Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 13, No. 4, 2025, 407-425.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Formation Control of Quadcopters with Variable Formation using Fuzzy-PID Controller

Seyedeh Mahsa Zakipour Bahambari and Saeed Khankalantary*

Department of Electrical Engineering, K.N. Toosi University of Technology, Tehran, Iran.

Article Info

Article History:
Received 05 March 2025
Revised 13 May 2025
Accepted 07 July 2025

DOI:10.22044/jadm.2025.15865.2702

Keywords:
Quadcopter, PID, Fuzzy-PID, Trajectory Tracking, Variable Formation Control.

**Corresponding auhor: s.kalantary@kntu.ac.ir (S.Kalantary).*

Abstract

This paper focuses on the design of advanced controllers and the implementation of magnetic and velocity tracking at the position and formation control levels for a group of quadcopters. Initially, PID controllers are developed based on the quadcopter structure, and then a constrained fuzzy-PID controller is introduced to steer the system to the desired position. The performance of this controller is compared with classical PID and fuzzy-PID controllers. This study examines the arrangement and formation coordination of six quadcopters under three different scenarios, evaluating their formation control and coordination. Each quadcopter has an internal controller responsible for maintaining formation accuracy and system stability. Due to the complexity of quadcopter dynamics, trajectory tracking is challenging research. fuzzy-PID controller is proposed to stabilize the quadcopter along predefined trajectories, utilizing speed information as input. Simulation results in the MATLAB/Simulink environment demonstrate that the fuzzy-PID controller outperforms the classical PID controller. exhibiting greater resistance to external disturbances across all axes, higher accuracy in reducing tracking errors, and improved stability. This superiority is particularly evident in multi-agent systems, emphasizing the significance of advanced control techniques in enhancing the regulation of both single and multi-agent quadcopters. Ultimately, this improves tracking performance while ensuring dynamic efficiency in uncertain environments.

1. Introduction

A quadcopter is an aircraft that uses four rotors to lift and propel itself, arranged in a cross configuration. Its primary maneuvers rely on manipulating the speeds of these rotors. Although it has only six degrees of freedom (DOF), it has four actuators, making it underactuated and prone to unstable dynamics. One key challenges of small aerial vehicles is hardware redundancy, which adds to this constraint and necessitates careful consideration in control system design, crucial achieving acceptable and efficient performance. Small unmanned aerial vehicles (UAVs) are increasingly used for various civilian and military applications[1]. A quadcopter is of vertical takeoff and landing (VTOL)aircraft, consisting of four motors, providing great stability and maneuverability. Advantages include reduced energy consumption and ease of control. Furthermore, quadcopters can be remotely piloted, making them suitable for hazardous operations. Multirotor quadcopters are more stable than fixed-wing aircraft in applications involving mapping, aerial monitoring, and disaster response [2]. Although a quadcopter is an unmanned aerial vehicle capable of flying without a pilot, a good control system essential for fulfilling this task. Such a system primarily focused on rotor speed control, keeps the aircraft flying smoothly and While linear control techniques often provide dequate stabilization for basic stability, quadcopters well-known complex dynamic models and are exceptionally susceptible to external disturbances such as wind or unpredictable environmental factors. Consequently, a nonlinear control system is required, which can be further enhanced through algorithms designed to help the controller adapt to challenging situations. Due to their higher adaptability and decision-making efficiency in changing environments, intelligent control methods are increasingly successful in augmenting traditional control approaches [3]. The control problem of quadcopters has been extensively studied in the literature using various control approaches. These include classical techniques such as Proportional-Integral-Derivative (PID), optimal methods like Linear Quadratic Regulator (LQR), and robust control strategies such as H-infinity control for linear systems. For nonlinear systems, common methods include backstepping, feedback linearization, and model predictive control. Comparative analyses between PID and LQR controllers on micro quadcopters have shown that while both approaches can stabilize the system in hover mode, PID control exhibits lower performance under varying operating conditions. This highlights the advantage of LQR in achieving more efficient and robust control across diverse operating conditions [4, 5]. The design of an effective position and attitude controller is crucial for managing such complex systems, leading to extensive research on both linear and nonlinear control techniques. Among these, the PID controller remains one of the most widely used methods [6-8]. Acknowledging the limitations of conventional PID controllers, researchers have explored various methods for automatic tuning to enhance performance and adaptability [9]. Among various methods for tuning PID parameters, the integration of fuzzy neural networks has emerged as a noteworthy solution. For instance, Davanipour et al. (2018) introduced a self-adjusting PID controller based on a fuzzy wavelet neural network, which exhibited enhanced adaptability and performance in dynamic environments[10]. Similarly, Tripathy et al. (2019) developed a fuzzy PID controller for load frequency regulation, utilizing Spider Monkey Optimization (SMO) to optimize parameters, leading to improved stability and responsiveness under fluctuating load conditions [11]. Furthermore, Wang et al. (2017) proposed an advanced fuzzy PID controller integrated with a predictive functional control framework, achieving greater precision and flexibility in dynamic systems [12]. While these studies have significantly contributed to simplifying the PID tuning process, many of the suggested techniques remain complex for practical applications, and further advancements in control performance are still necessary. Despite the numerous benefits of drones, stabilization remains a critical challenge, particularly in achieving precise regulation and robustness. These concerns have driven extensive research into optimizing quadcopter controller performance. Although PID controllers are commonly employed for this purpose, they exhibit limitations such as weak adaptability, high sensitivity to noise, insufficient robustness, inability to handle multiple objectives simultaneously, and substantial computational demands. Consequently, PID controllers alone may not always ensure accurate quadcopter stabilization [13-15]. In some scenarios, they may even lead to instability, necessitating the integration of intelligent control algorithms like fuzzy logic controllers to address these shortcomings. By combining the inference capability of fuzzy logic with the learning adaptability of PID control, this hybrid framework enhances system robustness against parameter variations and disturbances while improving stability and reliability across different operational conditions. Another advantage of fuzzy PID controllers is their ease of adjustment, as the controlled model can dynamically evolve. Comprehensive surveys on formation control for multi-agent systems, such as [16], have examined consensus-based formation strategies. However, these studies have not extensively covered notable research on inter-agent distance-based formation control. Reports on behavior-based formation control strategies can be found in [17, 18], while virtual structure-based approaches have been explored in [19, 20], Additionally, formation control techniques employing virtual structures have been discussed in [21, 22]. Each of these methodologies has distinct advantages and drawbacks. For instance, leader-follower strategies are relatively simple to implement but lack direct formation feedback and are less resilient due to their dependence on the leader's stability. Conversely, behavior-based approaches effectively prevent collisions, yet their mathematical formulation remains complex [23]. UAV swarm formation controllers based on a consensus-driven strategy were designed in [24], demonstrating that the consensus method maintained structural integrity even in the presence of communication disruptions. Seo et al. combined consensus protocols with output feedback linearization to address formation control challenges, thereby enabling partially time-varying formations. Turpin et al. [25] also studied the leader-follower strategy of deploying and maintaining Unmanned Surface Vehicles (USVs) networks [26]. This study addressed uncertainties and stochastic disturbances affecting network resilience, proposing a robust control strategy to stabilize the formation under these conditions. In contrast, [27] introduced a distributed optimal cooperative control (DOCC) approach for multi-agent systems (MASs), which considers constraints such as input saturation and collision avoidance. Furthermore, the authors in [28] presented an approach for localizing a radio frequency (RF) source by leveraging environmental obstacles. This method enhances localization accuracy by accounting for signal reflections from surfaces, especially in cases where direct line-of-sight is obstructed. Experimental results demonstrated that 3D localization accuracy significantly improved when both direct and reflected signals were considered, particularly in large environments.in [29], the problem of optimal adaptive leader-follower consensus for linear multi-agent systems is investigated, considering both known and unknown agent dynamics. A distributed control framework is proposed to ensure that all follower agents asymptotically track the leader's trajectory while minimizing a predefined performance cost. For agents with unknown dynamics, adaptive control laws are developed to estimate uncertain parameters in real time, ensuring stability and consensus convergence. The approach combines optimal control techniques with adaptive mechanisms, providing robustness and improved performance in the presence of system uncertainties.

The simulation results presented in this paper illustrate the efficacy of the proposed method across diverse conditions and disturbances. The key contributions of this work can be outlined as follows:

- Design of a fuzzy control system incorporating disturbance rejection for quadcopter control.
- Development of a comprehensive quadcopter model.
- Performance comparison between fuzzy controllers and PID controllers using Simulink, along with analysis and comparison of their responses in the presence of disturbance.
- Implementation of two different scenarios of single-integrator formation control on a nonlinear model with a Fuzzy-PID controller.

This study begins by introducing the nonlinear models of the quadcopter, Subsequently, a single-integrator formation control strategy is formulated for a fleet of six quadcopters, with clearly defined control inputs aimed at tracking a Helix and Variation Formation. Finally, simulation results and outputs for each phase are presented in the concluding section, enabling a thorough comparative analysis.

The rest of this paper is organized as follows: Section 2 deals with the mathematical modeng of the quadcopter. Section 3 discusses Controller Design. Section 4 addresses multi-agent quadcopter formation control. Simulations of the controllers and the formation strategies are presented in Section 5. After that, we deliver our final thoughts and conclusions in the last section.

2. Quadrotor Dynamics and Kinematics

In a standard quadcopter design, four rotors are arranged in a cross configuration. Throttle adjustments involve simultaneously increasing or decreasing the speeds of all four rotors equally, generating a vertical force (denoted as (u_1)in the body-fixed frame), which allows the quadcopter to ascend or descend. To control roll movement, the speed of the left rotor is increased while the right rotor’s speed is decreased, as shown in Figure 1. Similarly, this method is applied to the other two rotors to control pitch movement. For yaw control, the quadcopter modifies the speed of its front and rear rotors by rotating them counterclockwise, while the other two rotors rotate clockwise. This adjustment enables the quadcopter to perform a yaw maneuver by increasing the speed of the counterclockwise rotors and decreasing the speed of the clockwise rotors.

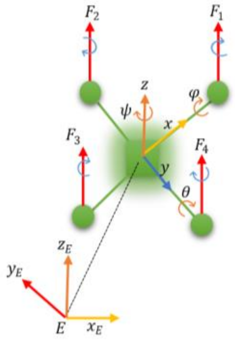


Figure 1. Quadcopter configuration.

To describe the motion of a quadcopter, two reference frames are defined, as illustrated in Figure 1. The quadcopter's orientation is determined by three Euler angles: roll (φ), pitch (θ), and yaw (ψ), which together form the vector $\Omega^T = (\varphi,\theta,\psi)$. These angles specify the quadcopter’s attitude in three-dimensional space. Additionally, its position in the inertial frame is represented by the vector. $r^T = (x,y,z)$. A rotation matrix R is utilized to transform vectors between the body-fixed frame and the inertial frame. This mathematical framework serves as the basis for analyzing and regulating the quadcopter's movement.

