Pages: 61-74 / Vol. 18 / Issue 2 / Summer-Fall 2025

Journal of Aerospace Science and Technology

Journal Homepage: https://jast.ias.ir

Original Research Article

Robust Intelligent Trajectory Tracking for Quadrotors: Integrating Backstepping and Feedback-Error-Learning Techniques

Esfandiar Baghelani [1*], Jafar Roshanian[2], and Mohammad Teshnehlab[3]

1, 2. Institute of Intelligent Control Systems, K. N. Toosi University of Technology, Tehran, Iran.

2, 3. Faculty of Electrical Engineering, K. N. Toosi University of Technology, Tehran, Iran.

ARTICLE INFO

Received: 29 August 2024
Revised: 01 March 2025
Accepted: 02 March 2025
Available online: 07 May 2025

KEYWORDS:

Trajectory Tracking,
Quadrotor
Backstepping
Feedback-Error-Learning

DOI:
https://dor.org/10.22034/jast.2025.475816.1206

ABSTRACT

This paper presents a novel hybrid control approach for quadrotor unmanned aerial vehicles (UAVs), combining the Lyapunov-based backstepping method for position control with a neural network-based feedback-error-learning (FEL) technique for attitude control. The proposed strategy marks the first implementation of such a hybrid approach on a quadrotor platform, aiming to enhance control performance by leveraging classical and learning-based methods. The study's primary objectives include implementing a backstepping controller for altitude and horizontal position control, utilizing the FEL technique for roll, pitch, and yaw angle control, and evaluating the hybrid method's effectiveness in improving trajectory tracking accuracy under various simulated uncertainties. Simulation results demonstrate the proposed method's superior performance in trajectory tracking, with metrics such as root mean square error (RMSE) and mean absolute error (MAE) indicating enhanced accuracy. The research contributes to aerial robotics by showcasing the feasibility and effectiveness of integrating robust and adaptive control techniques, offering potential improvements in quadrotor performance within complex and uncertain environments.

1 INTRODUCTION

In recent years, unmanned aerial vehicles (UAVs), especially quadrotors, have advanced significantly due to their versatile applications in surveillance, transportation, and search-and-rescue operations.

Quadrotors face unique control challenges due to their nonlinear dynamics, underactuation, and sensitivity to disturbances. They have six degrees of freedom (6DoF) but are controlled by only four motors, leading to a mismatch between control inputs and degrees of freedom. This underactuation, combined with the complex interactions between translational and rotational motions, makes quadrotor guidance and control a significant challenge in aerial robotics.

Researchers have explored various control strategies to improve quadrotor stability and performance. Traditional methods, such as proportional-integral-derivative (PID) [1] controllers, are common but often perform poorly under varying conditions. More advanced techniques have since been developed.

One such approach is the Lyapunov-based backstepping [2] method, which has emerged as a robust strategy for position control. The backstepping control technique, initially proposed

[*] Corresponding Author's Email: esfandiar.baghelani@email.kntu.ac.ir

Print ISSN:1735-2134 Online ISSN: 2345-3648

by Kotovich et al. [3], has emerged as a highly successful technique for stabilizing nonlinear systems. This approach utilizes the principles of Lyapunov stability proof theory [4] to achieve robust and reliable control of complex dynamical systems. Through rigorous mathematical analysis and careful consideration of system dynamics, this method offers a systematic framework for ensuring stability and performance in controlling nonlinear systems. The backstepping control method has demonstrated its effectiveness across various engineering applications, including aerospace and robotics, making it a valuable tool for addressing the challenges of nonlinear dynamics in modern engineering systems.

This study aims to present a novel hybrid control approach that combines the Lyapunov-based backstepping method for position control with the neural network-based FEL technique for attitude control of quadrotors. The proposed method is particularly innovative as it marks the first implementation of such a hybrid strategy on a quadrotor platform.

The primary objectives of this research are:

- To implement a backstepping controller for direct altitude control and indirect horizontal position control in the horizontal plane.
- Utilize the FEL technique with a classical PID controller to control roll, pitch, and yaw angles, thereby enhancing overall control performance.
- To evaluate the effectiveness of the proposed hybrid approach in improving trajectory tracking accuracy under various simulated uncertainties, including added noise, external disturbances, and sudden weight changes.

The performance of the proposed control strategy will be assessed using metrics such as the RMSE, MAE, IAE, ISE, ITAE, and ITSE for trajectory tracking, as well as the integral of absolute control signal (IAC) to evaluate control effort and energy consumption.

This research contributes to aerial robotics by demonstrating the feasibility and effectiveness of a hybrid control strategy that merges classical and learning-based methods. The findings of this study have potential implications for improving quadrotor performance in complex and uncertain environments, with applications ranging from precision trajectory tracking to urban navigation and obstacle avoidance. The remainder of this paper is structured as follows: we will begin with a review of relevant literature, followed by the formulation of the 6DOF quadrotor dynamics model. Subsequently, the design of the proposed hybrid control approach will be detailed. We will then present a simulation study to evaluate the proposed method and compare its performance against both a finely tuned, fixed-gain PID controller and an adaptive fuzzy logic-based PID controller (as described in [5]). Finally, a discussion of the results and concluding remarks will be provided.

2 LITERATURE REVIEW

Quadrotors have seen significant advancements in control systems, evolving from conventional methods to advanced intelligent techniques. Initially, traditional control strategies such as PID [1] controllers, linear control techniques like linear quadratic regulator (LQR) [6], and model predictive control (MPC) [7] were widely used. PID controllers are favored for their simplicity and effectiveness in stabilizing flight, yet they often struggle with dynamic disturbances. Linear control methods provide stability in controlled environments but are limited by their reliance on linear models. While more sophisticated and capable of handling constraints, MPC faces challenges due to its computational demands.

As the complexity of quadrotor applications increased, researchers began exploring advanced intelligent control methods. Fuzzy logic control (FLC) [5], [8] emerged as a solution for managing uncertainties and nonlinearities, outperforming traditional PID controllers in various conditions. Integrating neural networks with existing control systems allowed for learning complex input-output relationships, significantly enhancing performance in trajectory tracking and disturbance rejection. Reinforcement learning (RL) [9] has also gained traction, enabling quadrotors to adaptively learn optimal control policies through interactions with their environment, significantly improving capabilities such as autonomous navigation and obstacle avoidance. However, a key limitation is its reliance on extensive trial and error, which can be detrimental for inherently unstable systems.

Integrating conventional robust control methods with advanced intelligent techniques has emerged as a promising approach to enhance quadrotor performance [10]. Traditional methods, such as Lyapunov-based controls including backstepping and sliding mode control [11], are known for their robustness in handling uncertainties and nonlinear dynamics. However, they often lack optimality and adaptability in varying conditions. Conversely, learning-based methods, such as neural controllers, exhibit strong adaptive capabilities due to their learning abilities but may struggle with stability guarantees and predictable performance. By integrating the precise control offered by backstepping with the adaptive learning capabilities of neural networks, this novel method addresses the limitations of each technique.

FEL has emerged as a promising control strategy in robotics and autonomous systems, particularly for complex nonlinear systems such as quadrotor UAVs. The FEL approach, initially proposed by Kawato et al. [12], [13], combines a feedforward controller with a feedback controller to achieve adaptive control. In 1993, Gomi and Kawato [14] proposed a new learning scheme using FEL for a neural network model applied to adaptive nonlinear feedback control. In 1995, Hamavand et al. [15] presented a neural network-based control strategy for the trajectory control of robotic manipulators using FEL. In 1996, Teshnehlab et al. [16] proposed a neural network controller with a flexible structure based on FEL for multi-input multi-output (MIMO) systems. In 2000, Kalanovic et al. [17] developed an FEL neural network to control a powered trans-femoral prosthesis.

