

# Adaptive neural network-based trajectory tracking outer loop control for a quadrotor<sup>☆</sup>

Ivan Lopez-Sánchez, Jerónimo Moyrón, Javier Moreno-Valenzuela\*



## ARTICLE INFO

### Article history:

Received 7 April 2022

Received in revised form 22 August 2022

Accepted 24 August 2022

Available online 31 August 2022

Communicated by Hever Moncayo

## ABSTRACT

This manuscript introduces a novel adaptive neural network-based controller for trajectory tracking of quadrotors. This controller is conceived as an outer loop controller that interacts with an inner loop controller in a two-loop configuration. The inner loop in this two-loop configuration is assumed to be inaccessible and unmodifiable, which is a realistic hypothesis in the operation of commercial quadrotors. Under this situation, the proposed controller computes appropriate kinematic input commands for the inner loop to achieve trajectory tracking. One remarkable feature of the proposed algorithm is its robustness against parametric uncertainties from the inner loop. An exhaustive error convergence analysis is provided, thus guaranteeing the convergence of the trajectory tracking error. Experimental results and a comparison using other control schemes demonstrate the competitiveness of the proposed scheme, being the latter the best among the tested adaptive neural network-based schemes.

© 2022 Elsevier Masson SAS. All rights reserved.

## 1. Introduction

Currently, the industrial and scientific community's attention has been attracted to unmanned aerial vehicles (UAVs). Nowadays, quadrotors are the most representative UAV among such aircraft. They became attractive due to their capabilities and characteristics, e.g., vertical take-off and landing (VTOL), hovering, maneuverability, mechanical simplicity, compact size, portability, and thrust-to-weight ratio [1,2]. Compared with helicopters, which can also perform hovering and VTOL, quadrotors are easier to acquire or construct and cheaper to maintain. Because of this, quadrotors have been employed in many different applications, e.g., precision agriculture [3,4], surveillance [5], civil engineering [6], construction [7], maritime tasks [8,9], among others.

There are different control strategies reported in the literature employed to control quadrotors [2,10–20]. In the category of linear control, techniques such as proportional-integral-derivative control, linear quadratic regulator, linear model predictive control, and  $H_\infty$  were reported. The drawback of such strategies is the need to linearize the system dynamics, commonly around zero (hovering). This linearization limits the quadrotor's operation to small attitude angles and slow linear and angular velocities. In addition, many of the aerodynamic effects present during the hovering operation are neglected. On the other hand, in the nonlinear control category, there are control schemes such as feedback linearization, backstepping, sliding mode control, active disturbance rejection control, model predictive control, and adaptive control [21–24]. Learning-based intelligent control schemes such as fuzzy logic and neural networks also belong to the nonlinear control category.

Since the first appearance of the term intelligent control during the 1970 decade [25], the interest in studying such schemes has been growing. The works [26–31] provided the basis for current research in studying artificial neural networks and their applications in control theory. As a result, many contributions based on intelligent schemes concerning the control and stability of unmanned aircraft systems have been reported [32–39]. Intelligent control can be separated into two general schemes: fuzzy logic and neural networks. One advantage of these intelligent control techniques is their learning capability. Such techniques emulate at a certain level the human mental ability to adapt and react based on previous experiences. Fuzzy logic provides a response in agreement with pre-established criteria [12,14]. In contrast, neural networks find a relation or pattern between system inputs and outputs [12,14]. Recalling that quadrotors are nonlinear

<sup>☆</sup> This work was supported by CONACYT-Fondo Sectorial de Investigación para la Educación under project A1-S-24762 (Proyecto Apoyado por el Fondo Sectorial de Investigación para la Educación) and by SIP-IPN, Project number 20220436, México.

\* Corresponding author.

E-mail address: [moreno@citedi.mx](mailto:moreno@citedi.mx) (J. Moreno-Valenzuela).

systems under complex aerodynamic effects and unknown external disturbances, neural networks are a good alternative for controlling such vehicles.

Usually, neural networks pass through a training stage before their implementation, similar to the human learning process, based on previous experiences. Nevertheless, among the different classes of neural networks, there is a kind with online learning capabilities, better known as adaptive neural networks (ANNs). The use of ANNs is especially useful when accurate knowledge of the system parameters is not available or when the system is operating in changing environments [40]. Many applications have been found for ANN-based control, such as controlling a hydraulic knee exoskeleton with valve deadband and output constraint [41], a robot manipulator arm via image-based visual servoing control [42], a quarter car electrohydraulic active suspension system [43], a flexible joint robotic arm [44], and a tiltable V-tail morphing aircraft [45]. The ANN-based control has been applied in quadrotors to compensate for external disturbances and unmodeled dynamics [46–51]. It is noteworthy that all the previously cited works were developed assuming full access to the quadrotor dynamics in the sense that they can interact freely with the position and attitude dynamics. Additionally, those schemes assume a direct interaction with the actuators. Therefore, they compute forces and torques that must be transformed into angular velocities for the rotors. Though, such a level of access to the platform sensors and actuators may not be achievable on all platforms.

These days quadrotors are easy to acquire, and there are many options in the market. However, the amount of research-oriented quadrotors is smaller compared to the amount of entertainment-oriented quadrotors. Their main differences are the hardware, customization level, and software. In entertainment-oriented quadrotors, a control loop maintains the vehicle's stability serving as an intermediary between the pilot's input commands and the quadrotor. This control loop cannot be modified by the pilot, limiting access to some features that are free to access in the research-oriented platforms. Thus, the automation of this kind of vehicle can be difficult. Howbeit, as was shown in the works [52–57], this does not prevent employing commercial entertainment-oriented quadrotors in the research field. Works in the literature have demonstrated that a control scheme based on a two-loop configuration can be implemented on such platforms to achieve trajectory tracking tasks [58], payload transport [49], collision avoidance [59], and quadrotor formation [60]. In such works, inner and outer loops are integrated to work together in a two-loop configuration. The inner loop is an embedded controller installed by the manufacturer, and the outer loop controller is free to be designed. The development of such outer loop controllers is not exclusively for entertainment-oriented platforms. As shown in [61], they can also be implemented on research-oriented platforms by designing appropriate inner and outer loop controllers.

This work addresses the trajectory tracking problem of a quadrotor by using a two-loop configuration, considering that the inner loop is unmodifiable. This problem has also been addressed in [58,59,61–65] employing different control strategies in which the knowledge of the system parameters is needed to design the outer loop controller. Currently, few works address the control of a quadrotor under a two-loop configuration, where the inner loop is inaccessible and unmodifiable. Besides, many of them require knowledge of the parameters of the inner loop controller and the quadrotor. This implies a parameter identification process before implementing the outer loop controller, which is a challenging task. In addition, this parameter identification task is more demanding to perform on entertainment-oriented platforms because some data acquisition and interaction with the platform are limited. Our proposed novel algorithm aims to overcome this drawback and provide robustness to parameter uncertainty and unmodeled dynamics. As a contribution, we present an adaptive neural network controller for the trajectory tracking of quadrotors on which the control law is completely system-parameter-free. This controller is conceived as the outer loop controller in a two-loop configuration. The inner loop is considered inaccessible and unmodifiable. The proposed scheme is robust against parametric uncertainties from the inner loop and guarantees convergence of the tracking error signals. Furthermore, experimental results prove the competitiveness of the proposed scheme over other neural network-based algorithms. Additionally, our proposed outer loop controller is an improvement regarding [61]; the differences representing an advantage over it are the following:

- Knowledge of the system parameters is not needed.
- Integral action is included in the design to cope with the steady-state error, constant disturbances, and the neural network approximation error.
- The adaptive neural network uses a more straightforward activation function on which no parameters are needed to define it.
- Only the output weight matrix is updated online, reducing the algorithm's computational complexity.
- The proposed algorithm needs fewer update laws to operate.
- The convergence to zero of the error and its time derivative is theoretically ensured and derived easily.

The proposed algorithm is an ANN-based controller which serves as an outer control loop for the overall system formed by the quadrotor with an inaccessible inner loop controller. The inaccessible inner loop controller admits as inputs angular position commands for the roll and pitch angles and velocity commands for the vertical height and yaw rotation. Thus, the proposed outer loop controller provides the appropriate inputs for the inaccessible inner controller to achieve the quadrotor's trajectory tracking task. The ANN is housed in the proposed outer loop controller. It is an essential part of the scheme because it provides robustness to parameter uncertainties, external disturbance, and unmodeled dynamics.

The manuscript is organized as follows: Section 2 presents the mathematical model of the system formed by the quadrotor with an inner controller and the control goal. In Section 3 the development of the proposed controller and the error convergence analysis are described. The experimental results are presented in Section 4. Finally, in Section 5, the conclusions of this document are given.