2.1 Kinematics

The quadcopter's absolute position in the world frame is represented by the $x, y,$ and z axes, encapsulated in a column vector ξ. Its attitude, or angular orientation, in the world frame is described by Θ, Which includes the roll angle ϕ The pitch angle θ, . and the yaw angle ψ. Collectively, these angles are known as Euler angles.

$$\xi = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \Theta = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix} \tag{1}$$

The quadcopter's body frame is centered at its center of mass. In this frame, the linear velocities along the three principal axes are denoted as v_B,While the associated angular velocities are represented by v. More specifically, the rotational velocities around the x, y, and z axes are labeled as p, q, and r, respectively.

$$V_B = \begin{bmatrix} V_{B,x} \\ V_{B,y} \\ V_{B,z} \end{bmatrix} \qquad v = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2}$$

To monitor the state of the quadcopter, the positions and angular velocities measured in the body frame must be transformed into the world frame. For positions, this transformation between the two frames is accomplished using rotation matrices corresponding to each axis. These rotation matrices are as follows:

$$R_{z,\psi} = \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$R_{y,\theta} = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} \tag{3}$$
$$R_{x,\varphi} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & cos\varphi & -sin\varphi \\ 0 & sin\varphi & cos\varphi \end{bmatrix}$$

The overall rotation matrix from the body frame to the world frame is derived from this as follows:

$$R = \begin{bmatrix} c\psi.c\theta & c\psi.s\theta.s\varphi - s\psi.c\varphi & c\psi.s\theta.c\varphi + s\psi.s\varphi \\ s\psi.c\theta & s\psi.s\theta.s\varphi + c\psi.c\varphi & s\psi.s\theta.c\varphi - c\psi.s\varphi \\ -s\theta & c\theta.s\varphi & c\theta.c\varphi \end{bmatrix} \tag{4}$$

The inverse of the rotation matrix R, Which transforms coordinates from the world frame to the body frame, is equivalent to its transpose ($R^{-1} = R^T$) due to the orthogonality property of R. The angular velocities of the quadcopter, measured in the body frame, differ from their representation in the world frame. To express these.angular velocities in the world frame, they must be transformed using the rotation matrix η.

$$\dot{\Theta} = \eta v$$
$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin\varphi tan\theta & cos\varphi tan\theta \\ 0 & cos\varphi & -sin\varphi \\ 0 & sin\varphi / cos\theta & cos\varphi / cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5}$$

During operation, the quadcopter's motors generate an upward force, F, along the body frame's z-axis. This force is directly related to the square of each motor's angular velocity. Furthermore, as the motors rotate, they collectively produce torque, τ, around the three principal axes of the body frame. This torque is also proportional to the angular velocities of all four motors.

$$F = \sum_{I=1}^{4} f_i = C_T \sum_{i=1}^{4} \omega_i^2$$
$$F_B = \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} \tag{6}$$

The force generated by the i^{th} motor is denoted as f_i, where f_i is calculated using the thrust coefficient C_T and the motor's angular velocity. The total forces in the body frame are represented as a vector F_B.

The torques generated by the four motors include $\tau_\varphi, \tau_\theta$, and τ_ψ, which act along the x-axis, y-axis, and z-axis of the body frame, respectively. These torques contribute to the roll, pitch, and yaw motions of the quadcopter.

$$\tau = \begin{bmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} b\left(\omega_4^2 - \omega_2^2\right) \\ bl\left(\omega_3^2 - \omega_1^2\right) \\ d\left(-\omega_1^2 - \omega_3^2 + \omega_2^2 + \omega_4^2\right) \end{bmatrix} \tag{7}$$
$$T = \omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2$$

Where L being the distance between the rotor and the center of mass of the quadcopter, b represents the constant thrust coefficient , and d They can be explained in equation (7).

2.2 Dynamic model

The dynamical model of a quadrotor is represented by six equations. Three equations govern the quadrotor's translational motion, describing its $x, y,$

and z positions. The other three equations describe its rotational dynamics, capturing the roll, pitch, and yaw motions. These equations take the angular velocities of the four motors, ω, as inputs and determine the next state of the quadcopter. The dynamics of the quadcopter are formulated using Newton-Euler equations. Newton's equations represent the translational motion and are expressed in the inertial (world) frame, incorporating the gravitational force vector G.

$$m\ddot{\xi} = G + RF_B$$

$$m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R\begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} \tag{8}$$

It is possible to simplify (8) and isolate the vector $\ddot{\xi}$:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{F}{m}\begin{bmatrix} cos\psi sin\theta cos\varphi + sin\psi sin\varphi \\ sin\psi sin\theta cos\varphi - cos\psi sin\varphi \\ cos\theta cos\varphi \end{bmatrix} \tag{9}$$

The rotational dynamics of the quadcopter are described by Euler's equations of motion, as outlined in equation (10). This equation is most effectively formulated in the body frame because the control inputs are directly applied in this frame. Furthermore, measurements from onboard sensors, such as gyroscopes and accelerometers, are inherently aligned with the body frame, making it a practical choice for analyzing and controlling rotational motion. By working in the body frame, the system's rotational behavior can be more intuitively modeled and controller, leveraging the natural alignment of sensor data and control inputs.

$$\tau = j_r\dot{v} + v \times j_r v$$

$$\begin{bmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}\begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

$$\begin{bmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p} \\ I_{xx}\dot{q} \\ I_{xx}\dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx}p \\ I_{xx}q \\ I_{xx}r \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} I_{zz}qr - I_{yy}qr \\ -I_{zz}pr + I_{xx}pr \\ I_{yy}pq - I_{xx}pq \end{bmatrix}$$

In the rotational dynamics of the quadcopter, j_r represents the inertia matrix, which contains the mass moments of inertia for each axis: I_{xx}, I_{yy} ,and I_{zz}. By simplifying equation (10), it is possible to isolate the time derivative of the angular velocity vector, $\dot{v}=\left[\dot{p},\, \dot{q}, \dot{r}\right]^T$, leading to a more concise representation of the rotational motion in terms of its dynamics.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \tau_\varphi / I_{xx} \\ \tau_\varphi / I_{yy} \\ \tau_\varphi / I_{zz} \end{bmatrix} + \begin{bmatrix} \left(I_{yy} - I_{zz}\right)qr / I_{xx} \\ \left(I_{zz} - I_{xx}\right)pr / I_{yy} \\ \left(I_{xx} - I_{yy}\right)pq / I_{zz} \end{bmatrix} \tag{11}$$

Equation (11) can be converted to the inertial frame after being integrated with Equation (5). These Newton-Euler equations are considered straightforward since they exclude external forces and torques, and they can be expressed in the following equation form.

$$\begin{gathered} \dot{p} = \left(I_{yy} - I_{zz}\right)\frac{qr}{I_{xx}} + \frac{\tau_\varphi}{I_{xx}} - \frac{j_r\Omega_r}{I_{xx}}q \\ \dot{q} = \left(I_{zz} - I_{xx}\right)\frac{pr}{I_{yy}} + \frac{\tau_\theta}{I_{yy}} - \frac{j_r\Omega_r}{I_{yy}}p \\ \dot{r} = \left(I_{xx} - I_{yy}\right)\frac{pq}{I_{zz}} + \frac{\tau_\psi}{I_{zz}} \\ \dot{\varphi} = p + r.cos\theta tan\theta + q.sin\theta tan\theta \\ \dot{\theta} = q.cos\varphi - r.sin\varphi \\ \dot{\psi} = r.\frac{cos\varphi}{tan\theta} + q.\frac{sin\varphi}{cos\theta} \\ \ddot{x} = \left(\cos\varphi \cdot \sin\theta \cdot \cos\psi + sin\varphi.sin\psi\right).\frac{F}{m} \\ \ddot{y} = \left(\cos\varphi \cdot \sin\theta \cdot \cos\psi - sin\varphi.cos\psi\right).\frac{F}{m} \\ \ddot{z} = -g + \left(\cos\varphi \cdot \cos\theta\right).\frac{F}{m} \end{gathered} \tag{12}$$

That the torques that are created by the four motors consist of $\tau_\varphi, \tau_\theta$ and τ_ψ and T in the direction of the body $x, y\, and\, z$ axes.

3. Control Algorithm

In this section, the attitude control of the quadcopter is first analyzed, followed by the simultaneous control of both position and attitude. a PID algorithm is employed, for attitude control, while a fuzzy logic controller regulates the z-position. Feedback linearization is adopted to achieve precise attitude control. The proposed control strategy accounts for variations in the quadcopter's mass and moments of inertia, ensuring robust performance under changing conditions. The controllers are designed to enable the quadcopter to follow a predefined trajectory accurately. The structure of the proposed cascaded control system is illustrated in Figure 2.

Figure 2 provides a general overview of the quadcopter's control. The reference trajectory is first input into the system, and then passes through the position and attitude controllers. The inner control loop includes the x and y positions; due to the nature of quadcopter control, the output of this section computes the desired roll and pitch angles. These angles are then fed into their respective attitude controllers. The z-position controller operates separately, based on the specific dynamics and control laws of the quadcopter.

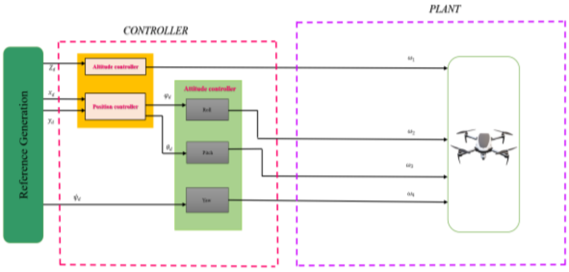


Figure 2. Control architecture.

Following these stages, the outputs of the *z*-position controller and the desired roll, pitch, and yaw angles are used, according to the quadcopter control laws, to compute the total thrust (*T*) and the moments in roll, pitch, and yaw. These values are then transformed into angular velocities (ω) in the subsequent block. It is important to note that these transformation relationships are given explicitly in Equation (15), which is part of the standard quadcopter control laws. Finally, the computed ω values are fed into the quadcopter model block, which outputs six state variables: three positions and three angles. These outputs are then fed back into the system in a closed-loop configuration to complete the control process. A clearer and more detailed representation of the control flow is provided in Figure 3 of the manuscript.

3.1 PID Controller

A robust and effective control strategy for the quadcopter can be achieved by employing a PID controller optimized using the Ziegler-Nichols methodology. The PID controller's simplicity and practicality make it a preferred choice for quadcopter control applications. Due to the complexity of the quadcopter dynamics and the interdependence of different control loops, we followed a step-by-step tuning approach: we first tuned the outer-loop controllers, which include the *x* and *y* position controllers, and then proceeded to tune the inner-loop controllers, which include the *z* position and attitude controllers.

This stepwise tuning process ensured that the interactions between different control loops were properly handled and the system performance was optimized.