In 2004, Nakanishi et al. [18] proposed the FEL method for nonlinear adaptive control. Presented the FEL controller for joint angle control using functional electrical stimulation (FES) in rehabilitation robotics in 2005. In 2006, Miyamura et al. [19] analyzed the stability of the FEL method with time delay. Using FEL, Ruan et al. [20] proposed an adaptive control scheme for inverted pendulum balancing. 2014 Taheri et al. [21] proposed the FEL controller with an adaptive neural network for low Earth orbit satellite tracking. Sabahi et al. [22] applied a type-2 fuzzy logic system with FEL for load frequency control, demonstrating improved performance over conventional methods. Khanesar et al. [23] 2015 developed the FEL control for magnetic satellites using type-2 fuzzy neural networks, showcasing the potential of combining FEL with advanced computational intelligence techniques. Kayacan et al. [24] introduced a self-learning disturbance observer for nonlinear systems in the FEL framework, addressing the challenge of time-varying disturbances. Asgari et al. [25] **2022** presented a stable FEL scheme for nonlinear nonaffine systems, demonstrating its ability to handle uncertainties and disturbances. In 2022**,** Figueiredo et al. [26] investigated using FEL for time-effective gait trajectory tracking in wearable exoskeletons, highlighting its potential for rehabilitation applications.

The key strength of FEL lies in its ability to learn the inverse dynamics of a system online, gradually reducing the reliance on the feedback controller [18]. This adaptive nature makes the FEL particularly suitable for systems with uncertainties and time-varying dynamics, such as quadrotors operating in diverse environments.

3 QUADROTOR 6DOF DYNAMIC MODEL

In the world-fixed reference $\mathrm{F_W}$, vector $\boldsymbol{p} = [x \quad y \quad z]^T$ represents the absolute position of the quadrotor at its center of gravity. The components of the velocity vector $v = [v_x \quad v_y \quad v_z]$ in (1) are expressed in the F_W. The dynamic model of the quadrotor's position within this fixed-world coordinate system is outlined in (2) [27]. The thrust vector T is generated by the four propellers. The gravitational acceleration on Earth is $g = 9.81[m/s^2]$, and the total mass of the quadrotor is denoted as m.

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \end{cases} \tag{1}$$

$$\begin{cases} \dot{v}_x = \dfrac{T}{m}(\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi)) \\ \dot{v}_y = \dfrac{T}{m}(\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi)) \\ \dot{v}_z = \dfrac{T}{m}\cos(\phi)\cos(\theta) - g \end{cases} \tag{2}$$

The quadrotor's attitude is represented by the vector $\boldsymbol{o} = [\phi \quad \theta \quad \psi]^T$ in the body frame F_B, known as

Euler angles. The angular velocity in the F_W is given by $\boldsymbol{\omega} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$, while the angular velocity in the F_B is denoted as $\boldsymbol{\omega_B} = [p \quad q \quad r]^T$. The dynamic model of the quadrotor's attitude is detailed in (3) and (4) [27].

$$\begin{cases} \dot{\phi} = p + \sin(\phi)\tan(\theta)\,q + \cos(\phi)\tan(\theta)\,r \\ \dot{\theta} = \cos(\phi)\,q - \sin(\phi)\,r \\ \dot{\psi} = \dfrac{\sin(\phi)}{\cos(\theta)}q + \dfrac{\cos(\phi)}{\cos(\theta)}r \end{cases} \tag{3}$$

The moments of inertia of the quadrotor around the three axes X_B, Y_B, and Z_B of the body coordinate system are denoted as I_x, I_y, and I_z respectively. The control inputs for the Euler roll, pitch, and yaw angles are represented by τ_ϕ, τ_θ, and τ_ψ in the control vector $\boldsymbol{u} = [T \quad \tau_\phi \quad \tau_\theta \quad \tau_\psi]$, respectively.

$$\begin{cases} \dot{p} = \dfrac{I_y - I_z}{I_x}qr + \dfrac{\tau_\phi}{I_x} \\ \dot{q} = \dfrac{I_z - I_x}{I_y}pr + \dfrac{\tau_\theta}{I_y} \\ \dot{r} = \dfrac{I_x - I_y}{I_z}pq + \dfrac{\tau_\psi}{I_z} \end{cases} \tag{4}$$

4 HYBRID CONTROL DESIGN

In this section, we present the design of a hybrid controller for quadrotor systems, integrating position control via the backstepping method and attitude stabilization through FEL. The interplay between these two methodologies will be elaborated in separate subsections, beginning with the backstepping calculations for position control. Subsequently, we will address the implementation of the FEL technique for attitude control. Figure 1 illustrates the block diagram of this hybrid approach, highlighting the uncertainties introduced by external disturbances and measurement noise.

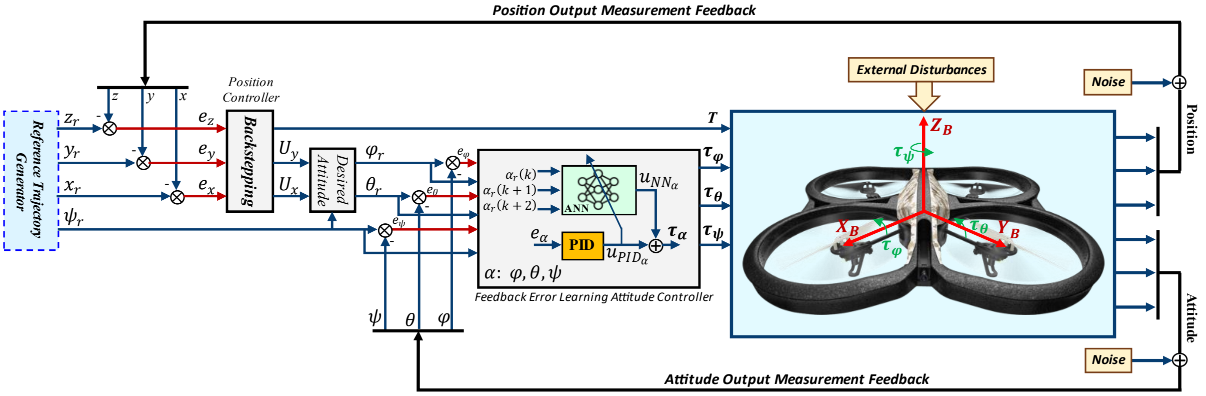

Figure 1. Block diagram of the Hybrid Controller of the quadrotor.

A. Backstepping Position Controller

This paper defines the position state variables for trajectory tracking control as tracking errors. Specifically, in the error vector $\boldsymbol{e} = \boldsymbol{p_r} - \boldsymbol{p}$ presented in (5), the subscript r denotes the desired position input value.

$$\begin{cases} e_x = x_r - x \\ e_y = y_r - y \\ e_z = z_r - z \end{cases} \tag{5}$$

The dynamic model of the quadrotor's position can be categorized into three separate state subspaces. The error component outlined in (5) functions as the state variable in these subspaces. Notably, the state variables x_1 and x_2 in (6) represent the tracking errors are associated with the desired position and velocity in the **X** direction.

$$\begin{cases} x_1 = e_x \\ x_2 = \dot{e}_x \end{cases} \tag{6}$$

Consequently, based on equations (2) and (5), the dynamic model of the position along the **X** direction within the state space can be expressed as follows:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \ddot{x}_r - \dfrac{T}{m}U_x \end{cases} \tag{7}$$

By analyzing (2) and (7), the virtual control input U_x can be formulated as shown in (8). Similarly, relations (9) to (14) are obtained.

$$U_x = cos(\phi)\, sin(\theta)\, cos(\psi) + sin(\phi)\, sin(\psi) \tag{8}$$

$$\begin{cases} x_3 = e_y \\ x_4 = \dot{e}_y \end{cases} \tag{9}$$

$$\begin{cases} \dot{x}_3 = x_4 \\ \dot{x}_4 = \ddot{y}_r - \frac{T}{m} U_y \end{cases} \tag{10}$$

$$U_y = cos(\phi)\, sin(\theta)\, sin(\psi) - sin(\phi)\, cos(\psi) \tag{11}$$

$$\begin{cases} x_5 = e_z \\ x_6 = \dot{e}_z \end{cases} \tag{12}$$

$$\begin{cases} \dot{x}_5 = x_6 \\ \dot{x}_6 = \ddot{z}_r - \frac{T}{m} U_z + g \end{cases} \tag{13}$$

$$U_z = cos(\phi)\, cos(\theta) \tag{14}$$

A.1. Thrust Control Design

Initially, the control input $u_1 = T$, representing the thrust controller, is designed using the backstepping method. To achieve this, the Lyapunov function candidate V_{x_5} is defined in (15) based on the state subspace outlined in (13).

$$V_{x_5} = \frac{1}{2} x_5^2 \tag{15}$$

According to Lyapunov's theory [28], the derivative of the function V_{x_5} must be negative definite [29], [30]. The expression for this derivative is provided in (16).

$$\dot{V}_{x_5} = x_5 \dot{x}_5 = x_5 x_6 \tag{16}$$

Assuming x_6 as a virtual controller, the necessary conditions for the derivative of the Lyapunov function candidate are established in (17).

$$x_6 = -k_5 x_5 \quad (k_5 \in R^+) \tag{17}$$

We introduce a new Lyapunov function candidate V_{x_5,x_6}, formulated by (17).

$$V_{x_5,x_6} = \frac{1}{2} x_5^2 + \frac{1}{2}(x_6 + k_5 x_5)^2 \tag{18}$$

The derivative of the function V_{x_5,x_6} must also be negative definite. The expression for this derivative can be found in (19).