**Notation:** The notation used in this manuscript is described next.  $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$  is the norm  $l_1$  of the vector  $\mathbf{x} \in \mathbb{R}^n$ .  $\|\mathbf{x}\| = \sqrt{\mathbf{x}^\top \mathbf{x}}$  is the Euclidean norm of the vector  $\mathbf{x} \in \mathbb{R}^n$ .  $\lambda_{\min}\{A\}$  and  $\lambda_{\max}\{A\}$  are the minimum and maximum eigenvalues of the symmetric matrix  $A \in \mathbb{R}^{n \times n}$ , respectively.  $\text{Tr}\{A\}$  is the trace of the matrix  $A \in \mathbb{R}^{n \times n}$  defined as  $\text{Tr}\{A\} = \sum_{i=1}^n A_{ii}$ .  $\|A\|_F^2 = \text{Tr}\{A^\top A\}$  is the Frobenius norm of the matrix  $A \in \mathbb{R}^{n \times n}$ . The Frobenius inner product is defined as  $\langle A, B \rangle_F = \text{Tr}\{A^\top B\}$  for any  $A$  and  $B$  matrices of compatible dimensions and according to the Schwarz inequality  $|\langle A, B \rangle_F| \leq \|A\|_F \|B\|_F$ .  $\text{diag}_n\{a\} \in \mathbb{R}^{n \times n}$  is a diagonal matrix with  $a$  in the elements of its diagonal.

## 2. Dynamic model, implications of the inner loop controller and control goal

This section presents the dynamic model of a quadrotor with respect to the inertial reference frame. Besides, the mathematical representation of the closed-loop system of the quadrotor with an inner loop controller is given. Furthermore, the inner loop controller is described, and the control goal is stated.

### 2.1. Quadrotor dynamic model

Based on [61,66–68], the dynamic model of a quadrotor in the inertial reference frame is given by

$$m\ddot{\mathbf{p}} + mg_z + D_p(\eta)\dot{\mathbf{p}} = \mathbf{r}_3(\eta)F, \quad (1)$$

$$M(\eta)\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} + D_\eta(\eta)\dot{\eta} = \Psi(\eta)^{-\top}\boldsymbol{\tau}, \quad (2)$$

where  $\mathbf{p} = [x \ y \ z]^\top \in \mathbb{R}^3$  and  $\eta = [\phi \ \theta \ \psi]^\top \in \mathbb{R}^3$  are the position and attitude vectors of the quadrotor, respectively, the constant  $m > 0$  is the mass of the vehicle,  $\mathbf{g}_z = [0 \ 0 \ g]^\top \in \mathbb{R}^3$ , being  $g$  the gravitational acceleration constant,  $D_p(\eta) \in \mathbb{R}^{3 \times 3}$  and  $D_\eta(\eta) \in \mathbb{R}^{3 \times 3}$  are positive definite matrices representing the aerodynamic drag and damping effects, respectively,  $\mathbf{r}_3(\eta) \in \mathbb{R}^3$  is the third column of the rotation matrix  $R(\eta) \in \mathbb{R}^{3 \times 3}$ ,  $M(\eta) \in \mathbb{R}^{3 \times 3}$  and  $C(\eta, \dot{\eta}) \in \mathbb{R}^{3 \times 3}$  are the inertia and Coriolis matrices, respectively,  $\Psi(\eta) \in \mathbb{R}^{3 \times 3}$  is a transformation matrix, while  $F \in \mathbb{R}$  and the vector  $\boldsymbol{\tau} = [\tau_\phi \ \tau_\theta \ \tau_\psi]^\top \in \mathbb{R}^3$  are the control inputs. Notice that this representation is restricted to attitude angles  $|\theta| < \pi/2$  and  $\phi, \psi \in \mathbb{R}$ . Thus, acrobatic maneuvering is out of the scope of this work.

### 2.2. Inner loop controller

As was described in [58,59,61–65,69], many commercial quadrotors available in the market nowadays have an embedded controller installed by the manufacturer. This controller stabilizes the vehicle and acts as an intermediary between the pilot commands and the quadrotor dynamic model's actual control inputs, simplifying flight tasks for the pilot. It is worth mentioning that this controller is out of reach and cannot be modified. However, it supports kinematic input commands, making it suitable to be employed as the inner loop controller in a two-loop configuration. Without loss of generality, it is assumed that the inner loop (embedded) controller has the following structure [61]

$$F = \frac{m}{\cos(\phi) \cos(\theta)} \left( g + \frac{\dot{z}_{\max} u_z - \dot{z}}{\tau_z} \right), \quad (3)$$

$$\boldsymbol{\tau} = \Psi(\eta)^\top [M(\eta)\tilde{\boldsymbol{\tau}} + C(\eta, \dot{\eta})\dot{\eta}], \quad (4)$$

$$\tilde{\boldsymbol{\tau}} = \begin{bmatrix} \omega_\phi^2 \phi_{\max} u_\phi - 2\xi_\phi \omega_\phi \dot{\phi} - \omega_\phi^2 \phi, \\ \omega_\theta^2 \theta_{\max} u_\theta - 2\xi_\theta \omega_\theta \dot{\theta} - \omega_\theta^2 \theta, \\ \frac{\dot{\psi}_{\max} u_\psi - \dot{\psi}}{\tau_\psi} \end{bmatrix}, \quad (5)$$

where  $\omega_\phi$  and  $\omega_\theta$  are the natural frequencies and  $\xi_\phi$  and  $\xi_\theta$  are the damping constants of second-order linear systems,  $\tau_z$  and  $\tau_\psi$  are time response constants of first-order linear systems,  $\theta_{\max}$ ,  $\phi_{\max}$ ,  $\dot{z}_{\max}$ , and  $\dot{\psi}_{\max}$  are positive normalizing constants for pitch and roll angles, vertical velocity, and change rate of the yaw angle, respectively, while  $u_\theta$ ,  $u_\phi$ ,  $u_z$ , and  $u_\psi$  are kinematic inputs for the inner loop controller. As it is discussed in the next section, the actual structure of the inner loop controller is not crucial as long as it supports the aforementioned kinematic input commands, and its resulting closed-loop dynamics can be represented by a second-order system with external disturbances.

### 2.3. Mathematical representation of a quadrotor with an inner loop controller

Following the procedure described in [61], the closed-loop system resulting from the quadrotor model in (1) and (2) with the inner loop controller in (3)–(5) leads to the following mathematical representation

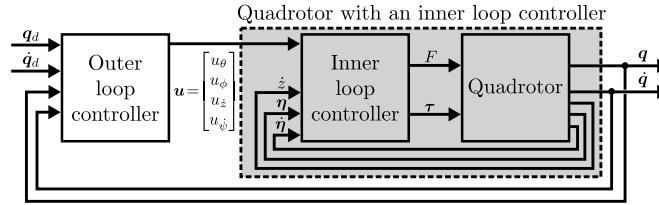
$$\ddot{\mathbf{q}} = T(\psi)K_u\mathbf{u} - K_v\dot{\mathbf{q}}, \quad (6)$$

where  $\mathbf{q} = [x \ y \ z \ \psi]^\top \in \mathbb{R}^4$  contains the position ( $x, y, z$ ) and the yaw angle  $\psi$  regarding the inertial reference frame, matrices

$$K_u = \begin{bmatrix} g\theta_{\max} & 0 & 0 & 0 \\ 0 & g\phi_{\max} & 0 & 0 \\ 0 & 0 & \frac{\dot{z}_{\max}}{\tau_z} & 0 \\ 0 & 0 & 0 & \frac{\dot{\psi}_{\max}}{\tau_\psi} \end{bmatrix}$$

and

$$K_v = \begin{bmatrix} d_x & 0 & 0 & 0 \\ 0 & d_y & 0 & 0 \\ 0 & 0 & \left(\frac{1}{\tau_z} + d_z\right) & 0 \\ 0 & 0 & 0 & \left(\frac{1}{\tau_\psi} + d_\psi\right) \end{bmatrix}$$



**Fig. 1.** Diagram of a quadrotor with an inaccessible inner loop controller.

are diagonal positive definite matrices containing the quadrotor and inner loop controller parameters where  $d_x$ ,  $d_y$  and  $d_z$  are constants relating the mass of the vehicle and the aerodynamic drag coefficients of the vehicle along the  $x$ ,  $y$  and  $z$  axis, respectively. The matrix  $T(\psi) \in \mathbb{R}^{4 \times 4}$  is a homogeneous transformation matrix defined as

$$T(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 & 0 \\ \sin(\psi) & -\cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and  $\mathbf{u} = [u_\theta \ u_\phi \ u_z \ u_\psi]^T \in \mathbb{R}^4$  is a kinematic control input vector, formed by the following elements:

- $u_\theta$  is an angular position command related to displacement about the  $x$ -axis.
- $u_\phi$  is an angular position command related to displacement about the  $y$ -axis.
- $u_z$  is a velocity command related to motion about the  $z$ -axis.
- $u_\psi$  is the yaw angle change rate command related to rotation about the  $z$ -axis.