In this strategy, the PID controller is utilized to maintain the quadrotor's orientation. Assuming stability in the rotational subsystem, with roll and pitch angles set to zero ($\varphi = 0 \, and \, \theta = 0$),the PID controller's individual components are represented by an equation. The altitude error signal, calculated as the difference between the desired altitude (zd) and the actual altitude (z), serves as input to the attitude controller. This attitude controller generates a control signal (u_1) to adjust the quadcopter's orientation and maintain its desired altitude.

$$u_i(t) = k_p e_i(t) + k_i \int_0^t e_i(t)dt + k_d \frac{de_i(t)}{dt} \qquad (13)$$

$$i = 1,2,3,4,5,6$$

The controller $u_1(t)$ is linked to the position on the Z-axis, while the controllers $u_2(t) to \, u_4(t))$ correspond to the angles $\varphi, \theta, and \, \psi$ (Roll, Pitch, Yaw) respectively. This setup allows for controlling three positional axes and the yaw direction, with roll and pitch direction controllers incorporated within. Effectively, the signal from the position controller determines the thrust vector in the inertial frame. The orientation of this vector establishes the reference points for the roll and pitch controllers, as illustrated in Figure 2. Moreover, the PID controller section internally computes the PID values for each component and feeds these computations into the power control system to derive T, τ_φ, τ_θ, and $\tau_{\psi.}$ These values are subsequently forwarded to the angular speed control interface to calculate the final values of w_1, w_2, w_3 and w_4, as depicted in Figure 3, representing the stages outlined earlier.

In the Quadcopter system, there are six positional states (x, y, z) and three angular orientations (φ, θ, ψ). However, only four control inputs are available, represented by the angular velocities of the four rotors, denoted as ω_i. The relationship between these states and the total thrust (T) and torque (τ) generated by the rotors is described in the Quadcopter's dynamics, as shown in Equation (12). The total thrust (T_1) primarily affects the acceleration along the Z-axis, crucial for maintaining the Quadcopter airborne. Meanwhile, torque $\tau_{\varphi 1}$ influences the angular acceleration around the φ angle, torque $\tau_{\theta 1}$ impacts the angular acceleration around the θ angle, and torque $\tau_{\psi 1}$ contributes to the angular acceleration around the ψ angle.

$$\begin{aligned} T_1 &= \left(g + u_1(t)\right) \times \frac{m}{cos\varphi . cos\theta} \\ \tau_{\varphi 1} &= \left(u_2(t)\right) \times I_{xx} \\ \tau_{\theta 1} &= \left(u_3(t)\right) \times I_{yy} \\ \tau_{\psi 1} &= \left(u_4(t)\right) \times I_{zz} \end{aligned} \qquad (14)$$

In this context, the gravity g, mass m, and moments of inertia I of the quadcopter are also taken into account. The correct angular velocities of the rotors, ω_i, can be determined using Equation (7), with the relevant values obtained from Equation (14).

$$\begin{aligned} \omega_1^2 &= \frac{T_1}{4K} - \frac{\tau_{\theta 1}}{2Kl} - \frac{\tau_{\psi 1}}{4b} \\ \omega_2^2 &= \frac{T_1}{4K} - \frac{\tau_{\varphi 1}}{2Kl} + \frac{\tau_{\psi 1}}{4b} \\ \omega_3^2 &= \frac{T_1}{4K} + \frac{\tau_{\theta 1}}{2Kl} - \frac{\tau_{\psi 1}}{4b} \\ \omega_4^2 &= \frac{T_1}{4K} - \frac{\tau_{\varphi 1}}{2Kl} + \frac{\tau_{\psi 1}}{4b} \end{aligned} \qquad (15)$$

To manage the two positional degrees of freedom (x, y) in quadcopter control, a hierarchical cascade structure with dual control loops is implemented. The inner control loop employs linear PID regulators to directly manage lateral adjustments along the x and y axes, leveraging their fast response characteristics to maintain dynamic stability. Simultaneously, the outer control loop uses additional PID controllers to calculate target roll (φ_d) and pitch (θ_d) angles, which act as reference inputs for the inner loop. These orientation commands enable indirect position control by translating horizontal thrust into tilt-induced movement. The outer loop thus bridges high-level positional goals with low-level attitude adjustments, while the inner loop executes rapid corrections. This layered architecture decouples positional and angular control objectives, allowing simultaneous precision in both trajectory tracking (via the outer loop) and stability during maneuvers (via the inner loop), ensuring accurate and robust aerial positioning.

$$\begin{aligned} \varphi_d &= sat(u_5) \\ \theta_d &= sat(u_6) \end{aligned} \qquad (16)$$

In our work on quadcopter control, specifically the position control in both the x and y axes the controllers are utilized to control position errors e_5 and e_6, defined as desired x and y positions (x desired and y desired) actual positions (x and y). Saturation blocks will bound the output signal (±0.05) as the outputs of the controllers to limit the control effort and to keep the system stable. These blocks are for controlling different parameters for the quadcopter, notably the roll (φ_d) and pitch (θ_d) angles. Block dialogs and system settings show that the simulation can be configured with adjustable parameters.

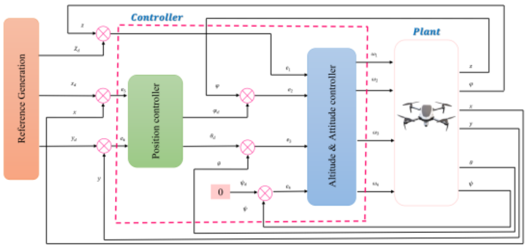


Figure 3. Full architecture of the quadcopter using Simulink.

3.2. Fuzzy Controller

Fuzzy logic has emerged as a powerful tool for incorporating human knowledge and managing uncertainties in control system processes [30]. A fuzzy self-tuning PID controller integrates fuzzy logic principles into a traditional PID framework [31, 32]. Fuzzy logic systems, a subset of artificial intelligence, are designed to manage uncertainty and imprecision by leveraging fuzzy set theory in decision-making. Unlike traditional binary logic, which operates strictly on true or false values, fuzzy logic incorporates intermediate truth levels, represented by membership values ranging from -1 to 1. This adaptability makes it particularly useful for handling complex and ambiguous real-world problems where precise boundaries are difficult to define. In the context of self-tuning a PID controller, a fuzzy relationship is established between the three PID gains—proportional, integral, and derivative—and the error terms "e" (current error) and "ed" (error rate of change), as illustrated in Figure 4. Similarly, for x and y position control, this relationship is depicted in Figure 5. By applying fuzzy control principles, the PID gains are continuously adjusted based on the varying values of "e" and "ed" to satisfy specific control objectives. This adaptive tuning mechanism enhances both the transient and steady-state performance of the system.

A set of seven fuzzy values is chosen to represent the linguistic variables for "e" (error) and "ed" (error derivative), as well as seven fuzzy values for the output. The membership functions employed in these controllers consist of a mix of triangular and Gaussian functions. Importantly, the widths of these fuzzy sets are non-uniform and have been fine-tuned through iterative trial-and-error testing, as depicted in Figure 6 for the x-position and Figure 7 for the y-position. The foundation of a fuzzy controller lies in its linguistic rule set, which is typically formulated based on expert knowledge and experience, though it can also be refined through experimental adjustments. The fuzzy rules utilized for adjusting the PID gains are outlined in Tables 1 and 2. Furthermore, the membership functions for all input and output variables have been carefully structured to achieve optimal control performance.

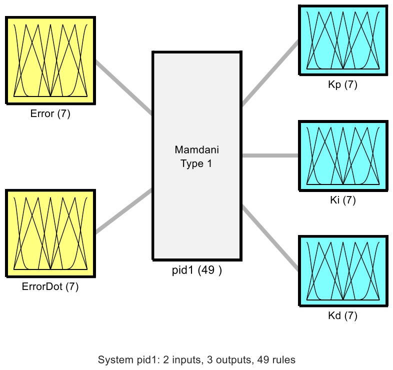


Figure 4. The fundamental design of a fuzzy controller for managing angles and z –position.

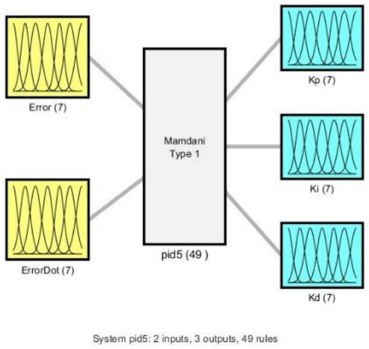


Figure 5. Basic structure of a fuzzy controller for x and y position.

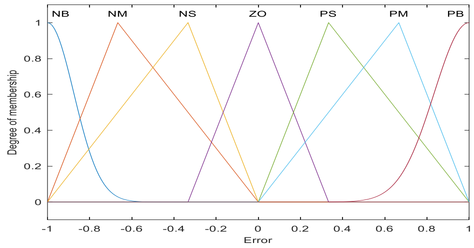


Figure 6. Membership function for all inputs and outputs for angles and z-position.

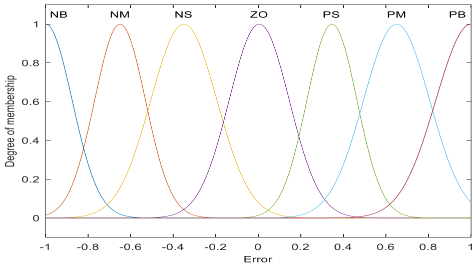


Figure 7. Membership function for all inputs and outputs for x and y position.

4. Formation control

In this section, the positions of all formation members would be detected by each quadrotor in the system with respect to a common reference coordinate system. Its control architecture comprises inner and outer control loops. The inner loop stabilizes the individual quadrotor platform and the outer loop achieves reference path tracking. Due to the large difference in time constant between these two loops, they can be treated independently with low-level and high-level dynamics [33,34], The high-level dynamics, are used for swarm control, as depicted in Figure 8. , which shows a schematic of the controlled nonlinear dynamics in formation.

Table 1. FUZZY RULE FOR KP, KI.

		Error						
		NB	NM	NS	ZO	PS	PM	PB
Error Rate	NB	M	S	VS	VVS	VS	S	M
	NM	B	M	S	VS	S	M	B
	NS	VB	B	M	S	M	B	VB
	ZO	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

Table 2. FUZZY RULE FOR KD.

		Error						
		NB	NM	NS	ZO	PS	PM	PB
Error Rate	NB	M	B	VB	VVB	VB	B	M
	NM	S	M	B	VB	B	M	S
	NS	VS	S	M	B	M	S	VS
	ZO	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

4-1. Communication Graph Between Agents in Multi-agent Control

The communication graph between agents plays an important role in multi-agent control, because it determines how information is balanced and exchanged among them. This graph can have various characteristics that influence the performance of the multi-agent system.

A communication graph is typically represented as $G=(V,E)$ where:

- V is a set of nodes, each representing an agent.
- E is a set of edges that represent communication links between agents. If $E\grave{o}(i,j)$, it means that agent i can share its information with agent j.

The selection of a suitable communication graph depends on various factors:

- **Type of system**: For example, in a group of quadcopters, a connected yet optimized graph might be preferred to reduce energy consumption.
- **Communication cost**: Fully connected graphs enable extensive communication but are resource-intensive.

- **Fault tolerance**: Denser graphs are generally more resilient to link failures.
- **Nature of the mission**: In leader-tracking missions, leader-follower graphs are more appropriate.asWXΘ

For controlling the agents in this article, the following communication graph is proposed: All agents receive path information from a leader and adjust their position and velocity to maintain a desired formation relative to the leader. Additionally, when agents are in close proximity to each other, they communicate to maintain safe distances and avoid collisions.

