}^2 - \Omega_2{}^2 + \Omega_3{}^2 - \Omega_4{}^2) \\ K_M(\Omega_1{}^2 + \Omega_2{}^2 - \Omega_3{}^2 - \Omega_4{}^2) \end{bmatrix} - \begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} \times I \begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} \quad (4)$$

After simplifying (4), the state space functions of rotational subsystem will be:

$$\begin{cases} \ddot{\phi} = \dot{\theta}\dot{\psi}\left(\frac{I_{yy} - I_{zz}}{I_{xx}}\right) + \frac{l}{I_{xx}}K_f\left(-\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 - \Omega_4{}^2\right) \\ \ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{I_{zz} - I_{xx}}{I_{yy}}\right) + \frac{l}{I_{yy}}K_f\left(\Omega_1{}^2 - \Omega_2{}^2 + \Omega_3{}^2 - \Omega_4{}^2\right) \\ \ddot{\psi} = \dot{\phi}\dot{\theta}\left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right) + \frac{1}{I_{zz}}K_M(\Omega_1{}^2 + \Omega_2{}^2 - \Omega_3{}^2 - \Omega_4{}^2) \end{cases} \quad (5)$$

2.2 Translational Subsystem

The translational subsystem is based on Newton's second law of motion in the Earth frame [7]:

$$m\ddot{r} = \begin{bmatrix}0\\0\\-mg\end{bmatrix} - F_a + RF_B \quad (6)$$

Where r refers to the distance between the quadrotor and the inertial frame; $r = [x\ y\ z]^{\mathrm{T}}$, m is the mass of the quadrotor,

and g is the gravitational acceleration ($g = 9.81\ m/s^2$), while F_a is represents the drag forces (negligible), R is the rotation matrix, and F_B refers to the nongravitational forces acting on the quadrotor.

The nongravitational forces are multiplied by the rotation matrix to transform the thrust forces from the body frame into the inertial frame.

In this case, we consider only the horizontal orientation without considering the rolling and the pitching, since the only nongravitational force acting of the quadrotor is the thrust produced by the propellers, hence we obtain:

$$\boldsymbol{F_B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{K_f(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2)} \end{bmatrix} \tag{7}$$

2.3 Rotation matrix

The quadrotor's orientation is defined by the rotation R from the body frame to the inertial frame. This orientation is characterized by roll (ϕ), pitch (θ), and yaw (ψ) angles, which respectively denote rotations about the X, Y, and Z-axes. After simplification it will be:

$$R = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\psi)\sin(\theta)\sin(\phi) - \cos(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ \cos(\theta)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\theta)\cos(\psi) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\theta)\cos(\psi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & cos(\phi)\,cos(\theta) \end{bmatrix} \tag{8}$$

Thus, by putting (7) and (8) in (6), the result will be:

$$m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} (\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\psi)\sin(\theta))(K_f(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2)) \\ (\cos(\phi)\sin(\psi)\sin(\theta) - \cos(\psi)\sin(\phi))\left(K_f\left(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2\right)\right) \\ \cos(\phi)\cos(\theta)\left(K_f\left(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2\right)\right) \end{bmatrix} \tag{9}$$

After simplifying (9), the state space functions of translational subsystem will be:

$$\begin{cases} \ddot{x} = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\frac{K_f\left(\Omega_1{}^2+\Omega_2{}^2+\Omega_3{}^2+\Omega_4{}^2\right)}{m} \\ \ddot{y} = (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\frac{K_f\left(\Omega_1{}^2+\Omega_2{}^2+\Omega_3{}^2+\Omega_4{}^2\right)}{m} \\ \ddot{z} = -g + (\cos\phi\cos\theta)\frac{K_f\left(\Omega_1{}^2+\Omega_2{}^2+\Omega_3{}^2+\Omega_4{}^2\right)}{m} \end{cases} \tag{10}$$

2.4 State Space Representation

Transforming the acquired mathematical model for the quadrotor into a state space model will help make the control problem easier to handle.

In this simulation, we take 12 state variables, i.e., position, velocity, and attitude of an airframe in the inertial frame, and angular rates:

So, X will be represented as a vector.

$$\mathrm{X} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}\ x_{12}]^T \tag{11}$$

Such that:

$$\mathrm{X} = [\phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}\ x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}\ ]^T \tag{12}$$

The control input is described by the vector [7]:

$$U = [\, U_1 \; U_2 \; U_3 \; U_4] \tag{13}$$

where:

$$U_1 = K_f\left(\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2\right) \tag{14}$$

$$U_2 = K_f(-\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 - \Omega_4{}^2 \tag{15}$$

$$U_3 = K_f\left(\Omega_1{}^2 - \Omega_2{}^2 + \Omega_3{}^2 - \Omega_4{}^2\right) \tag{16}$$

$$U_4 = K_M(\Omega_1{}^2 + \Omega_2{}^2 - \Omega_3{}^2 - \Omega_4{}^2) \tag{17}$$

U_1: It is responsible for the altitude and its rate of change $(z, \dot{z})$.