See Fig. 1 for a graphical representation of the system formed by the quadrotor dynamic model in closed-loop with an inner loop controller. Notice that equation (6) describes the behavior of the system in ideal conditions. However, in outdoor applications or more realistic circumstances, the vehicle is subject to external disturbances due to environmental phenomena (wind gusts or rain) and unmodeled dynamics, e.g., when the actual structure and parameters of the inner loop controller are not known. Therefore, the system (6) is represented as

$$\ddot{\mathbf{q}} = T(\psi)K_u\mathbf{u} - K_v\dot{\mathbf{q}} + \delta(t), \quad (7)$$

where  $\delta(t) \in \mathbb{R}^4$  is the vector of external disturbances and unmodeled dynamics, which is assumed to be bounded as

$$\|\delta(t)\| \leq \delta_0,$$

being  $\delta_0$  a positive constant [31,70–72]. Notice that although matrices  $T(\psi)$  and  $K_u$  are both symmetric, the product  $T(\psi)K_u \neq K_uT(\psi)$ . Therefore, for the design of the outer loop controller, we have considered the following assumption concerning the parameters of the inner loop controller.

**Assumption 1.** Consider that the maximum values for the roll and pitch angles are the same due to the inner loop controller operation, i.e.,  $\theta_{\max} = \phi_{\max}$ , the following holds

$$T(\psi)K_u = K_uT(\psi).$$

Assumption 1 can be easily fulfilled for those research-oriented quadrotors for which the structure of the inner loop controller is known and tunable, as the parameters  $\theta_{\max}$  and  $\phi_{\max}$  can be defined by the user. For the case of entertainment-oriented quadrotors, Assumption 1 establishes the class of inner loop controllers for which the proposed outer loop scheme can be applied. Notice that in both cases, the actual values of  $\theta_{\max}$  and  $\phi_{\max}$  are not needed to be known, but they have to be the same.

#### 2.4. Control goal

Assume that matrices  $K_u$  and  $K_v$  are unknown. The trajectory tracking error is defined as

$$\mathbf{e} = \mathbf{q}_d - \mathbf{q} \in \mathbb{R}^4, \quad (8)$$

where  $\mathbf{q}_d(t) = [x_d(t) \ y_d(t) \ z_d(t) \ \psi_d(t)]^T \in \mathbb{R}^4$  is the vector of desired trajectories which is assumed to be at least twice differentiable and bounded as follows [71,72]:

$$\begin{aligned} \left\| \mathbf{q}_d(t) \right\| &\leq q_{dB}, \forall t \geq t_0, \\ \left\| \dot{\mathbf{q}}_d(t) \right\| &\leq q_{dB}, \forall t \geq t_0, \\ \left\| \ddot{\mathbf{q}}_d(t) \right\| &\leq q_{dB}, \forall t \geq t_0, \end{aligned} \quad (9)$$

being  $q_{dB}$  a positive constant. The time derivative of the tracking error is defined as

$$\dot{\mathbf{e}} = \dot{\mathbf{q}}_d - \dot{\mathbf{q}} \in \mathbb{R}^4.$$

Then, the control goal is to design a model-parameter-free outer loop controller  $\mathbf{u}(t) = [u_\theta(t) \ u_\phi(t) \ u_z(t) \ u_\psi(t)]^\top \in \mathbb{R}^4$  such that

$$\lim_{t \rightarrow \infty} \begin{bmatrix} \mathbf{e}(t) \\ \dot{\mathbf{e}}(t) \end{bmatrix} = \mathbf{0} \quad (10)$$

be guaranteed despite parametric uncertainties and bounded time-varying external disturbances  $\delta(t)$ .

### 3. Outer loop adaptive neural network-based controller

The proposed scheme is based on ANNs using hyperbolic tangent as an activation function. Our scheme is specially developed to interact with the inner loop controller. It computes kinematic input commands in order to achieve trajectory tracking without prior knowledge of the quadrotor and inner loop parameters. Additionally, it is designed to be simple and easily implementable.

Considering the trajectory tracking error definition in (8) and the dynamics of a quadrotor with an inner controller in (7), the error dynamics is given by

$$\ddot{\mathbf{e}} = \ddot{\mathbf{q}}_d - T(\psi)K_u\mathbf{u} + K_v\dot{\mathbf{q}} - \delta(t). \quad (11)$$

In agreement with Assumption 1 and by multiplying the error dynamics in (11) by  $K_u^{-1}$  yields

$$K_u^{-1}\ddot{\mathbf{e}} = K_u^{-1}[\ddot{\mathbf{q}}_d + K_v\dot{\mathbf{q}} - \delta(t)] - T(\psi)\mathbf{u},$$

which can be expressed as

$$\Lambda\ddot{\mathbf{e}} = \mathbf{f}(\ddot{\mathbf{q}}_d, \dot{\mathbf{q}}) - \Lambda\delta(t) - T(\psi)\mathbf{u},$$

where  $\Lambda = K_u^{-1}$  and  $\mathbf{f}(\ddot{\mathbf{q}}_d, \dot{\mathbf{q}}) = \Lambda[\ddot{\mathbf{q}}_d + K_v\dot{\mathbf{q}}]$ . Notice that  $\mathbf{f}(\ddot{\mathbf{q}}_d, \dot{\mathbf{q}})$  contains the unknown matrices  $\Lambda$  and  $K_v$ . Taking advantage of the continuous function approximation property of neural networks [29,31,70–72],  $\mathbf{f}(\ddot{\mathbf{q}}_d, \dot{\mathbf{q}})$  can be approximated through a neural network as follows:

$$\mathbf{f}(\boldsymbol{\gamma}) = W^\top \boldsymbol{\sigma}(V^\top \boldsymbol{\gamma}) + \boldsymbol{\epsilon},$$

where  $\boldsymbol{\gamma} = [1 \ \mathbf{x}^\top]^\top \in \mathbb{R}^9$  is the neural network extended input vector,  $\mathbf{x} = [\ddot{\mathbf{q}}_d^\top \ \dot{\mathbf{q}}^\top]^\top \in \mathbb{R}^8$ ,  $V \in \mathbb{R}^{9 \times L_N}$  and  $W \in \mathbb{R}^{L_N \times 4}$  are the neural network ideal input and output weight matrices, respectively, that are bounded as  $\|W\|_F \leq W_0$  and  $\|V\|_F \leq V_0$ . The vector  $\boldsymbol{\sigma}(\boldsymbol{\varrho}) \in \mathbb{R}^{L_N}$  is the activation functions vector defined by  $\boldsymbol{\sigma}(\boldsymbol{\varrho}) = [\tanh(\varrho_1) \ \dots \ \tanh(\varrho_{L_N})]^\top$ , where  $\boldsymbol{\varrho} = V^\top \boldsymbol{\gamma} \in \mathbb{R}^{L_N}$ ,  $\boldsymbol{\epsilon} \in \mathbb{R}^4$  is the neural network approximation error bounded as  $0 < \|\boldsymbol{\epsilon}\| \leq \varepsilon_0$ , and  $L_N \in \mathbb{N}$  is the number of neurons forming the neural network. It is noteworthy to mention that  $\|\mathbf{x}(t)\|$  is bounded as

$$\|\mathbf{x}(t)\| \leq 2q_{dB} + \|\dot{\mathbf{e}}(t)\|, \forall t \geq t_0. \quad (12)$$

Boundedness of  $\boldsymbol{\epsilon}$  is guaranteed by the continuous function approximation property [29,31,70–72] as there exists  $b_x > 2q_{dB}$ , such that,  $\mathbf{x}(t)$  is contained inside a compact set  $\mathcal{S}_x = \{\mathbf{x}(t) \in \mathbb{R}^8 : \|\mathbf{x}(t)\| \leq b_x\}$  where  $q_{dB}$  is the bound in (9).

Under the assumption that the approximation property holds, the trajectory tracking error dynamics can be represented as

$$\Lambda\ddot{\mathbf{e}} = W^\top \boldsymbol{\sigma}(V^\top \boldsymbol{\gamma}) + \boldsymbol{\epsilon} - T(\psi)\mathbf{u} \quad (13)$$

where  $\boldsymbol{\epsilon} = \boldsymbol{\epsilon} - \Lambda\delta$  is bounded as  $\|\boldsymbol{\epsilon}\|_1 \leq \epsilon_0$ , with  $\epsilon_0 > 0$ .