V ̇_(x_5,x_6)=x_5 x ̇_5+(x_6+k_5 x_5)(x ̇_6+k_5 x ̇_5) (19)

By utilizing (16) and adding and subtracting the term $k_5 x_5^2$, followed by appropriate factoring, we arrive at (20).

$$\dot{V}_{x_5,x_6} = (x_6 + k_5 x_5)(x_5 + \dot{x}_6 + k_5 x_6) - k_5 x_5^2 \tag{20}$$

To establish the condition for the derivative of the Lyapunov function candidate V_{x_5,x_6}, it is necessary that condition (21) be satisfied.

$$x_5 + \dot{x}_6 + k_5 x_6 = -k_6(x_6 + k_5 x_5) \tag{21}$$

The T trust control is derived from (21) by substituting $\dot{x}_6$ from (13) by (22). Note that $k_6 \in R^+$.

$$T = \frac{m}{U_z}[\,\ddot{z}_r + g + (1 + k_5 k_6)x_5 + (k_5 + k_6)x_6] \tag{22}$$

A.2. Desired X-Y plane Attitude

Trajectory tracking in the X-Y plane is achieved through rolling and pitching. For this purpose, two virtual inputs**,** U_x and U_y are employed. Using the backstepping method, U_x is obtained similarly to the calculation of T in (23). Additionally, U_y is obtained in (24).

$$U_x = \frac{m}{T}[\,\ddot{x}_r + (1 + k_1 k_2)x_1 + (k_1 + k_2)x_2] \tag{23}$$

$$U_y = \frac{m}{T}[\,\ddot{y}_r + (1 + k_3 k_4)x_3 + (k_3 + k_4)x_4] \tag{24}$$

Note that $(k_1, k_2, k_3, k_4 \in R^+)$. When the control error approaches zero, the values of the Euler angles reach their desired values, and the virtual inputs U_x and U_y are at their desired levels according to (25) and (26).

$$U_{x_r} = cos(\phi_r)\, sin(\theta_r)\, cos(\psi_r) + sin(\phi_r)\, sin(\psi_r) \tag{25}$$

$$U_{y_r} = cos(\phi_r)\, sin(\theta_r)\, sin(\psi_r) - sin(\phi_r)\, cos(\psi_r) \tag{26}$$

Under the mentioned conditions, by multiplying $\sin(\psi_r)$ and $\cos(\psi_r)$ in (25) and (26), along with some mathematical operations of addition and subtraction, the desired roll angle ϕ_r is obtained in (27). Similarly, the target pitch angle θ_r can be derived in (28).

$$\phi_r = sin^{-1}\left(U_{x_r} sin(\psi_r) - U_{y_r} cos(\psi_r)\right) \tag{27}$$

$$\theta_r = sin^{-1}\left(\frac{U_{x_r} cos(\psi_r) + U_{y_r} sin(\psi_r)}{cos(\phi_r)}\right) \tag{28}$$

B. Feedback Error Learning Attitude Controller

We implement the FEL approach to achieve precise regulation of the roll, pitch, and yaw angles. This approach combines PID control with an inverse dynamics neural network that undergoes online training, enabling effective and adaptive management of the Euler angles. By leveraging this combination, the quadrotor can accurately follow desired trajectories, enhancing its performance and reliability in various operational conditions.

In the neural inverse model controller architecture, the feedforward relations transform the input into the neural control signal. The desired angle values at three distinct time points - the current step, one step in the future, and two steps in the future - are provided as inputs to the network. The input vector $\boldsymbol{X_{in}}$ is specified in relation (29), where k denotes the time step.

$$X_{in} = [\alpha_r(k), \alpha_r(k+1), \alpha_r(k+2)]^T \qquad (29)$$

The input vector is weighted using the first layer's weights and combined with the layer's bias to generate net^1, with the number of net^1 values corresponding to the number of neurons in this layer. These net^1 values are then processed through a hyperbolic tangent activation function, yielding the outputs of the first layer, designated as o^1. This procedure is similarly executed in the output layer, although here, the activation function is linear, and the Euler angles are substituted with the α. u_{NN_α} denotes the neural inverse dynamics model control signal.

$$\begin{cases} net^1 = W^1 X_{in} + b^1 \\ o^1 = tanh(net^1) \\ net^2 = W^2 o^1 + b^2 \\ u_{NN_\alpha} = net^2 \end{cases} \quad \alpha: \varphi, \theta, \psi \qquad (30)$$

Before the neural controller's input is transformed into a control signal during each iteration, its weights and biases are adjusted. The quadrotor's relative degree is denoted as $\bar{r} = 2$ [8]. Given this relative degree, the input to the neural controller is defined as (29). In the FEL method, $u_{pid_\alpha} = \tau_\alpha - u_{NN_\alpha}$, and the cost function is established by (31), aiming to minimize the classical control signal.

$$E = \frac{1}{2} u_{pid_\alpha}^2 \qquad (31)$$

According to the backpropagation learning algorithm [32], the training of weights and biases for the hidden layer and the output layer of the neural network is computed using chain derivatives, as outlined in (32) and (34).

$$\frac{\partial E}{\partial W^2} = \frac{\partial E}{\partial u_{pid_\alpha}} \frac{\partial u_{pid_\alpha}}{\partial u_{NN_\alpha}} \frac{\partial u_{NN_\alpha}}{\partial net^2} \frac{\partial net^2}{\partial W^2} \qquad (32)$$

After deriving and substituting the equivalent values for each derivative in (32), (33) is obtained.

$$\frac{\partial E}{\partial W^2} = -u_{pid_\alpha} \cdot o^1 \qquad (33)$$

In (34), by substituting the equivalent values of the chain derivatives, (35) is also derived.

$$\frac{\partial E}{\partial W^1} = \frac{\partial E}{\partial u_{pid_\alpha}} \frac{\partial u_{pid_\alpha}}{\partial u_{NN_\alpha}} \frac{\partial u_{NN_\alpha}}{\partial net^2} \frac{\partial net^2}{\partial o^1} \frac{\partial o^1}{\partial net^1} \frac{\partial net^1}{\partial W^1} \qquad (34)$$

$$\frac{\partial E}{\partial W^1} = -u_{pid_\alpha} \cdot W^2 \cdot \left(1 - o^{1^2}\right) \cdot \begin{bmatrix} \alpha_r(k) \\ \alpha_r(k+1) \\ \alpha_r(k+2) \end{bmatrix} \qquad (35)$$

The weights and biases are initialized randomly, with values ranging from -1 to +1. These weights and biases are updated in each iteration throughout the control process based on gradient descent calculations. The mathematical representation of the weight-updating process for each layer in the neural network is provided in (36) and (37). Specifically, (37) illustrates the generalized delta rule, where the learning rate η is essential for determining the speed and accuracy of learning, taking a value between 0 and 1.