This section describes the general structure of the multi-agent system, consisting of six quadcopters, and the way they are controlled in a leader-follower configuration.

System Structure: Leader-Follower:

- Number of agents: Six quadcopters
- Role distribution: One leader and five followers
- Interaction and communication diagram:
  - The leader has a motion reference and communicates unidirectionally with the followers: it sends commands or position data but does not receive any control input from the followers.
  - The followers receive the leader's position and also communicate with each other to avoid collisions.
  - As a result, the communication graph is as follows:
    - Node 1 (leader) → outgoing edges to all followers
    - Between followers (nodes 2 to 6), a fully connected bidirectional graph exists for repulsion force calculation.

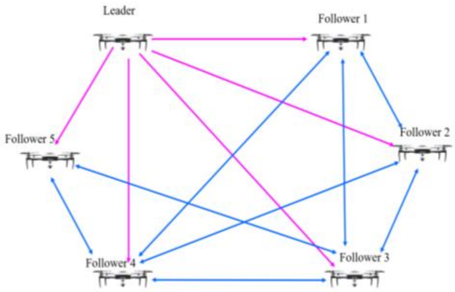


Figure 8. Communication graph between agents.

Figure 9 presents a schematic representation of how nonlinear dynamics are controlled within a group formation. This figure illustrates how each member synchronizes its behavior with the collective group through high-level control, while low-level control locally stabilizes each quadrotor. This efficient combination of local and collective control forms the foundation of successful performance in group flight systems. Ultimately, this hybrid control structure plays a crucial role in accomplishing complex, multi-agent flight missions by offering high levels of flexibility, stability, and adaptability.

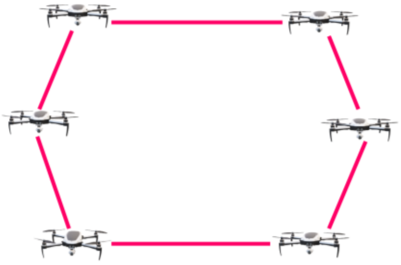


Figure 9. Schematic Formalization of Nonlinear Controlled System.

In addition to the overall control structure previously described, this section explores three distinct formation patterns within the group flight system. These formations are designed to enhance the system's efficiency, flexibility, and adaptability in response to varying environmental conditions and diverse mission requirements. Each formation possesses unique characteristics and specific applications. The helix formation is one of these structures, in which the group members are arranged to form a three-dimensional helix pattern. This configuration enables the even distribution of quadrotors across different altitudes and positions, making it highly suitable for missions that require extensive spatial coverage or imaging from multiple angles. In this formation, precise control over the altitude and horizontal position of each member is essential to maintain the spiral structure. The variation formation refers to a class of configurations in which the group's structure continuously and adaptively changes. Based on mission demands or environmental conditions, the spatial parameters of the agents are adjusted in real time. This type of formation offers high flexibility to respond to unexpected changes and requires the use of adaptive control algorithms and real-time decision-making at the group level.

For the sake of simplification, the control system is modeled as a double integrator, as expressed by the following equation. Accordingly, the control input for the leader and the other agents is defined as follows:

$$\ddot{q}_i = u_i \quad i = 1, 2, \ldots, N \tag{17}$$

For the leader factor we have:

$$u_1 \square \ddot{q}_{d1} + \left(\dot{q}_{d1} - \dot{q}_1\right) + \left(q_{d1} - q_1\right) \tag{18}$$

And for the leaders we have $\left(j = 1, 2, \ldots .N\right)$

$$u_i \square \sum_{j \in N_i} K_a \left\{ \left(q_j - q_i\right) + \left(\dot{q}_j - \dot{q}_i\right) \right\} \tag{19}$$

Theorm: Lyapunov-Based Stability Proof

We define the following Lyapunov function to analyze stability of the formation:

$$V = \frac{1}{2}\sum_{i=2}^{6} v_i - v_j^{\,2} + \frac{k_p}{2}\Sigma\left(q_i - q_j\right)^2 \tag{20}$$

This function is positive definite, and $V = 0$ only when all agents match the desired formation and move with the same velocity.

- **Derivative of the Lyapunov Function:**

We compute the time derivative of V:

Relative Velocity Error:

$$\dot{V}_1 = \sum_{i=2}^{6}\left(v_i - v_j\right)^T\left(\dot{q}_i - \dot{q}_1\right) \tag{21}$$

From the control law (1), for each follower:

$$\dot{v}_1 = u_i = -k_r\,\Sigma\left(q_i - q_j\right) - k_a\,\Sigma\left(v_i - v_j\right) \tag{22}$$

So :

$$\dot{V}_1 = \sum_{i=2}^{6}\left(v_i - v_1\right)^T\left[-k_r\,\Sigma\left(q_i - q_j\right) - k_a\,\Sigma\left(v_i - v_j\right)\right] \tag{23}$$

This expression will simplify using symmetry in the graph (undirected), and noting that:

- The terms involving position errors cancel with the derivative of the second term in V

1.1 Position Error Term

$$\dot{V}_2 = k_p\,\Sigma\left(q_i - q_j\right)^T\left(v_i - v_j\right) \tag{24}$$

This term also contributes to reducing the error since it promotes velocity alignment to correct position errors.

Putting both parts together:

$$\dot{V} = -k_a\,\Sigma v_i - v_j^{\,2} \tag{25}$$

This is:

- Negative semi-definite, as it's a sum of squared velocity differences
- Zero only when all agents have equal velocities.

4.6. Helix formation

This section focuses on the coordination of multiple quadrotors within a specified formation framework, leveraging the nonlinear dynamics inherent to quadrotor systems. Specifically, we consider a formation composed of six quadrotors, aiming for convergence of all agents to a predefined geometric configuration. We propose a formation control strategy grounded in a layered unified double-integrator formation model. A primary application of this approach, detailed in the subsequent section, involves the integration of the nonlinear system dynamics regulated in the secondary control layer with the overarching formation control scheme.

In this system, the leader is considered the primary agent that directly follows the reference trajectory. The position and velocity of this agent are controlled according to specific dynamic equations to minimize tracking error along the given path.

The desired trajectory is defined such that initially, the agents ascend from their initial positions while attempting to form a Hexagon formation up to an altitude of 5 meters. Then, they proceed along the path at a constant altitude, beginning to execute a spiral motion. In this operating system, a primary node is initially designated as the leader, which must follow the path outlined below.

For the leader factor we have:

$$u_1 \square\ \ddot{q}_{d1} + \left(\dot{q}_{d1} - \dot{q}_1\right) + \left(q_{d1} - q_1\right) \tag{26}$$

On the other hand, we know that:

$$q_{d1} = \left[x_d\ , y_d , z_d\right]^T \tag{27}$$

The vector $q_{d1} = \left[x_d\ , y_d , z_d\right]^T$ represents the desired position components, and the specific values for x_d $, y_d$, and z_d are provided in the simulation section.

And for the leaders we have $\left(j = 1, 2, \ldots .N\right)$

$$u_i \square \sum_{j \in N_i} a_{ij}\left[K_a\left\{\left(q_j - q_i - d_{ij}\right) + \left(\dot{q}_j - \dot{q}_i\right)\right\}\right] \tag{28}$$

where a_{ij}, d_{ij} are calculated from Equations (34) and (35).

4.2. Variation formation

Variation formation control is an advanced control methodology utilized in multi-agent systems, such as quadcopter swarms, to dynamically adapt the formation of the group based on mission objectives, environmental constraints, or system state variations. This approach ensures that a fleet of quadcopters can perform cooperative tasks effectively, such as search and rescue operations, environmental monitoring, or payload transport. The behavior of n quadcopters in the airspace is referred to as variation formation or in more common terms formation control, we are designing on the principle that with this one element we can create space in order to have n quadcopters to have a given configuration. This allows each quadcopter to adapt its position and orientation with respect to others, making it possible for the swarm to simultaneously execute tasks such as monitoring or transport in an environment while maintaining a desired formation pattern. In this section, system is in the general equations of double integrator with variation formation can be written as follows:

$$\ddot{q}_i = u_i \tag{29}$$

even though formation changes after t_s and the relation and distance between agents is determined through the following equation

$$d_i = \left(d_i^1 - d_i^2\right)\left(1 - e^{(t-t_s)}\right)u\left(t - t_s\right) \tag{30}$$

According to (30), j and i in d_i^j show the position of the quadcopter in varying formation and the i'th agent, respectively. Then, control input is applied to system as follows:

$$u_1 = \ddot{q}_{d2} + \left(\dot{q}_{d2} - \dot{q}_1\right) + \left(q_{d2} - q_1\right) + K\frac{q_1 - q_0}{r_1^2}\left(\frac{1}{r_1} - \frac{1}{r_a}\right) \tag{31}$$

where r_1 is the distance from each agent to the forward face of an obstacle with radius R:

$$r_1 \square q_1 - q_0 - R \tag{32}$$

where:

- r_a is the area of influence of the obstacle's repulsive potential field,
- q_{d2} is the reference trajectory of the agents.
- q_0 is the position of the center of the obstacle.

And for the leaders, we have $(j = 1, 2, \ldots N)$

$$u_i = \sum_{j \in N_i}\left[\lambda_1\left(q_j - q_i - d_i\right) + \lambda_2\left(\dot{q}_j - \dot{q}_i\right)\right] \tag{33}$$

5. Simulation and Results

We present three examples involving systems controlled by various combinations of PID and Fuzzy-PID controllers under both disturbed and undisturbed conditions. The next section evaluates the performance of Fuzzy-PID controllers in these scenarios. The simulation constants for the quadcopter are provided in Table 3. The discussion is organized into two main sections. The first section focuses on the design and comparison of two controllers to address internal issues. The second section explores three distinct formation control scenarios using the selected controller.

5.1.Comparison of PID and Fuzzy-PID Controllers

To enable simulation and performance evaluation, the mathematical model and corresponding control systems for the quadcopter were developed in MATLAB/Simulink. Initially, the quadcopter is considered to be in hover mode with specific angles of pitch, roll, and yaw. The system is then transitioned into a static state, where these angles reach their setpoints progressively. Both control methods are implemented, and their outcomes are compared. The gains of the classical PID controller were tuned using the Ziegler-Nichols method, and the resulting PID gain parameters are shown in Table 4, a Fuzzy-PID controller was developed for performance comparison with the classical PID controller. The data for the Fuzzy-PID controller are based on this strategy and given in Table 5.

Table 3. DisISGN PARAMETERS OF QUADCOPTER

symbol	Description and unit	Value
∅	Roll angle	-
θ	Pitch angle	-
φ	Yaw angle	-
m	Mass Of quadcopter	0.5
L	Center of quadcopter to center of propeller distance	0.2
I_{xx}	Body moment of inertia around the x-axis	4.85×10^{-3}
I_{yy}	Body moment of inertia around the y-axis	4.85×10^{-3}
I_{zz}	Body moment of inertia around the z-axis	8.81×10^{-3}
b	Thrust factor	3.36×10^{-5}
d	Drag factor	1.12×10^{-7}
j_r	Rotor inertia	2.92×10^{-6}

Table 4. PARAETER OF PIDAIN.