U_2: It is responsible for the roll rotation and its rate of change $(\phi, \dot{\phi})$.

U_3: It is responsible for the pitch rotation and its rate of change $(\theta, \dot{\theta})$.

U_4: It is responsible for the yaw rotation and its rate of change $(\psi, \dot{\psi})$.

Using (14) to (17) and replace them into the Rotational and Translational subsystems (5) and (10), respectively.

The final mathematical model, which defines the position of the quadrotor in space and its angular and linear velocities, will be as follows:

$$\begin{cases} \dot{x}_1 = \dot{\phi} = x_2 \\ \dot{x}_2 = \ddot{\phi} = x_4 x_6 a_1 + b_1 U_2 \\ \dot{x}_3 = \dot{\theta} = x_4 \\ \dot{x}_4 = \ddot{\theta} = x_2 x_6 a_2 + b_2 U_3 \\ \dot{x}_5 = \dot{\psi} = x_6 \\ \dot{x}_6 = \ddot{\psi} = x_2 x_4 a_3 + b_3 U_4 \\ \dot{x}_7 = \dot{x} = x_8 \\ \dot{x}_8 = \ddot{x} = \dfrac{U_1}{m}(\cos x_1 \sin x_3 \cos x_5 + \sin x_1 \sin x_5) \\ \dot{x}_9 = \dot{y} = x_{10} \\ \dot{x}_{10} = \ddot{y} = \dfrac{U_1}{m}(\cos x_1 \sin x_3 \sin x_5 - \sin x_1 \cos x_5) \\ \dot{x}_{11} = \dot{z} = x_{12} \\ \dot{x}_{12} = \ddot{z} = -g + \dfrac{U_1}{m}\cos(x_1)\cos(x_3) \end{cases} \tag{18}$$

Where:

$$a_1 = \frac{I_{yy} - I_{zz}}{I_{xx}} \;\; ; \;\; a_2 = \frac{I_{zz} - I_{xx}}{I_{yy}} \;\; ; \;\; a_3 = \frac{I_{xx} - I_{yy}}{I_{zz}}$$

$$b_1 = \frac{l}{I_{xx}} \qquad ; \qquad b_2 = \frac{l}{I_{yy}} \qquad ; \; b_3 = \frac{1}{I_{zz}}$$

Since the quadrotor is rotating around its hover point, the roll and pitch angles, represented by the values of ϕ and θ, are small enough to support the hypotheses that: $cos(\phi) = cos(\theta) = 1$ and $sin(\phi) = \phi$, and $sin(\theta) = \theta$. Thus:

$$\begin{cases} \ddot{x} = (\theta \cos\psi + \phi \sin\psi)\dfrac{U_1}{m} \\ \ddot{y} = (\theta \sin\psi - \phi \cos\psi)\dfrac{U_1}{m} \end{cases} \tag{19}$$

Hence, the angles ϕ_d and θ_d can be obtained from the following matrix form:

$$\begin{bmatrix} \theta_d \\ \phi_d \end{bmatrix} = \frac{m}{U_1}\begin{bmatrix} cos(\psi_d) & \sin(\psi_d) \\ sin(\psi_d) & -\cos(\psi_d) \end{bmatrix}\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \tag{20}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \frac{U_1}{m}\begin{bmatrix} U_{\mathrm{x}} \\ U_{\mathrm{y}} \end{bmatrix} \tag{21}$$

The vector in (21) represents a virtual input control from which we calculate the desired roll and pitch angles. In order to simplify the design approach, in this work the desired yaw angle ψ_d is considered to be equal to zero. Therefore, by replacing (21) in (20) and putting $\psi_d = 0$, the final equation will be as follows:

$$\begin{cases} \theta_d = U_{\mathrm{x}} \\ \phi_d = -U_{\mathrm{y}} \end{cases} \tag{22}$$

3. Methodology

Precise control methods are needed for the efficient and safe navigation of unmanned aerial vehicles (UAVs). Fuzzy logic control works with approximate or uncertain information, in contrast to standard control techniques that depend on exact mathematical models and crisp values. This makes it especially well-suited and effective for systems with ill-defined or unknown properties [8].

In this work, the Adaptive Fuzzy PID (AFPID) controller is presented as the main method, combining the principles of two baseline approaches: classical PID and Fuzzy PID (FPID) to enhance flexibility, robustness, and overall tracking performance.

3.1 Baseline Controllers (PID & Fuzzy PID)

As baseline controllers for UAV path tracking, we consider the classical PID controller and a Fuzzy-PID (FPID) regulator. These serve as fundamental methods and points of comparison before introducing the main approach, the Adaptive Fuzzy PID (AFPID) controller.