$$\nabla W^i(k) = \frac{\partial E}{\partial W^i}(k) \quad i = 1,2 \qquad (36)$$

$$W^i(k+1) = W^i(k) - \eta \, \nabla W^i(k) \qquad (37)$$

Attitude FEL Control

The classical feedback Attitude control within the FEL method is established through the PID approach outlined in (38) as u_{pid_α}. In this context k_{p_α} refers to the proportional gain, k_{i_α} signifies the integral gain, and k_{d_α} represents the derivative gain. Note that α is replaced by three angles: roll (φ), pitch (θ), and yaw (ψ).

$$u_{pid_\alpha} = k_{p_\alpha}.e_\alpha + k_{i_\alpha}.\Sigma\, e_\alpha.\Delta t + k_{d_\alpha}\frac{\Delta e_\alpha}{\Delta t} \quad (38)$$

The Δe represents the changes in error over the time interval Δt. The symbol Σ denotes the sum of errors over the period, time Δt. The final control signal (39) for the three Euler angles applied to the quadrotor combines the neural controller signal and the classical controller signal.

$$\tau_\alpha = u_{pid_\alpha} + u_{NN_\alpha} \quad (39)$$

Simulation Results

In the section on simulation results, we detail the 6DoF flight modeling and simulation of the Parrot AR.Drone 2.0 quadrotor, followed by an evaluation of the implementation outcomes for the previously designed controllers. The block diagram in Figure 1 illustrates that the measurement of position and attitude data is subject to noise interference. Furthermore, the quadrotor is impacted by external disturbances along the Z-axis. Roll, pitch, and yaw disturbances are also integrated into the flight dynamic equations for the simulation. This section investigates two types of trajectories: conical and square. In both subsections, we show that applying the FEL method for stabilizing and controlling the Euler angles has significantly enhanced trajectory tracking.

Let's examine a general nonlinear MIMO system described by its state-space representation [31] in (40)

$$\begin{cases} x(k+1) = f\big(x(k)\big) + g\big(x(k)\big)u(k) \\ y(k) \quad\; = h\big(x(k)\big) \end{cases} \quad (40)$$

y represents the output of the system. The system functions are defined as follows:

$f: R^{n_S} \to R^{n_S}$: This function describes the state transition of the system.

$g: R^{n_S} \to R^{n_S} \times R^{n_I}$: This function maps the current state and input to the next state.

$h: R^{n_S} \to R^{n_O}$: This function relates the system state to the output.

In this context, n_S, n_I, and n_O represent the dimensions of the state, input, and output spaces, respectively.

Consider a general nonlinear MIMO system subject to external disturbances [8], [31]. This system can be described by (41):

$$\begin{cases} x(k+1) = f\big(x(k)\big) + g\big(x(k)\big)u(k) + d(k) \\ y(k) \quad\; = h\big(x(k)\big) \end{cases} \quad (41)$$

The disturbance vector affecting the system is $d \in R^{n_S}$.

Let's investigate a general nonlinear MIMO system with noisy measurements. $N: R^{n_S} \to R^{n_O}$ represents the additive noise, such as additive white Gaussian noise, at time step k [31]. The system can be described by (42):

$$\begin{cases} x(k+1) = f\big(x(k)\big) + g\big(x(k)\big)u(k) \\ y(k) \quad\; = h\big(x(k)\big) + N(k) \end{cases} \quad (42)$$

In this study, to evaluate the effectiveness of the proposed method in trajectory tracking of the quadrotor, disturbances were introduced according to (43)-(47).

$$d_{Z_1}(t) = 0.5g \qquad 30 < t < 32.1 \quad (43)$$

$$d_{Z_2}(t) = -0.45g \qquad 55 < t < 58.1 \quad (44)$$

$$d_{Roll}(t) = 0.002\sin(t)\cos(2t) \quad (45)$$

$$d_{Pitch}(t) = 0.002 \qquad 20 < t < 23.1 \quad (46)$$

$$d_{Yaw}(t) = 0.105 \qquad 40 < t < 43.1 \quad (47)$$

Additionally, white noise with a zero mean and a variance of 0.01 was added to the position and attitude measurements.

This approach allows us to assess the impact of disturbances and noise on the system's performance and enables a more thorough analysis of the proposed method's effectiveness.

Following the trajectory tracking results for both conical and square paths, presented in figures across subsections A and B, we will report in Tables 1 and 2 the average percentage change of performance metrics. These tables will summarize the proposed FEL approach's performance relative to the fixed-gain PID and FLS-tuned PID controllers, based on data collected from simulations across conical, spiral, and square trajectories.

A. Conical Trajectory

The conical trajectory for the quadrotor is designed to guide its movement along a defined path that resembles a cone shape. This trajectory is characterized by specific relationships for the desired position and attitude, as outlined in relations (48) to (51).

$$x_r = 0.2t\sin(t/3) \tag{48}$$

$$y_r = 0.2t\cos(t/3) \tag{49}$$

$$z_r = 0.8t + 2 \tag{50}$$

$$\psi_r = 0.0873\sin(t/3) \tag{51}$$

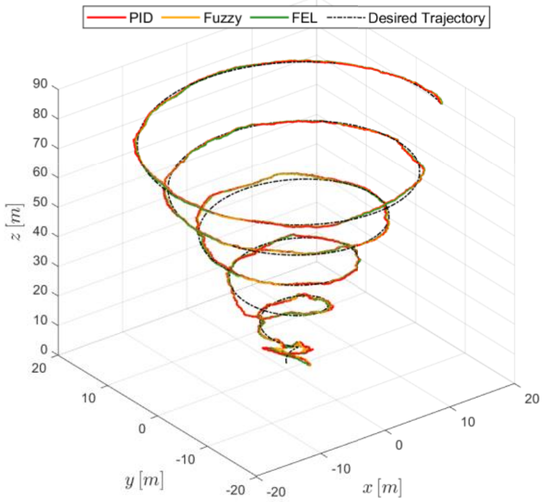

Fig. 2. 3D view of conical trajectory tracking by Parrot with PID vs FEL controller

Figure 2 illustrates the trajectory tracking of the quadrotor along a conical path using PID and FEL controllers in three-dimensional space within the F_W. Figure 3 also shows the quadrotor's position variations over time compared to the desired position.

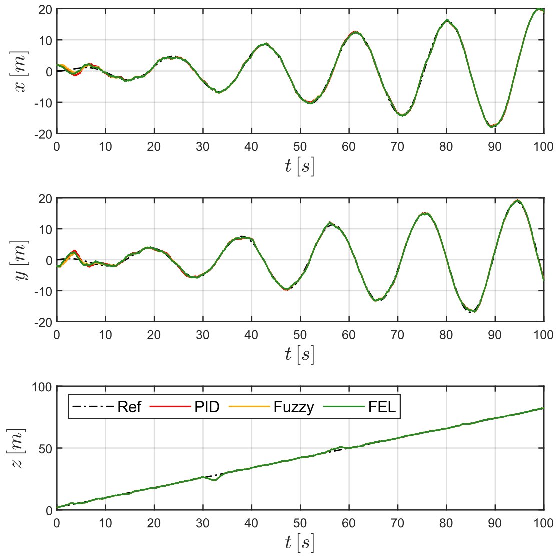

Fig. 3. The behavior of the three axes' positions in conical trajectory tracking over time.

Figure 4 displays the tracking error across the three axes. The proposed method exhibits lower tracking errors in trajectory tracking.

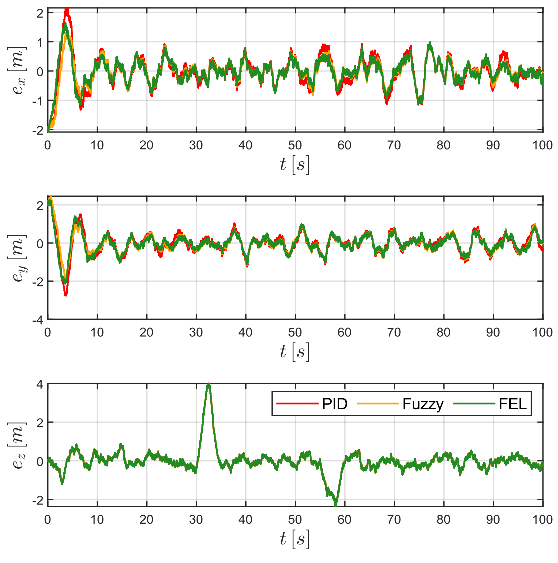

Fig. 4. Position errors in conical trajectory tracking

Figure 5 illustrates the variations in attitude, including the Euler angles, over time. The desired values for roll and pitch angles differ based on the control method used. This discrepancy affects the position controller's effectiveness in the performance of the attitude controller.

