sensors

MDPI

Article

Neural Network-Based Adaptive Attitude Controller for a Micro-Unmanned Aerial Vehicle

Gennaro Raspaolo [1], Salvatore Rosario Bassolillo [1], Luciano Blasi [1], Egidio D'Amato [2,*] and Immacolata Notaro [1]

[1] Department of Engineering, Universitá degli Studi della Campania "L. Vanvitelli", Via Roma, 29, 81031 Aversa, CE, Italy; gennaro.raspaolo@unicampania.it (G.R.); salvatore.bassolillo@gmail.com (S.R.B.); luciano.blasi@unicampania.it (L.B.); immacolata.notaro@unicampania.it (I.N.)

[2] Department of Science and Technology, Universitá degli Studi di Napoli "Parthenope", Centro Direzionale di Napoli, Isola C4, 80143 Napoli, NA, Italy

* Correspondence: egidio.damato@uniparthenope.it

Abstract

The deployment of micro-Unmanned Aerial Vehicles (UAVs) for civil applications has exponentially grown in the last decades due to their versatility, low cost and ease of use. However, their effective implementation requires an efficient attitude control system that ensures the tracking of a reference signal, even in the presence of uncertainties or external disturbances. The main aim of this paper is to present the design of a neural-network-augmented H_∞ controller for a micro-UAV, to compensate for dynamic uncertainties and environmental perturbations. The effectiveness of the proposed architecture is validated through both numerical sensitivity analyses and experimental tests on a Crazyflie 2.1+ quadrotor mounted on a tilting test bench.

Keywords: attitude control; robust control; disturbance rejection; micro quadcopter; neural networks

1. Introduction

Over the last few decades, UAVs have gained increasing attention due to their versatility and ease of use, both in civil and military applications. Their widespread adoption is mainly due to cost reductions and improvements in low-cost electronics, offering competitive products for a wide range of applications [1,2], including localization and mapping [3–5], environmental monitoring [6–8] and precision agriculture [9–11]. Multirotor drones have shown significant advantages over fixed-wing aircraft in hazardous environments [12–14] due to their Vertical Take-Off and Landing (VTOL) capability, maneuverability in confined spaces and the ability to hover over a specific target.

However, multirotor vehicles need a reliable and effective attitude controller to perform precise maneuvers, capable of ensuring vehicle stability even in adverse operating conditions, such as in the presence of wind gusts or disturbances due to unbalanced payloads [15].

Due to their straightforward implementation and ease of tuning, Proportional Integral Derivative (PID) controllers are widely considered as the baseline control architecture for both linear and nonlinear quadrotor dynamics [16–24].

The authors of [25] proposed a systematic model-based approach to the design of a UAV research platform, adopting a conventional PID controller for attitude stabilization purposes. A cascaded attitude control strategy is introduced in [26], featuring a PID-based inner loop for angular velocity and a Proportional Integral (PI)-based outer loop for attitude

Check for updates

Received: 21 September 2026
Revised: 23 October 2026
Accepted: 24 October 2026
Published: 27 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

regulation. However, these controllers exhibit poor performance in the presence of system uncertainties or strong external disturbances.

To address performance limitations of classical PID-based strategies, several alternative control architectures have been proposed in the literature. For instance, in [27], a state feedback controller based on the Linear Quadratic Regulator (LQR) paradigm was presented. An adaptive PID controller using sliding mode control approaches was proposed in [28]. To address the nonlinear nature of quadrotor dynamics, several nonlinear control methodologies have been developed. Backstepping control has emerged as an effective solution due to its recursive design procedure and suitability for cascaded systems. The authors of [29] proposed a robust backstepping controller enhanced by disturbance estimation techniques to improve attitude-tracking performance under uncertainties. Another popular nonlinear approach is sliding mode control (SMC), which is known for its robustness against model uncertainties and external disturbances. The application of SMC to quadrotor attitude regulation is investigated in [30], comparing its performance with other nonlinear control strategies.

More recently, optimal predictive and robust control techniques have gained increasing attention. model predictive control (MPC) explicitly incorporates system constraints and optimizes the control action over a finite prediction horizon. A multi-model predictive control framework for quadrotor attitude regulation is introduced in [31], achieving performance close to nonlinear MPC while significantly reducing computational load.

While these approaches can significantly improve tracking performance, their capabilities strongly rely on the availability of accurate models or extensive training data, and they generally do not provide explicit robustness guarantees against uncertainties or persistent disturbances. Furthermore, model-based nonlinear techniques, such as dynamic inversion, are highly sensitive to modeling errors, which may degrade controller performance in real-world operational scenarios [32,33].

In contrast, the robust $\mathcal{H}_\infty$ control technique has proven to guarantee closed-loop stability in the presence of bounded disturbances and model uncertainties [34,35]. Over the years, several researchers have adopted this kind of technique in UAV applications. The authors of [36] proposed a technique that utilizes feedback linearization for model nonlinearity, while a μ-synthesis controller is used to achieve robust performance in the presence of model uncertainties and external disturbances. The paper presented in [37] proposes the application of an H_∞ controller with static gains for the stabilization of an HAPD (High-Altitude Performance Demonstrator) with redundant control surfaces. To effectively stabilize a quadrotor against atmospheric disturbances during flight operations, the authors of [38] used the H_∞ paradigm to nullify the influence of atmospheric turbulence and ground effect on the UAV. The authors of [39] synthesized the H_∞ controller for both pitch and roll angles for a real quadcopter, first performing a continuous-time system identification on experimental data to capture a nominal model of the system as well as a multiplicative uncertainty.

Despite the robustness guarantees of the H_∞ controller, its performance relies on a fixed controller structure, which cannot adapt to unknown disturbances or aggressive maneuvers. In such scenarios, a quadcopter exhibits strong and unpredictable nonlinear behaviors, compromising the nominal performance of the vehicle. To cope with these limitations, adaptive control techniques offer a viable solution [40–42]. Among adaptive methodologies, Model Reference Adaptive Control (MRAC) has attracted considerable attention for its ability to compensate for parametric uncertainties. First introduced in the 1950s by Whitaker [43], MRAC has been successfully applied by several researchers to quadrotor platforms. In [44], the MRAC with a modified version of the MIT rule is used to control the altitude of a quadcopter. The authors of [45] used a PID to control the motor speed while a direct MRAC was used for the high-level control of attitude angles. The use

of the MRAC paradigm in the attitude loop of a hexacopter was proposed in [46] to adapt the control coefficients in case of failure or when the payload is released.

However, the classical MRAC-based control approach presents some limitations, such as dependence on precise models and sensitivity to parametric variations, especially in fast-dynamics applications [47–49]. To overcome these difficulties, several authors have focused on the use of neural network (NN)-based controllers, allowing them to create more robust and flexible adaptive models capable of managing in real time the uncertainties and nonlinearities typical of the behavior of a quadcopter [47,50].

In recent years, safety reasons have pushed researchers to develop hybrid architectures, combining a baseline controller with an NN compensator, with the latter able to provide an additive correction signal to enhance robustness and adaptability in the presence of model uncertainties and external disturbances [51]. Indeed, although NN-based controllers outperform traditional methods under external disturbances or model uncertainties once trained, the training phase itself requires a stable baseline control strategy [52]. The authors of [53] proposed a control scheme to stabilize the attitude of a mobile manipulating UAV (MM-UAV), using a combination of gain scheduling and Lyapunov-based MRAC. In [54], the authors presented a control architecture based on PID and MRAC to compensate for any UAV failures, with the MRAC controller able to provide an additive signal to guarantee the global asymptotic stability of the damaged system. The work proposed in [55] introduced a hybrid adaptive control technique based on MRAC as a compensator for the attitude control of a quadrotor, whose performance is analyzed in the presence of disturbances.

To further improve the capabilities of the above-mentioned approaches, H_∞ robustness can be combined with learning techniques to take advantage of both approaches. For instance, the authors of [56] use Lyapunov analysis to treat network reconstruction errors as disturbances within an H_∞ framework, utilizing monitoring logic to prevent system divergence. Likewise, the approach in [57] employs high-order neural networks to make nominal controllers more robust without requiring predefined bounds, at the cost of increased control effort and local validity. Additionally, in [58], the authors minimize computational burden through the adaptation of a single parameter, though this introduces the risk of chattering due to discontinuous control. In aeronautics, a combination of H_∞ and neural adaptive control has been used to solve different tasks. In [59], an auto-landing controller is designed using a high-order H_∞ coupled with adaptive compensation, whereas a quadcopter controller is presented and validated through simulations in [60]. The use of ridge functions for rocket autopilot in [61] reduces the number of tuning parameters; however, it necessitates the online update of multiple gain matrices. The aim of this paper is to bridge the gap between theoretical design and experimental validation on resource-constrained systems, while continuing to guarantee robust stability and performance.

A preliminary study on this topic was published by the authors as a conference paper in [62].

Specifically, the main contributions of this work are as follows:

- A two-tier hybrid architecture combining a mixed-sensitivity H∞ baseline controller, which guarantees robust stability and noise attenuation, with an online adaptive NN loop that compensates for external disturbances and parametric variations in real time.
- By adopting Displaced Rectified Linear Unit (DReLU) activations within a Lyapunov-stable adaptation framework, the controller adapts online from scratch with zero offline pre-training, achieving a reduction in embedded processor load compared to standard Radial Basis Function (RBF)-based networks.
- A simulation campaign maps the hyperparameter space (adaptation rate, σ-modification factor, and hidden-layer dimension), defining the trade-off between convergence rate, tracking precision, and avoidance of high-frequency oscillations.

- Real-time deployment and benchmark testing on the STM32F405 microcontroller of a Crazyflie 2.1+ micro-UAV (Bitcraze AB, Malmö, Sweden) subjected to persistent unbalancing torques induced by a $2 \cdot 10^{-3}$ kg weight added on one arm of the drone at a distance of $\approx 5 \cdot 10^{-2}$ m. The drone is constrained to attitude motion by being mounted on a dedicated 3-degree-of-freedom tilting test rig with a spherical joint, resulting in negligible friction and inertia variation.

The paper is organized as follows: Section 2 introduces the UAV dynamics model while Section 3 describes the proposed flight control algorithm. Section 4 presents the results of the sensitivity analysis, carried out by changing the NN parameters, alongside experimental tests performed on the Crazyflie 2.1+ platform, to prove the effectiveness of the proposed control architecture. Finally, Section 5 draws the conclusions of the work.

2. Quadrotor Dynamic Model

The quadcopter dynamics are modeled by adopting Euler's rigid body equations. Accordingly, the vehicle motion is described by considering its translational and rotational behavior with respect to two reference frames (see Figure 1):

- North–East–Down (NED) reference frame, $E = \{x_E, y_E, z_E\}$, located on the Earth's surface;
- Body-fixed reference frame, $B = \{x_B, y_B, z_B\}$, whose origin, O_B, is located at the center of gravity (CG) of the quadrotor.

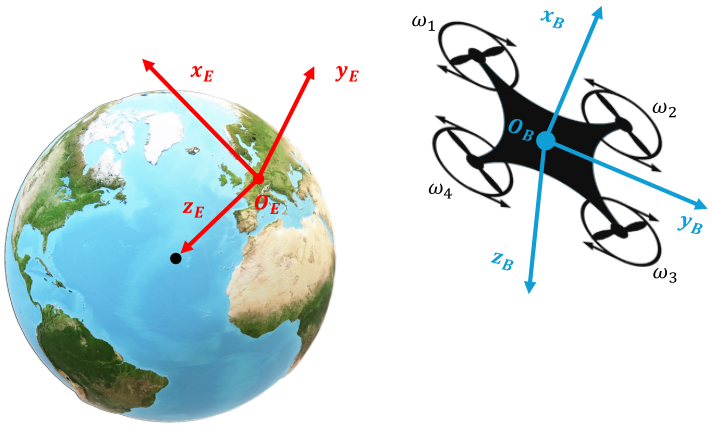


Figure 1. Coordinate control system definition: the Earth-fixed North–East–Down (NED) inertial frame $E = \{x_E, y_E, z_E\}$ and the aircraft body-fixed reference frame $B = \{x_B, y_B, z_B\}$ centered at the vehicle's center of gravity.

The transformation between these frames is expressed by the rotation matrix, $\boldsymbol{R}_{BE}(\boldsymbol{\Theta})$ (Equation (1)), which maps the Earth-fixed frame to the body frame as a function of the UAV Euler angles $\boldsymbol{\Theta} = [\phi, \theta, \psi]^T$,

$$\boldsymbol{R}_{BE}(\boldsymbol{\Theta}) = \begin{bmatrix} C_\theta C_\psi & C_\theta S_\psi & -S_\theta \\ C_\psi S_\theta S_\phi - S_\psi C_\phi & S_\psi S_\theta S_\phi + C_\psi C_\phi & C_\theta S_\phi \\ C_\psi S_\theta C_\phi + S_\psi S_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi & C_\theta C_\phi \end{bmatrix} \tag{1}$$

where $C_\eta = \cos(\eta)$ and $S_\eta = \sin(\eta)$, for $\boldsymbol{\eta} \in \{\phi, \theta, \psi\}$.

Let $\boldsymbol{V} = [u, v, w]^T$ and $\boldsymbol{\Omega} = [p, q, r]^T$ be the velocity and angular rate vectors expressed in the body frame, while $\boldsymbol{F}$ and $\boldsymbol{T}$ represent the external forces and torques acting on the vehicle with respect to the CG, respectively. The external forces depend on both the motion variables and the propellers' rotational speeds, $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3, \omega_4]^T$. Finally, let $\boldsymbol{p_E} = [x_E, y_E, z_E]^T$ represent the position of the quadrotor center of mass in the NED inertial reference frame.

The translational and rotational dynamics of the quadrotor are governed by Newton–Euler equations as follows:

$$m(\dot{\boldsymbol{V}} + \boldsymbol{\Omega} \times \boldsymbol{V}) = \boldsymbol{F}(\boldsymbol{V}, \boldsymbol{\Omega}, \boldsymbol{\Theta}, \boldsymbol{p_E}, \boldsymbol{\omega}) \tag{2}$$

$$\boldsymbol{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \boldsymbol{J}\boldsymbol{\Omega} = \boldsymbol{T}(\boldsymbol{V}, \boldsymbol{\Omega}, \boldsymbol{p_E}, \boldsymbol{\omega}) \tag{3}$$

Equations (2) and (3) represent the translational and rotational equilibrium, respectively, where m is the mass of the quadrotor and $\boldsymbol{J}$ is the inertia matrix.

