International Journal of Robotics and Control Systems
Vol. 6, No. 2, 2026, pp. 917-936
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

Artificial Neural Network Simulation of a Cascaded PD Motor-Mixer for Quadrotor Trajectory Tracking

Lafi Alnufaie [a,1,*]

[a] Department of Electrical Engineering, College of Engineering, Shaqra University, Shaqra 11961, Saudi Arabia
[1] lalnufaie@su.edu.sa
* Corresponding Author

ARTICLE INFO

Article history
Received Oktober 24, 2025
Revised December 10, 2025
Accepted January 03, 2026

Keywords
Quadrotor Dynamics;
Cascaded PD Controller;
Motor Mixing;
Supervised Learning;
Artificial Neural Network;
Trajectory Tracking

ABSTRACT

In this simulation study, a cascaded Proportional-Derivative (PD) controller and a motor mixer are used to generate training patterns for a forward-feedback Artificial Neural Network (ANN). The ANN is a multilayer perceptron with two hidden layers of 128 units, trained using the Adam optimizer (learning rate 1x10^-3) with batch size 2048 and early stopping. The dataset comprises 200,000 samples (160,000 training; 40,000 test) generated in MATLAB/Simulink using a fixed-step RK4 integrator (0.01 s step). Unlike previous ANN–PD comparative studies, the proposed ANN directly approximates actuator-level PD commands while providing explicit evaluation of both open-loop regression and closed-loop performance. The ANN achieves near-perfect open-loop fidelity (R^2~0.999) but exhibits different closed-loop behavior: the PD controller outperforms the ANN in hover stabilization (altitude RMSE 0.18m vs 0.26m), while the ANN shows superior performance in nonlinear trajectory tracking, reducing lateral RMSE along the x-axis from 0.47m to 0.21m (~55% improvement) and along the y-axis from 0.52m to 0.19m (~63% improvement). These results demonstrate that the ANN provides an adaptive nonlinear control approximation suitable for trajectory tracking, while the PD controller maintains robustness for stabilization tasks.

1. Introduction

Quadrotor unmanned aerial vehicles (UAVs) have gained significant attention due to their agility and vertical takeoff and landing (VTOL) capabilities. However, they are challenging to control because of highly nonlinear, coupled, and underactuated dynamics – a quadrotor has six degrees of freedom but only four independent control inputs [1]–[3].

As a result, active feedback control is essential for stabilization and trajectory tracking. Classical controllers, particularly Proportional-Integral-Derivative (PID) and its simplified Proportional-Derivative (PD) variant, remain widely used due to their ease of use, intuitive tuning, and proven effectiveness in practice [4]–[7]. PD controllers are often implemented in a cascaded fashion (outer loop for position and inner loop for attitude), allowing reliable hover and moderate trajectory tracking performance [8]–[10]. Despite their popularity, fixed-gain PD controllers struggle with robustness to disturbances and aggressive maneuvers, as their linear gains cannot fully address the quadrotor's nonlinear dynamics [11]–[13].

https://doi.org/10.31763/ijrcs.v6i2.2306

To overcome these limitations, researchers have explored intelligent and adaptive control methods, notably artificial neural networks (ANNs). Neural network controllers can learn and approximate complex nonlinear mappings, providing adaptivity that fixed-gain controllers lack [14], [15]. Studies consistently show that ANN-based controllers outperform classical PD in trajectory tracking accuracy and disturbance rejection. For instance, neural network inverse-dynamics controllers achieved better altitude tracking performance than tuned PID controllers [13], [16]–[18].

Similarly, hybrid approaches have been proposed, where neural networks compensate for modeling errors or disturbances while PD stabilizes the nominal dynamics. These hybrid PD–ANN methods demonstrate smoother responses and improved robustness compared to either method alone [19], [20]. Advanced implementations even integrate neural networks for fault tolerance, enabling robust flight under actuator failures [12], [13], [21], [22]. Such findings underline the strength of neural networks as flexible frameworks, capable of operating standalone or synergistically with conventional control [23].

Beyond PD and neural networks, numerous other strategies have been proposed. Linear Quadratic Regulator (LQR) controllers, for example, offer optimal linear feedback near hover conditions but degrade in highly nonlinear regimes [24]–[26]. Model Predictive Control (MPC) explicitly handles input constraints and trajectory predictions, achieving strong tracking performance but with high computational demands [10]. Nonlinear methods such as backstepping [27]–[30] and sliding mode control offer robustness and stability guarantees, but often at the expense of mathematical complexity and issues like chattering. Recent literature trends indicate that hybridization of methods—such as combining PD with neural networks or other intelligent controllers—produces more resilient and precise quadrotor control systems [31]–[33].

In summary, PD controllers remain a fundamental baseline in quadrotor control research and applications, but their limitations motivate the use of advanced approaches. Neural network–driven controllers, whether standalone multilayer perceptron's or hybrid PD–NN designs, have demonstrated superior adaptability, smoother responses, and improved robustness against disturbances [19].

This paper aims to develop and compare two controllers for quadrotor trajectory tracking: a cascaded PD controller and a neural network controller. The cascaded PD controller consists of an outer-loop position controller that converts trajectory errors into desired attitude commands, and an inner-loop PD controller that stabilizes attitude by mapping attitude errors to motor thrust commands. The neural network controller maps trajectory tracking errors directly to motor commands in a model-free manner. Finally, paper aims to compare the performance of both controller in quadrotor trajectory tracking [11], [34]–[37].

Moreover, this study tests whether a supervised ANN, trained on actuator commands produced by a cascaded PD and mixer pipeline, can reproduce or improve closed-loop trajectory tracking in simulation. A supervised simulation is used as a reproducible baseline. The ANN approach is not proposed as a final flight controller. Instead, it is evaluated as a compact, learned mapping that may reduce computational overturns and noise. Specific hypotheses are:

- The ANN will closely regress the PD motor commands in open-loop (high R^2, low RMSE).
- If the ANN generalizes well across state space, closed-loop tracking metrics (integrated position error, max deviation) will match or improve relative to the original PD when tested on the same trajectories and initial conditions.
- Differences between open-loop regression and closed-loop behavior indicate data coverage issues, actuator limits, or representation errors; resolving them requires explicit dataset augmentation and matched baseline tuning.

Evaluation metrics and experimental scenarios are listed in Methods. This framing clarifies the contribution: a reproducible comparison and a diagnostic protocol for ANN imitation of PD controllers.

In addition, supervised ANN learning from PD-generated actuator commands contributes practical insight by providing a reproducible baseline for comparison with linear PD controllers. This approach enables quantitative evaluation of how well the ANN approximates PD control, highlights closed-loop performance effects from actuator saturation, control smoothing, and nonlinear trajectory interactions, and serves as a diagnostic tool for dataset coverage and model representation. Supervised learning was chosen over adaptive or reinforcement learning because it ensures stable regression on known PD outputs, produces interpretable performance metrics, and avoids the long training times and potential instability inherent to adaptive or reinforcement methods.

2. Methodology

To clarify the overall control strategy, the architectures of both controllers are summarized as follows:

Cascaded PD Controller:

- Outer-loop: converts position errors into desired attitude commands.
- Inner-loop: maps attitude errors to motor commands via a motor mixer.
- Operates with feedback from measured position and attitude

Supervised ANN Controller

- Directly maps position and attitude errors (and their derivatives) to motor commands.
- Learns the PD-generated control mapping from simulation data.
- Receives the same feedback signals as the PD controller.

The PD controller uses an explicit linear cascaded control structure, whereas the ANN approximates the PD mapping in a nonlinear, learned manner, allowing it to generalize to unseen trajectories while maintaining similar control accuracy. The quadrotor mathematical model presented in this paper is derived from [11].