Based on the trajectory tracking error dynamics in (13), the following adaptive neural network-based controller is proposed

$$\mathbf{u} = T(\psi)^{-1} \left[ K_p \boldsymbol{\epsilon} + K_i \xi + K_d \dot{\boldsymbol{\epsilon}} + \hat{W}^\top \boldsymbol{\sigma}(V^\top \boldsymbol{\gamma}) + \Delta \text{sign}(\alpha \boldsymbol{\epsilon} + \dot{\boldsymbol{\epsilon}}) \right], \quad (14)$$

$$\dot{\xi} = \alpha \boldsymbol{\epsilon} + \dot{\boldsymbol{\epsilon}}, \quad (15)$$

along with the output weight matrix adaptation law given by

$$\dot{\hat{W}} = N \left[ \boldsymbol{\sigma}(V^\top \boldsymbol{\gamma})(\alpha \boldsymbol{\epsilon} + \dot{\boldsymbol{\epsilon}})^\top - \kappa \|\alpha \boldsymbol{\epsilon} + \dot{\boldsymbol{\epsilon}}\| \hat{W} \right], \quad (16)$$

where  $\xi \in \mathbb{R}^4$  is the integral of  $\alpha \boldsymbol{\epsilon} + \dot{\boldsymbol{\epsilon}}$ ,  $N \in \mathbb{R}^{L_N \times L_N}$ ,  $K_p \in \mathbb{R}^{4 \times 4}$ ,  $K_i \in \mathbb{R}^{4 \times 4}$ ,  $K_d \in \mathbb{R}^{4 \times 4}$ , and  $\Delta \in \mathbb{R}^{4 \times 4}$  are diagonal positive definite matrices, and  $\alpha$  and  $\kappa$  are strictly positive constants. Besides, considering any vector  $\mathbf{z} \in \mathbb{R}^4$ , the sign function vector is given by  $\text{sign}(\mathbf{z}) = [\text{sign}(z_1) \ \dots \ \text{sign}(z_4)]^\top$ , being

$$\text{sign}(z_i) = \begin{cases} 1, & \text{for } z_i > 0, \\ 0, & \text{for } z_i = 0, \\ -1, & \text{for } z_i < 0. \end{cases}$$

Further details about the derivation of the proposed controller and the output weight matrix adaptation law are given in Appendix A.

Finally, the closed-loop system is comprised of the controller (14)–(16) and the trajectory tracking error dynamics in (13), which is given as

$$\Lambda \ddot{\mathbf{e}} = -K_p \mathbf{e} - K_i \xi - K_d \dot{\mathbf{e}} - \Delta \text{sign}(\alpha \mathbf{e} + \dot{\mathbf{e}}) + \tilde{W}^\top \sigma + \epsilon, \quad (17)$$

$$\dot{\tilde{W}} = -N \left[ \sigma (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top - \kappa \|\alpha \mathbf{e} + \dot{\mathbf{e}}\| (W - \tilde{W}) \right], \quad (18)$$

$$\dot{\xi} = \alpha \mathbf{e} + \dot{\mathbf{e}}, \quad (19)$$

where  $\sigma = \sigma(V^\top \gamma)$  is used hereafter to simplify the notation, and the output weights estimation error  $\tilde{W}$  is defined as

$$\tilde{W} = W - \hat{W}.$$

**Proposition 1.** Let the desired trajectory  $\mathbf{q}_d(t)$  and its first and second time derivatives  $\dot{\mathbf{q}}_d(t)$  and  $\ddot{\mathbf{q}}_d(t)$  be bounded as in (9) and assume that  $\|\dot{\mathbf{e}}(t_0)\|$  is inside a compact set  $S_{\dot{\mathbf{e}}} = \{\dot{\mathbf{e}}(t) \in \mathbb{R}^e : \|\dot{\mathbf{e}}(t)\| \leq b_x - 2q_{dB}\}$  for some  $b_x > 2q_{dB}$ . Assume  $K_p$ ,  $K_i$ ,  $K_d$ ,  $\Lambda$ ,  $N$ , and  $\Delta$  to be diagonal positive definite matrices along with strictly positive constants  $\alpha$  and  $\kappa$  fulfilling the conditions

$$0 < \alpha < \frac{\lambda_{\min}\{K_d\} \lambda_{\min}\{\Lambda\}}{\lambda_{\max}\{\Lambda\}^2}, \quad (20)$$

$$\lambda_{\min}\{\Delta\} \geq \epsilon_0 + \frac{\kappa W_0^2}{4}. \quad (21)$$

Then, the solutions  $\mathbf{e}(t)$  and  $\dot{\mathbf{e}}(t)$  of the closed-loop system (17)-(19) converge to zero asymptotically. Additionally, the output weights estimation error  $\tilde{W}(t)$  and the state variable  $\xi(t)$  remain bounded for all time  $t \geq t_0$ .

**Proof.** By inequality (12),  $\|\dot{\mathbf{e}}(t_0)\| \in S_{\dot{\mathbf{e}}}$  implies  $\|\mathbf{x}(t_0)\| \in S_x$  and the function approximation property holds. Therefore, the closed-loop system can be represented as in (17)-(19). Now, consider the following non-negative function

$$L = \frac{1}{2} \left[ \mathbf{e}^\top (K_p + \alpha K_d) \mathbf{e} + 2\alpha \mathbf{e}^\top \Lambda \dot{\mathbf{e}} + \dot{\mathbf{e}}^\top \Lambda \dot{\mathbf{e}} + \xi^\top K_i \xi + \text{Tr} \left\{ \tilde{W}^\top N^{-1} \tilde{W} \right\} \right]. \quad (22)$$

Using the facts  $\mathbf{x}^\top A \mathbf{x} \geq \lambda_{\min}\{A\} \|\mathbf{x}\|^2$  and  $\mathbf{x}^\top A \mathbf{y} \geq -\lambda_{\max}\{A\} \|\mathbf{x}\| \|\mathbf{y}\|$ ,  $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ , a lower bound for  $L$  in (22) is given by

$$L \geq \frac{1}{2} \left[ (\lambda_{\min}\{K_p\} + \alpha \lambda_{\min}\{K_d\}) \|\mathbf{e}\|^2 - 2\alpha \lambda_{\max}\{\Lambda\} \|\mathbf{e}\| \|\dot{\mathbf{e}}\| + \lambda_{\min}\{\Lambda\} \|\dot{\mathbf{e}}\|^2 + \lambda_{\min}\{K_i\} \|\xi\|^2 + \text{Tr} \left\{ \tilde{W}^\top N^{-1} \tilde{W} \right\} \right],$$

which can be rewritten as

$$L \geq \frac{1}{2} \left[ \lambda_{\min}\{K_p\} \|\mathbf{e}\|^2 + \mathbf{s}^\top \Phi \mathbf{s} + \text{Tr} \left\{ \tilde{W}^\top N^{-1} \tilde{W} \right\} \right],$$

where  $\mathbf{s} = [\|\mathbf{e}\| \ \|\dot{\mathbf{e}}\| \ \|\xi\|]^\top \in \mathbb{R}^3$  and

$$\Phi = \begin{bmatrix} \alpha \lambda_{\min}\{K_d\} & -\alpha \lambda_{\max}\{\Lambda\} & 0 \\ -\alpha \lambda_{\max}\{\Lambda\} & \lambda_{\min}\{\Lambda\} & 0 \\ 0 & 0 & \lambda_{\min}\{K_i\} \end{bmatrix} \in \mathbb{R}^{3 \times 3}.$$

Satisfying condition (20) matrix  $\Phi$  is guaranteed to be positive definite, implying that  $L$  in (22) is a globally positive definite and radially unbounded function. Furthermore,  $L$  is also a decrescent function since it does not depend explicitly on time.

The time derivative of  $L$  in (22) along the trajectories of the closed-loop system (17)-(19) is given by

$$\dot{L} = -\alpha \mathbf{e}^\top K_p \mathbf{e} - \dot{\mathbf{e}}^\top (K_d - \alpha \Lambda) \dot{\mathbf{e}} - (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \Delta \text{sign}(\alpha \mathbf{e} + \dot{\mathbf{e}}) + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \tilde{W}^\top \sigma + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \epsilon + \text{Tr} \left\{ \tilde{W}^\top N^{-1} \dot{\tilde{W}} \right\}.$$

Substituting the equation (18) yields

$$\begin{aligned} \dot{L} = & -\alpha \mathbf{e}^\top K_p \mathbf{e} - \dot{\mathbf{e}}^\top (K_d - \alpha \Lambda) \dot{\mathbf{e}} - (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \Delta \text{sign}(\alpha \mathbf{e} + \dot{\mathbf{e}}) + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \tilde{W}^\top \sigma + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \epsilon \\ & + \text{Tr} \left\{ \kappa \|\alpha \mathbf{e} + \dot{\mathbf{e}}\| \tilde{W}^\top (W - \tilde{W}) - \tilde{W}^\top \sigma (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \right\}, \end{aligned}$$

by using the properties

$$\begin{aligned} \text{Tr}\{A + B\} &= \text{Tr}\{A\} + \text{Tr}\{B\}, \\ \text{Tr}\{cA\} &= c\text{Tr}\{A\}, \quad \forall c \in \mathbb{R}, \end{aligned}$$

and the facts

$$\begin{aligned} (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \tilde{W}^\top \sigma &= \text{Tr} \left\{ \tilde{W}^\top \sigma (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \right\}, \\ (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \Delta \text{sign}(\alpha \mathbf{e} + \dot{\mathbf{e}}) &= \|\Delta(\alpha \mathbf{e} + \dot{\mathbf{e}})\|_1, \end{aligned}$$

the time derivative of (22) along the trajectories of the closed-loop system is given by

$$\dot{L} = -\mathbf{e}^\top \alpha K_p \mathbf{e} - \dot{\mathbf{e}}^\top (K_d - \alpha \Lambda) \dot{\mathbf{e}} - \|\Delta(\alpha \mathbf{e} + \dot{\mathbf{e}})\|_1 + \kappa \|\alpha \mathbf{e} + \dot{\mathbf{e}}\| \text{Tr} \left\{ \tilde{W}^\top (W - \tilde{W}) \right\} + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \epsilon. \quad (23)$$