	K_p	K_i	K_d
ψ	1000	0.01	400
ϕ	80	0.01	12.3
θ	50	0.03	50
x	40	0	80
y	1	0	1
z	5	0.1	2

Table 5. INPUT RANG OF MEMBERSHIP FUNCTIONS IN FUZZY PID CONTROLLER.

	K_p	K_i	K_d
roll	[40 59]	[0 0.01]	[14 20]
pitch	[100 120]	[0 0.9]	[20 25]
yaw	[40 70]	[0.020.18]	[20 25]
x	[2 6]	[0 0.01]	[0.1 4]
y	[35 55]	[0 0.01]	[90 120]
z	[0.1 0.5]	[0 0.01]	[0.7 1.5]

The figures demonstrate the dynamic performance of the quadcopter's control system using both PID and fuzzy-PID controllers in the absence of external disturbances. Each subplot captures a distinct motion characteristic: Figure 10 illustrates the pitch response, Figure 11 shows the roll response, and Figure 12 presents the yaw response. Additionally, Figure 13 depicts the z-position behavior, while Figures 14 and 15 focus on the x-position and y-position, respectively. These plots provide a comparative analysis of the two controllers, emphasizing differences in settling times, oscillation levels, and steady-state precision. The findings indicate that the fuzzy-PID controller delivers smoother and more stable performance across various motion parameters, with its effectiveness influenced by specific system characteristics.

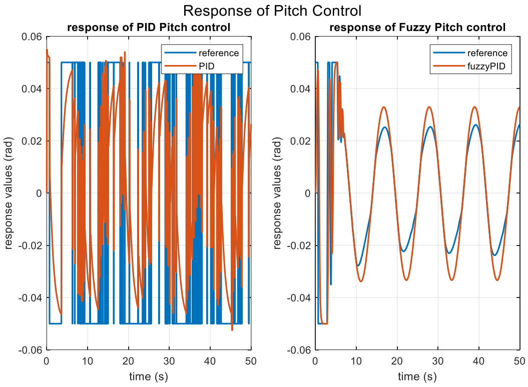


Figure 10. Pitch angle response without disturbances for (a) PID-Controller and (b) Fuzzy-PID Controller.

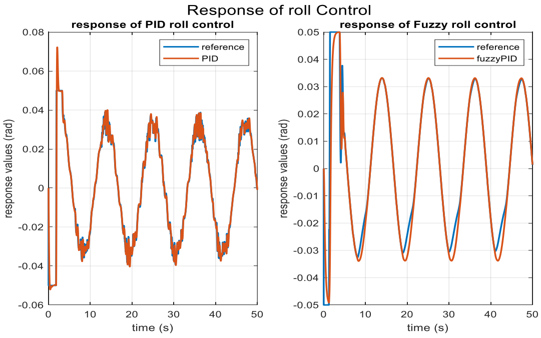


Figure 11. roll angle response without disturbances for (a) PID-Controller and (b) Fuzzy-PID Controller.

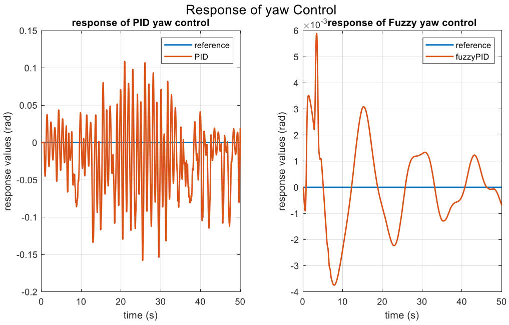


Figure 12. Yaw angle response without disturbances for (a) PID-Controller and (b) Fuzzy-PID Controller.

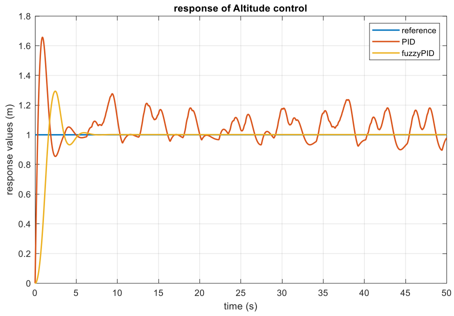


Figure 13. Response of z-position controllers without disturbance.

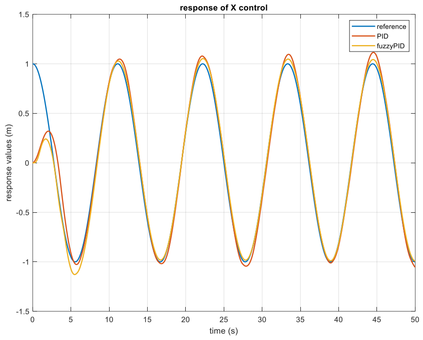


Figure 14. Response of x-position controllers without disturbance.

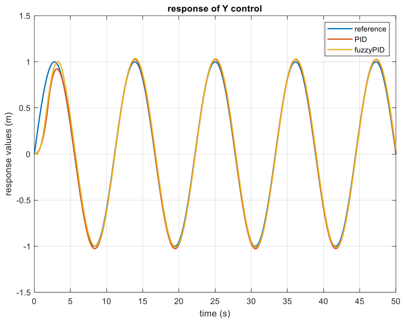


Figure 15. Response of y-position controllers without disturbance.

Furthermore, a disturbance with a step size of 0.6 is applied to both PID and Fuzzy-PID,controllers within the time range of 10 to 12 seconds. Figure 16 and Figure 17 compare the performance of PID and fuzzy-PID controllers under conditions with external disturbances, specifically for the pitch angle of the system. The figures demonstrate the dynamic response differences, where the fuzzy-PID controller shows enhanced stability and robustness, reducing oscillations and achieving quicker settling times. Figure 16 through Figure 21 further validate this by providing detailed responses across various disturbance scenarios, highlighting the superior adaptability of the fuzzy-PID controller in maintaining control accuracy and minimizing error. These comparisons underline the fuzzy-PID controller's ability to handle external perturbations effectively while maintaining smoother system behavior. Notably, the pitch angle experiences substantial overshoot initially with the classical PID controller, requiring approximately 5 seconds to stabilize in the presence of a disturbance.

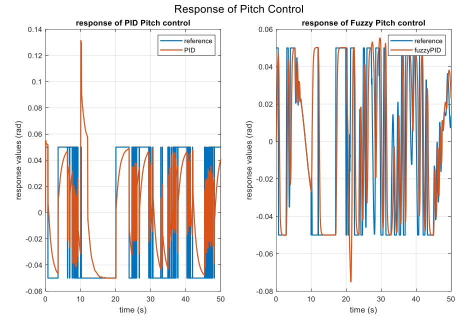


Figure 16. Pitch angle with disturbances for (a) PID-Controller and (b) Fuzzy-PID Controller.

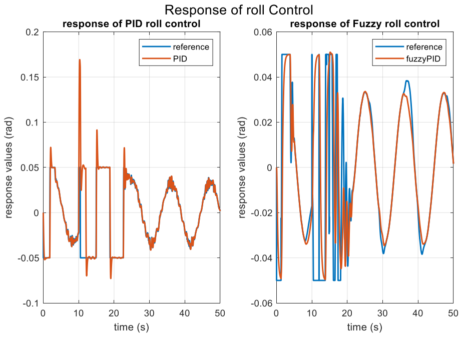


Figure 17. roll angle response with disturbances for (a) PID-Controller and (b) Fuzzy-PID Controller.

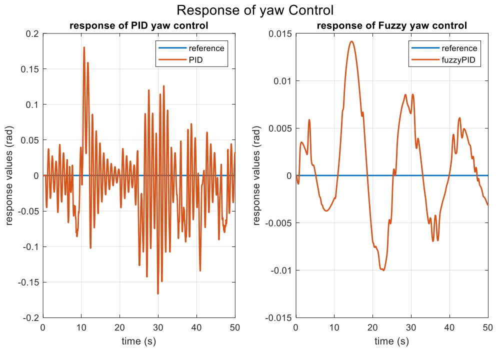


Figure 18. Yaw angle response with disturbances for (a) PID-Controller and (b) Fuzzy-PID Controller.

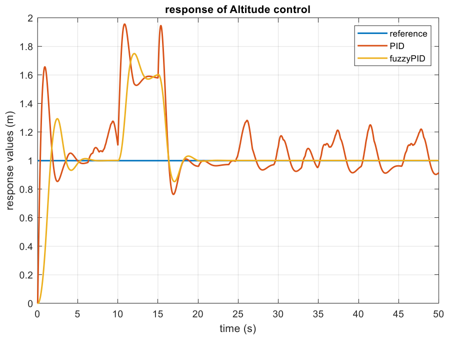


Figure 19. Response of Z-position controller with disturbance.

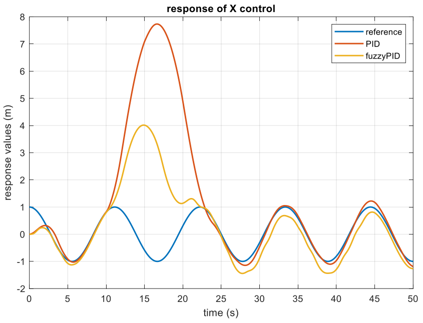


Figure 20. Response of x-position controllers with disturbance.

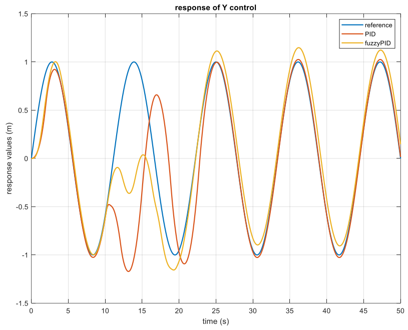


Figure 21. Response of y-position controllers with disturbance.

In contrast, the fuzzy-PID controller effectively mitigates the disturbance much faster, achieving stability within 2.5 seconds. This highlights the fuzzy-PID controller's superior performance in managing the pitch angle. For the roll angle, while the classical PID controller exhibits slightly less overshoot and undershoot compared to the fuzzy-PID controller, it tends to approach the instability threshold over time. Conversely, the fuzzy-PID controller successfully maintains control and achieves stability more rapidly in the presence of disturbances. This comparison demonstrates the fuzzy-PID controller's superiority in achieving quicker and more stable angular responses across multiple axes. All these changes have been calculated in Tables 6 .and Table 7, and for IAE and ISE in Tables 8 and 9.