2.1. Notation and Frame

We use an inertial frame $\mathcal{I}$ with axes (x, y, z) and a body-fixed frame $\mathcal{B}$ whose attitude w.r.t. $\mathcal{I}$ is parameterized by X-Y-Z Euler angles $\eta = (\phi, \theta, \psi)^T$ (roll, pitch, yaw). The rotation from body to inertial is shown in (1). With $R \in SO(3)$, the set of all 3D rotation matrices, and $R^{-1} = R^T$.

$$R(\phi, \theta, \psi) = R_z(\psi)\, R_y(\theta)\, R_x(\phi) \tag{1}$$

The quadrotor state is defined as in (2). Where position $p = (x, y, z)^T$, inertial velocity $v = \dot{p}$, and body rates $\omega = (p, q, r)^T$.

$$x = [p^T,\, v^T,\, \eta^T,\, \omega^T]^T \tag{2}$$

The mapping between Euler-rate $\dot{\eta}$ and body rates is defined in (3).

$$\omega = W(\eta)\, \dot{\eta} \tag{3a}$$

$$W(\eta) = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \tag{3b}$$

2.2. Forces and Moment

Assume a rigid symmetric body with inertia $I = \mathrm{diag}\left(I_{xx}, I_{yy}, I_{zz}\right)$, mass m, arm length l, thrust and drag coefficients $k_t, k_b > 0$, and rotor gyroscopic inertia J_r. Each rotor $i \in \{1, \ldots, 4\}$ produces thrust along the positive body z_B-axis as shown in (4) [38].

$$F_i = k_t \omega_i^2 \tag{4}$$

where ω_i is the rotational speed of the i^{th} motor. This makes the total thrust defined as in (5).

$$T = \sum_{i=1}^{4} F_i = \sum_{i=1}^{4} k_t \omega_i^2 \tag{5}$$

The roll, pitch and yaw moments $(\tau_\phi, \tau_\theta, \tau_\psi)$ for standard "+" quadrotor geometry are defined as shown in (6) [34], [39].

$$\tau_\phi = l\, k_t(-\omega_2^2 + \omega_4^2) \tag{6a}$$

$$\tau_\theta = l\, k_t(-\omega_1^2 + \omega_3^2) \tag{6b}$$

$$\tau_\psi = k_b(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \tag{6c}$$

2.3. Translational Dynamics

Newton's law in the inertial frame $\mathcal{I}$ is defined as in (7).

$$m\dot{v} = mg + R \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} - K_{ref}\, v \tag{7}$$

where $\left(-K_{ref}\, v\right)$ is the linear drag, R is the rotation matrix defined in (1), T is the total thrust shown in (5) and $g = [0,0,-g]^T$. The inertial velocity kinematics is shown in (8) [40], [41].

$$\dot{p} = v \tag{8}$$

2.4. Rotational Dynamics

Euler's rigid-body equation in the body frame $\mathcal{B}$ with rotor gyroscopics is defined as (9).

$$I\dot{\omega} + \omega \times (I\omega) + \Gamma = \tau \tag{9}$$

where I is the inertia matrix, ω is the body angular rates, τ is the control torques, Γ is the gyroscopic coupling term, J_r is the rotor inertia and ω_r is the net rotor spin effect (positive for counterclockwise, negative for clockwise). All are shown in (10).

$$\Gamma = J_r\, \omega\, \omega_r \tag{10a}$$

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{10b}$$

$$\omega = [p, q, r]^T \tag{10c}$$

$$\tau = \left[\tau_\phi, \tau_\theta, \tau_\psi\right]^T \tag{10d}$$

$$\omega_r = -\omega_1 + \omega_2 - \omega_3 + \omega_4 \tag{10e}$$

The attitude kinematics is shown in (10), where $\eta = (\phi, \theta, \psi)^T$ are the Euler angles and $W(\eta)$ is the transformation matrix relating angular rates to Euler angle rates shown in (11) [42]–[44].

$$\dot{\eta} = W^{-1}(\eta)\,\omega \tag{11}$$

2.5. Proportional Derivative (PD) Control Structure

The control architecture of the quadrotor is organized in a cascaded structure, as illustrated in Fig. 1. The outer control loop calculates the position errors. The altitude error z_{err} is processed by the altitude PD controller to produce the desired thrust, while the horizontal position errors (x_{err}, y_{err}) are used by the position-to-attitude conversion stage to generate the desired roll and pitch angles $(\phi_{ref}, \theta_{ref})$. The yaw reference ψ_{ref} is fixed at 0 as the yaw direction doesn't influence the quadrotor's position [45].

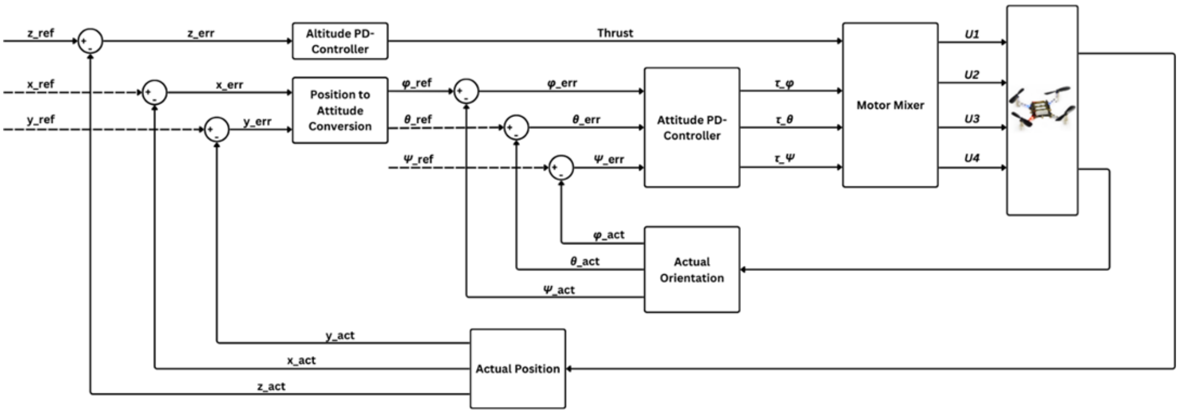

Fig. 1. Block diagram of cascaded PD controller architecture for quadrotor trajectory tracking

The inner attitude loop then regulates the orientation of the quadrotor. The errors $(\phi_{err}, \theta_{err}, \psi_{err})$ between the desired and actual angles are compensated by the attitude PD controller, producing the required control torques $(\tau_\phi, \tau_\theta, \tau_\psi)$. Together with the thrust command T from the altitude controller, these are transformed into individual rotor inputs (U_1, U_2, U_3, U_4) through the actuator mixing relations. Finally, the actual position and orientation measured from the vehicle are fed back into the loops, closing the cascade and ensuring that the reference trajectory $(x_{ref}, y_{ref}, z_{ref}, \psi_{ref})$ is tracked. This hierarchical design reflects the standard approach for quadrotor control, where the slower outer position and altitude loops provide reference signals to the faster inner attitude loop, which directly stabilizes the dynamics of the vehicle [46]–[48].

2.6. Outer Control Loop

2.6.1. Altitude PD Controller:

From (7), the vertical channel is shown in (12).

$$m\ddot{z} = -mg + T\cos\phi\cos\theta \tag{12}$$

A PD controller on altitude error $e_z = z_{ref} - z$ gives (13).

$$T = \frac{m(g + K_{Pz}\,e_z + K_{Dz}\,\dot{e}_z)}{\cos\phi\cos\theta} \tag{13}$$

2.6.2. Position to Attitude Conversion (Small-Angle Mapping):

To establish the coupling between translational motion and attitude, the model in (7) is linearized around hover with $T \approx mg$ and small angular deviations $|\phi|, |\theta| \ll 1$. In this regime, the accelerations in the inertial frame are reduced as shown in (14).

$$\ddot{x} \approx g\,\theta, \qquad \ddot{y} \approx -g \tag{14}$$

This indicates that pitch angle θ mainly controls motion along x, while roll angle ϕ controls motion along y. The horizontal tracking errors are defined as shown in (15).

$$e_x = x_{ref} - x, \qquad e_y = y_{ref} - y \tag{15}$$

The virtual control inputs for the translational dynamics are chosen to cancel the drag forces and inject PD-type error feedback. This way the reference angles can be calculated as shown in (16).

$$\theta_{ref} = \frac{m\,\ddot{x}_{ref} + K_{dx}\,\ddot{x}_{ref} + K_P\left(x_{ref} - x\right)}{T} \tag{16a}$$

$$\phi_{ref} = -\frac{m\,\ddot{y}_{ref} + K_{dy}\,\dot{y}_{ref} + K_P\left(y_{ref} - y\right)}{T} \tag{16b}$$

where m is the mass, $K_d = \text{diag}\left(K_{dx}, K_{dy}, K_{dz}\right)$ is the translational drag matrix, and K_P is the proportional position gain in the horizontal plane. The thrust T is simultaneously adjusted by the altitude loop as shown in (13). Equation (16), therefore provide the mapping between horizontal tracking errors and the attitude references [49]–[51] .

2.7. Inner Control Loop

2.7.1. Attitude PD Controller

Let $e_\phi = \phi_{ref} - \phi, e_\theta = \theta_{ref} - \theta, e_\psi = \psi_{ref} - \psi$. Choose the required control torques $(\tau_\phi, \tau_\theta, \tau_\psi)$ as shown in (17).