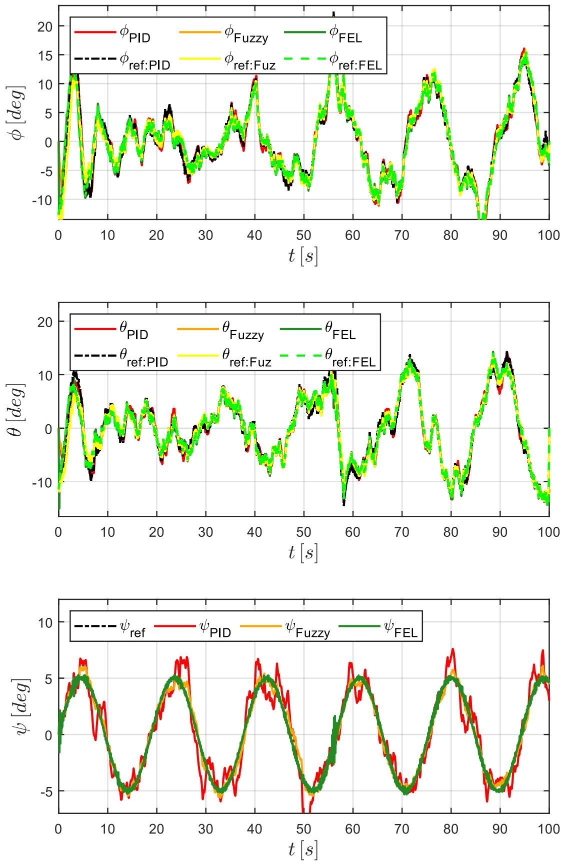

Figure 5. The behavior of the three Euler angles in conical trajectory tracking over time

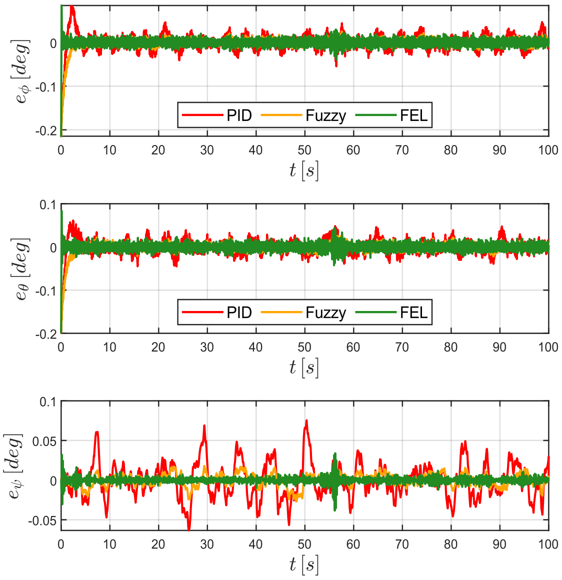

Figure 6. Attitude errors in conical trajectory tracking

Figure 6 shows the attitude control error. The results indicate that the FEL controller performs better.

Figure 7 illustrates the control vector of the quadrotor. The control torques for attitude in the FEL method indicate higher control effort and energy consumption associated with this approach.

Figure 8 compares the RMSE and MAE for the three angles, roll, pitch, and yaw, across the two controllers. The performance of the FEL controller is superior to that of the classical PID controller in these metrics.

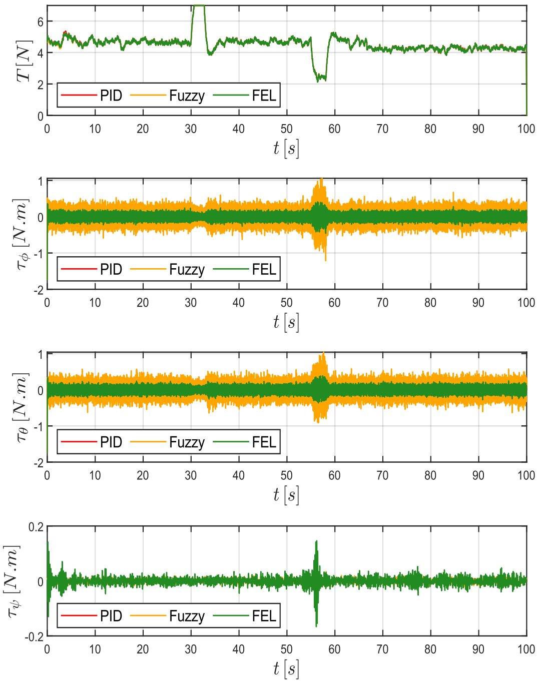

Figure 7. Control vector in conical trajectory tracking of Parrot with PID vs FEL controller.

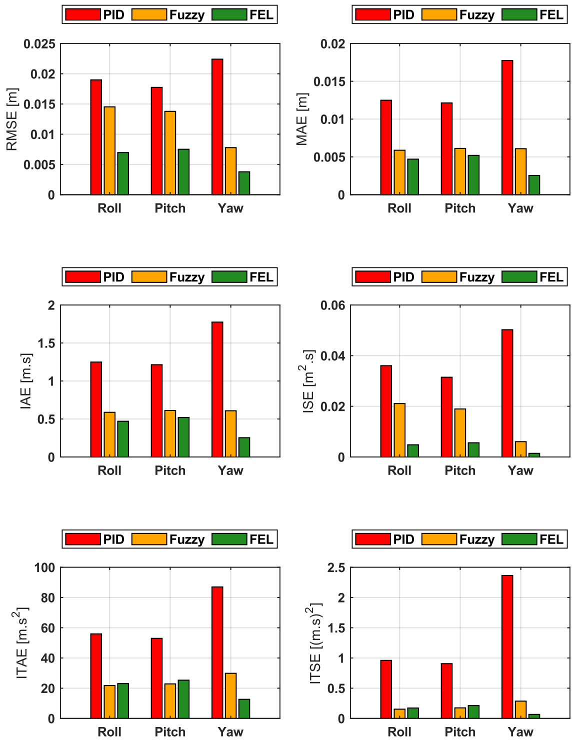

Figure 8. RMSE, MAE, IAE, ISE, ITAE, and ITSE of conical trajectory tracking attitude control

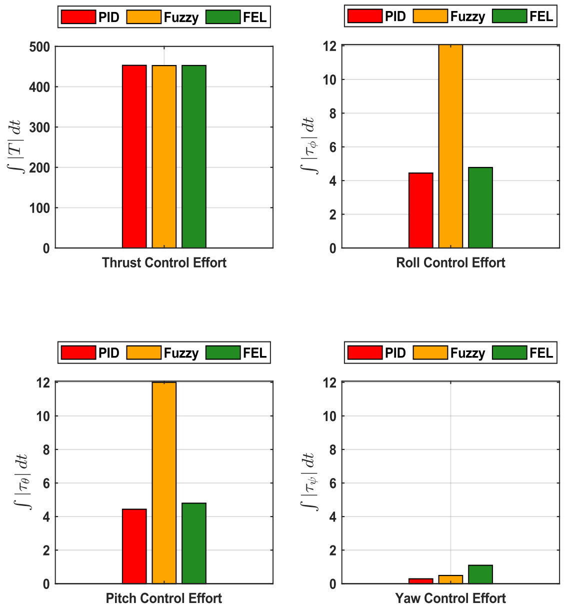

Figure 9. The integral absolute value of the control signal over time during the conical trajectory tracking.

Figure 9 illustrates the total absolute value of the control signal over time during the trajectory tracking, reflecting control effort and energy consumption. The FEL method shows a higher control effort, but this minimal difference compared to the classical method is acceptable and does not undermine the advantages of the FEL approach.

B. Square Trajectory

The square trajectory for the quadrotor is defined by specifying its coordinates in the X and Y dimensions while maintaining a constant altitude in the Z dimension. To ensure smooth transitions between the vertices, linear interpolation is employed, allowing the quadrotor to follow the desired path accurately over a specified time interval.