PID controllers are widely used in robotics applications, including path trajectory tracking. The key advantages of using PID controllers for these tasks are their ability to continuously minimize the error between the desired path/trajectory and the robot's actual position and orientation.

The PID controller can be presented as a mathematical equation given by:

$$u(t) = K_p e + K_i \int_0^t e \ dt + K_d \frac{d}{dt} e \tag{23}$$

K_P, K_D and K_I are the proportional, derivative and integral gains, respectively. However, in this work it required manual tuning of the gains by trial and error to achieve acceptable tracking performance. The Fuzzy PID (FPID) controller improves upon the classical PID by using a Mamdani fuzzy inference system (FIS) [9,10] to regulate the control action, which reduces the need for constant manual adjustment. figure 2 illustrates the key components of a fuzzy logic system [11].

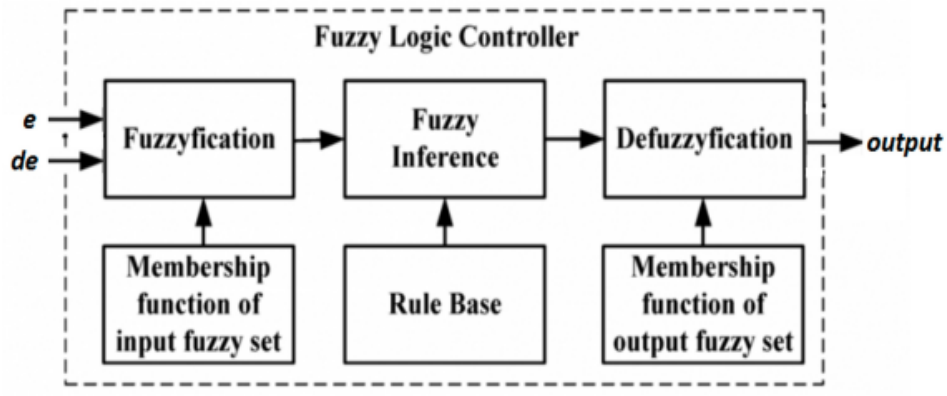


Figure 2. Block diagram of Fuzzy logic controller system

The FPID controller can be presented as a mathematical equation given by:

$$u(t) = K_p e + K_i \int_0^t e \ dt + K_d \frac{d}{dt} e + FIS(e, de) \tag{24}$$

Where FIS is the Mamdani fuzzy inference output. In MATLAB, this term is obtained using evalfis instruction. Nevertheless, the FPID still operates with fixed PID gains and its performance depends on the design of the fuzzy rule base and membership functions, so it cannot adjust parameters in real time.

3.2 Adaptive Fuzzy PID control (AFPID)

Adaptive control is a method used to allow the real-time system to self-correct its configuration to handle any external issue. As the name suggests, the Adaptive Fuzzy PID methodology combines both PID control methods with adaptive fuzzy logic [12].

The main objective is to tune the gains of the PID controller (proportional, derivative and integral) using fuzzy logic, allowing the system to adapt to any changes or issues that could occur. The fuzzy logic controller uses the error and rate of change of error to adaptably regulate the proportional K_p, integral K_I, and derivative K_D gain as illustrated in figure 3 [13].

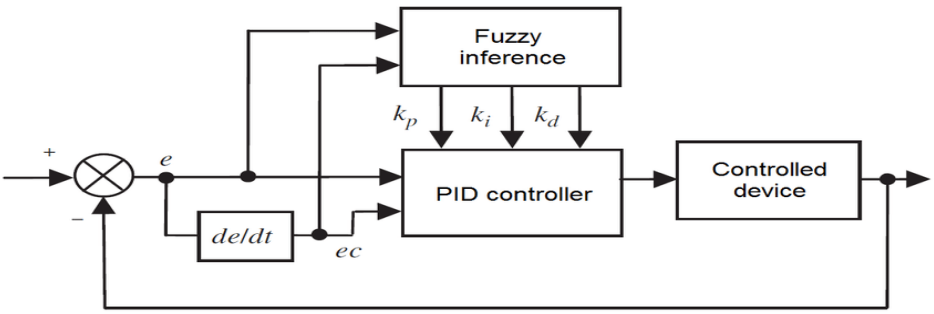


Figure 3. The design of Adaptive Fuzzy PID

For the control of UAV. figure 4 shows the design of the system controllers and the below sub-sections describe the work of each controller.