Further comments on the relation of  $\dot{L}$  in (23) with the proposed controller  $\mathbf{u}$  in (14)-(15) and the adaptation law  $\dot{\tilde{W}}$  in (16) are given in the Appendix. Since  $-\mathbf{x}^\top A \mathbf{x} \leq -\lambda_{\min}\{A\} \|\mathbf{x}\|^2$  and  $\mathbf{x}^\top A \mathbf{x} \leq \lambda_{\max}\{A\} \|\mathbf{x}\|^2$ , an upper bound for  $\dot{L}$  in (23) is

$$\dot{L} \leq -\alpha \lambda_{\min}\{K_p\} \|\boldsymbol{e}\|^2 - (\lambda_{\min}\{K_d\} - \alpha \lambda_{\max}\{\Lambda\}) \|\dot{\boldsymbol{e}}\|^2 - \lambda_{\min}\{\Delta\} \|\alpha \boldsymbol{e} + \dot{\boldsymbol{e}}\|_1 \\ + \kappa \|\alpha \boldsymbol{e} + \dot{\boldsymbol{e}}\| \operatorname{Tr}\{\tilde{W}^\top (W - \tilde{W})\} + \|\alpha \boldsymbol{e} + \dot{\boldsymbol{e}}\| \|\boldsymbol{\epsilon}\|_1.$$

From the Frobenius norm [71], it is known that

$$\operatorname{Tr}\{\tilde{W}^\top (W - \tilde{W})\} = \left| \langle \tilde{W}, W \rangle_F \right| - \|\tilde{W}\|_F^2, \\ \leq \|\tilde{W}\|_F \|W\|_F - \|\tilde{W}\|_F^2.$$

Furthermore, using the fact

$$\|\tilde{W}\|_F^2 - \|\tilde{W}\|_F \|W\|_F = \left( \|\tilde{W}\|_F - \frac{\|W\|_F}{2} \right)^2 - \frac{\|W\|_F^2}{4},$$

and recalling the following upper bounds

$$\|\boldsymbol{\epsilon}\|_1 \leq \epsilon_0, \\ \|\boldsymbol{W}\|_F \leq W_0, \\ \|\boldsymbol{x}\| \leq \|\boldsymbol{x}\|_1, \forall \boldsymbol{x} \in \mathbb{R}^n,$$

the upper bound for (23) is given by

$$\dot{L} \leq -\alpha \lambda_{\min}\{K_p\} \|\boldsymbol{e}\|^2 - (\lambda_{\min}\{K_d\} - \alpha \lambda_{\max}\{\Lambda\}) \|\dot{\boldsymbol{e}}\|^2 \\ - \kappa \|\alpha \boldsymbol{e} + \dot{\boldsymbol{e}}\| \left( \|\tilde{W}\|_F - \frac{W_0}{2} \right)^2 - \|\alpha \boldsymbol{e} + \dot{\boldsymbol{e}}\|_1 \left( \lambda_{\min}\{\Delta\} - \frac{\kappa W_0^2}{4} - \epsilon_0 \right),$$

which can be rewritten as

$$\dot{L} \leq -\boldsymbol{\rho}^\top Q \boldsymbol{\rho} - \kappa \|\alpha \boldsymbol{e} + \dot{\boldsymbol{e}}\| \left( \|\tilde{W}\|_F - \frac{W_0}{2} \right)^2 - \|\alpha \boldsymbol{e} + \dot{\boldsymbol{e}}\|_1 \left( \lambda_{\min}\{\Delta\} - \frac{\kappa W_0^2}{4} - \epsilon_0 \right), \quad (24)$$

where  $\boldsymbol{\rho} = [\|\boldsymbol{e}\| \ \|\dot{\boldsymbol{e}}\|]^\top \in \mathbb{R}^2$  and

$$Q = \begin{bmatrix} \alpha \lambda_{\min}\{K_p\} & 0 \\ 0 & \lambda_{\min}\{K_d\} - \alpha \lambda_{\max}\{\Lambda\} \end{bmatrix} \in \mathbb{R}^{2 \times 2}.$$

In order to guarantee  $\dot{L}$  in (23) to be negative semidefinite, conditions (20) and (21) must be satisfied. Notice that  $\lambda_{\min}\{K_d\}/\lambda_{\max}\{\Lambda\}$  must be satisfied to ensure  $Q$  to be positive definite. This holds if  $\alpha$  is selected as in (20) since  $\lambda_{\min}\{K_d\} \lambda_{\min}\{\Lambda\}/\lambda_{\max}\{\Lambda\}^2 \leq \lambda_{\min}\{K_d\}/\lambda_{\max}\{\Lambda\}$ . Thus, condition (20) also ensures  $Q$  to be a positive definite matrix and condition (21) ensures that the last term of (24) stands negative. Then, while conditions (20) and (21) are met,  $L > 0$  and  $\dot{L} \leq 0$ , implying that  $\boldsymbol{e}(t)$ ,  $\dot{\boldsymbol{e}}(t)$ ,  $\boldsymbol{\xi}(t)$ , and  $\tilde{W}(t)$  are bounded. In fact,  $\|\dot{\boldsymbol{e}}(t)\|$  can be explicitly bounded by  $L(t_0)$  as

$$\|\dot{\boldsymbol{e}}(t)\| \leq \sqrt{\frac{2L(t_0)}{\lambda_{\min}\{\Delta\}}}, \forall t \geq t_0.$$

At the same time by inequality (12), we have that

$$\|\boldsymbol{x}(t)\| \leq 2q_{dB} + \sqrt{\frac{2L(t_0)}{\lambda_{\min}\{\Delta\}}}, \forall t \geq t_0.$$

This means that there exists a compact level set  $\Omega$  containing the origin of the state space such that the inequality

$$L(t_0) \leq \frac{1}{2} \lambda_{\min}\{\Delta\} (b_x - 2q_{dB})^2$$

holds. This implies that for small enough initial conditions  $\boldsymbol{e}(t_0)$ ,  $\dot{\boldsymbol{e}}(t_0)$ ,  $\tilde{W}(t_0)$ , and  $\boldsymbol{\xi}(t_0)$  inside  $\Omega$ ,  $\dot{\boldsymbol{e}}(t) \in \mathcal{S}_{\dot{\boldsymbol{e}}}$  and  $\boldsymbol{x}(t) \in \mathcal{S}_x$ , for all  $t \geq t_0$ . Therefore, the function approximation property is always satisfied.

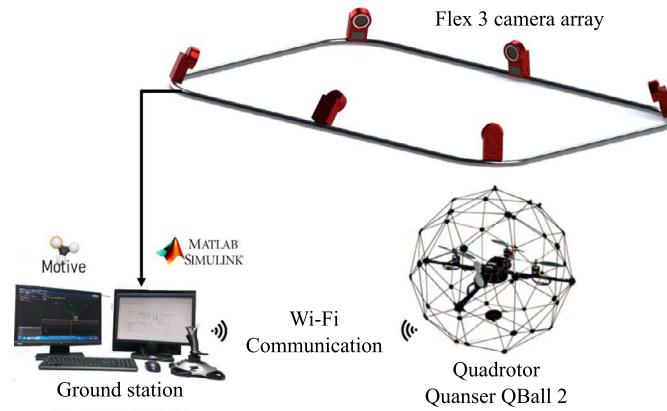
Finally, integrating both sides of (24) and solving for the left-hand side, the following is obtained

$$\frac{L(t_0)}{\lambda_{\min}\{Q\}} \geq \int_{t_0}^t \left\| [\|\boldsymbol{e}(t)\| \ \|\dot{\boldsymbol{e}}(t)\|]^\top \right\|^2 dt, \quad (25)$$

from which it is observed that the integral is bounded.