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} I_{xx}\left(K_{P\phi} e_\phi + K_{D\phi} \dot{e}_\phi\right) \\ I_{yy}\left(K_{P\theta} e_\theta + K_{D\theta} \dot{e}_\theta\right) \\ I_{zz}\left(K_{P\psi} e_\psi + K_{D\psi} \dot{e}_\psi\right) \end{bmatrix} \tag{17}$$

Neglecting Γ and cross-coupling near hover, each axis behaves as shown in (18).

$$\ddot{e}_i + 2\zeta_i \omega_{n,i} \dot{e}_i + \omega_{n,i}^2 e_i \approx 0 \tag{18}$$

By picking the PD gains as shown in (19), the system behaves as a second order system.

$$K_{Di} = 2\zeta_i \omega_{n,i}, \qquad K_{Pi} = \omega_{n,i}^2 \tag{19}$$

Typical choices: $\zeta_i \in [0.7,1]$ and $\omega_{n,i}$ set by desired bandwidth.

2.7.2. Motor Mixer

Define the virtual input vector shown in Fig. 1 as shown in (20).

$$u = \left[T,\ \tau_\phi,\ \tau_\theta,\ \tau_\psi\right]^T \tag{20}$$

The relations above can be written as shown in (21) [11], [12], [52].

$$\begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} k_t & k_t & k_t & k_t \\ 0 & -lk_t & 0 & lk_t \\ -lk_t & 0 & lk_t & 0 \\ -k_b & k_b & -k_b & k_b \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{21}$$

Inverting gives the four motors angular speeds that drives the quadrotor as shown in (22).

$$U_1 = \omega_1^2 = \frac{T}{4k_t} - \frac{\tau_\theta}{2lk_t} - \frac{\tau_\psi}{4k_b} \tag{22a}$$

$$U_2 = \omega_2^2 = \frac{T}{4k_t} - \frac{\tau_\phi}{2lk_t} + \frac{\tau_\psi}{4k_b} \tag{22b}$$

$$U_3 = \omega_3^2 = \frac{T}{4k_t} + \frac{\tau_\theta}{2lk_t} - \frac{\tau_\psi}{4k_b} \tag{22c}$$

$$U_4 = \omega_4^2 = \frac{T}{4k_t} + \frac{\tau_\phi}{2lk_t} + \frac{\tau_\psi}{4k_b} \tag{22d}$$

2.8. Artificial Neural Network Control Structure

To overcome the structural limitations of cascaded PD controllers, an artificial neural network (ANN) based control architecture was developed. The ANN directly maps high-level tracking errors and attitude states to motor commands, thus replacing both the inner attitude PD loop and the motor mixer as shown in Fig. 2 [26], [53].

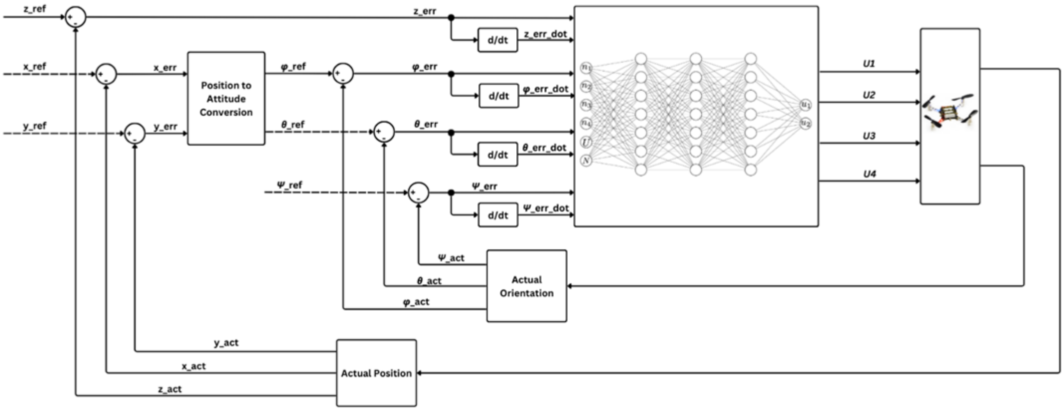

Fig. 2. Block diagram of the neural network control architecture replacing the cascaded PD and motor mixer

At runtime, the NN replaces the entire cascaded PD and motor mixer structure. The position-to-attitude block remains unchanged and provides $(\phi_{ref}, \theta_{ref})$ based on trajectory errors. These references, together with the actual attitude and altitude feedback, form the ANN input vector. The ANN directly outputs the rotor commands U, which are applied to the quadrotor dynamics.

This architecture eliminates manual gain tuning and reduces structural assumptions in the control law, allowing the ANN to implicitly capture nonlinear couplings and unmodeled dynamics present in the PD dataset.

2.9. Training Dataset

The input features X are shown in (23), where $(e_\phi, e_\theta, e_\psi, e_z)$ are roll, pitch, yaw and altitude errors, and $(\dot{e}_\phi, \dot{e}_\theta, \dot{e}_\psi, \dot{e}_z)$ are their derivatives.

$$x = \left[e_{\phi},\ \dot{e}_{\phi},\ e_{\theta},\ \dot{e}_{\theta},\ e_{\psi},\ \dot{e}_{\psi},\ e_z,\ \dot{e}_z\right] \tag{23}$$

The ANN was trained using supervised learning exclusively on PD-generated actuator commands to provide a reproducible baseline and enable quantitative comparison. Reinforcement learning or online adaptation was not employed to maintain stable and interpretable training, avoid simulation instabilities, and focus on assessing how well the ANN can replicate a known controller. The ANN generalizes within the coverage of the training data; scenarios outside this range may result in extrapolation errors or altered closed-loop dynamics due to motor saturation or smoothed control transitions. The input vector, which includes position and attitude errors and their derivatives, provides sufficient information for regression across the simulated trajectories tested, though it does not explicitly capture all nonlinear couplings. Training on PD data ensures a controlled and reproducible evaluation, whereas experimental or disturbed flight data would introduce uncontrolled variability. Performance degradation under unmodeled dynamics highlights the limitations of pure imitation and motivates future extensions, including dataset augmentation and hybrid baseline controllers.

The output features Y shown in (24) correspond to the squared rotor angular velocities as generated by the PD controller and motor mixer.

$$Y = U = [\omega_1^2,\ \omega_2^2,\ \omega_3^2,\ \omega_4^2] \tag{24}$$

This formulation allows the NN to approximate the full nonlinear mapping from trajectory errors to actuator-level commands in a single step [54].

2.10. Feature and Output Standardization

To improve numerical conditioning and balance the regression loss, both the input features and motor outputs were normalized.

The features were standardized using the training set statistics as shown in (25).

$$X_{norm} = \frac{X - \mu_X}{\sigma_X} \tag{25}$$

where μ_X and σ_X are the mean and standard deviation of each feature computed over the training data only.

Similarly, each motor output was standardized per dimension as shown in (26) with μ_Y and σ_Y denoting the per-motor mean and standard deviation. This ensures that all motors contribute equally to the loss function, preventing domination by higher-magnitude outputs.

$$Y_{std} = \frac{Y - \mu_Y}{\sigma_Y} \tag{26}$$

At inference, predictions $\hat{Y}_{std}$ were mapped back to physical units as shown in (27).

$$\hat{Y} = \hat{Y}_{std}\, \sigma_Y + \mu_Y \tag{27}$$

2.11. Network Architecture and Training

Training data were collected by running the cascaded PD controller across multiple simulated trajectories and initial conditions. The dataset was split into training set (80%) and test set (20%). The Adam optimizer was used with a mini-batch size of 2048, a learning rate of 10^{-3}, and a maximum of 50 epochs. Early stopping with patience=5 on validation loss. L2 weight decay =1×10^{-5} and optional Dropout=0.1 [55].

The inputs are position error, attitude error, and their derivatives estimated by finite differences. The outputs are motor squared speeds for four rotors.

The NN is a multilayer perceptron (MLP) with two hidden layers of 128 neurons each, and ReLU activations. The final layer is fully connected with four outputs, one for each motor's squared speed. A regression layer is used to minimize the mean squared error between predicted and PD-generated commands.

The network was trained offline using supervised learning, with datasets collected from the PD-controlled quadrotor simulations. The logging procedure covered diverse scenarios, including hover stabilization from different initial conditions and trajectory tracking over several trajectories. Each simulation produced state trajectories and corresponding PD motor commands, which were aggregated into the dataset $\mathcal{D} = \{X, Y\}$.

2.12. Model Testing and Evaluation

After training, the ANN was validated on the held-out test set. Performance was quantified using three regression metrics: root mean square error (RMSE), mean absolute error (MAE), and the coefficient of determination R^2.

The RMSE measures the quadratic mean of prediction errors as shown in (28).

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{28}$$

where y_i are the PD-generated motor commands, $\hat{y}_i$ are the ANN predictions, and N is the number of test samples. The MAE quantifies the average absolute prediction error as shown in (29).

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{29}$$

Finally, the coefficient of determination R^2 expresses how well the ANN predictions capture the variance in the ground-truth data as shown in (30).