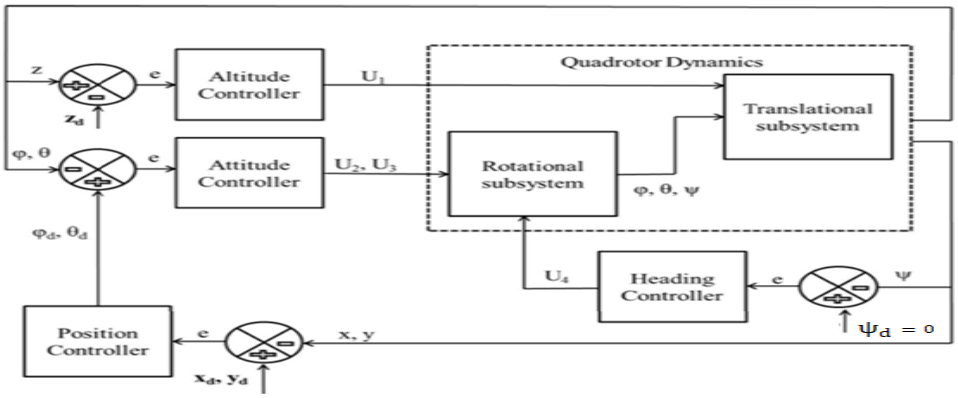


Figure 4. Block diagram of all system controllers.

- **Altitude Controller**

The feedback signal from the quadrotor dynamics subsystem will be compared with the desired z-position and this error signal e_z will be input to the altitude controller.

$$U_1(k) = K_{Pz_{FIC}} e_z(k) + K_{Dz_{FIC}}[e_z(k) - e_z(k-1)] + K_{Iz_{FIC}}\left[\sum_{i=0}^{k} e_z(i)\right] \quad (25)$$

$$+ evalfis\big(U_{z_{FIS}}, [e_z(k), e_z(k) - e_z(k-1)]\big)$$

Where $U_{z_{FIS}}$ is the fuzzy inference system that contains two inputs: error of the altitude "$\boldsymbol{z}$" and its derivative and provides the output value.

- **Position Controller**

The quadrotor subsystem provides feedback on x and y positions. When these positions are not in line with the desired positions, an error signal is sent to the position controller, which modifies the error to ensure that the quadrotor is moving in the correct direction. The outputs of x and y position controllers are U_{x} and U_{y}, respectively.

$$U_x(k) = K_{Px_{FIC}} e_x(k) + K_{Dx_{FIC}}[e_x(k) - e_x(k-1)] + K_{Ix_{FIC}}\left[\sum_{i=0}^{k} e_x(i)\right] + evalfis(U_{x_{FIS}}, [e_x(k), e_x(k) - e_x(k-1)]) \quad (26)$$

$U_{x_{FIS}}$ is the fuzzy inference system that contains two inputs: error of "x" and its derivative and provides the output value.

$$U_y(k) = K_{Py_{FIC}} e_y(k) + K_{Dy_{FIC}}[e_y(k) - e_y(k-1)] + K_{Iy_{FIC}}\left[\sum_{i=0}^{k} e_y(i)\right] + evalfis(U_{y_{FIS}}, [e_y(k), e_y(k) - e_y(k-1)]) \quad (27)$$

$U_{y_{FIS}}$ is the fuzzy inference system that contains two inputs: error of "y" and its derivative and provides the output value.

- **Attitude Controllers**

The attitude controller receives the roll and pitch angles as feedback from the dynamic subsystem, which compares to the desired roll and pitch, coming from the position controller. The roll and pitch controller are as follows:

$$U_2(k) = K_{P\phi_{FIC}} e_\phi(k) + K_{D\phi_{FIC}}[e_\phi(k) - e_\phi(k-1)] + K_{I\phi_{FIC}}\left[\sum_{i=0}^{k} e_\phi(i)\right] + evalfis(U_{2_{FIS}}, [e_\phi(k), e_\phi(k - e_\phi(k-1)]) \quad (28)$$

$U_{2_{FIS}}$ is the fuzzy inference system that contains two inputs: error of "roll" and its derivative and provides the output value.

$$U_3(k) = K_{P\theta_{FIC}} e_\theta(k) + K_{D\theta_{FIC}}[e_\theta(k) - e_\theta(k-1)] + K_{I\theta_{FIC}}\left[\sum_{i=0}^{k} e_\theta(i)\right] + evalfis(U_{3_{FIS}}, [e_\theta(k), e_\theta(k) - e_\theta(k-1)]) \quad (29)$$

$U_{3_{FIS}}$ is the fuzzy inference system that contains two inputs: error of "pitch" and its derivative and provides the output value.

- **Yaw controller**

The control input for the yaw angle is defined by this equation:

$$U_4(k) = K_{P\psi} e_\psi(k) + K_{D\psi}[e_\psi(k) - e_\psi(k-1)] + K_{I\psi}\left[\sum_{i=0}^{k} e_\psi(i)\right] + evalfis(U_{4_{FIS}}, [e_\psi(k), e_\psi(k) - e_\psi(k-1)]) \quad (30)$$

$U_{4_{FIS}}$ is the fuzzy inference system that contains two inputs: error of **"yaw"** and its derivative and provides the output value.