Convergence of  $\boldsymbol{e}(t)$  and  $\dot{\boldsymbol{e}}(t)$  can be demonstrated using functional spaces. Boundedness of  $\boldsymbol{e}(t)$  and  $\dot{\boldsymbol{e}}(t)$  implies  $\boldsymbol{e}(t), \dot{\boldsymbol{e}}(t) \in L_\infty^4$ . Moreover, (25) implies that  $\boldsymbol{e}(t)$  is a square-integrable function, i.e.,  $\boldsymbol{e}(t) \in L_2^4$ . From Lemma A.5 of [73] it is concluded that

$$\lim_{t \rightarrow \infty} \boldsymbol{e}(t) = 0. \quad (26)$$



**Fig. 2.** Quanser's QBall 2 quadrotor integrated with the motion capture system OptiTrack.

Convergence of  $\dot{\mathbf{e}}(t)$  can be also ensured because  $\ddot{\mathbf{e}}(t)$  in (17) is bounded, i.e.,  $\ddot{\mathbf{e}}(t) \in L_\infty^4$ . Therefore, considering (26),  $\mathbf{e}(t), \dot{\mathbf{e}}(t), \ddot{\mathbf{e}}(t) \in L_\infty^4$ , and Lemma A.6 of [73], it is concluded that

$$\lim_{t \rightarrow \infty} \dot{\mathbf{e}}(t) = 0. \quad (27)$$

Based on (26) and (27), the control objective in (10) is achieved. This completes the proof of Proposition 1.  $\square$

#### 4. Experimental results

Experiments were carried out in order to assess the performance of the proposed controller. Comparisons were made considering three controllers: a proportional-derivative (PD), an ANN with an integral sliding modes scheme, and another ANN. The experiments consisted of achieving the trajectory tracking task of an eight-shaped path at a fixed height. The performance of the schemes was evaluated by comparing the root mean squared (RMS) value of the tracking errors  $\mathbf{e}(t)$  and their time derivatives  $\dot{\mathbf{e}}(t)$  during the steady-state period. The experiments were performed using the QBall 2 quadrotor from Quanser® and the motion capture system OptiTrack®, available at the Control Systems Laboratory of CITEDI-IPN.

##### 4.1. Experimental platform

The experimental setup is composed of the QBall 2 quadrotor and the motion capture system OptiTrack®, as depicted in Fig. 2. The QBall 2 quadrotor is a research-oriented platform and has been used in many other works [61,74–79]. The system OptiTrack® measures the cartesian position of the quadrotor and the yaw angle in real-time through a set of six Flex 3 cameras. The roll and pitch angles and the angular velocities of the quadrotor are obtained by the inertial measurement unit (IMU) onboard the quadrotor, which is formed of a 3-axis gyroscope and a 3-axis accelerometer. The data acquisition rates of the IMU and the OptiTrack® system are 500 [Hz] and 30 [Hz], respectively. The controller is developed in MATLAB®-Simulink®, converted into an executable program through Quarc software, and uploaded via Wi-Fi to the quadrotor's onboard computer. The uploaded controller is executed into the onboard computer at a 500 [Hz] frequency. In this work, the parameters corresponding to the system formed by the QBall 2 quadrotor with the inner controller (3)-(5) are the following:

$$\begin{aligned} m &= 1.79 \text{ [kg]}, & g &= 9.81 \text{ [m/s}^2\text{]}, \\ \omega_\phi = \omega_\theta &= 13.944 \text{ [Hz]}, & \xi_\phi = \xi_\theta &= 1.593, \\ \tau_z = \tau_{\dot{\psi}} &= 0.728 \text{ [s]}, & K_u &= \text{diag}\{15.812, 15.812, 10.6, 2.38\}. \end{aligned} \quad (28)$$

##### 4.2. Controllers implemented in the comparison

###### 4.2.1. PD controller

The PD scheme performance was used as a reference point to assess the other schemes' performance. The PD scheme is given by

$$\mathbf{u} = [T(\psi)K_u]^{-1}[\ddot{\mathbf{q}}_d + K_1\mathbf{e} + K_2\dot{\mathbf{e}}],$$

being  $\mathbf{e}$  the trajectory tracking error defined in (8),  $K_1$  and  $K_2 \in \mathbb{R}^{4 \times 4}$  diagonal positive definite gain matrices for the proportional and derivative control actions, respectively, and  $K_u$  the matrix containing parameters of the inner loop controller (3)-(5). The PD outer loop controller will be referenced as OLPDC (outer loop PD controller) and was implemented using  $K_u$  as in (28) and the following gain matrices

$$K_1 = \text{diag}\{4.5, 3.0, 7.5, 3.0\},$$

$$K_2 = \text{diag}\{1.5, 1.5, 1.5, 1.5\}.$$

#### 4.2.2. Generalized regression adaptive neural network controller

The controller presented in [61] is a generalized regression adaptive neural network-based scheme. The generalized regression neural network is a special kind of the radial basis function neural network on which the activation function is a normalized Gaussian function. This scheme serves as an outer loop controller for the system formed by the quadrotor with an inner loop controller in (7) and is defined as

$$\mathbf{u} = [T(\psi)K_u]^{-1} \left[ \ddot{\mathbf{q}}_d - b_1 \tanh(b_2 \mathbf{h}) - \hat{W}_G^\top \sigma_{GR}(\mathbf{y}_G, \hat{\mathbf{c}}, \hat{\rho}) - K_r \text{sign}(\mathbf{h}) \right], \quad (29)$$

where  $\mathbf{h} = \Upsilon \mathbf{e}_G + \dot{\mathbf{e}}_G$  is the filtered error,  $\Upsilon$  and  $K_r \in \mathbb{R}^{4 \times 4}$  are diagonal positive definite matrices,  $\mathbf{e}_G = \mathbf{q} - \mathbf{q}_d$  is the tracking error,  $b_1$  and  $b_2$  are positive constants, and  $\hat{W}_G^\top \sigma_{GR}(\mathbf{y}_G, \hat{\mathbf{c}}, \hat{\rho})$  represents the generalized regression adaptive neural network. The activation function is given by

$$\sigma_{GRi}(\mathbf{y}_G, \hat{\mathbf{c}}, \hat{\rho}_i) = \frac{e^{\frac{\|\mathbf{y}_G - \mathbf{c}_i\|}{2\rho_i}}}{\sum_{j=1}^l e^{\frac{\|\mathbf{y}_G - \mathbf{c}_j\|}{2\rho_j}}}, \quad (30)$$

where  $\mathbf{y}_G$  is the input vector of the neural network,  $l$  is the number of neurons,  $\hat{\mathbf{c}}$  and  $\hat{\rho}$  are the center and standard deviation vector estimations, respectively,  $\hat{W}_G$  is the neural network output weight matrix estimation, being the estimates obtained through the following adaptation laws

$$\dot{\hat{W}}_G = b_3 \sigma_{GR}(\mathbf{y}_G, \hat{\mathbf{c}}, \hat{\rho}) \tanh(b_2 h_i), \quad (31)$$

$$\dot{\hat{\mathbf{c}}} = b_4 \sum_{i=1}^n \left[ \tanh(b_2 h_i) \left( \Delta \hat{\sigma}_c^\top \hat{W}_{Gi}(t) \right) \right], \quad (32)$$

$$\dot{\hat{\rho}} = b_5 \sum_{i=1}^n \left[ \tanh(b_2 h_i) \left( \Delta \hat{\sigma}_\rho^\top \hat{W}_{Gi}(t) \right) \right], \quad (33)$$

where

$$\Delta \hat{\sigma}_c = \frac{\partial \sigma_{GR}(\mathbf{y}_G, \mathbf{c}, \rho)}{\partial \mathbf{c}} \Bigg|_{\begin{array}{l} \rho = \hat{\rho} \\ \mathbf{c} = \hat{\mathbf{c}} \end{array}}, \quad (34)$$

$$\Delta \hat{\sigma}_\rho = \frac{\partial \sigma_{GR}(\mathbf{y}_G, \mathbf{c}, \rho)}{\partial \rho} \Bigg|_{\begin{array}{l} \rho = \hat{\rho} \\ \mathbf{c} = \hat{\mathbf{c}} \end{array}}, \quad (35)$$