Figure 10 illustrates the square trajectory tracking using PID and FEL controllers in three-dimensional space within the F_W.

In this section, like the conical trajectory section, Figures 11 to 17 present the results of implementing the proposed control method. Despite the differences in trajectory, the results are consistent with those observed in the previous section.

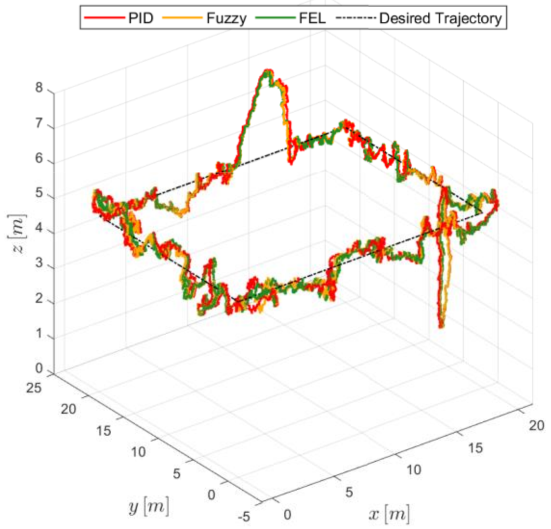

Figure 10. 3D view of square trajectory tracking by Parrot with PID vs FEL controller

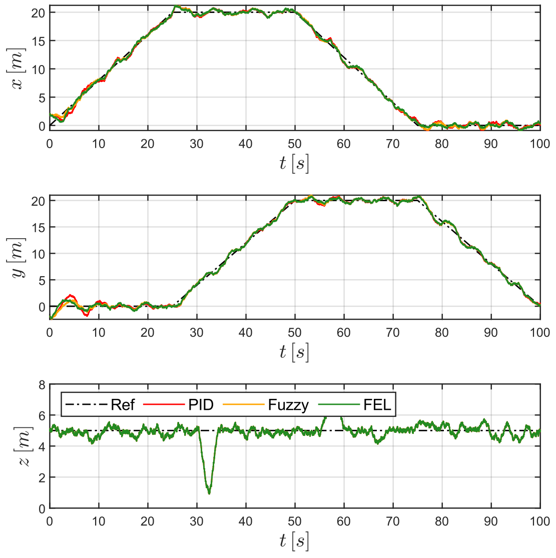

Figure 11. The behavior of the three axes' positions in square trajectory tracking over time.

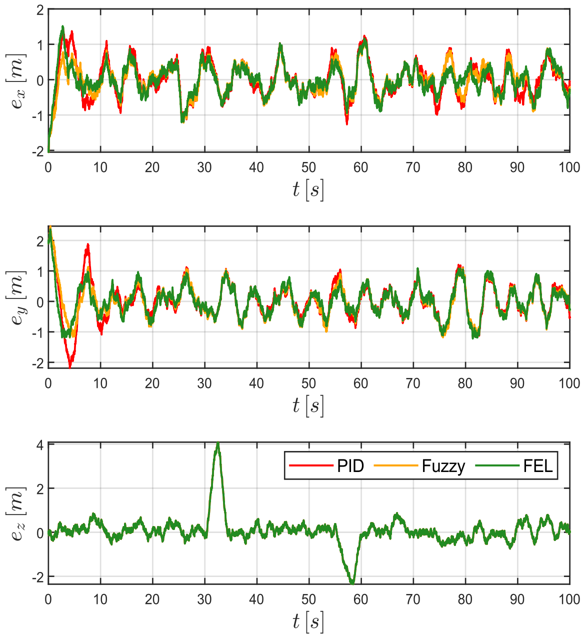

Figure 12. Position errors in square trajectory tracking

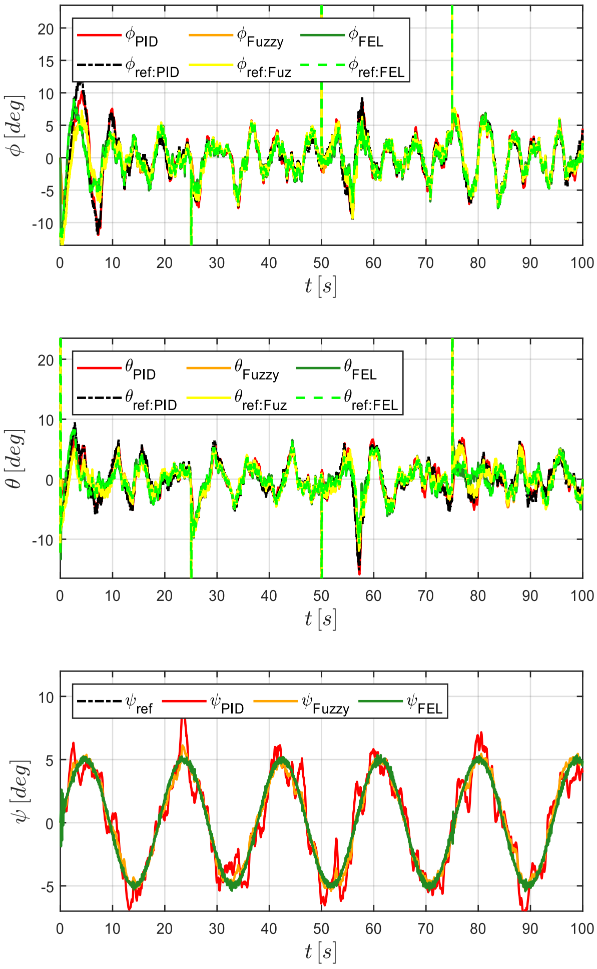

Figure 13. The behavior of the three Euler angles in a square trajectory over time

Table 1 presents the performance comparison of the proposed control method against a fixed-gain PID controller for quadrotor attitude control. Similarly, Table 2 compares the proposed method's performance against a FLS-tuned PID controller. These tables report the average percentage change in performance metrics, detailed in the table columns.

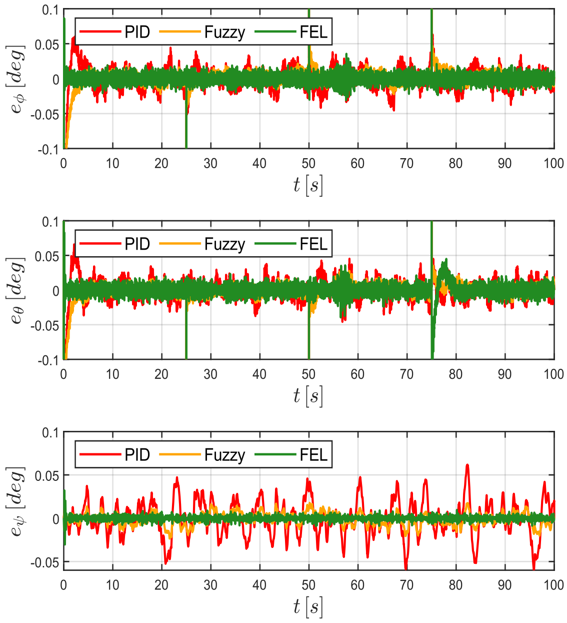

Figure 14. Attitude errors in square trajectory tracking

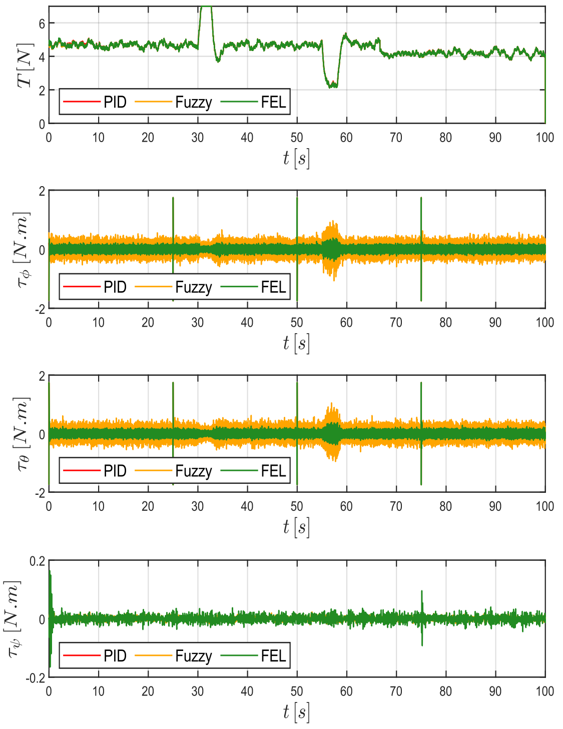

Figure 15. Control vector in square trajectory tracking of Parrot with PID vs FEL controller.