To complete the dynamic model, the relationships mapping the body-fixed velocities to the inertial frame derivatives are given by

$$\dot{\boldsymbol{p_E}} = \boldsymbol{R}_{BE}^{-1}(\boldsymbol{\Theta})\boldsymbol{V} \tag{4}$$

$$\dot{\boldsymbol{\Theta}} = \boldsymbol{H}^{-1}(\boldsymbol{\Theta})\boldsymbol{\Omega} \tag{5}$$

where the transformation matrix $\boldsymbol{H}(\boldsymbol{\Theta})$ is defined as

$$\boldsymbol{H}(\boldsymbol{\Theta}) = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \tag{6}$$

The external forces and torques ((2) and (3)) can be decomposed into their physical contributions as follows:

$$\boldsymbol{F}(\boldsymbol{V}, \boldsymbol{\Omega}, \boldsymbol{\Theta}, \boldsymbol{p_E}, \boldsymbol{\omega}) = \boldsymbol{F}_g(\boldsymbol{\Theta}) + \boldsymbol{F}_p(\boldsymbol{V}, \boldsymbol{\Omega}, \boldsymbol{p_E}, \boldsymbol{\omega}) + \boldsymbol{F}_a(\boldsymbol{V}, \boldsymbol{\Omega}, \boldsymbol{p_E}) \tag{7}$$

$$\boldsymbol{T}(\boldsymbol{V}, \boldsymbol{\Omega}, \boldsymbol{p_E}, \boldsymbol{\omega}) = \boldsymbol{T}_p(\boldsymbol{V}, \boldsymbol{\Omega}, \boldsymbol{p_E}, \boldsymbol{\omega}) + \boldsymbol{T}_a(\boldsymbol{V}, \boldsymbol{\Omega}, \boldsymbol{p_E}) \tag{8}$$

where $\boldsymbol{F}_g$ denotes the gravitational force, $\boldsymbol{F}_a$ and $\boldsymbol{T}_a$ account for aerodynamic forces and moments, respectively, expressed as nonlinear functions of the motion state [63]. The terms $\boldsymbol{F}_p$ and $\boldsymbol{T}_p$ denote the propulsive forces and moments generated by the combined action of the four motors, which constitute the primary control inputs of the vehicle.

Denoting with F_i^r the thrust generated by the i-th rotor, the total propulsive forces and torques are expressed in the body-fixed frame as

$$\begin{cases} \boldsymbol{F}_p = -\sum_{i=1}^{4} F_i^r \boldsymbol{e}_3 \\ \boldsymbol{T}_p = \begin{bmatrix} L_p \\ M_p \\ N_p \end{bmatrix} = \begin{bmatrix} b[(F_1^r + F_4^r) - (F_2^r + F_3^r)] \\ l[(F_1^r + F_2^r) - (F_3^r + F_4^r)] \\ \sum_{i=1}^{4}[(-1)^{i+1} T_i^r] \end{bmatrix} \end{cases} \tag{9}$$

where $\boldsymbol{e}_3 = [0, 0, 1]^T$ is the unit vector along the z-axis of the body frame, T_i^r represents the reactive torque produced by the i-th propeller, while b and l denote the distances of the rotor axes from the CG along the y_B and x_B axes, respectively (see Figure 2).

The forces and moments generated by the propellers depend on complex aerodynamic interactions and are generally difficult to model. However, near the hovering condition, a well-accepted simplified model expresses the thrust and drag torque of each rotor as a quadratic relation of its rotational speed ω_i via the coefficients k_f and k_t [64]. Under this assumption, the propulsive forces and moments can be compactly written in matrix form.

$$\begin{bmatrix} F_{pz} \\ L_p \\ M_p \\ N_p \end{bmatrix} = \begin{bmatrix} -k_f & -k_f & -k_f & -k_f \\ bk_f & -bk_f & -bk_f & bk_f \\ lk_f & lk_f & -lk_f & -lk_f \\ k_t & -k_t & k_t & -k_t \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{10}$$

The nonlinear quadrotor dynamics, defined in (2)–(5), can be linearized around a nominal steady-state hovering equilibrium. This operating condition is characterized by

null translational and angular velocity vectors ($\boldsymbol{V}_0 = \boldsymbol{0}, \boldsymbol{\Omega}_0 = \boldsymbol{0}$), a constant nominal altitude $z_{E,0}$, and flat attitude profiles ($\boldsymbol{\Theta}_0 = \boldsymbol{0}$), implying that the nominal propulsive thrust vector perfectly balances gravity.

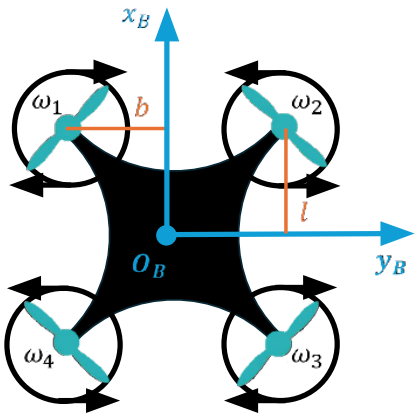


Figure 2. Schematic top view of the quadrotor airframe in X-configuration, detailing the counter-rotating propeller assignments (ω_1 to ω_4) and the respective structural moment arms (b, l) relative to the center of gravity.

To yield a tractable linear model for control synthesis, the following standard assumptions are introduced:

- The quadrotor structure is rigid and symmetric, implying a diagonal inertia matrix, i.e., $\boldsymbol{J} = \mathrm{diag}(J_{xx}, J_{yy}, J_{zz})$;
- The CG coincides with the geometric center of the vehicle;
- Aerodynamic forces and moments, other than the primary propulsive inputs, are neglected due to the low-speed flight regime near the hover state.

Under these assumptions, the attitude perturbation variables remain small ($\phi \approx 0$, $\theta \approx 0, \psi \approx 0$), which reduces the kinematic transformation matrices introduced in (4) and (5) to identity operators, namely $\boldsymbol{R}_{BE}(\boldsymbol{\Theta}) \approx \boldsymbol{I}_{3\times3}$ and $\boldsymbol{H}^{-1}(\boldsymbol{\Theta}) \approx \boldsymbol{I}_{3\times3}$.

By defining the perturbation state vector as $\boldsymbol{x}(t) = [\delta\boldsymbol{V}^T, \delta\boldsymbol{\Omega}^T, \delta\boldsymbol{p_E}^T, \delta\boldsymbol{\Theta}^T]^T \in \mathbb{R}^{12}$ and the control input variations as $\boldsymbol{u}(t) = [\delta F_{pz}, \delta L_p, \delta M_p, \delta N_p]^T \in \mathbb{R}^4$, the resulting small-disturbance dynamics can be compactly formulated in the standard state–space representation:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t), \end{aligned} \tag{11}$$

where the state matrix $\boldsymbol{A} \in \mathbb{R}^{12\times12}$ and the input matrix $\boldsymbol{B} \in \mathbb{R}^{12\times4}$ are analytically derived via Jacobian evaluation at the hover state, yielding the following sparse block-partitioned structures:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{A}_{V\Theta} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_V \\ \boldsymbol{B}_\Omega \\ \boldsymbol{0}_{3\times4} \\ \boldsymbol{0}_{3\times4} \end{bmatrix} \tag{12}$$

where the non-zero sub-matrices capture the dominant local gravity coupling and control allocation channels, defined respectively as

$$\boldsymbol{A}_{V\Theta} = \begin{bmatrix} 0 & g & 0 \\ -g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B}_V = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B}_\Omega = \begin{bmatrix} 0 & \frac{1}{J_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{J_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{J_{zz}} \end{bmatrix} \tag{13}$$

Assuming full state availability, the output mapping matrices reduce to

$$\boldsymbol{C} = \boldsymbol{I}_{12\times12}, \quad \boldsymbol{D} = \boldsymbol{0}_{12\times4} \tag{14}$$

3. Flight Control Algorithm

The scope of this paper is to analyze the attitude performance of a drone constrained to a platform. Consequently, a position controller is not required, and the remainder of the analysis focuses exclusively on the attitude dynamics.

To design the control algorithms for the attitude tracking loop, a reduced-order state space model is extracted from the linearized dynamics (11). Let $\tilde{\boldsymbol{x}}(t) = [\delta\boldsymbol{\Theta}^T, \delta\boldsymbol{\Omega}^T]^T = [\delta\phi, \delta\theta, \delta\psi, \delta p, \delta q, \delta r]^T \in \mathbb{R}^6$ denote the attitude state vector, and $\tilde{\boldsymbol{u}}(t) = [\delta L_p, \delta M_p, \delta N_p]^T \in \mathbb{R}^3$ represent the virtual control moments. Incorporating additive sensor noise $\boldsymbol{\nu}(t)$, the reduced-order attitude dynamics is formulated as

$$\begin{aligned} \dot{\tilde{\boldsymbol{x}}}(t) &= \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}(t) + \tilde{\boldsymbol{B}}\tilde{\boldsymbol{u}}(t) \\ \tilde{\boldsymbol{y}}(t) &= \tilde{\boldsymbol{C}}\tilde{\boldsymbol{x}}(t) + \boldsymbol{\nu}(t) \end{aligned} \tag{15}$$

where the sparse system matrices $\tilde{\boldsymbol{A}} \in \mathbb{R}^{6\times6}$, $\tilde{\boldsymbol{B}} \in \mathbb{R}^{6\times3}$, and the measurement matrix $\tilde{\boldsymbol{C}} \in \mathbb{R}^{6\times6}$ directly follow the structural partitions defined in (12).

The proposed control architecture consists of a robust linear H_∞ baseline controller augmented by an online adaptive neural network compensator. The general control architecture is shown in Figure 3. It has a cascade structure where the angular rate and the attitude feedback is separated to mimic the so-called Stability Augmentation System (SAS) and Control Augmentation System (CAS) [20].

To formalize this hierarchical structure, Equation (15) is partitioned into two vector equations corresponding to the attitude kinematics and the rotational dynamics:

$$\begin{aligned} \delta\dot{\boldsymbol{\Theta}}(t) &= \tilde{\mathbf{A}}_{11}\delta\boldsymbol{\Theta}(t) + \tilde{\mathbf{A}}_{12}\delta\boldsymbol{\Omega}(t) + \tilde{\mathbf{B}}_1\tilde{\boldsymbol{u}}(t) \\ \delta\dot{\boldsymbol{\Omega}}(t) &= \tilde{\mathbf{A}}_{21}\delta\boldsymbol{\Theta}(t) + \tilde{\mathbf{A}}_{22}\delta\boldsymbol{\Omega}(t) + \tilde{\mathbf{B}}_2\tilde{\boldsymbol{u}}(t) \end{aligned} \tag{16}$$

where the state matrices are partitioned accordingly as $\tilde{\mathbf{A}} = \begin{bmatrix} \tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12} \\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22} \end{bmatrix}$ and $\tilde{\mathbf{B}} = \begin{bmatrix} \tilde{\mathbf{B}}_1 \\ \tilde{\mathbf{B}}_2 \end{bmatrix}$ with $\tilde{\mathbf{A}}_{11}, \tilde{\mathbf{A}}_{21}, \tilde{\mathbf{A}}_{22}, \tilde{\mathbf{B}}_1 = \mathbf{0}$ and $\tilde{\mathbf{A}}_{12} = \mathbf{I}$.

The CAS (outer loop) acts on the attitude kinematics represented by the first row in (16). The output of this subsystem is a desired angular rate variation to drive the attitude tracking error to zero:

$$\boldsymbol{\delta\Omega}_{des}(t) = \delta\dot{\boldsymbol{\Theta}}_{ref}(t) + \mathbf{K}_{CAS}\boldsymbol{e}_{\boldsymbol{\Theta}}(t) \tag{17}$$

where $\boldsymbol{e}_{\boldsymbol{\Theta}} = \delta\boldsymbol{\Theta}_{ref}(t) - \delta\boldsymbol{\Theta}(t)$ is the attitude tracking error and $\mathbf{K}_{CAS}$ denotes the transfer function of the feedback controller $\mathbf{H}_{\infty,CAS}$.

$\delta\boldsymbol{\Omega}_{des}$ is then fed to the SAS (inner loop, i.e., the second row of (16)) which computes the virtual control moments $\tilde{\boldsymbol{u}}(t) = [\delta L_P, \delta M_P, \delta N_P]^T$ that drive the angular rate error to zero:

$$\tilde{\boldsymbol{u}}_{\mathbf{H}_\infty}(t) = \tilde{\mathbf{B}}_2^{-1}\left(\delta\dot{\boldsymbol{\Omega}}_{\text{des}}(t) + \mathbf{K}_{SAS}\boldsymbol{e}_{\boldsymbol{\Omega}}(t)\right) \tag{18}$$

where $\boldsymbol{e}_{\boldsymbol{\Omega}} = \delta\boldsymbol{\Omega}_{des}(t) - \delta\boldsymbol{\Omega}(t)$ is the tracking error of the angular rate, $\mathbf{K}_{SAS}$ is the transfer function of the feedback controller $\mathbf{H}_{\infty,SAS}$ and $\mathbf{B_2} = \mathbf{J}^{-1}$.

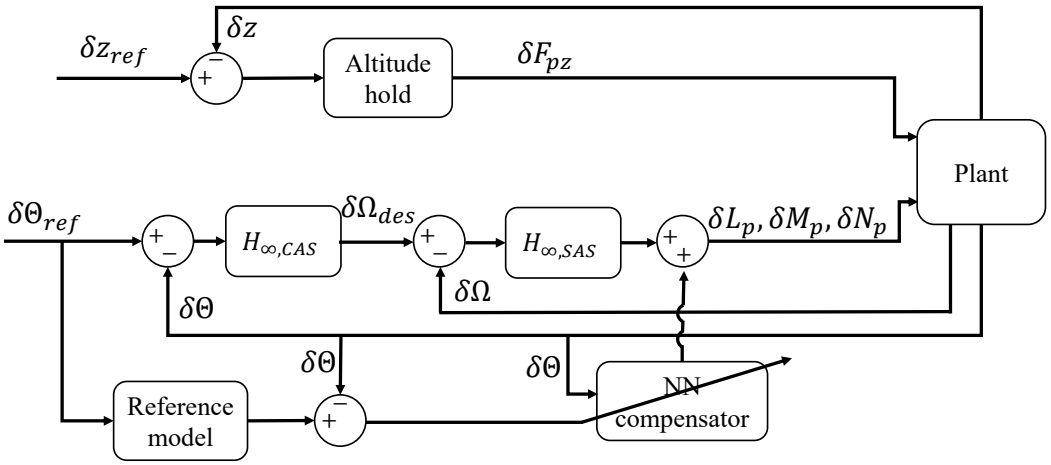


Figure 3. Global architecture of the proposed control system, illustrating the parallel decoupling layout between the inner attitude tracking loop augmented via the NN-based adaptive compensator and the baseline altitude-hold channel.

Each controller $j \in \{CAS, SAS\}$ is synthesized within the mixed-sensitivity H_∞ framework to satisfy tracking performance and robustness margins. Three dynamic weighting functions are considered to shape the closed-loop response (see Figure 4): $\boldsymbol{Q}_{e,j}(s)$ to minimize steady-state tracking error, $\boldsymbol{Q}_{u,j}(s)$ to penalize high-frequency control effort and $\boldsymbol{Q}_{y,j}(s)$ to mitigate measurement noise injection. Each shaping filter is cast into a minimal state-space realization.