In the previous control inputs, $e(k)$ represents the error between the z-reference and z-position $e(k) = ref\ (k) - pos(k)$, U_{FIS} is the fuzzy inference system that contains two inputs: error and its derivative and provides the output value and $K_{P_{FIC}}, K_{D_{FIC}}, K_{I_{FIC}}$ are the adaptable outputs gains (proportional, derivative and integral) of the PID controller.

4. Simulation Setup

The simulation is done utilizing Matlab. The PID controller was designed based on the equations (25) to (29). The quadrotor is simulated using the model described in the second section and the parameters of the model are listed in table 1 [14].

Table 1. The parameters of quadrotor model

m	0.486(Kg)
l	0.25(m)
g	9.81(m/s²)
I_x	$3.82 \times 10^{-3}(\text{Kg} \times m^2)$
I_y	$3.82 \times 10^{-3}(\text{Kg} \times m^2)$
I_z	$7.65 \times 10^{-3}(\text{Kg} \times m^2)$

In the first time, the values of PID parameters are manually tuned through trial-and-error; by taking values and see if they satisfy the desired result with minimizing the error.

In fuzzy system inference, Three linguistic values: negative

(N), zero (Z), and positive (P) are used for two inputs (error and rate of error) and one output with five linguistic values: great negative (GN), negative (N), zero (Z), positive (P) and great positive (GP). So, after accurate observation, rule base is defined as shown in table 2 [15].

Table 2. UAV controllers rule table .

e \ de	N	Z	P
N	GN	N	Z
Z	N	Z	P
P	Z	P	GP

Mamdani-type Fuzzy inference system with triangular and trapezoidal membership functions is used in our simulation, where the input range for both error and error's rate is defined from [-5, 5]. Whereas, there exist four controllers outputs for each Fuzzy logic system $\boldsymbol{U_1}$, $\boldsymbol{U_2}$, $\boldsymbol{U_3}$ and $\boldsymbol{U_4}$ with two virtual controllers $\boldsymbol{U_x}$ and $\boldsymbol{U_y}$; each one of them has its own output range: [-12, 12] for $\boldsymbol{U_1}$, [-3.05, 3.05] for $\boldsymbol{U_2}$ and $\boldsymbol{U_3}$, [-0.06, 0.06] for $\boldsymbol{U_4}$, and [-1, 1] for $\boldsymbol{U_x}$ and $\boldsymbol{U_y}$.

Therefore, in order to see the performance of the AFPID controller compared to well-tuned PID and FPID, we will take example path of quadrotor in 3D environment, and compare them in terms of the settling time, the steady state error and the total Root Mean Square error (RMSe) defined by (30).

$$RMSe = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left((e_x(i))^2 + (e_y(i))^2 + (e_z(i))^2\right)} \quad (30)$$

5. Results and Discussion

In the result part, we got the outcomes of the Adaptive Fuzzy PID response concerning the Y position, X position, roll, pitch, yaw (considered as zero in this context), and altitude Z in figure 5. The control inputs results are shown in figure 6 and the final simulated 3D UAV path in figure 7.