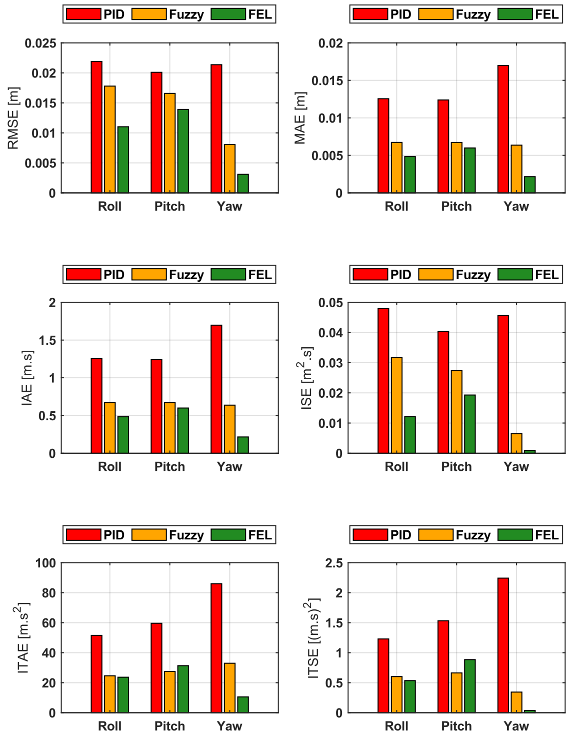

Figure 16. RMSE, MAE, IAE, ISE, ITAE, and ITSE of square trajectory tracking attitude control

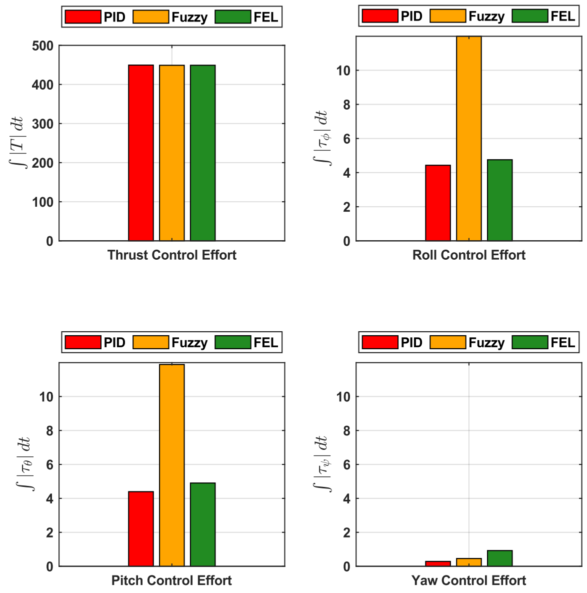

Figure 17. The integral absolute value of the control signal over

Table 1. Average percentage increase or decrease in performance metrics compared to the PID controller for conical, spiral, and square trajectory tracking. Positive values indicate improvement over the PID.

PID:	RMSE	MAE	IAE	ISE	ITAE	ITSE	IAC
Roll	+53.0	+57.3	+57.3	+77.6	+53.5	+71.0	**-7.5**
Pitch	+49.4	+55.9	+55.9	+72.7	+51.6	+63.6	**-9.8**
Yaw	+86.2	+87.0	+87.0	+98.1	+87.5	+98.3	**-238.9**

Table 2. Average percentage increase or decrease in performance metrics compared to the Fuzzy-PID controller for conical, spiral, and square trajectory tracking. Positive values indicate improvement over the Fuzzy-PID.

Fuzzy:	RMSE	MAE	IAE	ISE	ITAE	ITSE	IAC
Roll	+38.1	+16.8	+16.8	+60.9	**-6.3**	**-14.1**	+60.4
Pitch	+36.3	+13.3	+13.3	+57.2	**-11.1**	**-21.6**	+59.5
Yaw	+61.9	+63.5	+63.5	+85.3	+63.7	+86.3	**-107.1**

5 DISCUSSION

In this research, we aimed to develop a robust and adaptive intelligent controller for quadrotors by combining backstepping and FEL techniques. We selected backstepping due to its inherent robustness and the precise mathematical representation of the quadrotor’s positional dynamics. We conducted simulations that included sensor noise, external disturbances, and abrupt weight changes to evaluate the controller's performance under various uncertainties. The findings demonstrated the resilience of the backstepping control method in addressing these challenges. We integrated FEL with traditional PID controllers for the quadrotor's attitude control. Each neural inverse dynamic controller was designed with a single hidden layer containing 100 neurons, enabling online learning without data preprocessing.

As evident from Table 1, the proposed method demonstrates a significant improvement in all trajectory tracking performance indices related to error reduction. However, this improvement comes at the cost of increased control effort, particularly on the yaw channel. This heightened control effort is attributed to the presence of measurement noise, compelling the controller to exert more effort to compensate for perceived error increases. This issue could be mitigated through appropriate measurement noise filtering, employing denoising techniques in neural network training, and defining a cost function that considers the reduction of control effort. Table 2, which presents the average percentage change of metrics relative to the fuzzy PID controller, reveals a similar trend. Notably, for metrics that integrate time into the performance evaluation, such as ITAE and ITSE, the proposed method exhibits slightly inferior performance in the roll and pitch channels compared to fuzzy PID. This could be due to the fuzzy system’s enhanced capability to handle severe uncertainties over time. Conversely, in the roll and pitch channels, the proposed method demonstrates superior performance in reducing control effort compared to the fuzzy PID, likely because the fuzzy system exerts greater effort to counteract uncertainties, especially measurement noise. The comparative results in the yaw channel are consistent with those observed in the PID comparison.

This hybrid strategy significantly improved the trajectory-tracking capabilities of the quadrotor. It is important to note that the FEL method requires increased control effort, leading to higher energy consumption. This study marks a pioneering effort in applying this hybrid approach to quadrotor trajectory tracking. Future work aims to refine the neural network training cost function to minimize control effort, thereby maximizing the benefits of the proposed hybrid method for quadrotor trajectory tracking.

6 CONCLUSION

In conclusion, the fusion of traditional robust control techniques with intelligent online learning frameworks allows for creating a dependable and adaptive system for trajectory tracking in quadrotors. This combined control strategy effectively mitigates uncertainties and improves overall flight performance. Nonetheless, this approach does result in more significant energy usage and increased control demands. With the ongoing advancements in artificial intelligence and deep neural networks, these technologies are set to play a pivotal role in the evolution of hybrid intelligent control strategies within engineering.

Our findings demonstrate that learning-based methodologies can progressively enhance traditional control techniques, while the significance of robust control methods in ensuring system stability and control cannot be overstated.

CONFLICTS OF INTEREST

The authors of this paper declared no conflict of interest regarding the authorship or publication of this article.

REFERENCES

[1] I. Lopez-Sanchez and J. Moreno-Valenzuela, “PID control of quadrotor UAVs: A survey,” *Annu. Rev. Control*, vol. 56, no. June, p. 100900, 2023, https://doi.org/10.1016/j.arcontrol.2023.100900

[2] R. Wang and J. Liu, “Trajectory tracking control of a 6-DOF quadrotor UAV with input

saturation via backstepping," *J. Franklin Inst.*, vol. 355, no. 7, pp. 3288–3309, 2018, https://doi.org/10.1016/j.jfranklin.2018.01.039.

[3] P. V Kokotovic, M. Krstic, and I. Kanellakopoulos, "Backstepping to passivity: recursive design of adaptive systems," in *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, vol.4, 1992, pp. 3276–3280, https://doi.org/10.2514/1.13030.