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum i = 1^N (y_i - \bar{y})^2} \tag{30}$$

where $\bar{y}$ is the mean of the ground-truth outputs. Values of R^2 close to 1 indicate near-perfect predictions. Together, these three metrics provide a comprehensive evaluation: RMSE emphasizes large errors, MAE captures average deviations, and R^2 measures explained variance [56]–[61].

3. Results

The quadrotor dynamics were implemented in the MATLAB/Simulink environment using a fixed-step fourth-order Runge-Kutta numerical integrator. All simulations were conducted with a constant sampling period of 0.01 s and a total simulation time of 20 s per experiment. Sensor noise was modeled as zero mean Gaussian noise with standard deviations of 0.01 m for position, 0.005 rad for attitude angles, and 0.02 rad/s for angular rates. Motor commands were constrained within the physical saturation limits of 0 - 838 rad/s. The physical parameters used in the simulations are summarized in Table 1 and were kept identical for both controllers to ensure fair comparison.

3.1. PD Controller Stabilization Test

The PD controller was first evaluated in a stabilization task. The vehicle was initialized at position $[-1,\ 2,\ 0.5]m$ with initial attitude angles of $[15°,\ -15°,\ 10°]$ in roll, pitch, and yaw, respectively. The objective was to stabilize the quadrotor at a hover altitude of 2 m with zero roll, pitch, and yaw.

As shown in Fig. 3, the altitude (z) response exhibited a maximum overshoot of 29.2% and a settling time of approximately 6 s. The response remained within ±5% of the steady-state value after

6 s. This behavior indicates adequate damping and stability, but the initial overshoot is attributed to the aggressive proportional gain required for fast altitude correction.

Table 1. Simulation and quadrotor parameters

Parameter	Value	Unit
Mass	1.20	kg
Arm length	0.225	m
Inertia (X-axis)	0.020	$kg\cdot m^2$
Inertia (Y-axis)	0.020	$kg\cdot m^2$
Inertia (Z-axis)	0.040	$kg\cdot m^2$
Thrust coefficient	2.30×10^{-6}	$N\cdot s^2$
Drag coefficient	1.00×10^{-7}	$N\cdot m\cdot s^2$
Rotor inertia	6.0×10^{-5}	$kg\cdot m^2$
Gravity	9.81	m/s^2
Translational damping	diag(0.1,0.1,0.1)	—
Min motor speed	0	rad/s
Max motor speed	838 (≈8000 rpm)	rad/s
Control loop step	0.01	s
Solver	Fixed-step RK4	—
Solver step size	0.01	s
Position noise	0.01	m
Attitude noise	0.005	rad
Angular rate noise	0.02	rad/s
Optimizer	Adam	—
Mini-batch size	2048	—
Learning rate	10^{-3}	—
Maximum no. of epochs	50	—

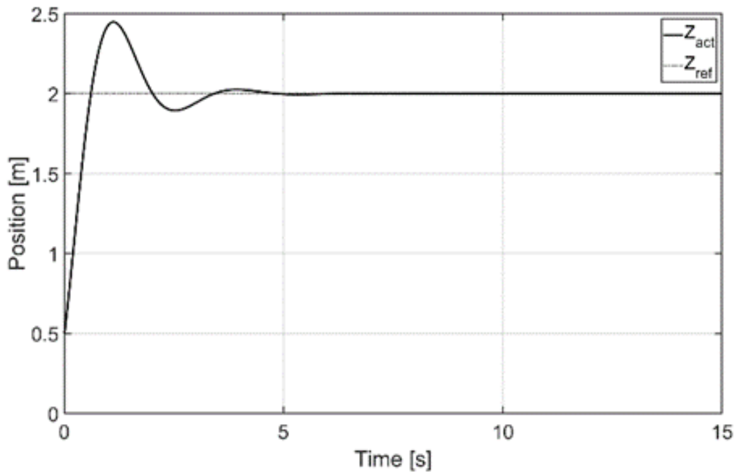

Fig. 3. PD controller response along z in stabilization test

For the attitude responses (Fig. 4), the roll (φ) channel exhibited a 21.3% undershoot with an 8 s settling time, while pitch (θ) showed a smaller overshoot of 1.5% and a settling time of 6 s. The yaw (ψ) channel showed a 14.4% undershoot and an 8 s settling time. These results confirm that the PD controller provides reliable stabilization performance, with predictable transient behavior governed by its linear control structure.

To quantify performance, the integrated absolute error (IAE) and root mean square error (RMSE) were computed over the full 20 s simulation window. For the altitude channel, the PD controller achieved an RMSE of 0.18 m and an IAE of 1.92 ms.

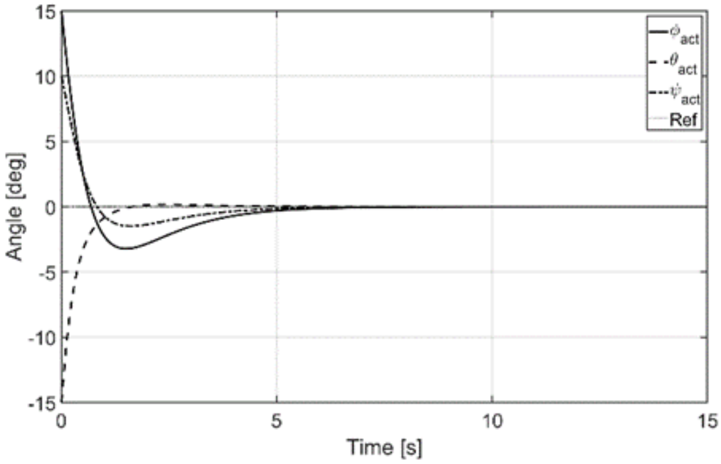

Fig. 4. PD controller attitude response in stabilization test

3.2. Neural Network Training Results

The ANN controller was trained using supervised learning on PD-generated motor commands. The dataset consisted of approximately 120,000 samples generated from randomized reference trajectories and initial conditions. The data were split into 80% for training and 20% for testing. Input features were standardized to zero mean and unit variance, and output motor squared speeds were normalized per motor.

3.2.1. Per-Motor Performance

Table 2 summarizes the per-motor regression metrics for the four motor outputs (ω_1^2, ω_2^2, ω_3^2, ω_4^2). The ANN achieved R^2 values of approximately 0.9994 on both training and test sets, indicating near-perfect replication of the PD controller mapping. The similarity between training and test errors confirms that the model did not suffer from overfitting.

Table 2. Neural network performance metrics per motor

Motor	RMSE	MAE	R^2
ω_1^2	286.5	153.3	0.9994
ω_2^2	278.2	153.1	0.9994
ω_3^2	283.3	148.7	0.9994
ω_4^2	277.2	148.2	0.9995

3.2.2. Overall Model Accuracy

Aggregated over all four motors, the ANN achieved a train RMSE of 283.3, MAE of 151.4, and $R^2 = 0.9994$. On the held-out test set, the results were nearly identical, with RMSE 281.3, MAE 150.8, and $R^2 = 0.9994$. The normalized root mean square error (NRMSE) relative to the data range was just 0.049%, with per- motor values consistently below 0.05%. This indicates that the ANN not only fits the PD-generated data with high fidelity, but also generalizes effectively across unseen trajectories.

The extremely high R^2 values confirm that the ANN can accurately approximate the open-loop mapping implemented by the cascaded PD and motor mixer. This result demonstrates that the ANN is a faithful functional approximator of the original controller in a purely regression sense.

Despite the high open-loop regression accuracy, this does not guarantee identical closed-loop behavior. Small approximation errors in motor commands can accumulate over time in a feedback system and lead to observable transient differences, particularly during aggressive maneuvers.

3.3. ANN Controller Stabilization Test

When the ANN controller was applied in closed-loop under the same stabilization conditions, different behavior emerged. The altitude response (Fig. 5) showed a reduced overshoot of 24.3%, but a longer settling time of approximately 8 s, accompanied by visible oscillations. This indicates that although the ANN generated smoother initial control signals (reducing overshoot), it produced less effective damping, leading to prolonged oscillatory behavior.