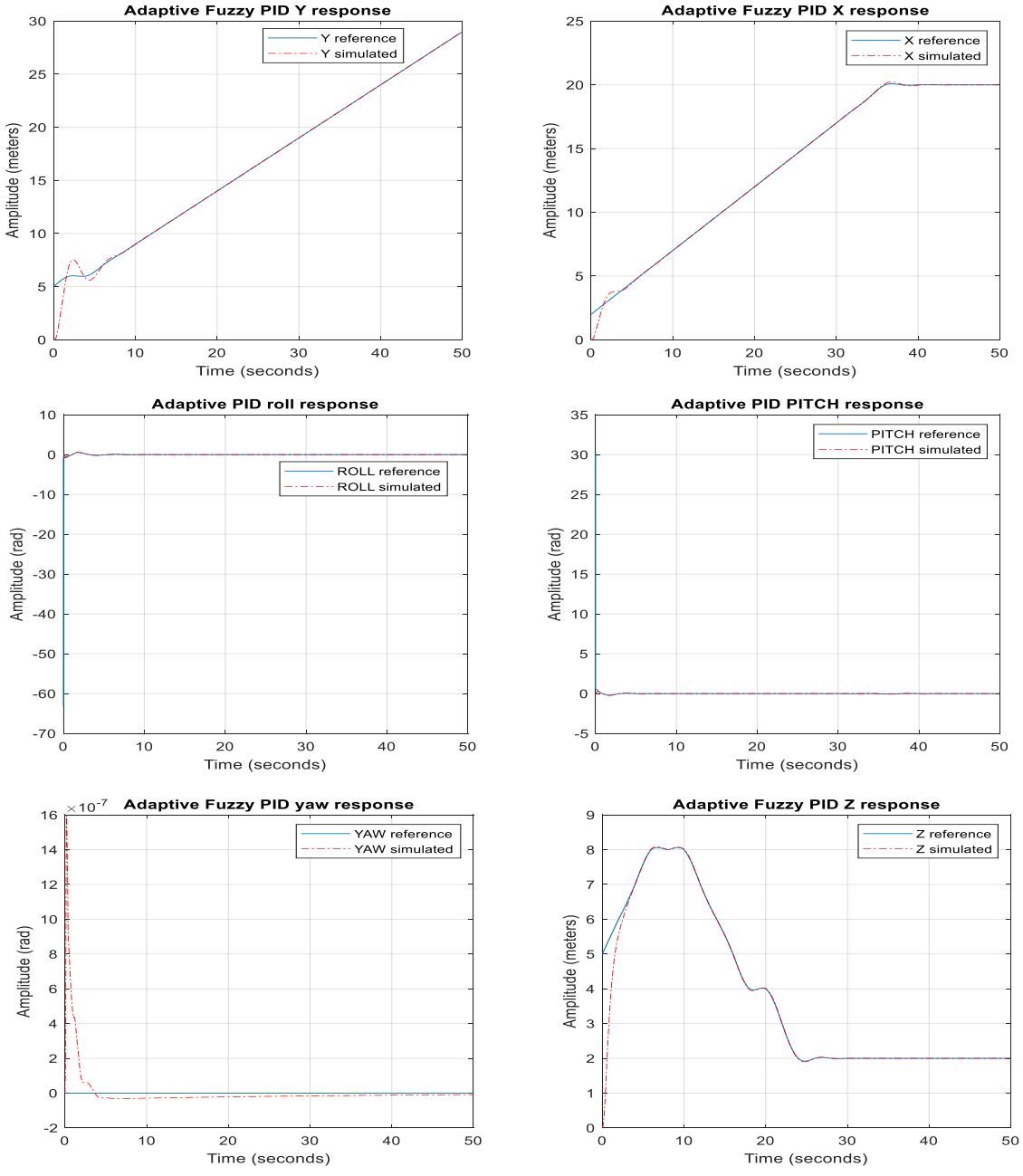


Figure 5. Adaptive Fuzzy PID Response for the First Path: Y-X Positions, Roll, Pitch, Yaw, and Altitude Z

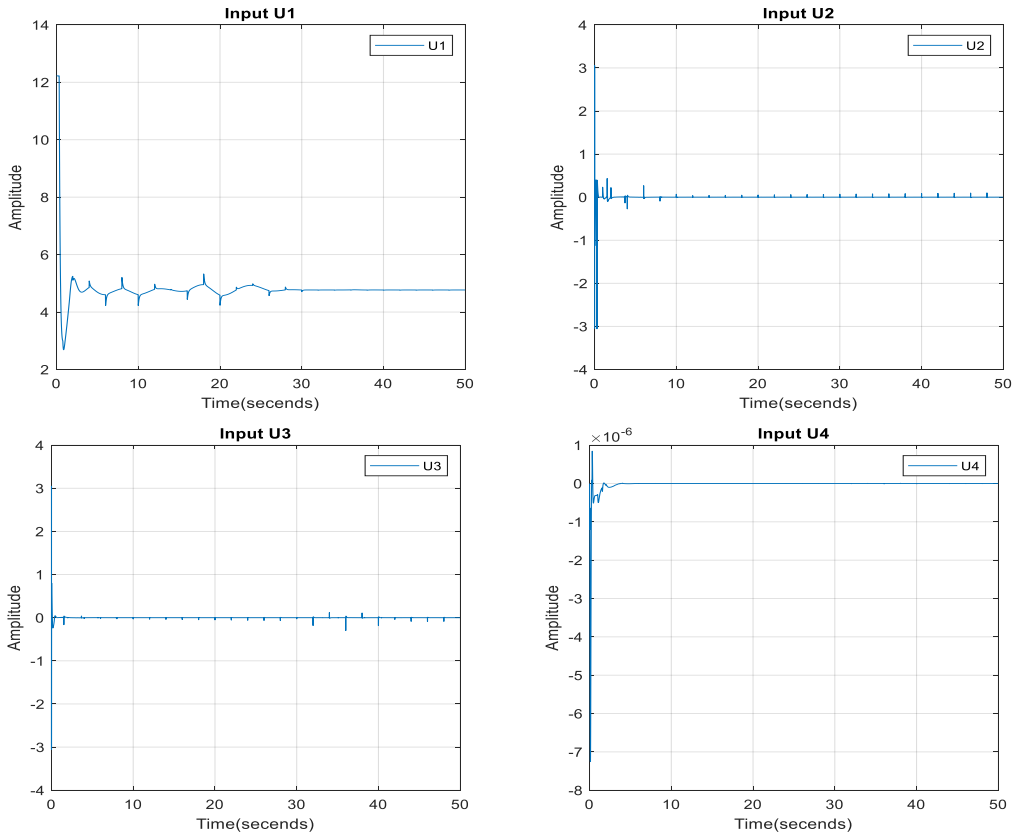


Figure 6. Control inputs Adaptive Fuzzy PID response.