$b_3$ ,  $b_4$ , and  $b_5$  are positive constants. The outer loop generalized regression adaptive neural network-based controller in (29)-(35) will be referred to as GRNN (generalized regression neural network-based controller) and was implemented with the following gains

$$b_1 = 1.5, \quad b_2 = 1.0, \quad b_3 = 15, \quad K_u = \text{diag}\{15.812, 15.812, 10.6, 2.38\},$$

$$b_4 = 0.1, \quad b_5 = 0.1, \quad l = 5, \quad K_r = \text{diag}\{0.001, 0.001, 0.001, 0.001\},$$

$$\Upsilon = \text{diag}\{3, 2, 5, 2\}.$$

#### 4.2.3. Adaptive neural network integral sliding mode controller

The controller described below was introduced in [80]. This control scheme is conceived for the quadrotor dynamic model in (1)-(2), implying free access to the vehicle dynamics and actuators. It also has a two-loop configuration, but in this case, there are no limitations regarding the inner loop, and the outer loop controller computes no kinematic input commands. In other words, this scheme was developed for what in this document is referred to as a research-oriented quadrotor. In this control scheme, the inner loop is an integral sliding mode scheme that handles the attitude dynamics, and the outer loop is a radial basis function adaptive neural network-based scheme that handles the position dynamics. The radial basis function adaptive neural network-based position controller is given by

$$F = u_{p3}, \quad (36)$$

$$\phi_d(t) = \sin^{-1} [u_{p1} \sin(\psi) - u_{p2} \cos(\psi)], \quad (37)$$

$$\theta_d(t) = \sin^{-1} [(u_{p1} \cos(\psi) + u_{p2} \sin(\psi)) / \cos(\phi_d)], \quad (38)$$

$$\mathbf{u}_p = F \begin{bmatrix} \cos(\psi) \sin(\theta) \cos(\phi) + \sin(\psi) \sin(\phi) \\ \sin(\psi) \sin(\theta) \cos(\phi) - \cos(\psi) \sin(\phi) \\ 1 \end{bmatrix}, \quad (39)$$

$$\mathbf{u}_p = H(\boldsymbol{\eta})^{-1} \left[ \ddot{\mathbf{p}}_d + \mathbf{g}_z + \hat{W}_p^\top \sigma_{RBF}(\mathbf{y}_p) + k_v \mathbf{r} \right], \quad (40)$$

matrix  $H(\boldsymbol{\eta}) = 1/m \text{diag}\{1, 1, \cos(\phi) \cos(\theta)\} \in \mathbb{R}^{3 \times 3}$ ,  $\mathbf{r} = \Lambda \mathbf{e}_p + \dot{\mathbf{e}}_p \in \mathbb{R}^3$  is the filtered error, being  $\Lambda = \Lambda^\top > 0$ , and  $\mathbf{e}_p = \mathbf{p}_d - \mathbf{p} \in \mathbb{R}^3$  the position error, where  $\mathbf{p}_d \in \mathbb{R}^3$  is the position reference signal, and  $k_v \in \mathbb{R}$  is a strictly positive constant. The desired acceleration term  $\ddot{\mathbf{p}}_d = [\ddot{x}_d, \ddot{y}_d, \ddot{z}_d]^\top$  was introduced by us to improve the performance of the controller. The term  $\hat{W}_p^\top \sigma_{RBF}(\mathbf{y}_p)$  is the radial basis function neural network where  $\mathbf{y}_p$  is the input vector of the neural network with an activation function defined by

$$\sigma_{\text{RBF}}(\gamma_p) = e^{\left(-\frac{\|\gamma_p - c_i\|^2}{\rho_i^2}\right)},$$

where  $c_i$  is the center and  $\rho_i$  is the width of the Gaussian function. The matrix  $\hat{W}_p$  is the output weight matrix estimate, which obtained through the following adaptation law

$$\dot{\hat{W}}_p = A\sigma_{\text{RBF}}(\gamma_p)\mathbf{r}^\top, \quad (41)$$

being  $A$  a symmetric positive definite matrix. The integral sliding mode attitude controller is defined as

$$\tau = M(\eta)\dot{\mathbf{v}}_\eta + C(\eta, \dot{\eta})\dot{\eta} + \rho_\eta \text{sign}(\mathbf{s}_\eta) + k_\eta M\tilde{\mathbf{v}}, \quad (42)$$

where  $\mathbf{e}_\eta = \eta_d - \eta$  is the attitude error,  $\eta_d$  the desired attitude,  $\tilde{\mathbf{v}} = \mathbf{v}_\eta - \dot{\eta}$  is the angular velocity error,  $\mathbf{v}_\eta = \dot{\eta}_d + k_w \mathbf{e}_\eta + \rho_w \text{sign}(\mathbf{s}_w)$ ,  $\mathbf{s}_w = \mathbf{e}_\eta + k_w \int_0^\eta \mathbf{e}_\eta$  and  $\mathbf{s}_\eta = \tilde{\mathbf{v}} + k_\eta \int_0^\eta \tilde{\mathbf{v}}$  are the integral sliding surfaces, and  $\rho_\eta$ ,  $k_\eta$ ,  $\rho_w$ , and  $k_w$  are strictly positive constants. Deeper details about this control scheme can be obtained in [80]. The adaptive neural network integral sliding mode controller in (36)-(42) will be denoted as ANNISM (adaptive neural network integral sliding mode controller) and was implemented using the following values

$$\begin{aligned} \mathbf{c} &= [-1.5, -1, -0.5, 0, 0.5, 1, 1.5]^\top, & k_w &= 4.5, \\ \boldsymbol{\rho} &= [5, 5, 5, 5, 5, 5]^\top, & \rho_w &= 1 \times 10^{-6}, \\ \Lambda &= \text{diag}\{2.55, 2.55, 3.55\}, & k_\eta &= 15, \\ A &= \text{diag}_7\{0.15\}, & \rho_\eta &= 1 \times 10^{-4}, \\ k_v &= 1.257. \end{aligned}$$

#### 4.2.4. Proposed controller

The controller in (14)-(16) will be denoted as OLANN (outer loop adaptive neural network controller) in the remainder of the document and was implemented with the following gains:

$$\begin{aligned} K_p &= \text{diag}\{0.22, 0.165, 0.3, 0.5\}, & N &= \text{diag}_{10}\{0.189\}, \\ K_i &= \text{diag}\{0.0041, 0.0041, 0.1655, 0.1655\}, & \alpha &= 0.2417, \\ K_d &= \text{diag}\{0.165, 0.1375, 0.32, 0.1\}, & \kappa &= 2.5, \\ \Delta &= \text{diag}_4\{0.001\}, & L_N &= 10. \end{aligned} \quad (43)$$

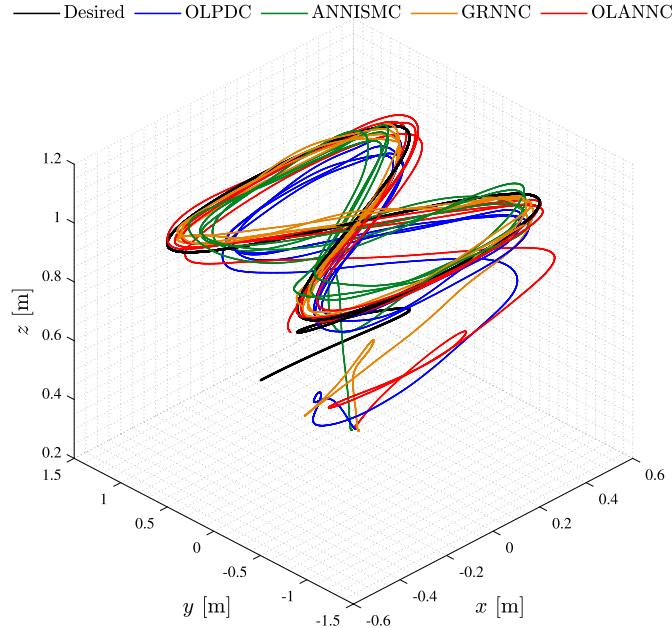
Random numbers in the set  $[-1, 1]$  were used to define the input weight matrix  $V$ . The output weight matrix  $\hat{W}$  was initialized with zeros. The matrix  $N$  governs the change rate of the output weights; greater values will increase this feature. Nevertheless, it could produce aggressive maneuvering and high-frequency oscillations in the attitude dynamics, especially during the take-off. Small values are recommended for implementation purposes. Notice that initializing the output weights estimates as zero helps avoid the NN contributing considerably at the beginning of the experiment. This lets the non-neural part of the scheme stabilize the system and diminish the tracking errors while the NN starts learning. Notice that the complexity of the proposed controller relies on the following characteristics of the neural network: the number of hidden layers, the number of neurons, and the number of outputs. Therefore, the number of differential equations solved online on each iteration is the product of the number of neurons and the number of outputs.