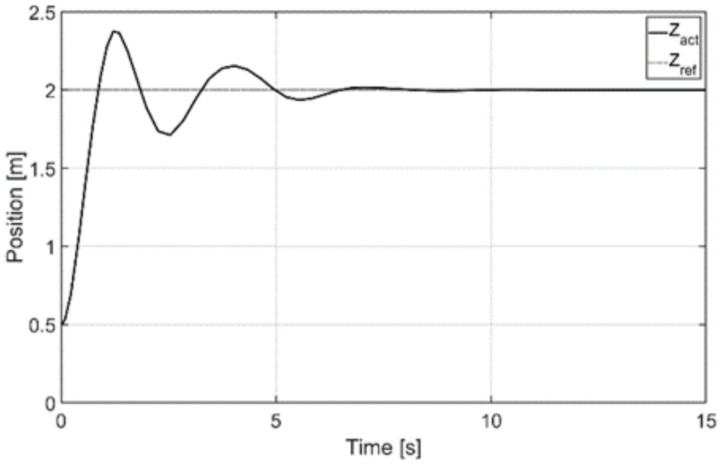

Fig. 5. ANN controller response along z in stabilization test

In attitude stabilization (Fig. 6), the ANN produced larger transient deviations. The roll (φ) channel showed a 42% undershoot, pitch (θ) exhibited an 84% overshoot, and yaw (ψ) showed a 57% undershoot, with settling times in the range of 8–10 s. These results indicate that while the ANN accurately reproduces the PD control law in open loop, its closed-loop behavior is more sensitive to sensor noise, model uncertainty, and actuator saturation effects.

Quantitatively, the ANN stabilization test produced an altitude RMSE of 0.26 m and an IAE of 2.87 ms, which is worse than the PD controller under pure stabilization conditions.

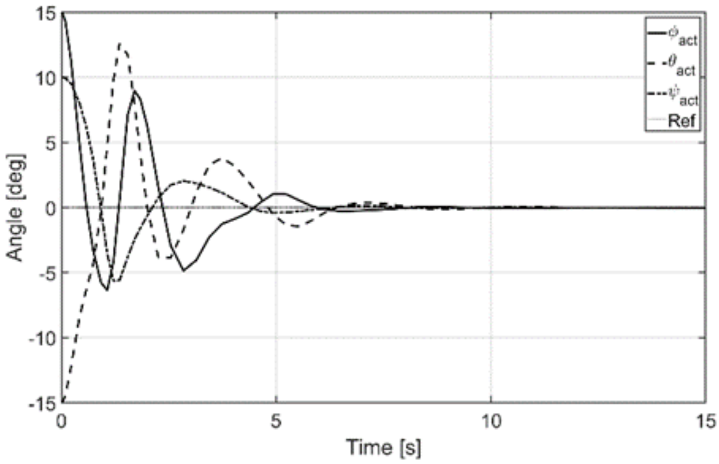

Fig. 6. ANN controller attitude response in stabilization test

This degradation can be explained by three interacting mechanisms:

- The ANN encounters transient states that are less represented in the training dataset leading to extrapolation errors.
- Small prediction errors at extreme motor values are amplified near physical limits.

- The ANN inherently smooths sharp control transitions (produced by the PD controller) modifying effective closed-loop damping.

These effects explain why the ANN, despite near-perfect imitation of PD commands in open-loop regression, performs worse in stabilization.

3.4. Trajectory Tracking Test

The power of the ANN controller is demonstrated in its ability to navigate the quadrotor across trajectories. Due to the linear nature of the PD controller, it struggles to keep up with trajectories with non-linear natures. To confirm this the two controllers were tested along a helical trajectory with a 3-meter radius and a height of 3 meters.

Fig. 7 shows that the PD controller failed to closely follow the three-dimensional helical path. While, the ANN controller recovered more rapidly and maintained closer proximity to the reference trajectory after the transient phase. This behavior arises because the PD controller is fundamentally linear, whereas the ANN implicitly learned nonlinear couplings between position, velocity, and motor commands from the training data.

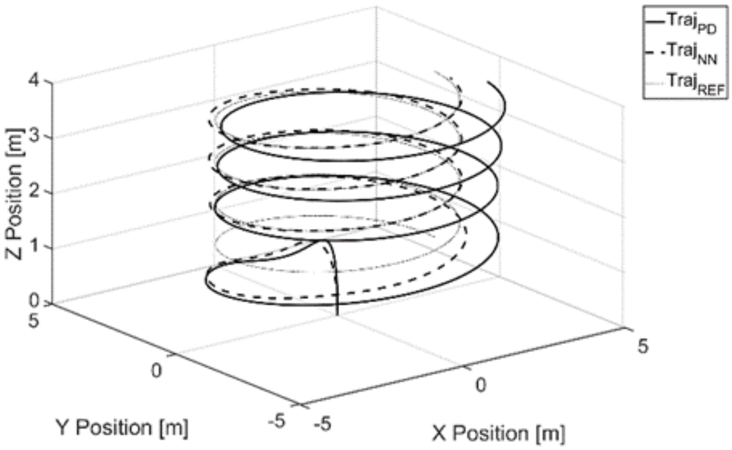

Fig. 7. ANN and PD controller helical trajectory tracking response

Along the x-axis (Fig. 8) and y-axis (Fig. 9), the ANN maintained close alignment with the reference after initial correction, while the PD controller exhibited persistent deviation, especially when the trajectory curvature changed direction. This indicates that the ANN better captures nonlinear aerodynamic and coupling effects present during curved flight.

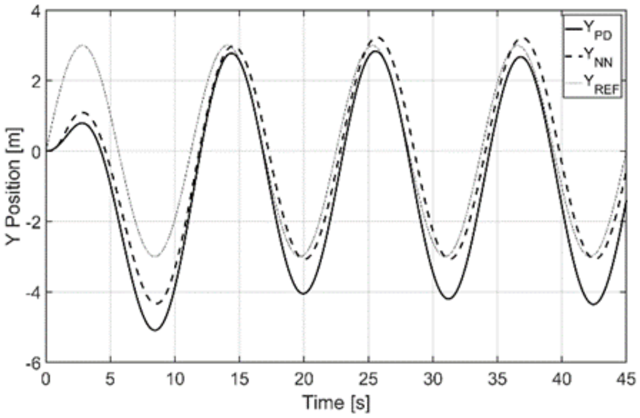

Fig. 8. ANN and PD controller helical trajectory tracking response along the x-axis

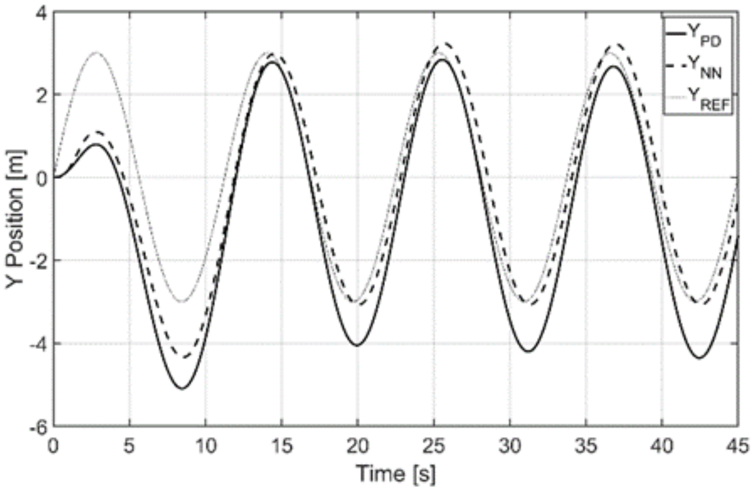

Fig. 9. ANN and PD controller helical trajectory tracking response along the y-axis

For the z-axis (Fig. 10), which is a simple linear ramp, the PD controller achieved nearly perfect tracking, while the ANN showed small oscillations. This confirms that the ANN does not universally outperform the PD controller; rather, its advantage is scenario-dependent.

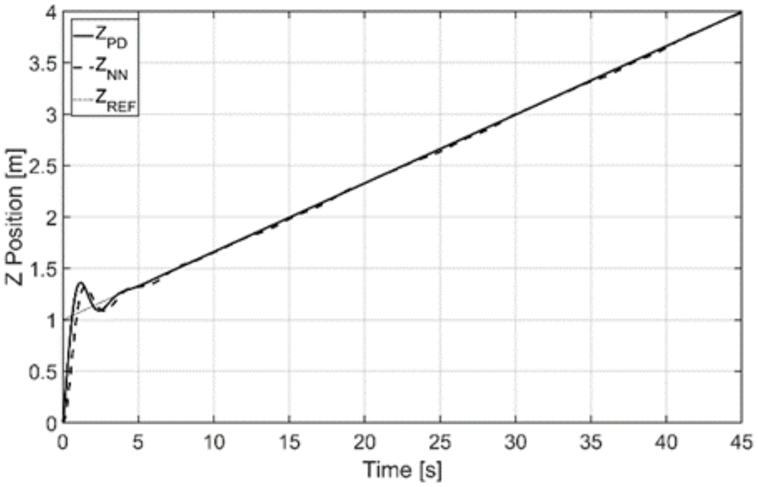

Fig. 10. ANN and PD controller helical trajectory tracking response along the z-axis

For the x-axis tracking, the ANN achieved an RMSE of 0.21 m compared to 0.47 m for the PD controller. Along the y-axis, the ANN achieved an RMSE of 0.19 m versus 0.52 m for the PD controller. Along the z-axis, the PD controller achieved superior performance due to the linear nature of the reference signal, with an RMSE of 0.12 m compared to 0.15 m for the ANN.