Consider

$$\tilde{\boldsymbol{x}}_{CAS} = \delta\boldsymbol{\Theta},\ \boldsymbol{e}_{CAS} = \boldsymbol{e}_{\boldsymbol{\Theta}},\ \boldsymbol{u}_{CAS} = \delta\boldsymbol{\Omega}_{des},\ \boldsymbol{y}_{CAS} = \delta\boldsymbol{\Theta}$$
$$\tilde{\boldsymbol{x}}_{SAS} = \delta\boldsymbol{\Omega},\ \boldsymbol{e}_{SAS} = \boldsymbol{e}_{\boldsymbol{\Omega}},\ \boldsymbol{u}_{SAS} = \tilde{\boldsymbol{u}}_{\boldsymbol{H}_\infty},\ \boldsymbol{y}_{SAS} = \delta\boldsymbol{\Omega}$$

By defining the augmented generalized state vector as $\boldsymbol{\xi_j} = [\tilde{\boldsymbol{x}_j}^T, \boldsymbol{x}_{e,j}^T, \boldsymbol{x}_{u,j}^T, \boldsymbol{x}_{y,j}^T]^T$, the interconnection layout is encapsulated into the standard generalized plant $\boldsymbol{P}(s)$:

$$\begin{bmatrix} \dot{\boldsymbol{\xi}}_j(t) \\ \boldsymbol{\tau}_j(t) \\ \boldsymbol{y}_{\boldsymbol{K},j}(t) \end{bmatrix} = \begin{bmatrix} \mathcal{A}_j & \mathcal{B}_{1,j} & \mathcal{B}_{2,j} \\ C_{1,j} & \mathcal{D}_{11,j} & \mathcal{D}_{12,j} \\ C_{2,j} & \mathcal{D}_{21,j} & \mathcal{D}_{22,j} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_j(t) \\ \boldsymbol{R}_j(t) \\ \tilde{\boldsymbol{u}}_j(t) \end{bmatrix} \tag{19}$$

where $\boldsymbol{\tau}_j = [\boldsymbol{\tau}_{1,j}^T, \boldsymbol{\tau}_{2,j}^T, \boldsymbol{\tau}_{3,j}^T]^T$ represents the regulated performance outputs, $\boldsymbol{y}_{\boldsymbol{K},j}$ is the input vector fed to the controller and R_j is the reference signal. The dynamic output feedback control law is structured as $\tilde{\boldsymbol{u}}_j(t) = \boldsymbol{K}_j \boldsymbol{y}_{\boldsymbol{K},j}(t)$.

The mixed-sensitivity synthesis [65] aims to compute an optimal stabilizing feedback controller $\boldsymbol{K}_j$ that minimizes the H_∞ norm of the closed-loop transfer matrix $\boldsymbol{Q}_{R\tau,j} : \boldsymbol{R}_j \to \boldsymbol{\tau}_j$:

$$\min_{\boldsymbol{K}_j} \|\boldsymbol{Q}_{R\tau,j}(s)\|_{H_\infty} \leq \gamma_{\infty,j} \tag{20}$$

where $\gamma_{\infty,j} > 0$ denotes the disturbance attenuation performance bound and $\boldsymbol{Q}_{R\tau,j}(s)$ is structured as

$$\boldsymbol{Q}_{R\tau,j}(s) = \begin{bmatrix} \boldsymbol{Q}_{e,j}(s)\boldsymbol{S}_j(s) \\ \boldsymbol{Q}_{u,j}(s)\boldsymbol{K}_j(s)\boldsymbol{S}_j(s) \\ \boldsymbol{Q}_{y,j}(s)\boldsymbol{T}_j(s) \end{bmatrix} \tag{21}$$

In Equation (21), $\boldsymbol{S}_j(s) = (\boldsymbol{I} + \boldsymbol{G}_j(s)\boldsymbol{K}_j(s))^{-1}$ and $\boldsymbol{T}_j(s) = \boldsymbol{G}_j(s)\boldsymbol{K}_j(s)(\boldsymbol{I} + \boldsymbol{G}_j(s)\boldsymbol{K}_j(s))^{-1}$ denote the sensitivity and complementary sensitivity functions of the nominal loop, respectively, with $\boldsymbol{G}_j(s)$ representing the transfer matrix of the reduced attitude plant (15).

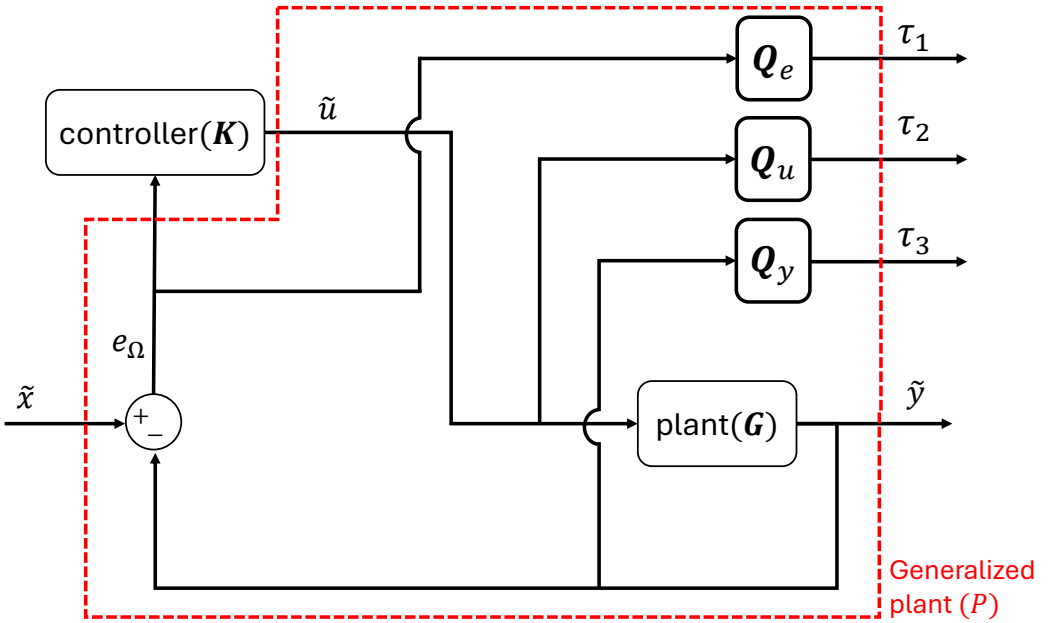


Figure 4. Standard mixed-sensitivity H_∞ optimization framework setup, detailing the closed-loop interconnection of the reduced attitude plant $\boldsymbol{G}(s)$, the baseline controller $\boldsymbol{K}(s)$, and the dynamic shaping weight filters $\boldsymbol{Q}_e(s)$, $\boldsymbol{Q}_u(s)$, and $\boldsymbol{Q}_y(s)$ embedded within the generalized plant $P(s)$.

Adaptive Neural Network Augmentation

To preserve tracking performance when operating far from the nominal hover condition or in the presence of severe unmodeled nonlinearities, the linear baseline control input $\tilde{\boldsymbol{u}}_{H_\infty}(t)$ is augmented by an adaptive NN-based control scheme. Under unmodeled perturbations, the actual nonlinear attitude dynamics is formulated as

$$\dot{\tilde{\boldsymbol{x}}}(t) = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}(t) + \tilde{\boldsymbol{B}}\boldsymbol{\Lambda}\left(\tilde{\boldsymbol{u}}_\Theta(t) + \boldsymbol{f}_T(\tilde{\boldsymbol{x}})\right) \tag{22}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{3\times3}$ is a positive-definite diagonal matrix modeling actuator effectiveness uncertainties, $\tilde{\boldsymbol{u}}_\Theta(t) = \tilde{\boldsymbol{u}}_{H_\infty}(t) + \boldsymbol{u}_{\mathrm{NN}}(t)$ is the total control command, and $\boldsymbol{f}_T(\tilde{\boldsymbol{x}}) \in \mathbb{R}^3$ represents the lumped unmodeled nonlinearities and exogenous disturbances.

The control objective is to force the perturbed state $\tilde{\boldsymbol{x}}(t)$ to asymptotically track the state $\boldsymbol{x}_r(t) \in \mathbb{R}^6$ of a stable linear reference model, defined by

$$\dot{\boldsymbol{x}}_r(t) = \boldsymbol{A}_r\boldsymbol{x}_r(t) + \boldsymbol{B}_r\boldsymbol{r}(t) \tag{23}$$

where $\boldsymbol{A}_r \in \mathbb{R}^{6\times6}$ is Hurwitz, $\boldsymbol{B}_r \in \mathbb{R}^{6\times3}$, and $\boldsymbol{r}(t) \in \mathbb{R}^3$ is the bounded command input vector.

The adaptive compensation term $\boldsymbol{u}_{\mathrm{NN}}(t)$ is parameterized via a Single-Layer Feedforward NN consisting of N hidden neurons and a linear output layer (see Figure 5):

$$\boldsymbol{u}_{\mathrm{NN}}(t) = -\boldsymbol{W}^T(t)\boldsymbol{\Phi}(\tilde{\boldsymbol{x}}) \tag{24}$$

where $\boldsymbol{\Phi}(\tilde{\boldsymbol{x}}) = [\Phi_1(\tilde{\boldsymbol{x}}), \ldots, \Phi_N(\tilde{\boldsymbol{x}}), 1]^T \in \mathbb{R}^{N+1}$ defines the vector of activation functions augmented with a unit bias, and $\boldsymbol{W}(t) \in \mathbb{R}^{(N+1)\times3}$ is the dynamically updated weight matrix:

$$\boldsymbol{W}(t) = \begin{bmatrix} w_{1,1}(t) & w_{1,2}(t) & w_{1,3}(t) \\ \vdots & \vdots & \vdots \\ w_{N+1,1}(t) & w_{N+1,2}(t) & w_{N+1,3}(t) \end{bmatrix} \tag{25}$$

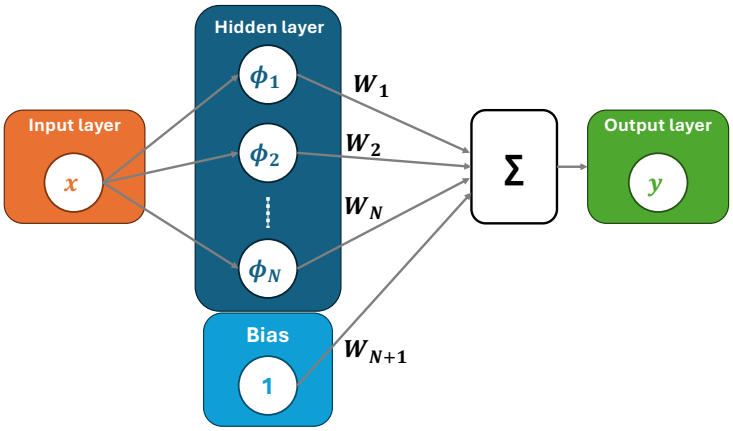


Figure 5. Structural topology of the augmented Single-Layer Feedforward NN compensator, showing the input layer, the hidden-layer neuron allocation (Φ_1 to Φ_N), the unit bias injection and the output-layer adaptive weight distribution $\boldsymbol{W}(t)$.

Let $\boldsymbol{e}(t) = \tilde{\boldsymbol{x}}(t) - \boldsymbol{x}_r(t)$ and $\boldsymbol{e}_{\boldsymbol{W}}(t) = \boldsymbol{W}_{opt} - \boldsymbol{W}(t)$ denote the state tracking error and the weight estimation error, respectively. $\boldsymbol{W}_{opt}$ represents the vector of optimal weight for which the ideal compensation term satisfies $\boldsymbol{W}_{opt}^T \boldsymbol{\Phi}(\tilde{\boldsymbol{x}}) = \boldsymbol{f}_T(\tilde{\boldsymbol{x}})$. The weight update law is designed by considering a Lyapunov function candidate $\boldsymbol{V}(t)$ that accounts for both tracking error and the weight estimation error. Due to the presence of neural network approximation error and external disturbances, the condition $\dot{\boldsymbol{V}} \leq 0$ cannot be guaranteed everywhere. Instead, to add robustness and avoid the drift of neural network weights, a leakage term α is introduced, leading to a dissipative Lyapunov derivative inequality from which it is possible to demonstrate Uniform Ultimate Boundedness (UUB) of the tracking error [66–68].

$$\dot{\boldsymbol{W}}(t) = \Gamma_{\mathrm{NN}} \boldsymbol{\Phi}(\tilde{\boldsymbol{x}}) \boldsymbol{e}^T(t) \boldsymbol{P} \tilde{\boldsymbol{B}} - \alpha \boldsymbol{W}(t) \tag{26}$$

where $\Gamma_{\mathrm{NN}} > 0$ represents the scalar adaptation rate, $\alpha > 0$ is the sigma-modification damping coefficient, and $\boldsymbol{P} = \boldsymbol{P}^T > 0$ is the unique solution to the algebraic Lyapunov equation $\boldsymbol{A}_r^T \boldsymbol{P} + \boldsymbol{P} \boldsymbol{A}_r = -\boldsymbol{Q}$, computed for a given symmetric positive-definite matrix $\boldsymbol{Q}$.

4. Results

To assess the performance of the proposed architecture, a comprehensive evaluation campaign, comprising a numerical sensitivity analysis and experimental tests, has been carried out. The benchmarking platform chosen for this work is the Crazyflie 2.1+ micro-quadrotor, whose nominal geometric, inertial and mass parameters are summarized in Table 1.

Table 1. Nominal geometric, mass, and inertial properties of the Crazyflie 2.1+ micro-quadrotor utilized for numerical synthesis and experimental benchmarking.

Parameter	Symbol	Value
Total Mass	m	0.031 kg
Rotor-to-CG Distance (x-axis)	l	0.049 m
Rotor-to-CG Distance (y-axis)	b	0.049 m
Inertia Tensor Diagonal	$\boldsymbol{J}$	$\mathrm{diag}(1.0, 1.0, 3.0) \times 10^{-5}$ kg · m^2

Angles and angular rates are controlled in the baseline controller with a cascade structure. To design the control system, the weighting functions $\boldsymbol{Q}_e(s)$, $\boldsymbol{Q}_u(s)$, and $\boldsymbol{Q}_y(s)$ are defined as first-order transfer functions.