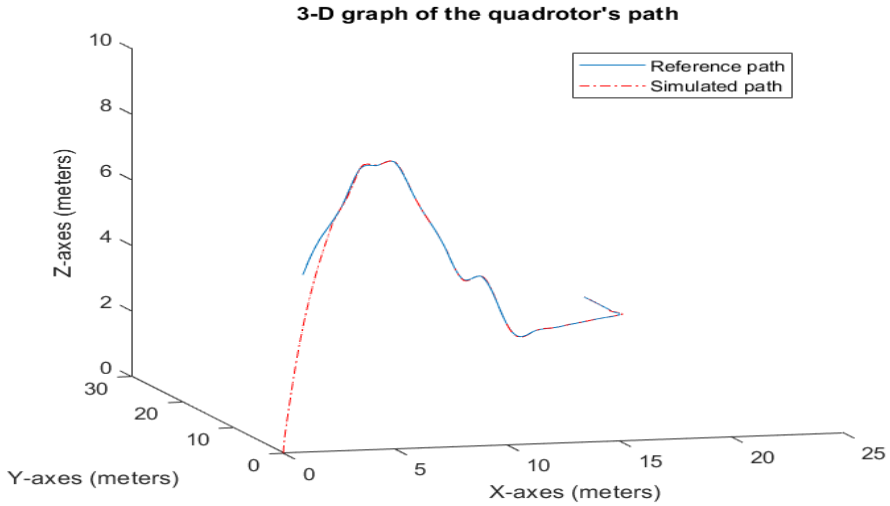


Figure 7. Adaptive Fuzzy PID Control Simulation of a 3D UAV's Trajectory.

From figure 5, we can notice that the X and Y response, both have a small overshoot from 0s to 5s and from 0s to 8s, respectively. For the altitude-Z, roll and pitch response, we can see that the controller follows directly the desired path with no overshoot.

Concerning the controller's outputs in figure 6, $\boldsymbol{U_1}$ has a fluctuation due to the sudden changes in the altitude Z. For $\boldsymbol{U_2}$,$\boldsymbol{U_3}$ and $\boldsymbol{U_4}$, we can notice that they were unstable a bit in the beginning but after that they maintain in the value of '0'.

Table 3 shows the performance of three controller methods: PID, FPID, and AFPID, across three positions X, Y, and Z. All controllers successfully drive the UAV to the desired setpoint without steady-state error. The comparison of settling times and RMSe shows that AFPID achieves slightly better performance than both FPID and PID.

Table 3. PID, FPID and AFPID Response for Trajectory Tracking: Settling Time and RMSe.

Controller Type	Settling Time (X-pos)	Settling Time (Y-pos)	Settling Time (Z-pos)	RMSe
PID	5s	6.2s	3.5s	1.0185
FPID	4.5s	5.75s	3.5s	0.9064
AFPID	3.5s	5.7s	3s	0.9063

All simulated plots demonstrate convergence to the reference one without any steady state error in all three control types.

The closeness of the results could be explained by the well-tuned parameters of the PID. The manual tuning method is an easy way for PID controllers to obtain a reasonable result. 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. Due to their little knowledge of the process plant, PID controllers are unable to adjust themselves automatically when the system faces a certain change. Hence, other approaches such us the Fuzzy logic technique, could be 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.

To obtain a better comparison of the control methods' adaptability in the presence of Gaussian disturbance over time from 10 seconds until 20 seconds. The effects of the disturbances are shown in figure 8. AFPID was the best, followed by FPID, with PID controller performing the least effectively.

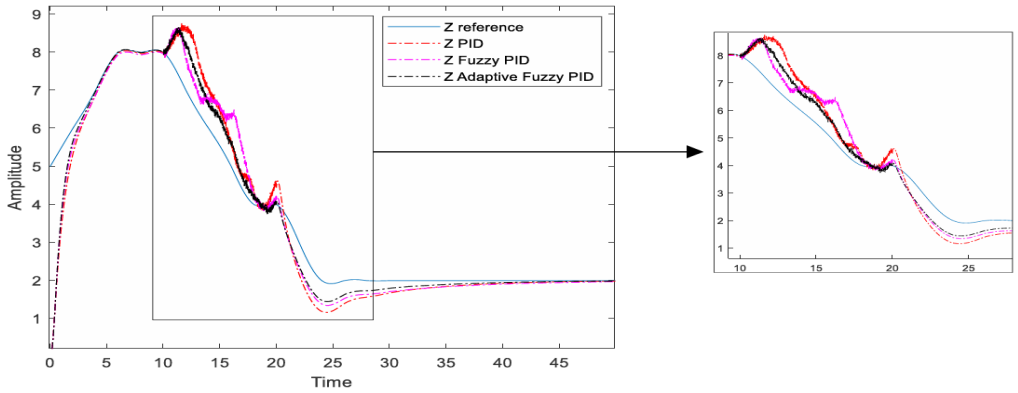


Figure 8. The response of PID, Fuzzy PID and Adaptive Fuzzy PID with Gaussian disturbance.