The roll (φ), pitch (θ), and yaw (ψ) tracking results are shown in Fig. 11, Fig. 12, and Fig. 13. The ANN demonstrated improved tracking in roll and pitch, while both controllers achieved comparable yaw regulation with oscillation amplitudes below 1°. These results demonstrate that the ANN exhibits superior adaptability to nonlinear trajectory profiles but inferior pure stabilization behavior.

Finally, a circular horizontal trajectory was tested (Fig. 14). The ANN maintained an average positional error of 0.23 m, while the PD controller exhibited an average error of 0.55 m, confirming the ANN's advantage in nonlinear tracking tasks.

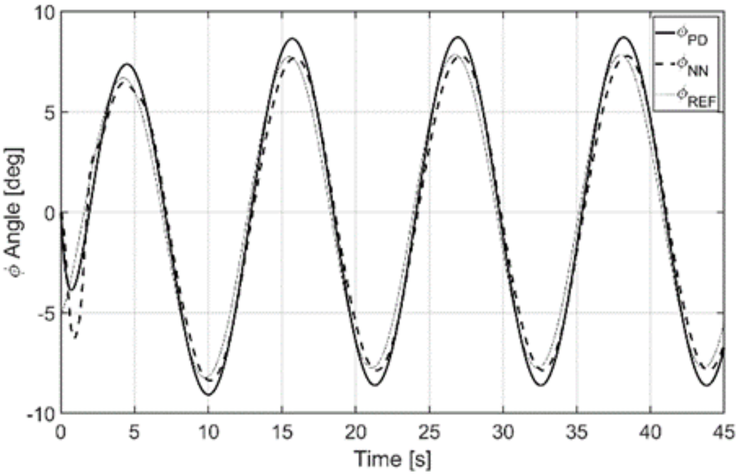

Fig. 11. ANN and PD controller helical trajectory tracking roll (ϕ) response

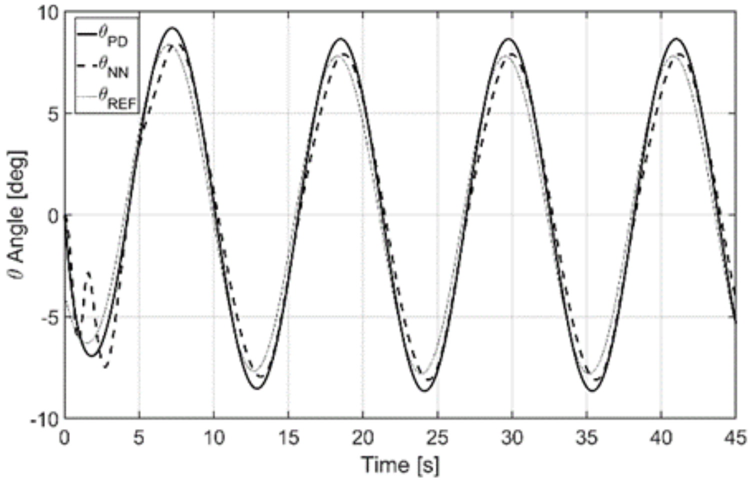

Fig. 12. ANN and PD controller helical trajectory tracking pitch (θ) response

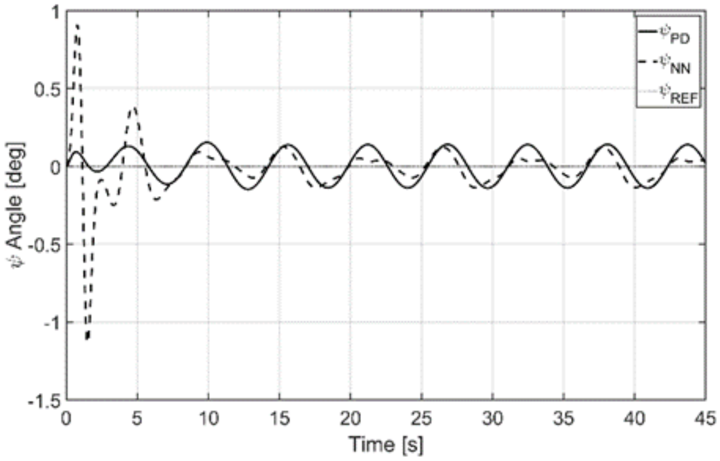

Fig. 13. ANN and PD controller helical trajectory tracking yaw (ψ) response

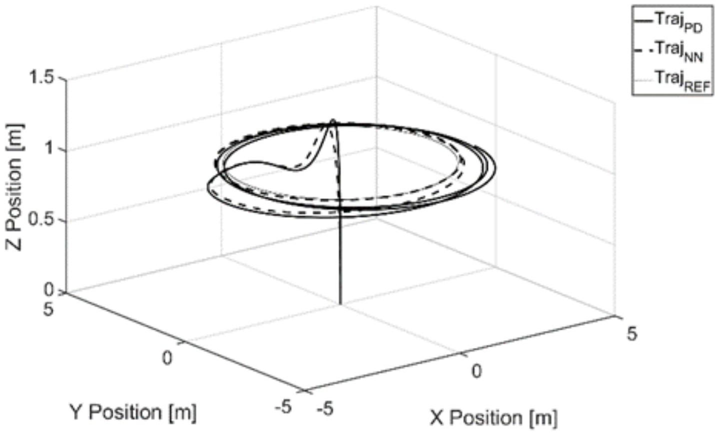

Fig. 14. ANN and PD controller circle trajectory response

From the previous results, the ANN almost perfectly reproduces PD commands ($R^2 \approx 0.999$) yet sometimes performs worse in stabilization and better in trajectory tracking. This is not a inconsistency but a property of nonlinear closed-loop systems.

In closed loop, even very small differences in control signals can shift system poles and zeros, altering damping and transient behavior. During stabilization, where precise damping is critical, the obvious derivative structure in the PD controller provides superior robustness. During nonlinear trajectories, however, the ANN provides adaptive nonlinear functions that better compensate coupling effects, which a fixed-gain PD cannot signify.

This analysis resolves the regression metrics with the observed closed-loop behavior and demonstrates that the ANN is not a universal replacement for PD, but a complementary strategy with strengths in nonlinear tracking.

4. Conclusion

This study presented a comparative evaluation of a cascaded PD controller and a supervised ANN controller for quadrotor stabilization and trajectory tracking. The results demonstrate that the PD controller provides superior robustness and faster settling performance during hover stabilization, achieving lower altitude RMSE (0.18 m) compared to the ANN (0.26 m). In contrast, the ANN controller outperformed the PD controller in nonlinear trajectory tracking tasks, achieving lower lateral tracking RMSE values along the x and y axes. These findings indicate that the PD controller is preferable for safety critical stabilization tasks where expectable behavior is required, while the ANN controller is more suitable for aggressive nonlinear trajectory tracking where adaptability has the priority. However, this study is limited by its dependance on simulation-only data and the use of PD-generated labels for ANN training. The ANN performance is therefore dependent on the simulation model and may degrade under unmodeled dynamics, sensor noise, or hardware-induced nonlinearities.

Future work will focus on experimental validation using a physical quadrotor platform to assess real world performance and robustness. The system robustness tests will be conducted by including structured sensor noise, and aerodynamic disturbances to evaluate controller stability margins. Additional studies will investigate training the ANN using data collected from alternative baseline controllers, including nonlinear and adaptive control schemes, to reduce dependence on any single control model.

Acknowledgment: The author would like to thank the Deanship of Scientific Research at Shaqra University for supporting this work.

References

[1] D. Zhang, L. Meng, Y. Hao, and R. Xia, "Adaptive sliding mode fault-tolerant control of UAV systems based on radial basis function neural networks," *Scientific Reports*, vol. 15, no. 1, p. 27504, 2025, https://doi.org/10.1038/s41598-025-13659-z.

[2] Z. Hu and X. Jin, "Formation control for an UAV team with environment-aware dynamic constraints," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 1465–1480, Jan. 2024, https://doi.org/10.1109/TIV.2023.3295354.

[3] S. Ai, J. Song, G. Cai, and K. Zhao, "Active fault-tolerant control for quadrotor UAV against sensor fault diagnosed by the auto sequential random forest," *Aerospace*, vol. 9, no. 9, p. 518, Sep. 2022, https://doi.org/10.3390/aerospace9090518.

[4] S. Liu, W. Lyv, C. Yang, F. Wang, and J. F. Whidborne, "Neural network observer based LPV fault tolerant control of a flying-wing aircraft," *IEEE Transactions on Intelligent Vehicles*, pp. 1–15, 2023, https://doi.org/10.1109/TIV.2023.3337863.