The weighting matrices are structured in diagonal forms as

$$\begin{aligned}
\boldsymbol{Q}_{e,CAS}(s) &= \mathrm{diag}([Q^1_{e,CAS}, Q^2_{e,CAS}, Q^3_{e,CAS}]) \\
\boldsymbol{Q}_{u,CAS}(s) &= \mathrm{diag}([Q^1_{u,CAS}, Q^2_{u,CAS}, Q^3_{u,CAS}]) \\
\boldsymbol{Q}_{y,CAS}(s) &= \mathrm{diag}([Q^1_{y,CAS}, Q^2_{y,CAS}, Q^3_{y,CAS}]) \\
\boldsymbol{Q}_{e,SAS}(s) &= \mathrm{diag}([Q^1_{e,SAS}, Q^2_{e,SAS}, Q^3_{e,SAS}]) \\
\boldsymbol{Q}_{u,SAS}(s) &= \mathrm{diag}([Q^1_{u,SAS}, Q^2_{u,SAS}, Q^3_{u,SAS}]) \\
\boldsymbol{Q}_{y,SAS}(s) &= \mathrm{diag}([Q^1_{y,SAS}, Q^2_{y,SAS}, Q^3_{y,SAS}])
\end{aligned}$$

The general structure used for $Q^k_{e,j}$ with $k \in \{1,2,3\}$ is

$$Q^k_{e,j}(s) = \epsilon^k_j \cdot \frac{s + \frac{M^k_j}{\epsilon^k_j} \cdot \omega^{k,b}_j}{s + \omega^{k,b}_j} \tag{27}$$

where M^k_j denotes the low-frequency gain, ϵ^k_j represents the high-frequency gain, and $\omega^{k,b}_j$ is the target bandwidth frequency. The control input weights $Q^k_{u,j}(s)$ and the robustness weights $Q^k_{y,j}(s)$ share another generalized first-order formulation:

$$Q^k_{u,j}(s) = G^{k,\infty}_{u,j} \cdot \frac{s + \omega^{k,z}_{u,j}}{s + \omega^{k,p}_{u,j}} \tag{28}$$

$$Q^k_{y,j}(s) = G^{k,\infty}_{y,j} \cdot \frac{s + \omega^{k,p}_{y,j}}{s + \omega^{k,p}_{y,j}} \tag{29}$$

where $G^{k,\infty}_{u,j}$ is the asymptotic high-frequency gain. The zero frequency $\omega^{k,z}_{u,j}$ ($\omega^{k,z}_{y,j}$) and pole frequency $\omega^{k,p}_{u,j}$ ($\omega^{k,p}_{y,j}$) are uniquely specified by defining a target intermediate gain $G^{k,1}_{u,j}$ ($G^{k,1}_{y,j}$) at a transition frequency $\omega^{k,c}_{u,j}$ ($\omega^{k,c}_{y,j}$), alongside the low-frequency gain limit $G^{k,0}_{u,j}$ ($G^{k,0}_{y,j}$):

$$\omega^{k,p}_{u,j} = \omega^{k,c}_{u,j} \sqrt{\frac{\left(G^{k,1}_{u,j}\right)^2 - \left(G^{k,0}_{u,j}\right)^2}{\left(G^{k,\infty}_{u,j}\right)^2 - \left(G^{k,1}_{u,j}\right)^2}}, \quad \omega^{k,z}_{u,j} = \omega^{k,p}_{u,j} \cdot \frac{G^{k,0}_{u,j}}{G^{k,\infty}_{u,j}} \tag{30}$$

$$\omega^{k,p}_{y,j} = \omega^{k,c}_{y,j} \sqrt{\frac{\left(G^{k,1}_{y,j}\right)^2 - \left(G^{k,0}_{y,j}\right)^2}{\left(G^{k,\infty}_{y,j}\right)^2 - \left(G^{k,1}_{y,j}\right)^2}}, \quad \omega^{k,z}_{y,j} = \omega^{k,p}_{y,j} \cdot \frac{G^{k,0}_{y,j}}{G^{k,\infty}_{y,j}} \tag{31}$$

The complete set of parameters for both the inner (p,q,r-axes) and the outer loops (ϕ,θ,ψ-axes) is summarized in Tables 2–4. The H_∞ bounds obtained with these parameters are $\gamma_{\infty,CAS} \approx 167$ and $\gamma_{\infty,SAS} \approx 2$.

Table 2. Design parameters for the performance weighting functions $\boldsymbol{Q}_e(s)$ of the inner (p,q) and outer (ϕ,θ) loops.

j	k	ϵ^k_j	M^k_j	$\omega^{k,b}_j$
SAS	1	10^3	$2\pi \cdot 1.0$ rad/s	10^{-2}
SAS	2	10^3	$2\pi \cdot 1.0$ rad/s	10^{-2}
SAS	3	10^{-2}	$2\pi \cdot 10^{-2}$ rad/s	-
CAS	1	10^4	$2\pi \cdot 1.0$ rad/s	10^{-3}
CAS	2	10^4	$2\pi \cdot 1.0$ rad/s	10^{-3}
CAS	3	10^2	$2\pi \cdot 10^{-2}$ rad/s	-

Table 3. Design parameters for the control input $\boldsymbol{Q}_u(s)$ weighting functions of the inner (p, q) and outer (ϕ, θ) loops.

j	k	$G_{u,j}^{k,0}$	$\omega_{u,j}^{k,c}$	$G_{u,j}^{k,1}$	$G_{u,j}^{k,\infty}$
SAS	1	100	$2\pi \cdot 1.0$ rad/s	1000	10^6
SAS	2	100	$2\pi \cdot 1.0$ rad/s	1000	10^6
SAS	3	0	$2\pi \cdot 1.0$ rad/s	-	1.0
CAS	1	0.01	$2\pi \cdot 0.25$ rad/s	0.1	1.0
CAS	2	0.01	$2\pi \cdot 0.25$ rad/s	0.1	1.0
CAS	3	-	-	-	

Table 4. Design parameters for the robustness $\boldsymbol{Q}_y(s)$ weighting functions of the inner (p, q) and outer (ϕ, θ) loops.

j	k	$G_{y,j}^{k,0}$	$\omega_{y,j}^{k,c}$	$G_{y,j}^{k,1}$	$G_{y,j}^{k,\infty}$
SAS	1	0.2	$2\pi \cdot 1.0$ rad/s	2.0	50
SAS	2	0.2	$2\pi \cdot 1.0$ rad/s	2.0	50
SAS	3	-	-	-	
CAS	1	0.06	$2\pi \cdot 1.0$ rad/s	0.6	100
CAS	2	0.06	$2\pi \cdot 1.0$ rad/s	0.6	100
CAS	3	-	-	-	

To evaluate the tracking performance and statistical dispersion of the flight control system, the continuous-time states are discretized considering a constant sampling period T_s, such that the generic time instant is expressed as $t_k = kT_s$, where $k \in \mathbb{N}$ denotes the discrete-time sample index.

The reference model used to train the neural network is represented by the following transfer function:

$$G_{ref}(s) = \frac{16}{s^2 + 8s + 16} \tag{32}$$

Denoting with $\eta(k) \triangleq \eta(kT_s)$ and $\eta_{\text{ref}}(k) \triangleq \eta_{\text{ref}}(kT_s)$ the generic component of the actual and reference attitude vectors sampled at the k-th step, respectively, the discrete-time tracking error is defined as

$$e_\eta(k) = \eta(k) - \eta_{\text{ref}}(k) \tag{33}$$

To quantitatively assess the tracking accuracy over a finite observation time window composed of n_k total samples, the Mean Absolute Error (MAE) and the Standard Deviation (SD) are introduced as performance metrics [69]:

$$MAE_\eta = \frac{1}{n_k} \sum_{k=1}^{n_k} |e_\eta(k)| \tag{34}$$

$$SD_\eta = \sqrt{\frac{1}{(n_k - 1)} \sum_{k=1}^{n_k} \left(e_\eta(k) - \mu_\eta\right)^2} \tag{35}$$

where $\mu_\eta = \frac{1}{n_k} \sum_{k=1}^{n_k} e_\eta(k)$ represents the mean algebraic tracking error.

Moreover, the smoothness of the control signal is assessed through the Total Variation Δ_{TV} [70], defined as the sum:

$$\Delta_{TV} = \sum_{k=1}^{n_k} |u_{k+1} - u_k| \tag{36}$$

where u_k is the control signal at the k-th time step. A lower value of Δ_{TV} indicates a smoother control profile, which is desirable to minimize actuator chattering.

4.1. Numerical Sensitivity Analysis

The baseline robust H_∞ synthesis loop was derived via a digital twin of the micro-UAV implemented within a MATLAB/Simulink R2025b (MathWorks, Natick, MA, USA) environment, operating at a fixed sample time of $T_s = 2 \times 10^{-3}$ s. To model parametric uncertainties and exogenous disturbances, an unbalancing CG displacement was injected, generating an unmodeled persistent pitching and rolling moment of $M_d = 9.81 \cdot 10^{-4}$ N · m, which replicates a weight of $2 \cdot 10^{-3}$ kg mounted at $5 \cdot 10^{-2}$ m from the CG.

To ensure enhanced distribution capability within the disturbance domain, the adaptive neural network augmentation utilizes DReLU, a piecewise linear activation function, which is an enhanced version of the standard Rectified Linear Unit (ReLU) function [71].

To address cross-channel coupling terms, a multidimensional neural network mapping the six-dimensional state $\tilde{\mathbf{x}}$ is needed. However, in this paper, to reduce RAM consumption, cross-channel couplings are considered negligible and three separate NNs are used, one for every angle, mapping the experienced disturbance to a specific attitude configuration. The activation function is parameterized as

$$\mathrm{DReLU}(x) = \max(\zeta_{\mathrm{bias}}, x) \tag{37}$$

where $\zeta_{\mathrm{bias}} \in \left[-\frac{\pi}{6} : \frac{\pi}{6}\right]$ represents the specific activation threshold.

The numerical validation campaign is structured to replicate the real-time online learning behavior of the adaptive controller, emphasizing its ability to operate without any prior offline pre-training phase. At the beginning of the simulation, the NN weights are initialized to zero ($\boldsymbol{W}(0) = \mathbf{0}$), meaning that the adaptive component provides no control action initially, leaving the stabilization completely to the baseline H_∞ loop.

Since the network weights must be updated dynamically from scratch in-flight, the simulation profile consists of an uninterrupted flight sequence divided into two subsequent operational phases: an initial online training window and a subsequent closed-loop evaluation phase. While the neural network learning remains continuously active, the division of the flight profile into training and evaluation phases emulates a realistic operational scenario. Specifically, before the actual flight, the adaptive compensator is given 82 s to learn the disturbances, after which a hover is maintained for an additional 18 s to evaluate learning retention.

Specifically, the training profile comprises five consecutive doublets with increasing peak-to-peak amplitudes of 10 deg, 20 deg, and 30 deg, with a period of $T_d = 5$ s. The evaluation test phase incorporates a cascade of four doublets (10 deg, 20 deg, 30 deg at $T_d = 5$ s and a persistent 30 deg pulse of $T_d = 15$ s), followed by a trapezoidal command profile with a constant rate of 3 deg/s (see Figure 6), which is used to evaluate the performance metrics to avoid taking into account the jump of the doublets.

4.1.1. Sensitivity Analysis #1: Hyperparameter Tuning (Γ_{NN}, α)

A preliminary investigation was performed to characterize the joint influence of the learning rate Γ_{NN} and the sigma-modification decay parameter α on tracking capability. The analyzed grid configuration is specified in Table 5.

Table 5. Discrete search grid for the adaptive learning rate Γ_{NN} and the sigma-modification parameter α evaluated during Sensitivity Analysis I.

Hyperparameter	Explored Set
Γ_{NN}	$[1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}]$
α	$[1 \times 10^{-4}, 1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}]$

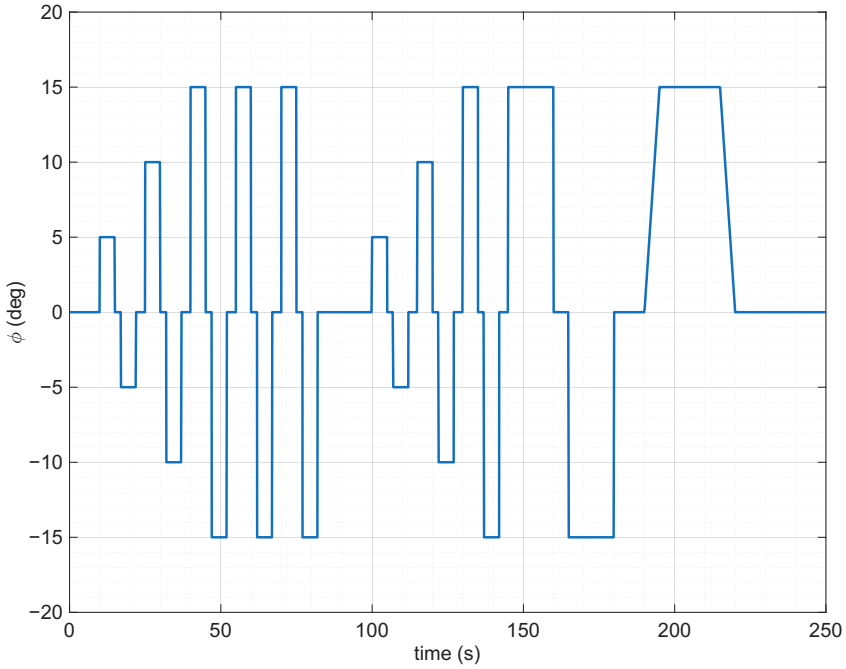


Figure 6. Multi-step reference command history $\phi_{\text{ref}}(t)$ applied across the continuous time window, comprising the initial online adaptation doublets ($0\text{ s} \leq t \leq 90\text{ s}$), the closed-loop evaluation doublets ($90\text{ s} \leq t \leq 190\text{ s}$), and the final smoothness tracking trapezoidal profile.

The closed-loop tracking behavior during both the training and testing phase is illustrated in Figures 7–10. As shown in Figures 7 and 8, a lower adaptation gain ($\Gamma_{\text{NN}} = 1 \times 10^{-4}$) slows down the error convergence rate, yielding worse tracking profiles. Conversely, excessive values ($\Gamma_{\text{NN}} = 5 \times 10^{-3}$) induce high-frequency oscillatory components and chattering phenomena as shown by the increment in Δ_{TV}, compromising the local loop stability. On the other hand, excessive leakage regulation via the decay factor α generates a more regular signal, as shown by the decrease in SD, at the cost of a steady-state tracking error resulting in a higher MAE.

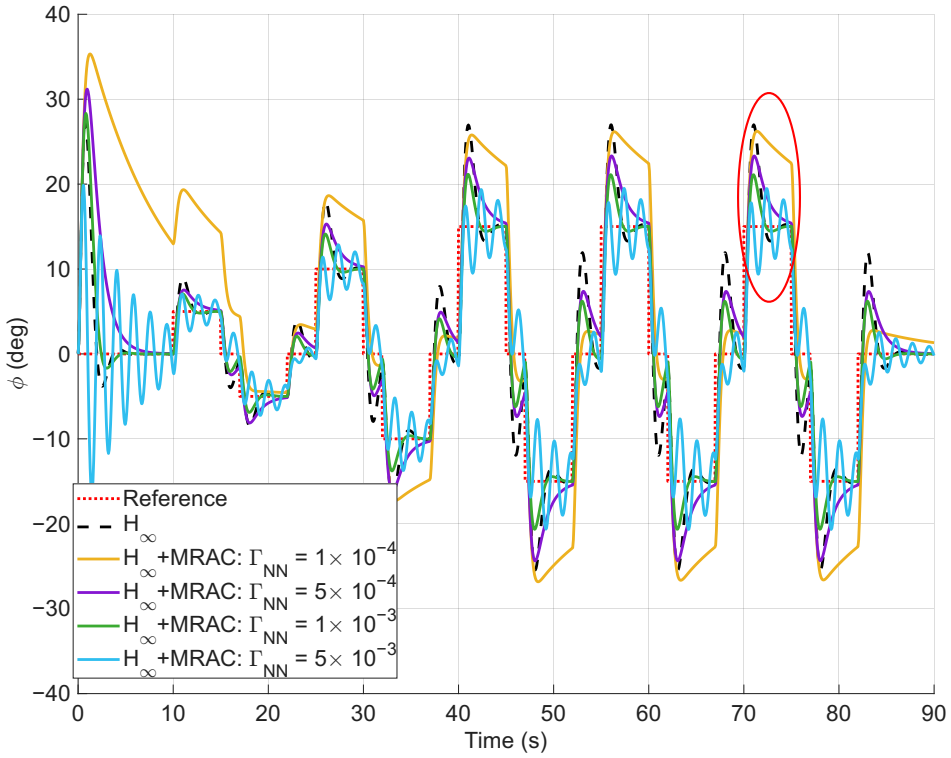
These results are confirmed by the numerical indices shown in Tables 6 and 7. Significant metric degradation occurs for hyperparameter choices exceeding $\Gamma_{\text{NN}} > 1 \times 10^{-3}$ and $\alpha > 1 \times 10^{-3}$.