In overall, the Adaptive fuzzy PID, fuzzy PID and PID controllers proved that they are adapted to the quadrotor when flying. However, the adaptive fuzzy PID controller includes the adaptive mechanism that continuously adjusts the parameters of the controller in real-time. Instead of tuning the parameters manually, this is the main difference that makes AFPID better than FPID and PID.

6. Conclusion

In this paper, we investigated the Adaptive Fuzzy PID (AFPID) control strategy for quadrotor UAVs, with classical PID and fuzzy PID (FPID) used as baseline controllers. The results showed that all controllers achieved stable trajectory tracking without steady-state error and performed approximately the same due to well-tuned PID parameters. However, AFPID demonstrated superior performance, providing faster settling times and greater robustness against disturbances thanks to its ability to adjust parameters in real time. These improvements confirm the effectiveness of integrating adaptiveness with fuzzy reasoning in UAV control. For future work, we will focus on extending the AFPID approach to more complex flight scenarios and validating its performance in real-world experiments.

7. References

[1] T. Ahmad, A. E. Morel, N. Cheng, K. Palaniappan, P. Calyam, K. Sun, and J. Pan, Future UAV/Drone systems for intelligent active surveillance and monitoring, ACM Comput. Surv., Aug. 2025. doi: 10.1145/3760389.

[2] S. Simon, Drones for automated parcel delivery: Use case analysis, Transp. Res. Interdiscip. Perspect., vol. 40, 2024, Art. no. 100931. doi: 10.1016/j.trip.2024.100931.

[3] Nandini and S. R. Singh, Application of drones technology in agriculture: A modern approach, J. Sci. Res. Rep., vol. 30, no. 7, pp. 142–152, 2024. doi: 10.9734/jsrr/2024/v30i72131.

[4] A. Noordin, M. A. A. Basri, and Z. Mohamed, Adaptive PID control via sliding mode for position tracking of quadrotor MAV: Simulation and real-time experiment evaluation, Aerospace, vol. 10, no. 6, Art. no. 512, May 2023. doi: 10.3390/aerospace10060512.

[5] A. G. Melo et al., Fuzzy gain-scheduling PID for UAV position and altitude controllers, Sensors, vol. 22, no. 6, Art. no. 2173, Mar. 2022. doi: 10.3390/s22062173.

[6] Y. Ye, Homeostasis of a quadrotor UAV based on fuzzy adaptive PID controller, in Proc. 2024 Int. Conf. Intell. Auto. and Adv. AI (ICIAAI), Atlantis Press, 2024. doi: 10.2991/978-94-6463-540-9_53.

[7] F. Fahmizal, Altitude control of quadrotor using fuzzy self-tuning PID controller, in Proc. IEEE Int. Conf. Control, Robot. Integr. Syst. (ICCRIS), 2017. doi: 10.1109/ICCRIS.2017.8068415.

[8] H. Hagras, General Type-2 Fuzzy Logic Systems to Enable Better Uncertainty Handling for Real World Application, The Computational Intelligence Center. Lecture presented at the The computational intelligence center.

[9] E. H. Mamdani, S. Assilian. "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller". International Journal of Man-Machine Studies 7,January 1975

[10] Mathwork, Mamdani and Sugeno Fuzzy Inference System, https://www.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html.

[11] M. Makhtoumi, Active Vibration Control of Launch Vehicle on Satellite Using Piezoelectric Stack Actuator, arXiv [Physics.Space-Ph], 2023. arXiv. http://arxiv.org/abs/1903.07396.

[12] M. A. Sattar, Adaptive Fuzzy Control of Quadrotor, M.S. thesis, Rochester Institute of Technology, Dubai, 2017. [Online]. Available: https://repository.rit.edu/theses/9618/

[13] M. A. Chancán León, S. Leal Braga, and J. C. Cuisano Egúsquiza, Combustion air temperature and humidity control for engine testing, in Proc. VII Congresso Nacional de Engenharia Mecânica (CONEM), São Luís, Maranhão, Brazil, Jul.–Aug. 2012, pp. 1–7.

[14] H. Boudjedir, Dual Neural Network for Adaptive Sliding Mode Control of Quadrotor Helicopter Stabilization, International Journal of Information Sciences and Techniques 2, no. 4 (July 31, 2012). https://doi.org/10.5121/ijist.2012.2401.

[15] 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 (ICCCEEE), Khartoum, Sudan, 2021.