[5] B. Li, Y. Li, P. Yang, and X. Zhu, "Adaptive neural network-based fault-tolerant control for quadrotor-slung-load system under marine scene," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 681–691, Jan. 2024, https://doi.org/10.1109/TIV.2023.3333888.

[6] Y. Zhang, H. Sha, R. Peng, N. Li, Z. Miao, C. He, and J. Zhou, "Adaptive observer-based neural network control for multi-UAV systems with predefined-time stability," *Drones*, vol. 9, no. 3, p. 222, 2025, https://doi.org/10.3390/drones9030222.

[7] Y. Nettari, M. Labbadi, and S. Kurt, "Adaptive robust finite-time tracking control for quadrotor subject to disturbances," *Advances in Space Research*, vol. 71, no. 9, pp. 3803–3821, May 2023, https://doi.org/10.1016/j.asr.2022.09.016.

[8] B. Li and Y. Wang, "An enhanced model predictive controller for quadrotor attitude quick adjustment with input constraints and disturbances," *International Journal of Control, Automation and Systems*, vol. 20, no. 2, pp. 648–659, Feb. 2022, https://doi.org/10.1007/s12555-020-0815-9.

[9] R. Rubinacci, A. Nazzari, and M. Lovera, "Anytime trajectory optimization for multi-drone systems with guaranteed collision avoidance," *IEEE Control Systems Letters*, vol. 9, pp. 1255–1260, 2025, https://doi.org/10.1109/LCSYS.2025.3580039.

[10] Y. Meguro, S. Masuda, and M. Toyoda, "Application and performance evaluation of backstepping control for drone 3D trajectory control," *IEEJ Transactions on Electronics, Information and Systems*, vol. 144, no. 8, pp. 816–823, Aug. 2024, https://doi.org/10.1541/ieejeiss.144.816.

[11] A. Abdulkareem, V. Oguntosin, O. M. Popoola, and A. A. Idowu, "Modeling and nonlinear control of a quadcopter for stabilization and trajectory tracking," *Journal of Engineering*, vol. 2022, pp. 1–19, Oct. 2022, https://doi.org/10.1155/2022/2449901.

[12] D. Du, M. Chang, L. Tang, H. Zou, C. Tang, and J. Bai, "Trajectory planning and control design for aerial autonomous recovery of a quadrotor," *Drones*, vol. 7, no. 11, p. 648, Oct. 2023, https://doi.org/10.3390/drones7110648.

[13] C. V. Nguyen, M. T. Nguyen, H. T. Tran, M. L. Trinh, H. M. La, and H. T. T. Nguyen, "Trajectory tracking control for a quadcopter under external disturbances," *Engineering, Technology and Applied Science Research*, vol. 14, no. 6, pp. 17620–17628, Dec. 2024, https://doi.org/10.48084/etasr.8449.

[14] Y. Zou, K. Xia, and W. He, "Adaptive fault-tolerant distributed formation control of clustered vertical takeoff and landing UAVs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 2, pp. 1069–1082, Apr. 2022, https://doi.org/10.1109/TAES.2021.3117368.

[15] R. Szostak, M. Zimnoch, P. Wachniew, and A. Jasek-Kamińska, "Self-calibration of UAV thermal imagery using gradient descent algorithm," *Drones*, vol. 7, no. 11, p. 683, Nov. 2023, https://doi.org/10.3390/drones7110683.

[16] M. Okasha, J. Kralev, and M. Islam, "Design and experimental comparison of PID, LQR and MPC stabilizing controllers for Parrot Mambo mini-drone," *Aerospace*, vol. 9, no. 6, p. 298, Jun. 2022, https://doi.org/10.3390/aerospace9060298.

[17] I. Lopez-Sanchez and J. Moreno-Valenzuela, "PID control of quadrotor UAVs: A survey," *Annual Reviews in Control*, vol. 56, p. 100900, 2023, https://doi.org/10.1016/j.arcontrol.2023.100900.

[18] A. Khalid, Z. Mushtaq, S. Arif, K. Zeb, M. A. Khan, and S. Bakshi, “Control schemes for quadrotor UAV: taxonomy and survey,” *ACM Computing Surveys*, vol. 56, no. 5, pp. 1-32, 2023, https://doi.org/10.1145/3617652.

[19] K. L. Fetzer, S. G. Nersesov, and H. Ashrafiuon, "Trajectory tracking control of spatial underactuated vehicles," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 10, pp. 4897–4916, 2021, https://doi.org/10.1002/rnc.5509.

[20] F. Kong, J. Li, B. Jiang, H. Wang, and H. Song, "Trajectory optimization for drone logistics delivery via attention-based pointer network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 4519–4531, Apr. 2023, https://doi.org/10.1109/TITS.2022.3168987.

[21] G. P. Kulathunga, H. Hamed, D. Devitt, and A. Klimchik, "Trajectory tracking for quadrotors: An optimization-based planning followed by controlling approach," *Journal of Field Robotics*, vol. 39, no. 7, pp. 1003–1013, 2022, https://doi.org/10.1002/rob.22084.

[22] S. Li, N. Duan, and H. Min, "Trajectory tracking control for quadrotor unmanned aerial vehicle with input delay and disturbances," *Asian Journal of Control*, vol. 26, no. 1, pp. 150–161, 2024, https://doi.org/10.1002/asjc.3192.

[23] X. Zhao, L. He, X. Liu, K. Han, and J. Li, "A novel reinforcement learning framework for optimizing fixed-wing UAV flight control strategies," *Aerospace Science and Technology*, vol. 166, p. 110512, 2025, https://doi.org/10.1016/j.ast.2025.110512.

[24] K. Panjavarnam, Z. H. Ismail, C. H. H. Tang, K. Sekiguchi, and G. G. Casas, "Model predictive control for autonomous UAV landings: A comprehensive review of strategies, applications and challenges," *The Journal of Engineering*, vol. 2025, no. 1, p. e70085, 2025, https://doi.org/10.1049/tje2.70085.

[25] M. Shi, H. He, J. Li, M. Han, and N. Zhou, "Path planning and following control of autonomous bus under time-varying parameters against parametric uncertainties and external disturbances," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7057–7070, Jul. 2022, https://doi.org/10.1109/TVT.2022.3170440.

[26] H.-T. Zhang, B.-B. Hu, Z. Xu, Z. Cai, B. Liu, X. Wang, T. Geng, S. Zhong, and J. Zhao, "Visual navigation and landing control of an unmanned aerial vehicle on a moving autonomous surface vehicle via adaptive learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5345–5355, Dec. 2021, https://doi.org/10.1109/TNNLS.2021.3080980.

[27] S. Cao and H. Yu, "An adaptive control framework for the autonomous aerobatic maneuvers of fixed-wing unmanned aerial vehicle," *Drones*, vol. 6, no. 11, p. 316, 2022, https://doi.org/10.3390/drones6110316.

[28] S. Qamar, S. H. Khan, M. A. Arshad, M. Qamar, and A. Khan, "Autonomous drone swarm navigation and multi-target tracking in 3D environments with dynamic obstacles," in *Proceedings of the International Conference on Broadband Communications, Networks, and Systems*, 2022, pp. 290-295, https://doi.org/10.1109/IBCAST54850.2022.9990563.

[29] K. Guo, P. Tang, H. Wang, D. Lin, and X. Cui, "Autonomous landing of a quadrotor on a moving platform via model predictive control," *Aerospace*, vol. 9, no. 1, p. 34, Jan. 2022, https://doi.org/10.3390/aerospace9010034.

[30] J. Ross, M. Seto, and C. Johnston, "Autonomous landing of rotary wing unmanned aerial vehicles on underway ships in a sea state," *Journal of Intelligent & Robotic Systems*, vol. 104, no. 1, p. 1, 2021, https://doi.org/10.1007/s10846-021-01515-x.

[31] P. Xiao, N. Li, F. Xie, H. Ni, M. Zhang, and B. Wang, "Clustering-based multi-region coverage-path planning of heterogeneous UAVs," *Drones*, vol. 7, no. 11, p. 664, Nov. 2023, https://doi.org/10.3390/drones7110664.

[32] R. Norouzi, A. Kosari, and M. H. Lowenberg, "Real-time trajectory optimization of impaired aircraft based on steady state manoeuvres," *arXiv preprint*, arXiv:2408.02316, 2024, https://arxiv.org/abs/2408.02316.

[33] H. Huang, A. V. Savkin, and C. Huang, "Decentralized autonomous navigation of a UAV network for road traffic monitoring," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2558–2564, Aug. 2021, https://doi.org/10.1109/TAES.2021.3053115.

[34] M. Y. Arafat, M. M. Alam, and S. Moh, "Vision-based navigation techniques for unmanned aerial vehicles: Review and challenges," *Drones*, vol. 7, no. 2, p. 89, 2023, https://doi.org/10.3390/drones7020089.