Table 6. Quantitative performance metrics (MAE_ϕ, SD_ϕ) evaluated from 195 s to 220 s as a function of the adaptation rate Γ_{NN} under a constant leakage factor $\alpha = 1 \times 10^{-4}$.

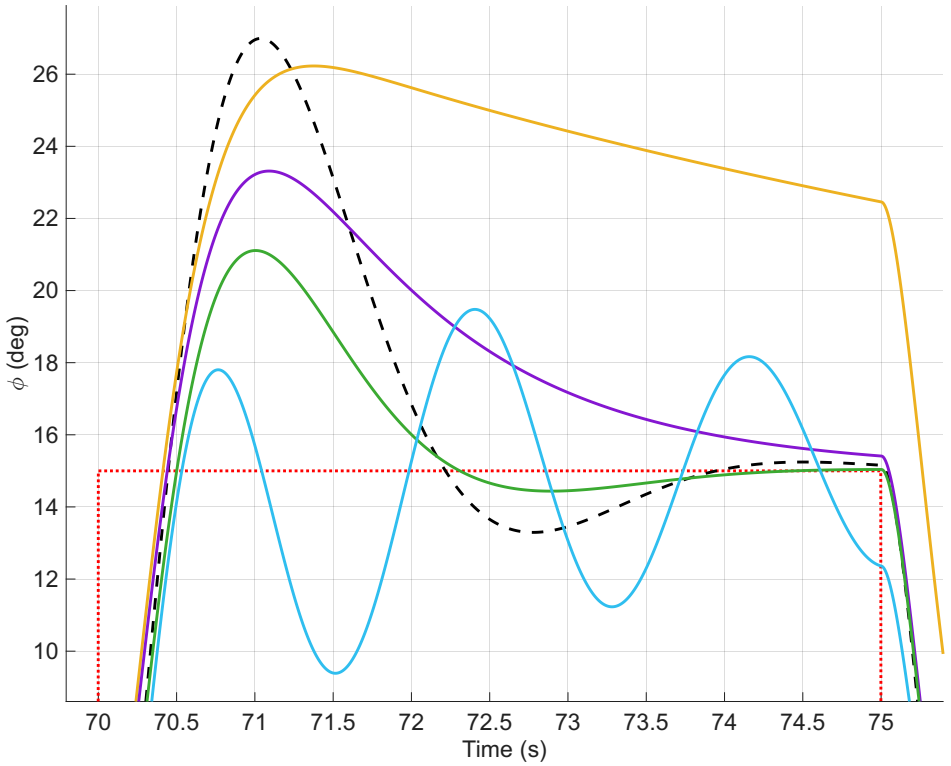
$\boldsymbol{\Gamma_{NN}}$	$\boldsymbol{MAE_\phi}$ [deg]	$\boldsymbol{SD_\phi}$ [deg]	$\boldsymbol{\Delta_{TV}}$ [Nm]
$1 \cdot 10^{-4}$	3.766	2.357	$4.5424 \cdot 10^{-6}$
$5 \cdot 10^{-4}$	0.287	0.666	$3.4211 \cdot 10^{-6}$
$1 \cdot 10^{-3}$	0.051	0.208	$3.5573 \cdot 10^{-6}$
$5 \cdot 10^{-3}$	0.032	0.186	$1.2178 \cdot 10^{-5}$

Table 7. Quantitative performance metrics (MAE_ϕ, SD_ϕ) evaluated from 195 s to 220 s as a function of the leakage parameter α under a constant adaptation rate $\Gamma_{\text{NN}} = 1 \times 10^{-3}$.

$\boldsymbol{\alpha}$	$\boldsymbol{MAE_\phi}$ [deg]	$\boldsymbol{SD_\phi}$ [deg]	$\boldsymbol{\Delta_{TV}}$ [Nm]
$1 \cdot 10^{-4}$	0.051	0.208	$3.557 \cdot 10^{-6}$
$1 \cdot 10^{-3}$	0.090	0.207	$3.556 \cdot 10^{-6}$
$1 \cdot 10^{-2}$	0.470	0.204	$3.556 \cdot 10^{-6}$
$1 \cdot 10^{-1}$	3.957	0.167	$3.504 \cdot 10^{-6}$

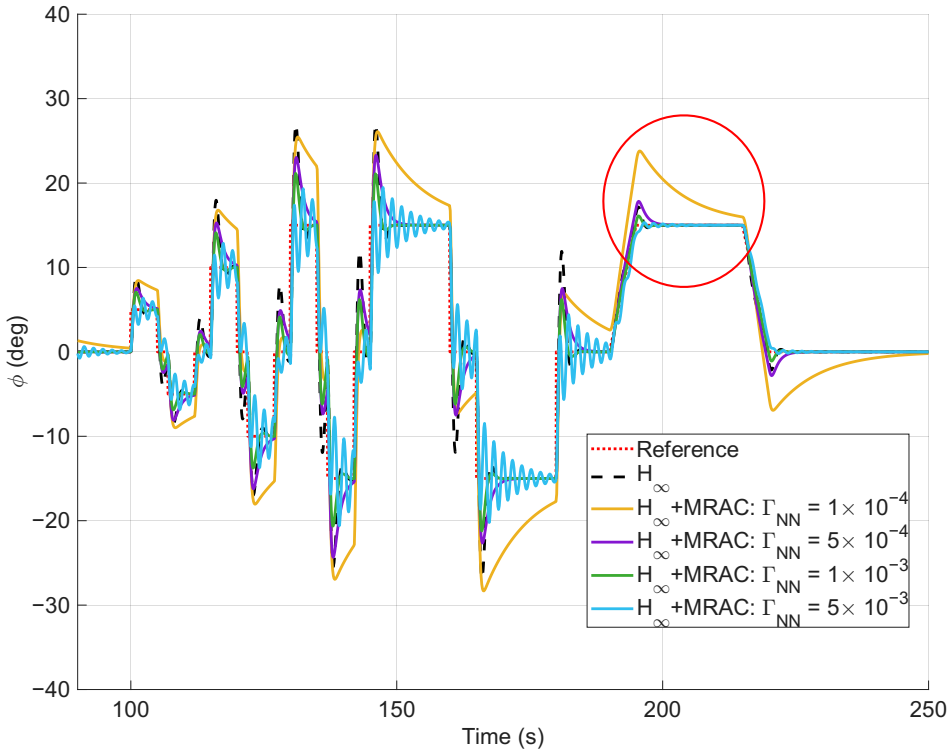


(**a**)

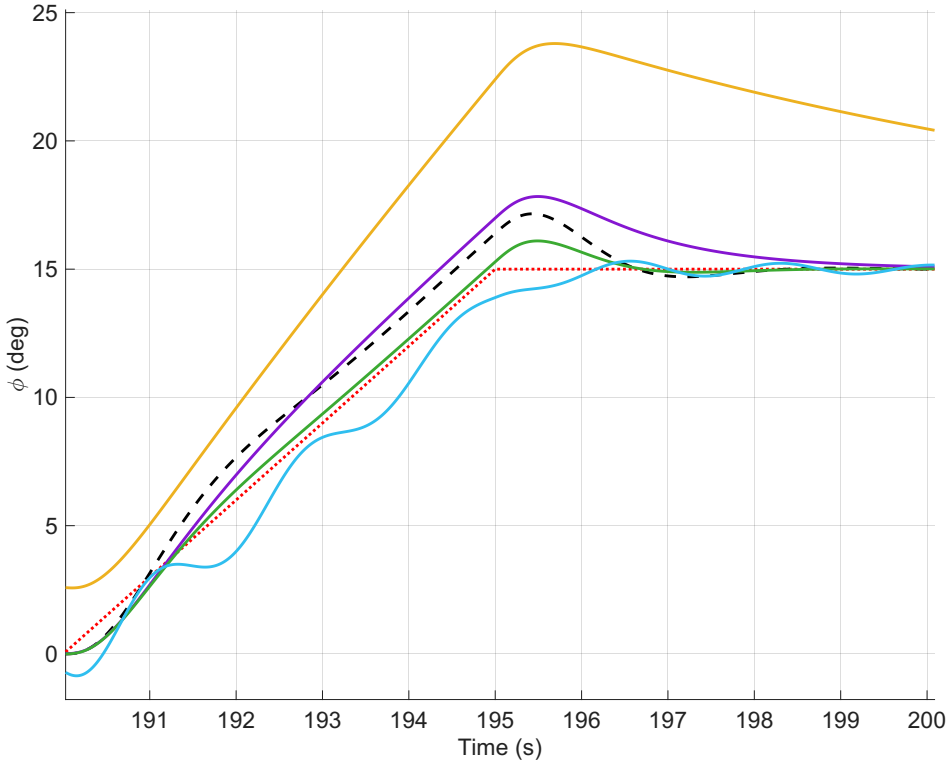


(**b**)

Figure 7. (**a**) Transient tracking response of the roll angle $\phi(t)$ during the online training phase across a range of adaptation rates Γ_{NN} (with $\alpha = 1 \times 10^{-4}$); the red circle indicates the region magnified in (**b**). (**b**) The magnified section highlights the settling behavior and high-frequency oscillations at peak values.

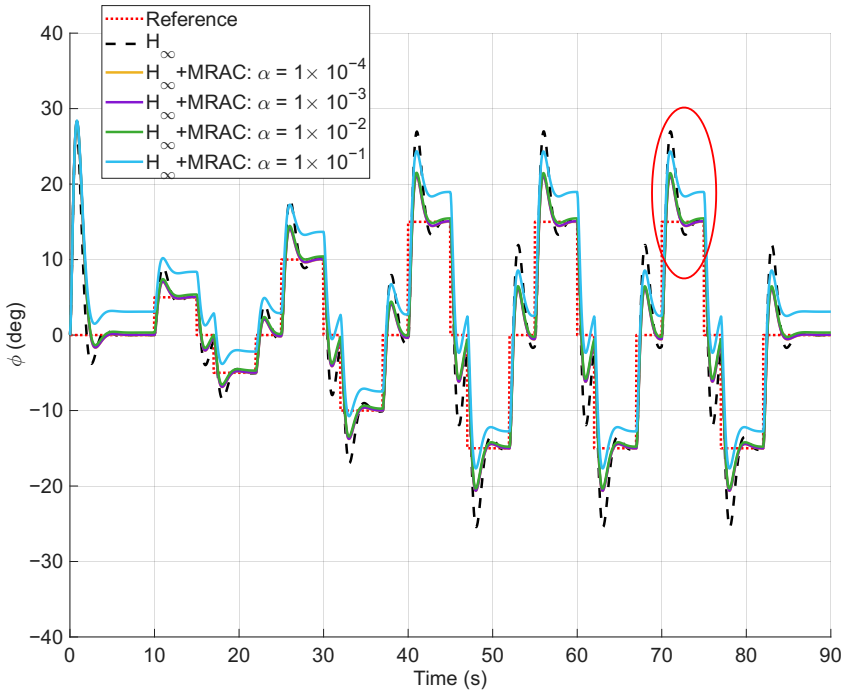


(**a**)

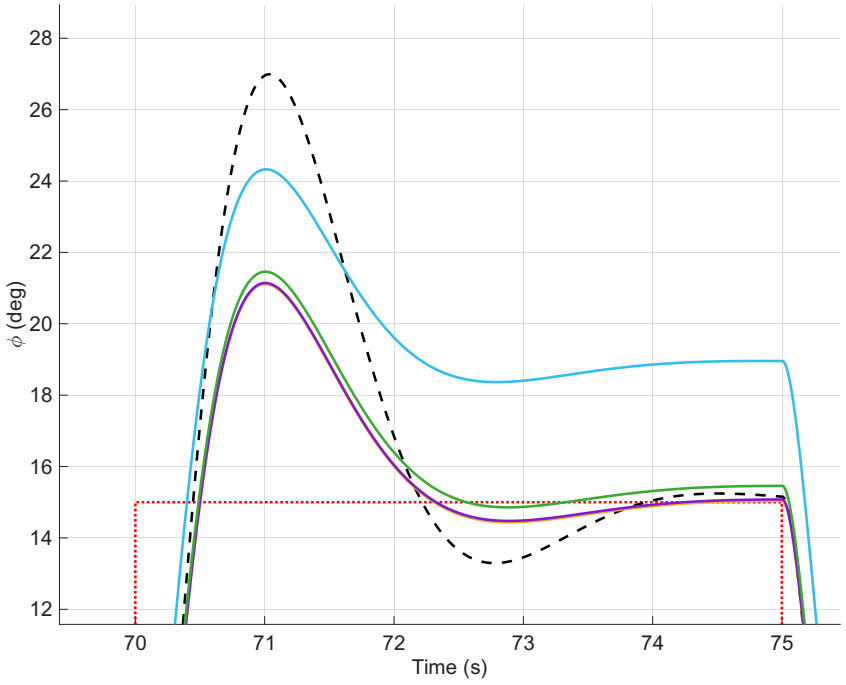


(**b**)

Figure 8. (**a**) Closed-loop attitude response validation of the roll angle $\phi(t)$ during the testing phase across different adaptation rates Γ_{NN} (with $\alpha = 1 \times 10^{-4}$); the red circle indicates the region magnified in (**b**). (**b**) The magnified section highlights tracking offsets and steady-state error mitigation on the persistent step command.