Table 6. comparison of classical PID and Fuzzy-PID controllers for angle control.

angle	RT-PID	RT-Fuzzy-PID	OS-PID	OS-Fuzzy-PID	ST-PID	ST-Fuzzy-PID
roll	0.3	0.15	0.15	0	1	0.5
Pitch	0.3	0.15	0.05	0.002	0.7	0.3
yaw	0.4	0.3	0.4	0.3	2.5	2

Table 7. Comparison Between Classical PID and Fuzzy-PID Controllers for Position Control.

position	RT-PID	RT-Fuzzy-PID	OS-PID	OS-Fuzzy-PID	ST-PID	ST-Fuzzy-PID
x	0.3	0.15	1.2	0.001	1	0.6
y	1.2	0.3	0.4	0.3	5.1	0.6
z	1.5	0.2	0	0	3.5	0.5

Table 8. COMPARISON OF IAE and ISE for Angles PID AND Fuzzy-PID Controller.

angle	IAE-PID	IAE-Fuzzy-PID	ISE-PID	ISE-Fuzzy-PID
Roll	2.5	0.3	2	0.1
Pitch	1.8	0.2	1.5	0.1
Yaw	3.5	3	2	1.8

Table 9. COMPARISON OF IAE and ISE for Position PID AND Fuzzy-PID Controller

position	IAE-PID	IAE-Fuzzy-PID	ISE-PID	ISE-Fuzzy-PID
x	3	1	2.5	0.5
y	5	1.2	4	0.6
z	4	0.8	3.5	0.4

5.2. Formation Controller

The results reveal that the Fuzzy-PID controller effectively rejects disturbances and accurately controls all outputs. Ultimately, we explore a problem involving six agents and conduct comprehensive simulations to implement formation control on a nonlinear model.

At this juncture, it is essential to outline three scenarios to illustrate the controller's performance. The forthcoming discussion will encompass these scenarios, showcasing the robustness and precision of the Fuzzy-PID controller in various formation control tasks.

5.2.1 Helix Formation

In this scenario, we chose these parameters for the helix formation, adjoint matrix is shown in (34).

$$A=\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{34}$$

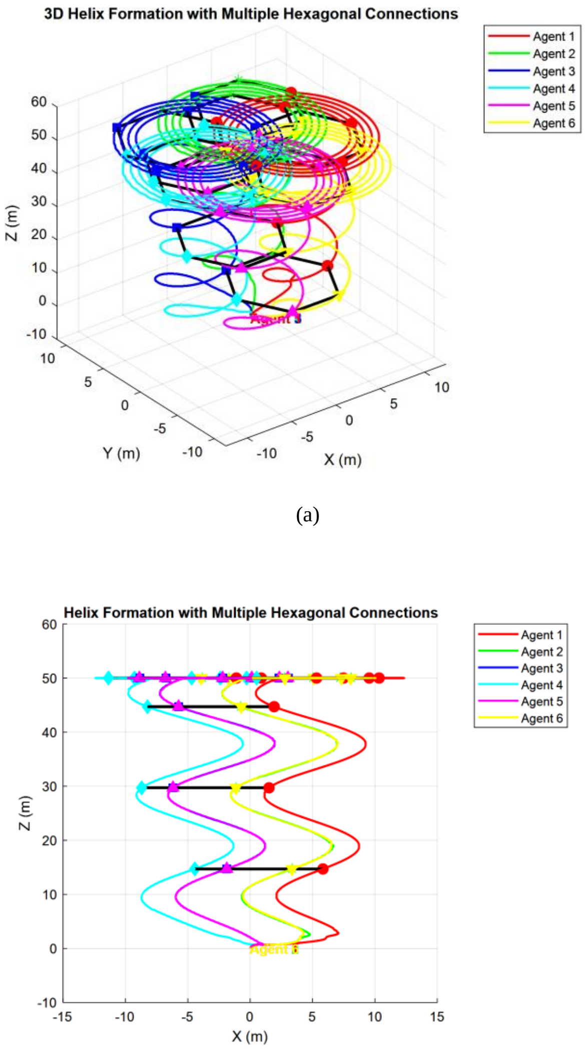
Additionally, the desired distance matrix has been defined below:

$$d=\begin{bmatrix} 0 & 1.1756 & 1.9021 & 1.9021 & 1.1756 & 1 \\ 1.1756 & 0 & 1.1756 & 1.9021 & 1.9021 & 1 \\ 1.9021 & 1.1756 & 0 & 1.1756 & 1.9021 & 1 \\ 1.9021 & 1.9021 & 1.1756 & 0 & 1.1756 & 1 \\ 1.1756 & 1.9021 & 1.9021 & 1.1756 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{35}$$

Furthermore, to achieve the formation among agents, The unknown object trajectory for the three dimensions x, y, and z is established in Equation (36)

$$\begin{aligned} x_d &= 5\sin\frac{\pi}{20}t \\ y_d &= 5\cos\frac{\pi}{20}t \\ z_d &= 0.1t \end{aligned} \tag{36}$$

Figure 22 illustrates the helix formation configuration through two perspectives: 3D and 2D representations.



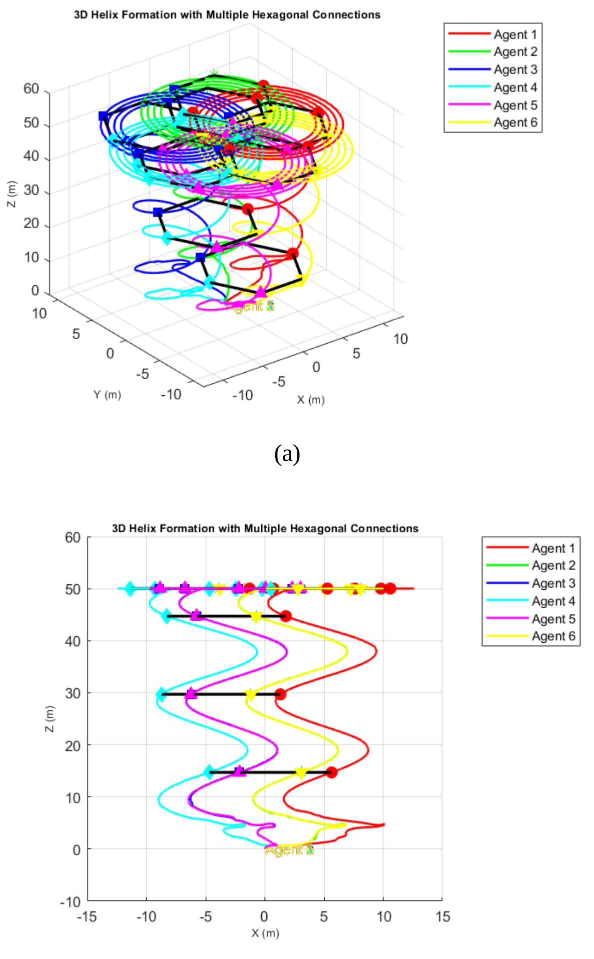
(a)

(b)

Figure 22. Agent’s formation in desired path (a)- 3D view of the helix trajectory for agents, (b) -2D planar projection of the helix trajectory.

Figure 22. (a) depicts the 3D visualization of the helix trajectory, demonstrating how all agents maintain their spatial alignment along the curved three-dimensional path. This perspective highlights the dynamic coordination of the agents in achieving the desired helix formation. Conversely, Figure 22. (b) provides a 2D projection of the same formation, focusing on the planar alignment of agents. This top-down view simplifies the trajectory, emphasizing the positional relationships and movement of the agents within the formation. Together, these representations validate that the six controlled nonlinear dynamic factors successfully reach the desired helix configuration. The results confirm that using system dynamics with internal controllers and specified formulas enables precise path-following, ensuring the agents achieve the defined structure in both 3D and 2D spaces.

After that, in another scenario, we added a disturbance with a step signal entered in time 10s and with amplitude of 1 and duration of 0.8s; outputs are shown in Figure 23. This figure illustrates the helix formation configuration under the presence of external disturbances, represented in both 3D and 2D perspectives. Figure 23. (a) shows the 3D visualization of the trajectory, where the agents successfully maintain their alignment along the desired helix path despite the imposed disturbances. This perspective highlights the robustness of the control system in compensating for dynamic perturbations and ensuring stability. Figure 23. (b) provides the 2D planar projection of the disturbed helix trajectory. While slight deviations from the ideal path are noticeable, the agents demonstrate the capability to adjust their positions and realign with the formation dynamically.

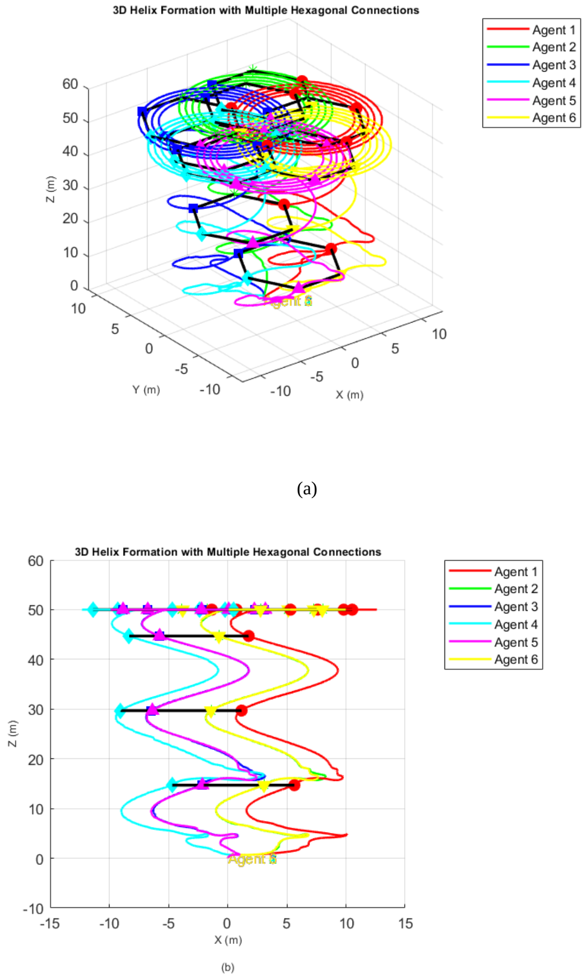


(a)

(b)

Figure 23. Agent's formation in desired path with one Disturbance (a)-3D view of the trajectory with disturbance compensation, (b)-2D planar projection of the trajectory under disturbances.

Finally, in another scenario, two disturbances were applied to our formation system. The first disturbance occurred within the time range of 10 to 12 seconds with a step size of 0.1, and the second disturbance was introduced between 50 and 52 seconds with a step size of 0.6. The results demonstrated that the quadcopter system is capable of handling multiple disturbances effectively, maintaining its formation, and continuing along its helix trajectory, as illustrated in Figure 24. This figure demonstrates the helix formation under the influence of two disturbances applied simultaneously to the system.