[4] S. Sastry, *Lyapunov Stability Theory BT - Nonlinear Systems: Analysis, Stability, and Control*, S. Sastry, Ed., New York, NY: Springer New York, 1999, pp. 182–234. https://doi.org/10.1007/978-1-4757-3108-8.

[5] Z.-Y. Zhao, M. Tomizuka, and S. Isaka, "Fuzzy gain scheduling of PID controllers," *IEEE Trans. Syst. Man. Cybern.*, vol. 23, no. 5, pp. 1392–1398, 1993, https://doi.org/10.1109/21.260670.

[6] L. Martins, C. Cardeira, and P. Oliveira, "Linear Quadratic Regulator for Trajectory Tracking of a Quadrotor," *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 176–181, 2019, https://doi.org/10.1016/j.ifacol.2019.11.195.

[7] K. Alexis, G. Nikolakopoulos, and A. Tzes, "On Trajectory Tracking Model Predictive Control of an Unmanned Quadrotor Helicopter Subject to Aerodynamic Disturbances," *Asian J. Control*, vol. 16, no. 1, pp. 209–224, Jan. 2014, https://doi.org/10.1002/asjc.587.

[8] A. Sarabakha and E. Kayacan, "Online Deep Fuzzy Learning for Control of Nonlinear Systems Using Expert Knowledge," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 7, pp. 1492–1503, 2020, https://doi.org/10.1109/TFUZZ.2019.2936787

[9] G. Wen, W. Hao, W. Feng, and K. Gao, "Optimized Backstepping Tracking Control Using Reinforcement Learning for Quadrotor Unmanned Aerial Vehicle System," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 52, no. 8, pp. 5004–5015, 2022, https://doi.org/10.1109/TSMC.2021.3112688

[10] C. Kwan and F. L. Lewis, "Robust backstepping control of nonlinear systems using neural networks," *IEEE Trans. Syst. Man, Cybern. Part ASystems Humans.*, vol. 30, no. 6, pp. 753–766, 2000, https://doi.org/10.1109/3468.895898.

[11] F. Chen, R. Jiang, K. Zhang, B. Jiang, and G. Tao, "Robust Backstepping Sliding-Mode Control and Observer-Based Fault Estimation for a Quadrotor UAV," *IEEE Trans. Ind. Electron.*, vol. 63, no. 8, pp. 5044–5056, 2016, https://doi.org/10.1109/TIE.2016.2552151.

[12] H. Miyamoto, M. Kawato, T. Setoyama, and R. Suzuki, "Feedback-error-learning neural network for trajectory control of a robotic manipulator," *Neural Networks*, vol. 1, no. 3, pp. 251–265, 1988, https://doi.org/10.1016/0893-6080(88)90030-5.

[13] M. Kawato, *Feedback-Error-Learning Neural Network for Supervised Motor Learning*. Elsevier B.V., 1990. https://doi.org/10.1016/B978-0-444-88400-8.50047-9.

[14] H. Gomi and M. Kawato, "Neural network control for a closed-loop System using Feedback-error-learning," *Neural Networks*, vol. 6, no. 7, pp. 933–946, 1993, https://doi.org/10.1016/S0893-6080(09)80004-X.

[15] Z. Hamavand and H. M. Schwartz, "Trajectory Control of Robotic Manipulators by Using a Feedback-Error-Learning Neural Network," *Robotica*, vol. 13, no. 5, pp. 449–459, 1995, https://doi.org/10.1017/S0263574700018282.

[16] M. Teshnehlab and K. Watanabe, "Neural network controller with flexible structure based on feedback-error-learning approach," *J. Intell. Robot. Syst.*, vol. 15, no. 4, pp. 367–387, 1996, https://doi.org/10.1007/BF00437602.

[17] V. D. Kalanovic, D. Popovic, and N. T. Skaug, "Feedback error learning neural network for trans-femoral prosthesis," *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 1, pp. 71–80, 2000, https://doi.org/10.1109/86.830951.

[18] J. Nakanishi and S. Schaal, "Feedback error learning and nonlinear adaptive control," *Neural Networks*, vol. 17, no. 10, pp. 1453–1465, 2004, https://doi.org/10.1016/j.neunet.2004.05.003.

[19] A. Miyamura Ideta, "Stability of feedback error learning method with time delay," *Neurocomputing*, vol. 69, no. 13–15, pp. 1645–1654, 2006, https://doi.org/10.1016/j.neucom.2005.04.011.

[20] X. Ruan, M. Ding, D. Gong, and J. Qiao, "Online adaptive control for inverted pendulum balancing based on feedback-error-learning," *Neurocomputing*, vol. 70, no. 4–6, pp. 770–776, 2007, https://doi.org/ 10.1016/j.neucom.2006.10.012.

[21] A. Taheri, M. A. Shoorehdeli, H. Bahrami, and M. H. Fatehi, "Implementation and control of X-Y pedestal using dual-drive technique and feedback error learning for LEO satellite tracking," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 4, pp. 1646–1657, 2014, https://doi.org/10.1016/j.neucom.2006.10.012

[22] K. Sabahi, S. Ghaemi, and S. Pezeshki, "Application of type-2 fuzzy logic system for load frequency control using feedback error learning approaches," *Appl. Soft Comput. J.*, vol. 21, pp. 1–11, 2014, https://doi.org/10.1109/TCST.2013.2281838.

[23] M. A. Khanesar, E. Kayacan, M. Reyhanoglu,

and O. Kaynak, “Feedback Error Learning Control of Magnetic Satellites Using Type-2 Fuzzy Neural Networks With Elliptic Membership Functions,” *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 858–868, 2015, https://doi.org/10.1109/TCYB.2015.2388758.

[24] E. Kayacan, J. M. Peschel, and G. Chowdhary, “A self-learning disturbance observer for nonlinear systems in feedback-error learning scheme,” *Eng. Appl. Artif. Intell.*, vol. 62, pp. 276–285, 2017, https://doi.org/10.1016/j.engappai.2017.04.013.

[25] M. Asgari and M. Asgari, “Feedback error learning neural network for stable control of nonlinear nonaffine systems,” *Trans. Inst. Meas. Control*, vol. 44, no. 13, pp. 2545–2556, 2022, https://doi.org/10.1177/01423312221086076

[26] J. Figueiredo, P. N. Fernandes, J. C. Moreno, and C. P. Santos, “Feedback-Error Learning for time-effective gait trajectory tracking in wearable exoskeletons,” *Anat. Rec.*, vol. 306, no. 4, pp. 728–740, Apr. 2023, https://doi.org/10.1002/ar.25031

[27] A. Sarabakha and E. Kayacan, “Online deep learning for improved trajectory tracking of unmanned aerial vehicles using expert knowledge,” in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 7727–7733, https://doi.org/10.1109/ICRA.2019.8.794314

[28] H. K. Khalil, “Lyapunov’s Stability Theory, BT - Encyclopedia of Systems and Control,” *J. Baillieul and T. Samad, Eds.*, London: Springer London, 2013, pp. 1–6. https://doi.org/10.1007/978-1-4471-5102-9_77-1.

[29] P. Singh, D. K. Giri, and A. K. Ghosh, “Robust backstepping sliding mode aircraft attitude and altitude control based on adaptive neural network using symmetric BLF,” *Aerosp. Sci. Technol.*, vol. 126, p. 107653, 2022, https://doi.org/10.1016/j.ast.2022.107653.

[30] S. Yu, X. Fan, J. Qi, L. Wan, and B. Liu, “Attitude control of quadrotor UAV based on integral backstepping active disturbance rejection control,” *Trans. Inst. Meas. Control*, vol. 46, no. 4, pp. 703–715, Jul. 2023, https://doi.org/10.1177/01423312231185423.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, https://doi.org/10.1038/323533a0.

HOW TO CITE THIS ARTICLE

E. Baghelani, J. Roshanian, and M Teshnehlab, “Robust Intelligent Trajectory Tracking for Quadrotors: Integrating Backstepping and Feedback-Error-Learning Technique,” *Journal of Aerospace Science and Technology*, Vol. 18, Issue 2, 2025, pp. 61-74.

DOI: 10.22034/jast.2025.475816.1206

URL: https://jast.ias.ir/article_220377.html