(**a**)



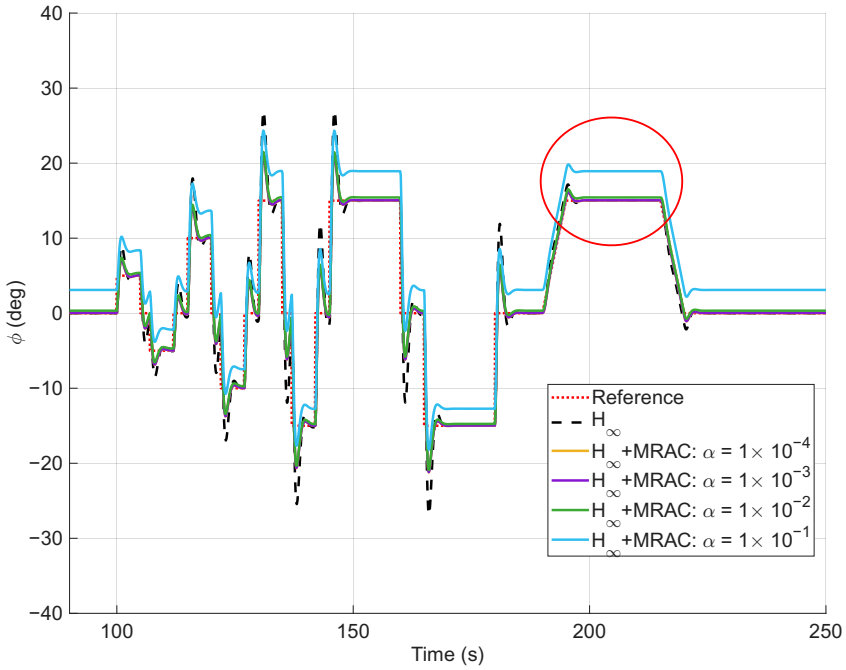
(**b**)

Figure 9. (**a**) Transient tracking response of the roll angle $\phi(t)$ during the online training phase across different leakage damping parameter α (with $\Gamma_{\mathrm{NN}} = 1 \times 10^{-3}$); the red circle indicates the region magnified in (**b**). (**b**) The magnified section emphasizes the drift suppression transient at $t = 70$ s.

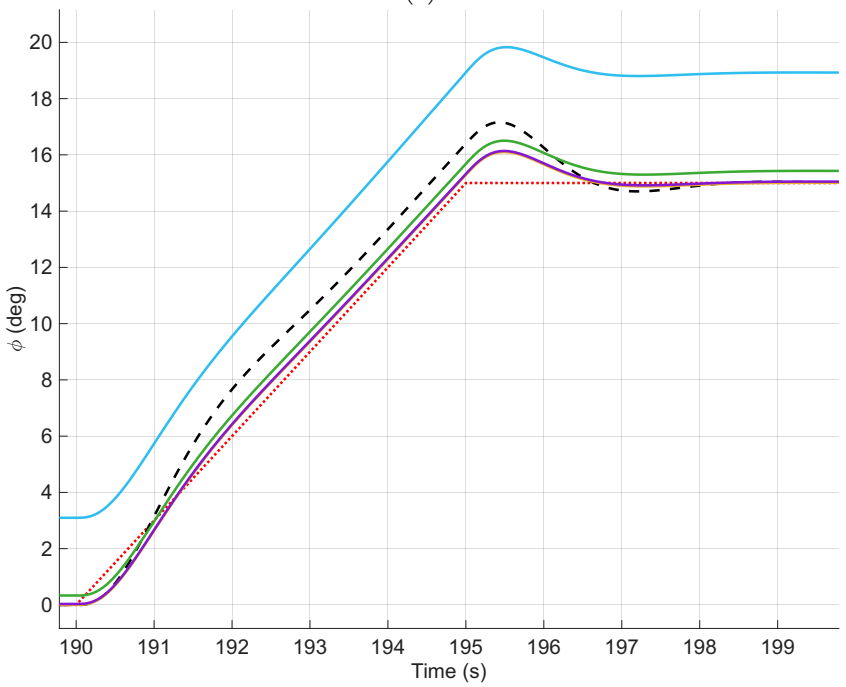
4.1.2. Sensitivity Analysis #2: Hidden-Layer Sizing (N)

The second campaign focused on optimizing the hidden-network-layer dimension, exploring $N \in [5, 10, 15, 20, 25, 30, 35]$.

The comparison is depicted in Figures 11 and 12, while Table 8 reports the comparison in terms of the performance indices, defined in (34) and (35), considering different hidden-layer configurations.



(**a**)



(**b**)

Figure 10. (**a**) Closed-loop attitude response validation of the roll angle $\phi(t)$ during the testing phase for different values of the leakage damping parameter α (with $\Gamma_{\mathrm{NN}} = 1 \times 10^{-3}$); the red circle indicates the region magnified in (**b**). (**b**) The magnified section details the degradation in tracking precision caused by excessive weighting.

Increasing the hidden dimension up to $N = 15$ significantly enhances approximation capacity. However, expanding the dimension further ($N \geq 25$) triggers local high-frequency oscillations given the selected baseline parameters ($\Gamma_{\mathrm{NN}}, \alpha$), leading to performance limitations [72].

As depicted in Table 8, while the steady-state tracking error saturates, the Total Variation Δ_{TV} further increases, indicating that the over-parameterized network attempts to fit high-frequency transient tracking errors and measurement noise rather than purely

reconstructing the underlying lumped disturbance, resulting in actuator chattering. Since persistent excitation (PE) cannot be guaranteed uniformly across every discrete DReLU region during rapid transient maneuvers, an excessive number of localized basis functions allows individual weights to rapidly over-compensate for instantaneous transient errors along the trajectory. This leads to neural coupling transients before the σ-modification damping term is able to stabilize the weights.

Table 8. Tracking accuracy metrics (MAE_ϕ, SD_ϕ) evaluated from 195 s to 220 s across different hidden-layer dimensions (N) to determine the optimal onboard balance between modeling precision and computational burden.

Number of Neurons	MAE_ϕ [deg]	SD_ϕ [deg]	Δ_{TV} [Nm]
5	0.420	0.856	$3.5648 \cdot 10^{-6}$
10	0.111	0.347	$3.4402 \cdot 10^{-6}$
15	0.051	0.208	$3.5573 \cdot 10^{-6}$
20	0.014	0.113	$3.7627 \cdot 10^{-6}$
25	0.002	0.086	$3.9083 \cdot 10^{-6}$
30	0.008	0.076	$4.1426 \cdot 10^{-6}$
35	0.013	0.080	$4.3299 \cdot 10^{-6}$

While the configurations for 15 and 20 neurons yield close metric performance, the $N = 15$ topology was selected for deployment. This choice is the result of practical design trade-off between performance and resource requirements: while increasing the number of neurons past 15 results in a reduction in MAE (as shown in Table 8), using fewer neurons translates into a smaller number of adaptive parameters to store and compute at each controller cycle, reducing memory usage and computational burden. Moreover, with less neurons, a better damped signal during transient response is achieved, as shown in Figure 11.

4.2. Experimental Results

To demonstrate the real-time feasibility of the proposed approach, the DReLU-parameterized network was implemented on the embedded architecture of a Crazyflie 2.1+ micro-UAV and benchmarked against a literature-standard RBF adaptive scheme [73]. The Crazyflie firmware was compiled using the arm-none-eabi-gcc toolchain targeting the ARM Cortex-M4 architecture with the size-optimization flag -Os (as specified in the standard firmware). RBF architectures provide localized function approximation fields written as

$$\mathrm{RBF}(x) = e^{-\sigma(x+\beta)^2} \tag{38}$$

where σ and β represent the RBF kernel width and center vector, respectively. Despite their mathematical flexibility, RBF schemes scale poorly with hardware constraints due to high computational overhead of evaluating exponential functions. The baseline synthesis parameter sets for both architectures are organized in Table 9, featuring parameters adapted from standard verification configurations [74]. It is worth noting that the two neural networks have comparable architecture as they use the same number of weights and neurons, placed at identical positions to map the space equivalently.

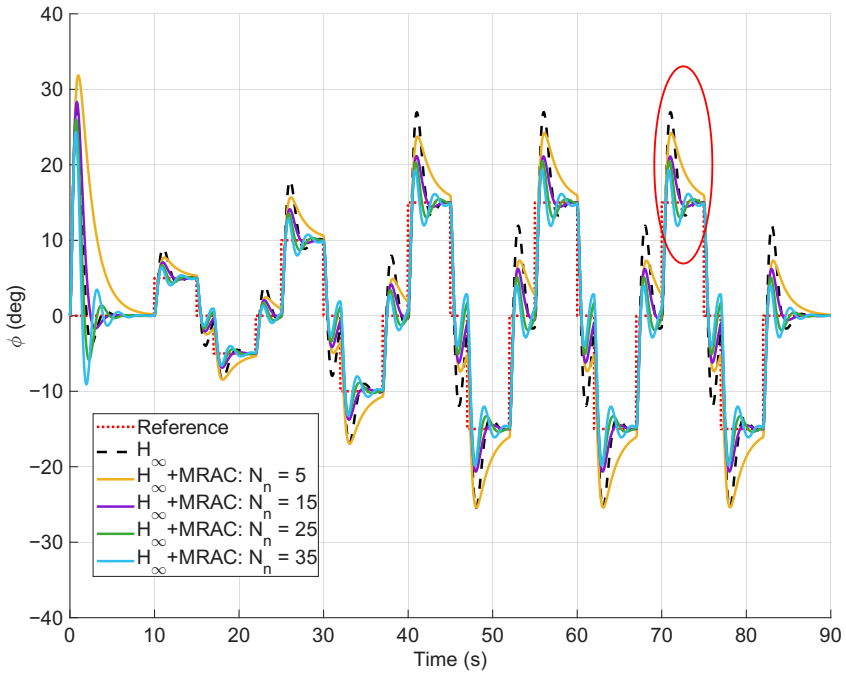
A comparison between the baseline controller and the baseline-plus-compensation architecture was omitted because the standard H_∞ controller employed here lacks an integral component, making such a benchmark unfair.

The test setup used a low-friction tilting calibration platform restricting the vehicle to a three-degrees-of-freedom constrained rotational body (Figure 13). The unmodeled

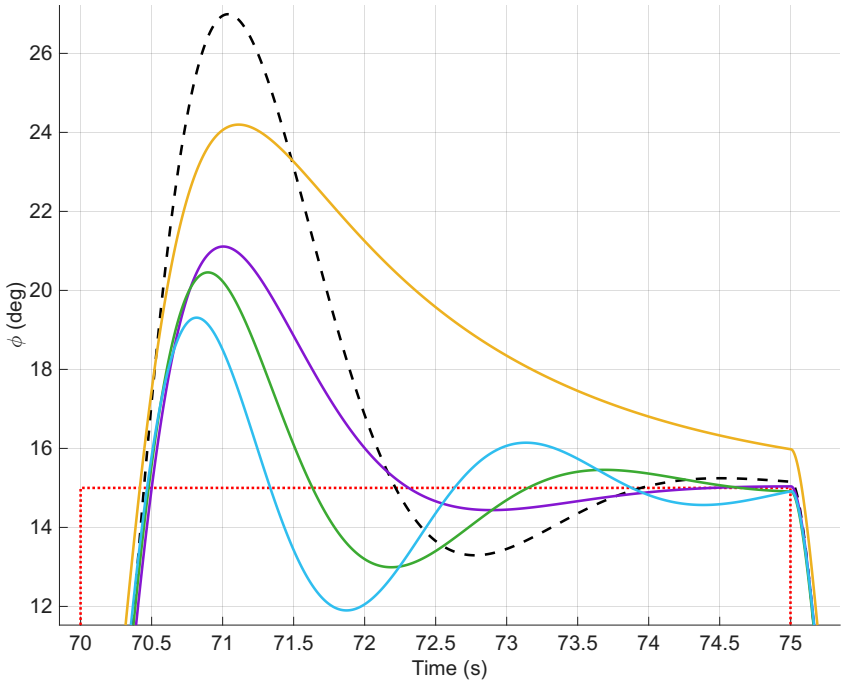
asymmetrical disturbance torque was physically generated by adding a calibrated payload mass of 2×10^{-3} kg beneath the M1 rotor axis at $\approx 5 \cdot 10^{-2}$ m from the center of mass.

The training and test phases use the same reference signals as the numerical assessment depicted in Figure 6.

The closed-loop control histories for both validation schemes are shown in Figure 14. Both adaptive architectures successfully suppress the persistent unbalancing roll moment, showing minimal overshoots and negligible cross-channel coupling into the pitching and yawing fields.

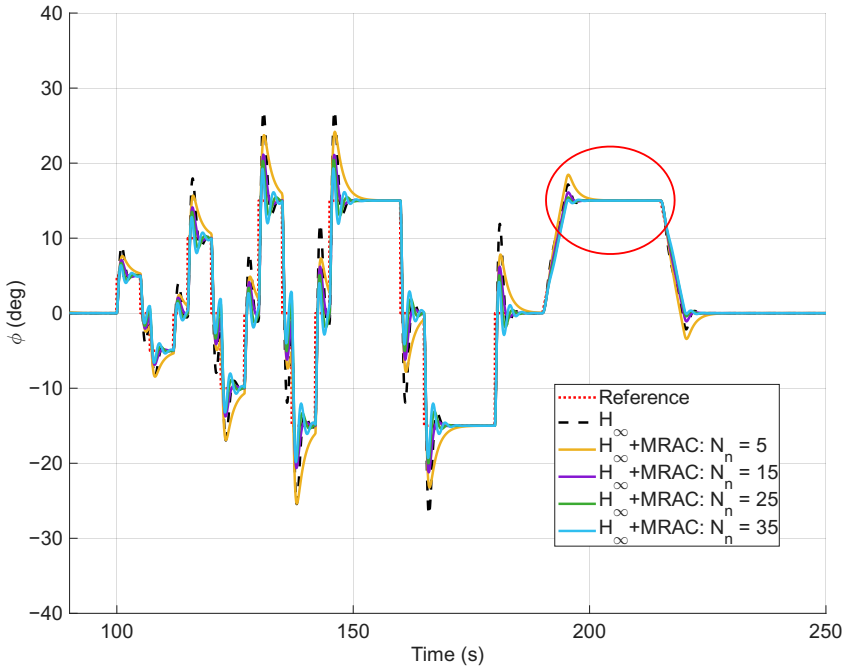


(**a**)

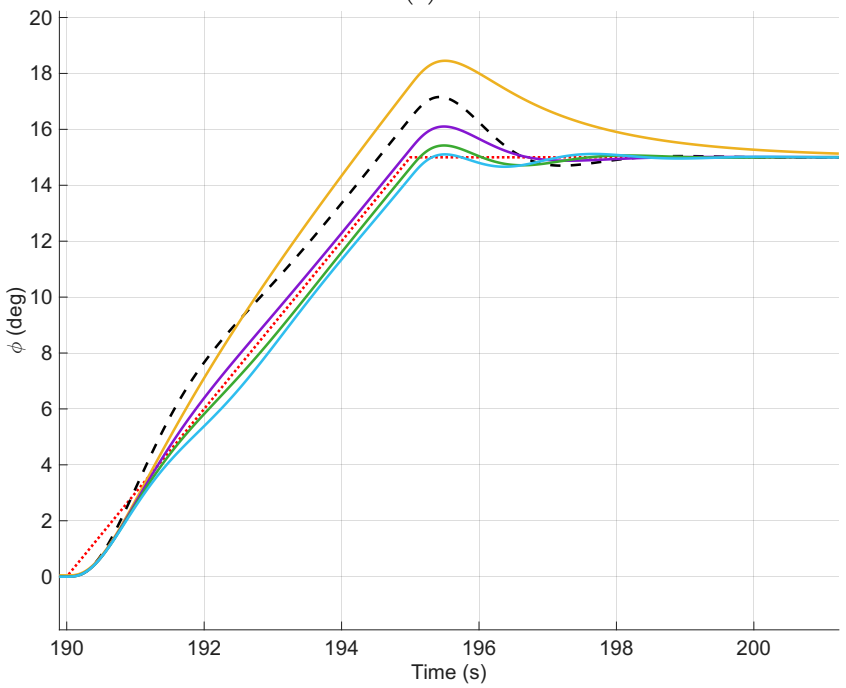


(**b**)

Figure 11. (**a**) Training phase response under variations in the hidden-layer size N; the red circle indicates the region magnified in (**b**). (**b**) The magnified plot emphasizes the under-modeling behavior of $N = 5$ versus the optimal convergence of the selected $N = 15$ node topology.