(a)

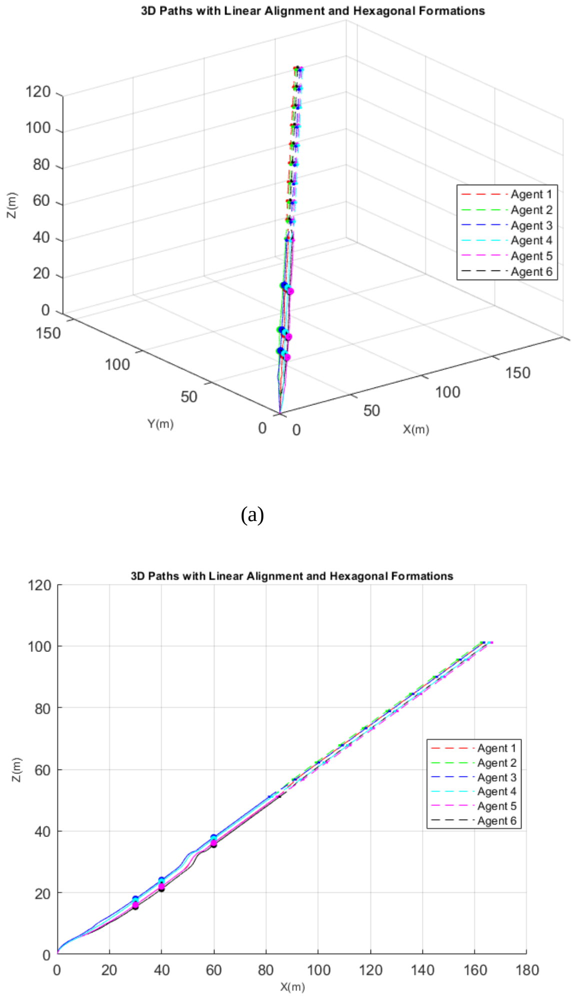
(b)

Figure 24. Agent's formation in desired path with Two Disturbance (a)- 3D representation showcasing trajectory adaptation in response to disturbances, (b)-2D projection highlighting planar alignment and recovery to maintain the helix structure.

Figure 24 (a) illustrates the 3D representation of the trajectory, showing how the agents dynamically adapt to the dual disturbances while maintaining the helix configuration. This perspective highlights the robustness of the controlled dynamics in a three-dimensional space. On the other hand, Figure 24 (b) presents the 2D projection, providing a clearer view of the agents' planar positions and their ability to recover and maintain the desired formation. The results validate that the proposed control system effectively compensates for multiple disturbances, enabling precise path-following and helix structure maintenance in both 3D and 2D views.

5.2.2. Variation formation

In this section, time-varying formation is considered. Before the 50s, the formation was as a line, and afterward, it was changed to a Hexagon shape, as illustrated in Figure 25. This figure presents the variation formation configuration through two distinct perspectives: 3D and 2D representations.



(a)

(b)

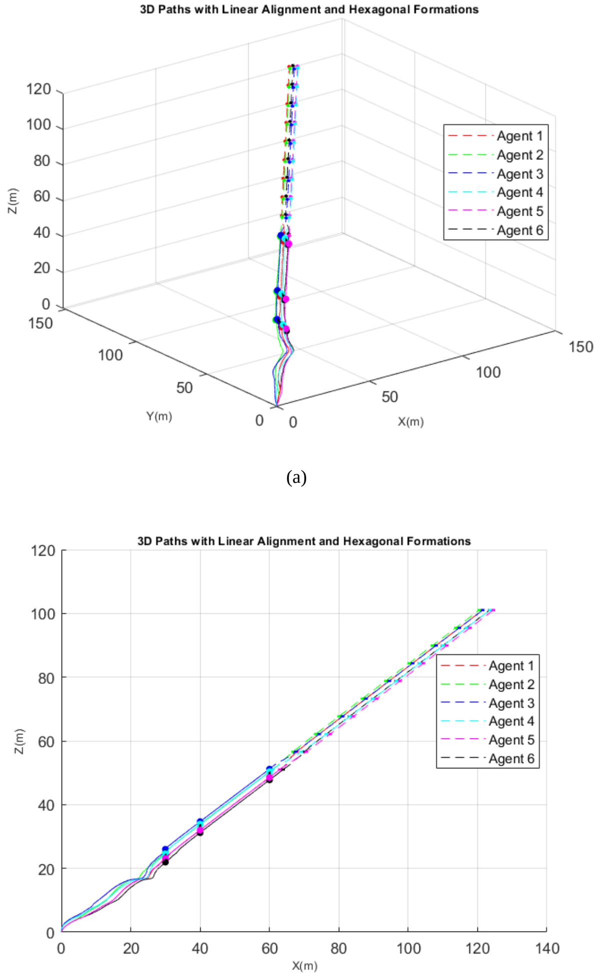
Figure 25. pentagonal formation in variation formation (a)-3D view illustrating the spatial adaptation and dynamic formation of agents; (b) 2D projection emphasizing the positional relationships and planar movements of agents within the variation formation.

Figure 25 (a) illustrates the 3D view of the variation formation, where the agents adapt their spatial alignment dynamically, showcasing the flexibility of the system in achieving the desired formation. This visualization highlights the smooth transitions and continuous adjustments of the agents along the curved three-dimensional trajectory. On the other hand, Figure 25 (b) offers a 2D projection of the same variation formation, providing a planar perspective that emphasizes the positional relationships and movement of the agents. The Z trajectory is considered zero in the plotting. Parameters and gains regarding this formation are shown in Table 10.

Table 10. variation formation parameters.

symbol	*parameter*	*value*
λ_1	*Constant coefficient*	5
λ_2	*Constant coefficient*	5
t_s	*Formation transition time*	50
r_a	*Obstacle radios*	5
q_o	*Circle position*	$(10,10,2.5)$
K	*Constant coefficient*	10
k_a	*Constant coefficient*	10
τ	*Exponential rate*	10
d_1^1	*Desired distance*	$\begin{bmatrix}0 & 6\end{bmatrix}^T$
d_1^2	*Desired distance*	$\begin{bmatrix}0 & 2\end{bmatrix}^T$
d_2^1	*Desired distance*	$\begin{bmatrix}0 & 4\end{bmatrix}^T$
d_2^2	*Desired distance*	$\begin{bmatrix}2\cos(18) & 2\sin(18)\end{bmatrix}^T$
d_3^1	*Desired distance*	$\begin{bmatrix}0 & 2\end{bmatrix}^T$
d_3^2	*Desired distance*	$\begin{bmatrix}-2\cos(18) & 2\sin(18)\end{bmatrix}^T$
d_4^1	*Desired distance*	$\begin{bmatrix}0 & -2\end{bmatrix}^T$
d_4^2	*Desired distance*	$\begin{bmatrix}2\cos(54) & -2sin(54)\end{bmatrix}^T$
d_5^1	*Desired distance*	$\begin{bmatrix}0 & -4\end{bmatrix}^T$
d_5^2	*Desired distance*	$\begin{bmatrix}2\cos(54) & -2sin(54)\end{bmatrix}^T$
d_6^1	*Desired distance*	$\begin{bmatrix}0 & -6\end{bmatrix}^T$
d_6^2	*Desired distance*	$\begin{bmatrix}-2\cos(18) & -2cos(18)\end{bmatrix}^T$
q_{d2}	reference trajectory	$[50,50,5]^T$

In the subsequent scenario, a disturbance was introduced in the form of a step signal at the 10-second mark, with an amplitude of 1 and a duration of 0.8 seconds. The resulting outputs are presented in Figure 26. This figure illustrates the variation formation under the influence of a single disturbance, represented in two perspectives: 3D and 2D. Figure 26 (a) shows the 3D representation of the formation, where the disturbance introduces temporary deviations in the agents' trajectories along the spatial path. Despite these deviations, the system effectively guides the agents back to the desired configuration, ensuring the integrity of the variation formation. Figure 26 (b) provides a 2D projection of the same scenario, highlighting the planar deviation and subsequent recovery of the agents. This view emphasizes the positional adjustments made by the agents in response to the disturbance. Both perspectives confirm the robustness of the system in maintaining formation stability despite external disruptions.

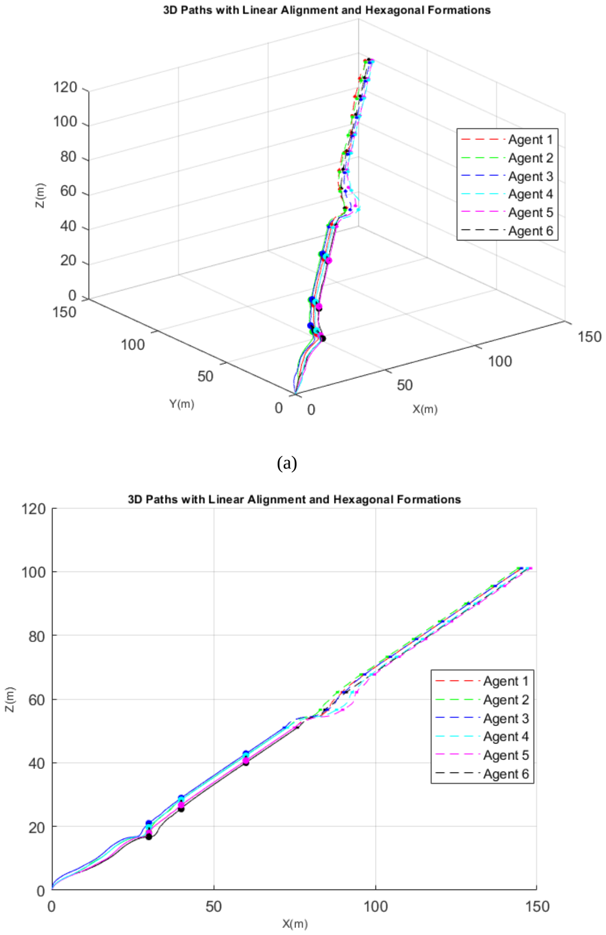


(a)

(b)

Figure 26. Variation formation under the influence of a one disturbance, with (a) showing the 3D representation and (b) the 2D projection, illustrating the system's resilience in maintaining the desired configuration.

In a final scenario, two disturbances were applied to the formation system. The first disturbance was introduced between 10 and 12 seconds with a step size of 0.1, while the second disturbance occurred between 50 and 52 seconds with a step size of 0.6. The results showed that the quadcopter system successfully managed both disturbances, maintaining formation integrity and continuing along its helical trajectory, as depicted in Figure 27. This figure demonstrates the variation formation under the influence of two simultaneous disturbances, presented in both 3D and 2D perspectives. Figure 27 (a) depicts the 3D visualization of the formation, showcasing how the agents momentarily deviate from their spatial trajectories due to the disturbances. Despite these perturbations, the system efficiently readjusts the agents' paths, successfully restoring the intended formation along the three-dimensional variation trajectory. Figure 27 (b) illustrates the 2D projection of the same scenario, providing a clear view of the agents' deviations and the corrective measures undertaken by the system in a planar context. This projection emphasizes the system's ability to counteract the combined effects of multiple disturbances, ensuring stability and alignment. Both representations validate the robustness of the control system in maintaining the variation formation under complex dynamic conditions.