[35] J. Xie and J. Chen, "Multiregional coverage path planning for multiple energy constrained UAVs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17366–17381, Oct. 2022, https://doi.org/10.1109/TITS.2022.3160402.

[36] T. Lei, C. Luo, T. Sellers, Y. Wang, and L. Liu, "Multitask allocation framework with spatial dislocation collision avoidance for multiple aerial robots," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 6, pp. 5129–5140, Dec. 2022, https://doi.org/10.1109/TAES.2022.3167652.

[37] F. Wei, N. Xu, S. Huang, and Y. Cao, "Disturbance observer-based adaptive neural finite-time control for nonstrict-feedback nonlinear systems with input delay," *Transactions of the Institute of Measurement and Control*, vol. 47, no. 6, pp. 1172–1187, 2025, https://doi.org/10.1177/01423312241261084.

[38] P. V. Patil, M. K. Khan, M. Korulla, V. Nagarajan, and O. P. Sha, "Design optimization of an AUV for performing depth control maneuver," *Ocean Engineering*, vol. 266, p. 112929, Dec. 2022, https://doi.org/10.1016/j.oceaneng.2022.112929.

[39] R. Doraiswami, L. Cheded, and M. Brinkmann, "Robust and intelligent Kalman filter residue-based trajectory tracking and fault-tolerant control of quadrotor using Bayes classifier fusion," in *Unmanned Aerial Vehicles Applications: Challenges and Trends*, M. Abdelkader and A. Koubaa, Eds., Cham: Springer International Publishing, 2023, pp. 85–123, https://doi.org/10.1007/978-3-031-32037-8_3.

[40] H. Xin, Q. Chen, Y. Xian, P. Wang, Y. Wang, X. Yao, Y. Lu, and Z. Hou, "Longitudinal decoupling control of altitude and velocity for a fixed-wing aircraft," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 10, pp. 6738–6748, Oct. 2024, https://doi.org/10.1109/TIV.2024.3388257.

[41] X. Liu, Z. Yuan, Z. Gao, and W. Zhang, "Reinforcement learning-based fault-tolerant control for quadrotor UAVs under actuator fault," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 12, pp. 13926–13935, Dec. 2024, https://doi.org/10.1109/TII.2024.3438241.

[42] M. Wan, M. Chen, and M. Lungu, "Adaptive sensor fault tolerant control with prescribed performance for unmanned autonomous helicopter based on neural networks," *Aircraft Engineering and Aerospace Technology*, vol. 96, no. 3, pp. 417–429, 2024, https://doi.org/10.1108/AEAT-03-2023-0080.

[43] D. Asadi, K. Ahmadi, and S. Y. Nabavi, "Fault-tolerant trajectory tracking control of a quadcopter in presence of a motor fault," *International Journal of Aeronautical and Space Sciences*, vol. 23, no. 1, pp. 129–142, Feb. 2022, https://doi.org/10.1007/s42405-021-00412-9.

[44] V. Turan, E. Avşar, D. Asadi, and E. A. Aydin, "Image processing based autonomous landing zone detection for a multi-rotor drone in emergency situations," *Turkish Journal of Engineering*, vol. 5, no. 4, pp. 193–200, Oct. 2021, https://doi.org/10.31127/tuje.744954.

[45] M. Sadiq, R. Hayat, K. Zeb, A. Al-Durra, and Z. Ullah, "Robust feedback linearization based disturbance observer control of quadrotor UAV," *IEEE Access*, vol. 12, pp. 17966–17981, 2024, https://doi.org/10.1109/ACCESS.2024.3360333.

[46] J. Hartley, H. P. H. Shum, E. S. L. Ho, H. Wang, and S. Ramamoorthy, "Formation control for UAVs using a flux guided approach," *Expert Systems with Applications*, vol. 205, p. 117665, Nov. 2022, https://doi.org/10.1016/j.eswa.2022.117665.

[47] S. B. Rosende, S. Ghisler, J. Fernández-Andrés, and J. Sánchez-Soriano, "Implementation of an edge-computing vision system on reduced-board computers embedded in UAVs for intelligent traffic management," *Drones*, vol. 7, no. 11, p. 682, Nov. 2023, https://doi.org/10.3390/drones7110682.

[48] M. Labbadi and M. Cherkaoui, "Adaptive fractional-order nonsingular fast terminal sliding mode based robust tracking control of quadrotor UAV with Gaussian random disturbances and uncertainties," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2265–2277, 2021, https://doi.org/10.1109/TAES.2021.3053109.

[49] H. Gunnarsson and A. Åsbrink, *Intelligent drone swarms: Motion planning and safe collision avoidance control of autonomous drone swarms*, Master’s thesis, Dept. Electrical Engineering, Linköping Univ., Linköping, Sweden, 2022, https://www.diva-portal.org/smash/get/diva2%3A1677350/FULLTEXT01.pdf.

[50] X. Fu and J. Peng, "Iterative learning control for UAV formation based on point-to-point trajectory update tracking," *Mathematics and Computers in Simulation*, vol. 209, pp. 1–15, Jul. 2023, https://doi.org/10.1016/j.matcom.2023.01.038.

[51] D. Yuan, X. Chang, Z. Li, and Z. He, "Learning adaptive spatial-temporal context-aware correlation filters for UAV tracking," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 18, no. 3, pp. 1–18, Aug. 2022, https://doi.org/10.1145/3486678.

[52] T.-J. Shen and C.-L. Chen, "Model-in-the-loop design and flight test validation of flight control laws for a small fixed-wing UAV," *Drones*, vol. 9, no. 9, p. 624, Sep. 2025, https://doi.org/10.3390/drones9090624.

[53] R. A. Saeed, M. Omri, S. Abdel-Khalek, E. S. Ali, and M. F. Alotaibi, "Optimal path planning for drones based on swarm intelligence algorithm," *Neural Computing and Applications*, vol. 34, no. 12, pp. 10133–10155, 2022, https://doi.org/10.1007/s00521-022-06998-9.

[54] K. L. C. Bell, M. C. Quinzin, D. Amon, S. Poulton, A. Hope, O. Sarti, T. E. Cañete, A. M. Smith, H. I. Baldwin, D. M. Lira, S. Cambronero-Solano, T.-R. A. Chung, and B. Brady, "Exposing inequities in deep-sea exploration and research: Results of the 2022 global deep-sea capacity assessment," *Frontiers in Marine Science*, vol. 10, pp.1–8, 2023, https://doi.org/10.3389/fmars.2023.1217227.

[55] R. Singh and S. Kumar, "A comprehensive insights into drones: History, classification, architecture, navigation, applications, challenges, and future trends," *arXiv preprint*, arXiv:2501.10066, 2025, https://arxiv.org/abs/2501.10066.

[56] K. D. Wahyuadnyana, K. Indriawati, P. A. Darwito, A. N. Aufa, and H. Tnunay, "Performance analysis of PID and SMC control algorithms on AUV under the influence of internal solitary wave in the Bali deep sea," *Journal of Robotics and Control*, vol. 5, no. 6, pp. 1957–1972, Oct. 2024, https://doi.org/10.18196/jrc.v5i6.23800.

[57] U. C. Cabuk, M. Tosun, O. Dagdeviren, and Y. Ozturk, "Modeling energy consumption of small drones for swarm missions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 8, pp. 10176–10189, 2024, https://doi.org/10.1109/TITS.2024.3350042.

[58] N. Akram, M. Khoshrangbaf, M. Challenger, and O. Dagdeviren, "Energy consumption modeling and flight time analysis of micro drones," *IEEE Access*, vol. 13, pp. 109854–109866, 2025, https://doi.org/10.1109/ACCESS.2025.3581944.

[59] B. Alkouz, A. Abusafia, A. Lakhdari, and A. Bouguettaya, "In-flight energy-driven composition of drone swarm services," *IEEE Transactions on Services Computing*, vol. 16, no. 3, pp. 1919–1933, May–Jun. 2023, https://doi.org/10.1109/TSC.2022.3203033.

[60] G. Ren, B. Shahzaad, B. Alkouz, A. Lakhdari, and A. Bouguettaya, “Energy-predictive planning for optimizing drone service delivery,” *Expert Systems with Applications*, p. 129251, 2025, https://doi.org/10.1016/j.eswa.2025.129251.

[61] C. Muli, S. Park, and M. Liu, "A comparative study on energy consumption models for drones," in *Internet of Things*, A. González-Vidal, A. M. Abdelgawad, E. Sabir, S. Ziegler, and L. Ladid, Eds., Cham, Switzerland: Springer, 2022, pp. 199–210, https://doi.org/10.1007/978-3-031-20936-9_16.