(**a**)



(**b**)

Figure 12. (**a**) Tracking accuracy validation during the evaluation phase across different hidden-layer configurations N; the red circle indicates the region magnified in (**b**). (**b**) The magnified plot highlights tracking degradation and oscillatory issues occurring at higher dimensions ($N \geq 25$).

In both cases, the memory usage is linked to the number of adaptive weights updated online, impacting the RAM. With 15 neurons, considering floating point representation, the neural network occupies only 60 bytes, which is a negligible percentage of the total memory of the considered processor. The primary distinction lies in the processing costs. Tables 10 and 11 summarize the results, computed as the mean of ten independent tests. To measure the computational load of the two controllers running at a fixed frequency of 250 Hz, the execution time of the entire control task was measured at each cycle using hardware timers and compared against the measured duration of the Central Processing

Unit (CPU) cycle. The CPU utilization metrics—measured directly on the STM32F405 microcontroller (ARM Cortex-M4 operating core)—highlight that the linear nature of the DReLU network maintains the execution burden at 1.464%, achieving a 52.6% reduction in computational resource consumption relative to the RBF structure (3.094%) without sacrificing tracking quality.

Table 9. Core parameterization and operational initialization profiles compiled for the real-time embedded implementation of the DReLU and benchmark RBF adaptive algorithms. Both architectures use the same number of neurons.

Parameter	Value
Neuron Count (N)	15
DReLU Learning Rate (Γ_{DReLU})	1×10^{-3}
DReLU Decay Factor (α_{DReLU})	1×10^{-4}
DReLU Bias (ζ_{bias})	$\left[-\frac{\pi}{6} : \frac{\pi}{6}\right]$
RBF Learning Rate (Γ_{RBF})	5.0
RBF Decay Factor (α_{RBF})	1×10^{-3}
Initial Weights ($\boldsymbol{W}(0)$)	$\mathbf{0}_{(N+1)\times 3}$
RBF Centers (β)	$\left[-\frac{\pi}{6} : \frac{\pi}{6}\right]$
RBF Scaling Factor (σ)	0.08



Figure 13. Real-time validation testbench setup: the Crazyflie 2.1+ micro-UAV is mounted to a low-friction tilting gimbal system restricting the vehicle to rotational attitude motions, featuring a 2×10^{-3} kg calibrated offset payload to generate an asymmetrical unmodeled disturbance moment.

Table 10. Onboard tracking performance indices and processing core footprints evaluated on the embedded STM32F405 microcontroller, highlighting the resource consumption efficiency of the proposed DReLU architecture over the RBF benchmark. The metrics in this table were evaluated for the last trapezoidal signal from 195 s to 220 s to avoid including the discontinuity of the doublets in the computation.

Activation Function	MAE_{ϕ} [deg]	SD_{ϕ} [deg]	$CPU\%$	$CPU\%$ (Worst Case)
Proposed DReLU	1.114	1.020	1.464	1.710
Standard RBF	1.231	1.172	3.094	3.472

Table 11. Experimental results for θ and ψ angles, evaluated for the last trapezoidal signal from 195 s to 220 s.

Activation Function	MAE_{θ} [deg]	SD_{θ} [deg]	MAE_{ψ} [deg]	SD_{ψ} [deg]
Proposed DReLU	0.789	0.964	1.130	1.413
Standard RBF	1.405	1.717	2.523	2.807

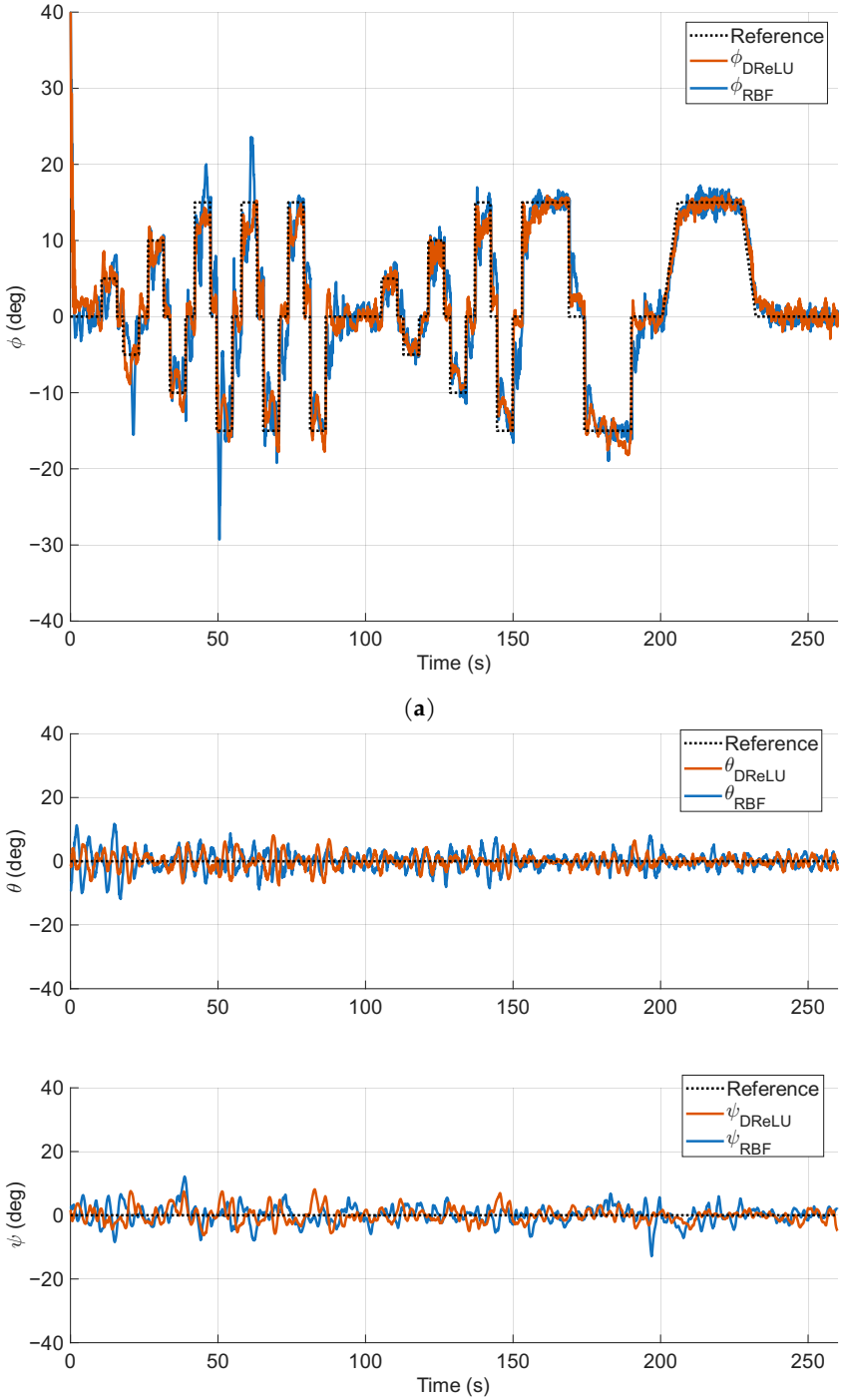


Figure 14. Experimental real-time flight telemetry histories (Roll ϕ (**a**) given the reference signal of Figure 6; Pitch θ and Yaw ψ (**b**) given a constant zero reference signal) under persistent unbalancing moment, contrasting the baseline H_{∞} controller performance augmented via the proposed linear DReLU network against the augmentation with the traditional RBF framework.

5. Conclusions

This paper presents the design, implementation and experimental validation of an augmented robust flight control architecture tailored for resource-constrained micro-UAV platforms. In particular, this research focuses on attitude control by validating the algorithms on a custom platform, thus avoiding free flight tests and the need for a position controller. This work extends a preliminary study [62], where the use of a neural network compensator was discussed and a baseline PID was compared to a PID+NN architecture. The proposed framework integrates a baseline robust linear H_∞ optimal stabilizer with an online NN-based loop parameterized via a Single-Layer Feedforward NN. By exploiting DReLU activation functions, the adaptive compensator ensures real-time mitigation of severe unmodeled nonlinearities and persistent exogenous perturbations while maintaining an extremely low computational footprint, making it fully compatible with low-power embedded microcontrollers.

A systematic two-stage validation campaign was carried out to assess the performance margins of the synthesized control loop. First, a comprehensive numerical sensitivity analysis was conducted using a linear simulator identified from experimental results on the vehicle. This simulation phase characterized the coupled effects of the online adaptation gain Γ_{NN}, the sigma-modification leakage factor α, and the hidden-layer size N, successfully mapping the hyperparameter space to optimize learning convergence speed while avoiding oscillatory transients. Crucially, the network weights were initialized from scratch without any prior offline pre-training, demonstrating true on-the-fly resilience.

The numerical findings were subsequently validated through real-time physical experimental tests using the Crazyflie 2.1+ micro-quadrotor platform constrained to an attitude tilting test bench. Under a persistent, highly asymmetrical unbalancing torque generated via physical payload displacement, the proposed DReLU-based augmented architecture successfully suppressed tracking offsets with minimal overshoot and negligible cross-channel dynamic coupling. When benchmarked against a literature-standard RBF adaptive network, the proposed scheme achieved a 52.6% reduction in embedded CPU execution core load while preserving comparable tracking accuracy metrics.

Future research will investigate the scalability of the DReLU adaptive framework to the outer loop by synthesizing an augmented position control layer. Additionally, further efforts will focus on evaluating network resilience under highly dynamic time-varying external moments, such as severe aerodynamic wind shear and complex ground effect interactions.

Author Contributions: Conceptualization, E.D., I.N. and G.R.; methodology, E.D., I.N. and G.R.; software, G.R.; validation, S.R.B. and G.R.; formal analysis, I.N. and G.R.; investigation, S.R.B., E.D. and G.R.; resources, E.D.; data curation, S.R.B. and G.R.; writing—original draft preparation, S.R.B. and G.R.; writing—review and editing, L.B., E.D. and I.N.; visualization, S.R.B., L.B. and G.R.; supervision, L.B. and E.D.; project administration, E.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. de Oliveira Evald, P.J.D.; Aoki, V.M.; da Silva, C.B.; Cardoso, D.S.; Pinheiro, P.M.; da Costa Botelho, S.S.; Junior, P.L.J.D. A Review on Quadrotor Attitude Control Strategies. *Int. J. Intell. Robot. Appl.* **2024**, *8*, 230–250. [CrossRef]
2. Hildmann, H.; Kovacs, E. Review: Using Unmanned Aerial Vehicles (UAVs) as Mobile Sensing Platforms (MSPs) for Disaster Response, Civil Security and Public Safety. *Drones* **2019**, *3*, 59. [CrossRef]
3. Nex, F.; Remondino, F. UAV for 3D mapping applications: A review. *Appl. Geomat.* **2014**, *6*, 1–15. [CrossRef]
4. Gupta, A.; Fernando, X. Simultaneous Localization and Mapping (SLAM) and Data Fusion in Unmanned Aerial Vehicles: Recent Advances and Challenges. *Drones* **2022**, *6*, 85. [CrossRef]
5. Tagarakis, A.C.; Filippou, E.; Kalaitzidis, D.; Benos, L.; Busato, P.; Bochtis, D. Proposing UGV and UAV Systems for 3D Mapping of Orchard Environments. *Sensors* **2022**, *22*, 1571. [CrossRef]
6. Ren, H.; Zhao, Y.; Xiao, W.; Hu, Z. A review of UAV monitoring in mining areas: Current status and future perspectives. *Int. J. Coal Sci. Technol.* **2019**, *6*, 320–333. [CrossRef]
7. Malaver, A.; Motta, N.; Corke, P.; Gonzalez, F. Development and Integration of a Solar Powered Unmanned Aerial Vehicle and a Wireless Sensor Network to Monitor Greenhouse Gases. *Sensors* **2015**, *15*, 4072–4096. [CrossRef]
8. Camarillo-Escobedo, R.; Flores, J.L.; Marin-Montoya, P.; García-Torales, G.; Camarillo-Escobedo, J.M. Smart Multi-Sensor System for Remote Air Quality Monitoring Using Unmanned Aerial Vehicle and LoRaWAN. *Sensors* **2022**, *22*, 1706. [CrossRef]
9. Mogili, U.R.; Deepak, B.B.V.L. Review on Application of Drone Systems in Precision Agriculture. *Procedia Comput. Sci.* **2018**, *133*, 502–509. [CrossRef]
10. Zhang, H.; Wang, L.; Tian, T.; Yin, J. A Review of Unmanned Aerial Vehicle Low-Altitude Remote Sensing (UAV-LARS) Use in Agricultural Monitoring in China. *Remote Sens.* **2021**, *13*, 1221. [CrossRef]
11. Sassu, A.; Gambella, F.; Ghiani, L.; Mercenaro, L.; Caria, M.; Pazzona, A.L. Advances in Unmanned Aerial System Remote Sensing for Precision Viticulture. *Sensors* **2021**, *21*, 956. [CrossRef]
12. Jaimes, A.; Kota, S.; Gomez, J. An approach to surveillance an area using swarm of fixed wing and quad-rotor unmanned aerial vehicles UAV (s). In Proceedings of the IEEE International Conference on System of Systems Engineering, Monterey, CA, USA, 2–4 June 2008; pp. 1–6.
13. Sa, I.; Hrabar, S.; Corke, P. Inspection of Pole-like Structures Using a Visual-Inertial Aided VTOL Platform with Shared Autonomy. *Sensors* **2015**, *15*, 22003–22048. [CrossRef]
14. Rattanaamporn, S.; Perera, A.; Nguyen, A.; Ngo, T.B.; Chahl, J. Drone Imaging and Sensors for Situational Awareness in Hazardous Environments: A Systematic Review. *J. Sens. Actuator Netw.* **2025**, *14*, 98. [CrossRef]
15. Hassani, H.; Mansouri, A.; Ahaitouf, A. Robust Attitude Control of a Disturbed Multi-Rotor Aircraft: Experimental Comparisons. In *Proceedings of the International Conference on Digital Technologies and Applications*; Lecture Notes in Networks and Systems; Springer: Berlin/Heidelberg, Germany, 2024; pp. 464–473. [CrossRef]
16. Lim, H.; Park, J.; Lee, D.; Kim, H.J. Build your own quadrotor: Open-source projects on unmanned aerial vehicles. *IEEE Robot. Autom. Mag.* **2012**, *19*, 33–45. [CrossRef]
17. Goodarzi, F.; Lee, D.; Lee, T. Geometric nonlinear PID control of a quadrotor UAV on SE (3). In Proceedings of the European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; pp. 3845–3850.
18. Liu, S.; Hou, Z.; Zheng, J. Attitude adjustment of quadrotor aircraft platform via a data-driven model free adaptive control cascaded with intelligent PID. In Proceedings of the Chinese Control and Decision Conference (CCDC), Yinchuan, China, 28–30 May 2016; pp. 4971–4976.
19. Jiao, Q.; Liu, J.; Zhang, Y.; Lian, W. Analysis and design the controller for quadrotors based on PID control method. In Proceedings of the 2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Nanjing, China, 18–20 May 2018; pp. 88–92.
20. Meier, L.; Tanskanen, P.; Fraundorfer, F.; Pollefeys, M. The pixhawk open-source computer vision framework for mavs. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *38*, 13–18. [CrossRef]
21. Candan, F.; Beke, A.; Kumbasar, T. Design and Deployment of Fuzzy PID Controllers to the nano quadcopter Crazyflie 2.0. In Proceedings of the 2018 Innovations in Intelligent Systems and Applications (INISTA), Thessaloniki, Greece, 3–5 July 2018; pp. 1–6. [CrossRef]
22. Lopez-Franco, C.; Gomez-Avila, J.; Alanis, A.Y.; Arana-Daniel, N.; Villaseñor, C. Visual Servoing for an Autonomous Hexarotor Using a Neural Network Based PID Controller. *Sensors* **2017**, *17*, 1865. [CrossRef]
23. Andrade, F.A.A.; Guedes, I.P.; Carvalho, G.F.; Zachi, A.R.L.; Haddad, D.B.; Almeida, L.F.; de Melo, A.G.; Pinto, M.F. Unmanned Aerial Vehicles Motion Control with Fuzzy Tuning of Cascaded-PID Gains. *Machines* **2022**, *10*, 12. [CrossRef]
24. Wen, S.; Zhang, Q.; Deng, J.; Lan, Y.; Yin, X.; Shan, J. Design and Experiment of a Variable Spray System for Unmanned Aerial Vehicles Based on PID and PWM Control. *Appl. Sci.* **2018**, *8*, 2482. [CrossRef]
25. Hoffmann, F.; Goddemeier, N.; Bertram, T. Attitude estimation and control of a quadrocopter. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 1072–1077.