(a)

(b)

Figure 27. Variation formation under the influence of two simultaneous disturbances, with (a) **showing the 3D representation,** (b) **the 2D projection, demonstrating the system's robustness in restoring the desired configuration.**

6. Conclusion

This study investigated the control of a quadcopter system by exploring both classical PID and fuzzy-PID control strategies. Initially, the nonlinear dynamics of the quadcopter were analyzed, followed by linearization to derive its state-space representation. The primary control objective was to achieve precise position control of the nonlinear quadcopter model using six individual PID controllers. These controllers were tuned to effectively manage all system outputs, with their parameters optimized to ensure stable and accurate responses. The resulting output plots confirmed that the selected PID gains were appropriately configured to regulate the system's behavior. A constrained fuzzy-PID controller was developed to improve performance while operating within system limitations. A comparative evaluation between the fuzzy-PID and conventional PID controllers demonstrated that the fuzzy-PID approach provided superior output regulation, particularly under constrained conditions. This controller exhibited a greater capability to accommodate physical limitations, ensuring smoother control inputs and better adaptability to nonlinear system dynamics. Additionally, the study explored the formation control of six quadcopters, incorporating their nonlinear characteristics. Various theoretical formation strategies were analyzed and subsequently implemented using a nonlinear fuzzy-PID controller. Simulation results confirmed the effectiveness of this method, successfully achieving formation control objectives while preserving system stability. A key feature of this research was the integration of the Ziegler-Nichols tuning method for initial PID parameter adjustment, followed by fine-tuning through a fuzzy supervisory controller. This two-step strategy was employed to independently regulate the three Euler angles of the quadcopter. Extensive MATLAB/Simulink simulations highlighted the benefits of the fuzzy-PID approach, such as reduced reliance on an exact system model, greater robustness against disturbances, and enhanced handling of nonlinear dynamics. The fuzzy-PID controller outperformed the classical PID controller in critical performance aspects, including lower overshoot, shorter settling time, and smoother transient responses. Notably, the design of the fuzzy controller, encompassing its membership functions and rule base, was guided by expert knowledge and iterative optimization.

References

[1] A. R. Girard, A. S. Howell, and J. K. Hedrick, "Border patrol and surveillance missions using multiple unmanned air vehicles," in Proc. 43rd IEEE Conf. Decis. Control (CDC)*, vol. 1, 2004, pp. 620-625.

[2] Ö. Dündar, M. Bilici, and T. Ünler, "Design and performance analyses of a fixed wing battery VTOL UAV," Eng. Sci. Technol., Int. J., vol. 23, no. 5, pp. 1182-1193, 2020.

[3] M. M. Ferdaus et al., "Redpac: A simple evolving neuro-fuzzy-based intelligent control framework for quadcopter," in Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE), 2019, pp. 1-7.

[4] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), vol. 3, 2004, pp. 2451-2456.

[5] P. E. I. Pounds, "Design, construction and control of a large quadrotor micro air vehicle," Ph.D. dissertation, Dept. Eng., Australian Nat. Univ., Canberra, Australia, 2007.

[6] P. Burggräf et al., "Quadrotors in factory applications: Design and implementation of the quadrotor's P-PID cascade control system," SN Appl. Sci., vol. 1, no. 7, p. 722, 2019.

[7] S. Abdelhay and A. Zakriti, "Modeling of a quadcopter trajectory tracking system using PID controller," Procedia Manuf., vol. 32, pp. 564-571, 2019.

[8] R. Miranda-Colorado and L. T. Aguilar, "Robust PID control of quadrotors with power reduction analysis," ISA Trans., vol. 98, pp. 47-62, 2020.

[9] D. Park et al., "Online tuning of PID controller using a multilayer fuzzy neural network design for quadcopter attitude tracking control," Front. Neurorobot., vol. 14, p. 619350, 2021.

[10] M. Davanipour et al., "Chaotic self-tuning PID controller based on fuzzy wavelet neural network model," Iran. J. Sci. Technol., Trans. Electr. Eng., vol. 42, pp. 357-366, 2018.

[11] T. Xu, "PID control and simulation of moving mass quadcopter UAV," in J. Phys.: Conf. Ser., vol. 2489, no. 1, 2023, p. 012015.

[12] H. Wang et al., "Power control in UAV-supported ultra dense networks: Communications, caching, and energy transfer," IEEE Commun. Mag., vol. 56, no. 6, pp. 28-34, 2018.

[13] H.-W. Lee and C.-S. Lee, "Research on logistics of intelligent unmanned aerial vehicle integration system," J. Ind. Inf. Integr., vol. 36, p. 100534, 2023.

[14] N. Abbas et al., "A survey: Future smart cities based on advance control of Unmanned Aerial Vehicles (UAVs)," Appl. Sci., vol. 13, no. 17, p. 9881, 2023.

[15] S. Sai et al., "A comprehensive survey on artificial intelligence for unmanned aerial vehicles," IEEE Open J. Veh. Technol., 2023.

[16] Y. Cao, W. Ren, and Z. Meng, "Decentralized finite-time sliding mode estimators and their applications in decentralized finite-time formation tracking," Syst. Control Lett., vol. 59, no. 9, pp. 522-529, 2010.

[17] X. Wang, V. Yadav, and S. Balakrishnan, "Cooperative UAV formation flying with obstacle/collision avoidance," IEEE Trans. Control Syst. Technol., vol. 15, no. 4, pp. 672-679, 2007.

[18] B. Yun et al., "Design and implementation of a leader-follower cooperative control system for unmanned helicopters," J. Control Theory Appl., vol. 8, pp. 61-68, 2010.

[19] R. Sharma and D. Ghose, "Collision avoidance between UAV clusters using swarm intelligence techniques," Int. J. Syst. Sci., vol. 40, no. 5, pp. 521-538, 2009.

[20] J. Wang and M. Xin, "Integrated optimal formation control of multiple unmanned aerial vehicles," IEEE Trans. Control Syst. Technol., vol. 21, no. 5, pp. 1731-1744, 2012.

[21] I. Bayezit and B. Fidan, "Distributed cohesive motion control of flight vehicle formations," IEEE Trans. Ind. Electron., vol. 60, no. 12, pp. 5763-5772, 2012.

[22] A. Kushleyev et al., "Towards a swarm of agile micro quadrotors," Auton. Robots, vol. 35, no. 4, pp. 287-300, 2013.

[23] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," IEEE Trans. Control Syst. Technol., vol. 9, no. 6, pp. 777-790, 2001.

[24] A. Abdessameud and A. Tayebi, "Formation control of VTOL unmanned aerial vehicles with communication delays," Automatica, vol. 47, no. 11, pp. 2383-2394, 2011.

[25] M. Turpin, N. Michael, and V. Kumar, "Decentralized formation control with variable shapes for aerial robots," in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2012, pp. 23-30.

[26] A. Azarbahram et al., "Leader-Follower Formation Control of Uncertain USV Networks Under Stochastic Disturbances," Int. J. Ind. Electron. Control Optim., vol. 5, no. 2, pp. 133-142, 2022.

[27] S. Khankalantary, I. Izadi, and F. Sheikholeslam, "Robust ADP-based solution of a class of nonlinear multi-agent systems with input saturation and collision avoidance constraints," ISA Trans., vol. 107, pp. 52-62, 2020.

[28] N. H. Sahrir and M. A. Mohd Basri, "Modelling and manual tuning PID control of quadcopter," in Control, Instrumentation and Mechatronics: Theory and Practice, Springer, 2022, pp. 346-357.

[29] F. Tatari and M. B. Naghibi-Sistani, "Optimal adaptive leader-follower consensus of linear multi-agent systems: Known and unknown dynamics," J. Artif. Intell. Data Min., vol. 11, no. 2, pp. 45-60, Jun. 2023. doi: 10.1234/jaidm.2023.123456.

[30] A. Surriani and M. Arrofiq, "Altitude control of quadrotor using fuzzy self tuning PID controller," in Proc. 5th Int. Conf. Instrum., Control, Autom. (ICA), 2017, pp. 67-72.

[31] T. J. Ross, Fuzzy Logic with Engineering Applications, 3rd ed. Hoboken, NJ: Wiley, 2005.

[32] E. Abbasi, M. Mahjoob, and R. Yazdanpanah, "Controlling of quadrotor UAV using a fuzzy system for tuning the PID gains in hovering mode," in Proc. 10th Int. Conf. Adv. Comput. Entertain. Technol., 2013, pp. 1-6.

[33] J.-W. Lee et al., "Adaptive altitude flight control of quadcopter under ground effect and time-varying load: Theory and experiments," J. Vib. Control, vol. 29, no. 3-4, pp. 571-581, 2023.

[34] M. De Queiroz, X. Cai, and M. Feemster, Formation Control of Multi-Agent Systems: A Graph Rigidity Approach. Hoboken, NJ: Wiley, 2019.

کنترل آرایش بندی کوادکوپترها با آرایش متغیر با زمان برای کنترل کننده فازی- PID

سیده مهسا ذکی پور و سعید خان کلانتری*

گروه آموزشی دانشکده مهندسی برق، دانشگاه صنعتی خواجه نصیر الدین طوسی، تهران، ایران.

ارسال ۲۰۲۵/۰۳/۰۵ بازنگری ۲۰۲۵/۰۵/۱۳؛ پذیرش ۲۰۲۵/۰۷/۰۷

چکیده:

در این مقاله، به طراحی و پیاده‌سازی استراتژی‌های کنترلی پیشرفته برای کنترل موقعیت و آرایش‌یابی یک سیستم چندعامله متشکل از ربات‌های پرنده (کوادکوپترها) می‌پردازد. هدف اصلی، دستیابی به‌دقت بالا در دنبال‌کردن مسیرهای مرجع و حفظ پایداری آرایش در حضور نامعینی‌های سیستمی و اغتشاشات خارجی است. در ابتدا، کنترل‌کننده‌های PID کلاسیک بر اساس مدل دینامیکی استاندارد کوادکوپتر طراحی شده‌اند. بااین‌حال، به دلیل محدودیت این کنترل‌کننده‌ها در مواجهه با غیرخطی‌بودن و اغتشاشات، یک کنترل‌کننده ترکیبی فازی -PID محدودشده پیشنهاد شده است تا عملکرد سیستم بهبود یابد.کنترل‌کننده پیشنهادی، با بهره‌گیری از منطق فازی و سادگی PID، امکان تنظیم خودکار پارامترهای کنترلی متناسب با شرایط متغیر سیستم را فراهم می‌آورد. شبیه‌سازی‌های گسترده در محیط MATLAB/Simulink نشان‌دهنده برتری عملکرد کنترل‌کننده فازی -PID در مقایسه با کنترل‌کننده‌های PID کلاسیک و فازی -PID استاندارد است. این کنترل‌کننده در تمام محورهای حرکتی، دقت بالاتری در کاهش خطای دنبال‌کردن مسیر و پایداری بهتری در حفظ آرایش گروهی شش کوادکوپتر از خود نشان می‌دهد. این تحقیق اهمیت ویژه استفاده از تکنیک‌های کنترل هوشمند در سیستم‌های چندعاملی را به‌ویژه در معماری‌های توزیع‌شده که به‌اشتراک‌گذاری اطلاعات محلی میان عوامل متکی هستند، موردتأکید قرار می‌دهد. رویکرد ارائه‌شده موجب بهبود قابل‌توجه عملکرد کنترل در محیط‌های پویا و نامطمئن گردیده و مسیر را برای کاربردهای واقعی آینده هموار می‌سازد.

کلمات کلیدی: کوادکوپتر، کنترل کننده PID، کنترل کننده فازی - PID ، مسیریابی، کنترل آرایش بندی متغییر با زمان.