26. Bo, G.; Xin, L.; Hui, Z.; Ling, W. Quadrotor helicopter attitude control using cascade PID. In Proceedings of the 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, China, 28–30 May 2016; pp. 5158–5163.
27. Panomrattanarug, B.; Higuchi, K.; Mora-Camino, F. Attitude Control of a Quadrotor Aircraft Using LQR State Feedback Controller with Full Order State Observer. In Proceedings of the The SICE Annual Conference 2013, Nagoya, Japan, 14–17 September 2013; pp. 2041–2046.
28. Noordin, A.; Mohd Basri, M.A.; Mohamed, Z.; Mat Lazim, I. Adaptive PID controller using sliding mode control approaches for quadrotor UAV attitude and position stabilization. *Arab. J. Sci. Eng.* **2021**, *46*, 963–981. [CrossRef]
29. Chen, J.; Long, Y.; Li, T. Robust backstepping control for attitude tracking of a quadrotor based on integral linear extended state observer under input saturation. *Nonlinear Dyn.* **2024**, *112*, 4573–4584. [CrossRef]
30. Adigbli, P.; Grand, C.; Mouret, J.B.; Doncieux, S. Nonlinear attitude and position control of a micro quadrotor using sliding mode and backstepping techniques. In Proceedings of the 7th European Micro Air Vehicle Conference (MAV07), Toulouse, France, 17–21 September 2007; pp. 1–9.
31. Izadi, M.; Shayan, Z.; Faieghi, R. Multi-model predictive attitude control of quadrotors. In Proceedings of the IEEE 20th International Conference on Automation Science and Engineering (CASE), Bari, Italy, 28 August–1 September 2024; pp. 3830–3835.
32. D'Amato, E.; Tartaglione, G.; Blasi, L.; Mattei, M. Nonlinear Dynamic Inversion with Neural Networks for the Flight Control of a Low Cost Tilt Rotor Uav. In Proceedings of the 30th Congress of the International Council of the Aeronautical Sciences. International Council of the Aeronautical Sciences ICAS, Daejeon, Republic of Korea, 25–30 September 2016.
33. Safwat, E.; Zhang, W.; Kamel, A.; Kassem, M. Robustness analysis of modified incremental nonlinear dynamic inversion for small UAVs. *Autom. Control Comput. Sci.* **2020**, *54*, 128–138. [CrossRef]
34. Xie, L.; de Souza, C.E. Robust H_∞ control for linear systems with norm-bounded time-varying uncertainty. In Proceedings of the 29th IEEE Conference on Decision and Control, Honolulu, HI, USA, 5–7 December 1990; pp. 1034–1035.
35. Hui, N.; Guo, Y.; Han, X.; Wu, B. Robust H-infinity dual cascade MPC-based attitude control study of a quadcopter UAV. *Actuators* **2024**, *13*, 392. [CrossRef]
36. Hamza, A.; Mohamed, A.H.; El-Badawy, A. Robust H-Infinity Control for a Quadrotor UAV. In Proceedings of the AIAA Scitech Forum, San Diego, CA, USA, 3–7 January 2022; p. 2033. [CrossRef]
37. Scordamaglia, V.; Sollazzo, A.; Mattei, M. Fixed Structure Flight Control Design of an Over-Actuated Aircraft in the Presence of Actuators with Different Dynamic Performance. *IFAC Proc. Vol.* **2012**, *45*, 127–132. [CrossRef]
38. Jafar, A.; Fasih-UR-Rehman, S.; Fazal-UR-Rehman, S.; Ahmed, N.; Shehzad, M.U. A robust H_∞ control for unmanned aerial vehicle against atmospheric turbulence. In Proceedings of the 2nd International Conference on Robotics and Artificial Intelligence (ICRAI), Rawalpindi, Pakistan, 1–2 November 2016; pp. 1–6.
39. Noormohammadi-Asl, A.; Esrafilian, O.; Arzati, M.A.; Taghirad, H.D. System Identification and H∞-Based Control of Quadrotor Attitude. *Mech. Syst. Signal Process.* **2020**, *135*, 106358. [CrossRef]
40. Lavretsky, E.; Wise, K.A. *Robust and Adaptive Control with Aerospace Applications*; Advanced Textbooks in Control and Signal Processing; Springer: London, UK, 2013. [CrossRef]
41. Santos, M.C.P.; Rosales, C.D.; Sarapura, J.A.; Sarcinelli-Filho, M.; Carelli, R. An adaptive dynamic controller for quadrotor to perform trajectory tracking tasks. *J. Intell. Robot. Syst.* **2019**, *93*, 5–16. [CrossRef]
42. Whitehead, B.; Bieniawski, S. Model reference adaptive control of a quadrotor UAV. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, ON, Canada, 2–5 August 2010; p. 8148.
43. Whitaker, H.; Yamron, J.; Kezer, A.; M.I.T. Instrumentation Laboratory. *Design of Model Reference Adaptive Control Systems for Aircraft*; Report Massachusetts Institute of Technology Instrumentation Laboratory R; M.I.T. Instrumentation Laboratory: Cambridge, MA, USA, 1958.
44. Rothe, J.; Zevering, J.; Strohmeier, M.; Montenegro, S. A Modified Model Reference Adaptive Controller (M-MRAC) Using an Updated MIT-rule for the Altitude of a UAV. *Electronics* **2020**, *9*, 1104. [CrossRef]
45. Khalil, A.; Jaradat, M.A.; Mukhopadhyay, S.; Abdel-Hafez, M.F. Autonomous Control of a Hybrid Rolling and Flying Caged Drone for Leak Detection in HVAC Ducts. *IEEE/ASME Trans. Mechatronics* **2023**, *29*, 366–378. [CrossRef]
46. Ale Isaac, M.S.; Luna, M.A.; Ragab, A.R.; Ale Eshagh Khoeini, M.M.; Kalra, R.; Campoy, P.; Flores Peña, P.; Molina, M. Medium-Scale Uavs: A Practical Control System Considering Aerodynamics Analysis. *Drones* **2022**, *6*, 244. [CrossRef]
47. Nguyen, N.T. Model-reference adaptive control. In *Model-Reference Adaptive Control: A Primer*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 83–123.
48. Anderson, R.B.; Marshall, J.A.; L'Afflitto, A. Novel Model Reference Adaptive Control Laws for Improved Transient Dynamics and Guaranteed Saturation Constraints. *J. Frankl. Inst.* **2021**, *358*, 6281–6308. [CrossRef]
49. Lavretsky, E.; Wise, K.A. Robust adaptive control. In *Robust and Adaptive Control: With Aerospace Applications*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 469–506.
50. Lara, D.; Romero, G.; Ibarra, R.; Sánchez, A.; Pegard, C. Onboard System for Flight Control of a Small UAV. In Proceedings of the World Automation Congress 2012, Puerto Vallarta, Mexico, 24–28 June 2012; pp. 1–6.

51. Zhang, D.; Wei, B. Convergence Performance Comparisons of PID, MRAC, and PID+ MRAC Hybrid Controller. *Front. Mech. Eng.* **2016**, *11*, 213–217. [CrossRef]
52. Li, M.; Cai, Z.; Zhao, J.; Wang, J.; Wang, Y. Disturbance Rejection and High Dynamic Quadrotor Control Based on Reinforcement Learning and Supervised Learning. *Neural Comput. Appl.* **2022**, *34*, 11141–11161. [CrossRef]
53. Orsag, M.; Korpela, C.M.; Bogdan, S.; Oh, P.Y. Hybrid adaptive control for aerial manipulation. *J. Intell. Robot. Syst.* **2014**, *73*, 693–707. [CrossRef]
54. Yang, X.; Hu, X.; Ye, H.; Liu, W.; Shen, H. Fraction-Order MRAC Method Based Fault Tolerant Control for Plant Protection UAV with Actuator Failure and Uncertainty. *Int. J. Control. Autom. Syst.* **2023**, *21*, 2623–2633. [CrossRef]
55. Tanveer, A.; Afzaal, N.; Murawwat, S.; Aleem, S.; Sayeda, F. Mitigating Flight Performance Errors in UAVs through Hybrid MRAC controller. In Proceedings of the International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE), Lahore, Pakistan, 2–4 December 2022; pp. 1–8.
56. Li, Q.; Tong, S.; Chai, T. H_∞ tracking of unknown nonlinear systems using neural network. In Proceedings of the 12th IEEE International Symposium on Intelligent Control, Istanbul, Turkey, 16–18 July 1997; pp. 199–204. [CrossRef]
57. Rovithakis, G. Reinforcing robustness using high order neural network controllers. In Proceedings of the 37th IEEE Conference on Decision and Control, Tampa, FL, USA, 16–18 December 1998; Volume 1, pp. 1052–1057. [CrossRef]
58. Yang, Y.S.; Wang, X.F. Adaptive H_∞ tracking control for a class of uncertain nonlinear systems using radial-basis-function neural networks. *Neurocomputing* **2007**, *70*, 932–941. [CrossRef]
59. Li, Y.; Sundararajan, N.; Saratchandran, P.; Wang, Z. Robust neuro-H_∞ controller design for aircraft auto-landing. *IEEE Trans. Aerosp. Electron. Syst.* **2004**, *40*, 158–167. [CrossRef]
60. Nicol, C.; Macnab, C.; Ramirez-Serrano, A. Robust neural network control of a quadrotor helicopter. In Proceedings of the Canadian Conference on Electrical and Computer Engineering, Niagara Falls, ON, Canada, 4–7 May 2008; pp. 001233–001238. [CrossRef]
61. Lin, C.K.; Wang, S.D. An adaptive H_∞ controller design for bank-to-turn missiles using ridge Gaussian neural networks. *IEEE Trans. Neural Netw.* **2004**, *15*, 1507–1516. [CrossRef]
62. Bassolillo, S.R.; Raspaolo, G.; Blasi, L.; D'Amato, E.; Notaro, I. Neural Network Based Model Reference Adaptive Attitude Control for a Micro Unmanned Air Vehicle. In Proceedings of the 11th International Conference on Control, Decision and Information Technologies (CoDIT), Split, Croatia, 15–18 July 2025; Volume 1, pp. 521–526. [CrossRef]
63. Madani, T.; Benallegue, A. Adaptive control via backstepping technique and neural networks of a quadrotor helicopter. *IFAC Proc. Vol.* **2008**, *41*, 6513–6518. [CrossRef]
64. Shastry, A.K.; Kothari, M.; Abhishek, A. Development of flight dynamics model of quadrotor. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, USA, 8–12 January 2018; p. 1850.
65. Chaudhuri, B.; Pal, B.C.; Zolotas, A.C.; Jaimoukha, I.M.; Green, T.C. Mixed-Sensitivity Approach to H_∞ Control of Power System Oscillations Employing Multiple FACTS Devices. *IEEE Trans. Power Syst.* **2003**, *18*, 1149–1156. [CrossRef]
66. Humaidi, A.; Hameed, A. Robust MRAC for a wing rock phenomenon in delta wing aircrafts. *AUT J. Model. Simul.* **2017**, *49*, 113–122. [CrossRef]
67. Jurado, F.; Ollervides-Vazquez, E.J. Decentralized Robust Direct MRAC via e-Modification for the Pose of a Quadrotor UAV. *Appl. Sci.* **2025**, *15*, 11713. [CrossRef]
68. Ioannou, P.; Sun, J. *Robust Adaptive Control*; Dover Books on Electrical Engineering Series; Dover Publications, Incorporated: Garden City, NY, USA, 2014; pp. 1–12. [CrossRef]
69. Chicco, D.; Warrens, M.J.; Jurman, G. The Coefficient of Determination R-squared Is More Informative than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation. *PeerJ Comput. Sci.* **2021**, *7*, e623. [CrossRef]
70. Klán, P.; Gorez, R. PI Controller Design for Actuator Preservation. *IFAC Proc. Vol.* **2008**, *41*, 5820–5824. [CrossRef]
71. Macêdo, D.; Zanchettin, C.; Oliveira, A.L.; Ludermir, T. Enhancing Batch Normalized Convolutional Networks Using Displaced Rectifier Linear Units: A Systematic Comparative Study. *Expert Syst. Appl.* **2019**, *124*, 271–281. [CrossRef]
72. Slama, S.; Errachdi, A.; Benrejeb, M. Online MRAC Method Using Neural Networks Based on Variable Learning Rate for Non-Linear Systems. *Int. J. Comput. Appl. Technol.* **2019**, *60*, 215. [CrossRef]
73. Liu, M.; Dong, X.; Li, Q.; Ren, Z. Model Reference Adaptive Control of a Quadrotor UAV based on RBF Neural Networks. In Proceedings of the IEEE CSAA Guidance, Navigation and Control Conference (CGNCC), Xiamen, China, 10–12 August 2018; pp. 1–6.
74. Chen, W.; Ding, Y.; Weng, F.; Liang, C.; Li, J. Global fast terminal fuzzy sliding mode control of quadrotor UAV based on RBF neural network. *Sensors* **2025**, *25*, 1060. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.