#### 4.3. Trajectory tracking

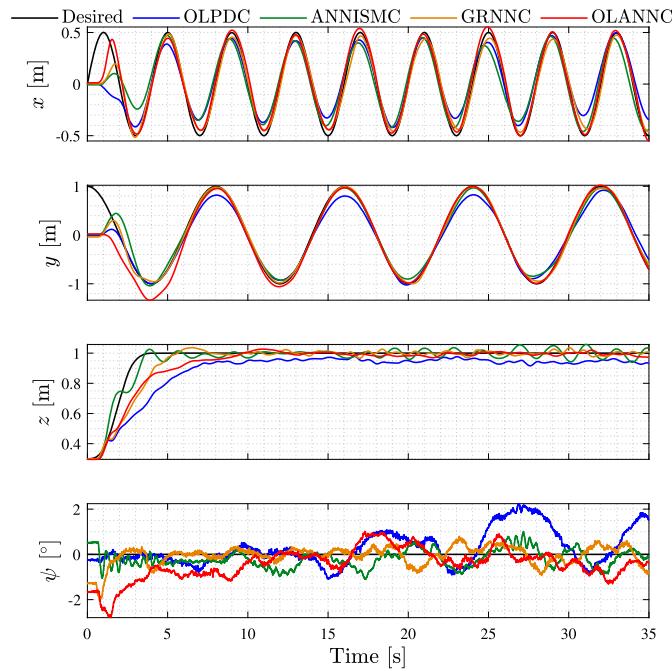
A trajectory tracking task was used to validate the proposed controller's functionality and assess its performance. The assigned trajectory draws an eight-shaped path described by the following equations

$$\begin{aligned} x_d(t) &= 0.5 \sin\left(\frac{2\pi}{4}t\right) [\text{m}], \\ y_d(t) &= \cos\left(\frac{2\pi}{8}t\right) [\text{m}], \\ z_d(t) &= \begin{cases} 1 - 0.7e^{-0.1t^3} [\text{m}], & t \leq 5, \\ 1 [\text{m}], & t > 5, \end{cases} \\ \psi_d(t) &= 0.0 [\text{°}]. \end{aligned} \quad (44)$$

The paths drawn by the quadrotor by implementing the controllers OLPDC, ANNISM, GRNN, and OLANN are depicted in Fig. 3. The time evolution of  $x(t)$ ,  $y(t)$ ,  $z(t)$ , and  $\psi(t)$  is presented in Fig. 4, where it can be observed that all the schemes accomplish the assigned task. Nonetheless, the ANNC-based controllers performed better than the OLPDC, especially in height  $z(t)$ . In Fig. 5, the tracking errors of the position and yaw angle signals are shown, from which it is easier to appreciate that the tracking errors obtained with the proposed OLANN scheme remain close to zero. Notice that the transient period of the proposed OLANN is larger than the one for the other schemes; this is because it takes some time for the neural network to adapt due to the absence of system parameters in the controller. Nevertheless, once the transient has vanished, the tracking errors are competitive regarding the other schemes. The control actions obtained with each implemented control scheme are depicted in Fig. 6, representing the total thrust  $F(t)$  and the torques on each rotation angle  $\tau(t) = [\tau_\phi(t) \ \tau_\theta(t) \ \tau_\psi(t)]^\top$ . Notice that the control actions obtained with each algorithm are very similar from one to another.



**Fig. 3.** Paths drawn by the quadrotor during the experimental tests by implementing the OLPDC, ANNISMNC, GRNNNC, and proposed OLANNNC schemes. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

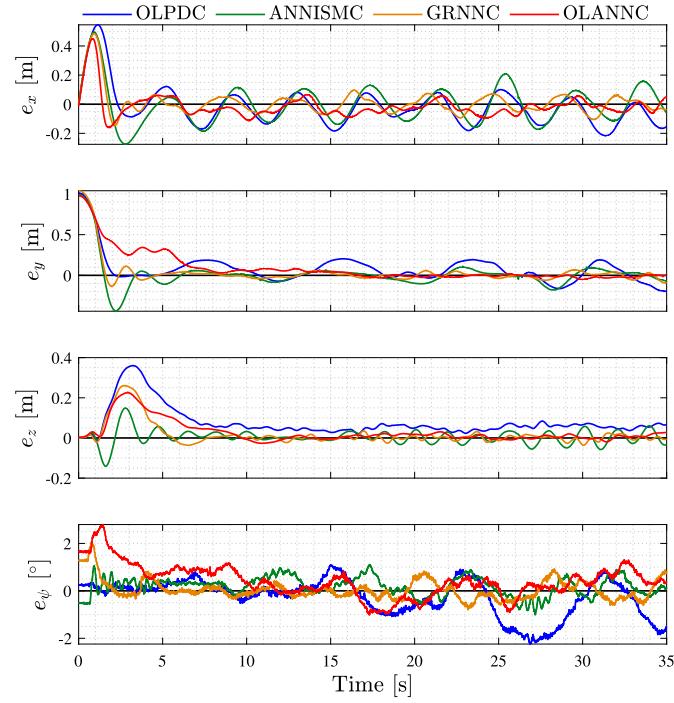


**Fig. 4.** Position and yaw angle signals of the quadrotor during the experimental implementation of the OLPDC, ANNISMNC, GRNNNC, and proposed OLANNNC schemes.

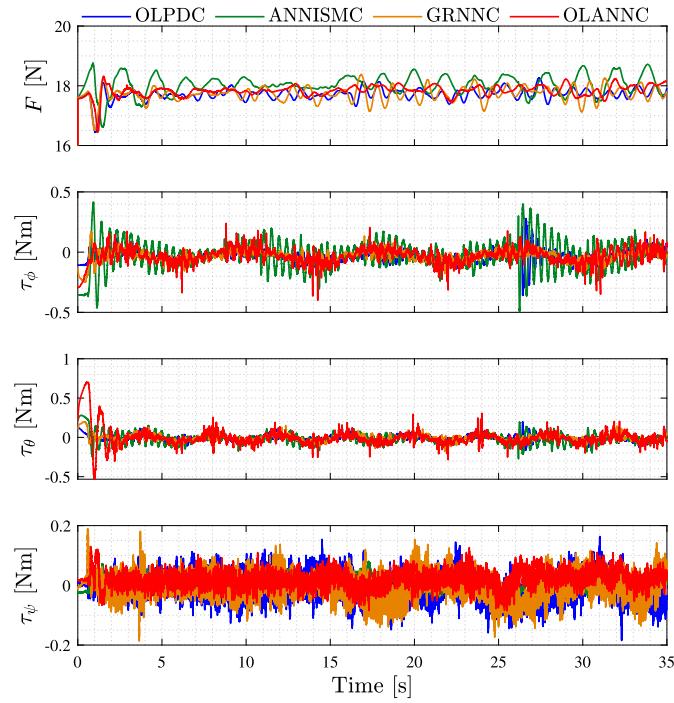
The RMS value of the tracking errors  $e(t)$  and their derivatives  $\dot{e}(t)$  was computed to provide a quantitative index to compare the performance of each implemented scheme. Additionally, the percentage of improvement for all the RMS error values was computed, taking as a comparing basis the performance of the OLPDC. Besides, the mean value of all the percentage of improvements for each controller was computed. The percentage of improvement was obtained with the following expression

$$P_{\text{imp}}\%(\zeta) = \frac{\text{RMS}(\text{OLPDC}) - \text{RMS}(\zeta)}{\text{RMS}(\text{OLPDC})} \times 100\%,$$

where  $\zeta$  represents the RMS error values of each signal obtained with the ANNISMNC, GRNNNC, and OLANNNC schemes. The RMS error values were computed with the data within the time interval  $10 \text{ [s]} \leq t \leq 35 \text{ [s]}$  corresponding to the steady-state period. These results are presented in Table 1, where the highest percentages of improvement are in bold font. As can be observed, all the ANN-based controllers provided a better performance regarding the OLPDC concerning only the tracking errors. This is readily appreciated in the percentage of improvement since all the obtained values are positive. However, only the proposed OLANNNC scheme provided positive values regarding the velocity and yaw angle change rate errors. It is noteworthy that, although the proposed OLANNNC scheme did not provide the highest



**Fig. 5.** Time evolution of the tracking errors during the experimental tests by implementing the OLPDC, ANNISMIC, GRNNNC, and proposed OLANNC schemes.



**Fig. 6.** Control actions provided by the OLPDC, ANNISMIC, GRNNNC, and proposed OLANNC schemes during the experimental tests.

percentage of improvement for each signal, its overall performance is the best. Besides, we must recall that no parameters of the quadrotor and the inner controller were included in this scheme, being not the case for the other ANN-based schemes. In addition, in Table 2, the mean  $\mu_F$  and standard deviation  $\sigma_F$  values of the total thrust  $F(t)$  computed by each scheme into the time interval  $10 \text{ [s]} \leq t \leq 35 \text{ [s]}$  are presented. The maximum variation among the total thrust means values are 1.8%, while for the standard deviation, the maximum variation is 44%. This variation can be due to mainly two factors: the structure of the controllers and the peaks in the total thrust signal. Notice that, in contrast to the ANNISMIC scheme, the OLPDC, GRNNNC, and OLANNC algorithms are outer loop schemes with the same inner loop controller. The ANNISMIC scheme presents the largest values for the mean and standard deviation, which can be due to this variation in the scheme's structure.

**Table 1**

RMS values of the tracking error  $e(t)$  and its time derivative  $\dot{e}(t)$  obtained by implementing the OLPDC, ANNISM, GRNNC, and proposed OLANNC schemes tracking the eight-shaped path in the experimental tests.

Signal	OLPDC	ANNISM	$P_{imp}\%$	GRNNC	$P_{imp}\%$	OLANNC	$P_{imp}\%$
$e_x$ [m]	0.0981	0.0952	2.93	0.0480	51.11	0.0475	<b>51.63</b>
$e_y$ [m]	0.1104	0.0637	42.36	0.0348	68.47	0.0330	<b>70.12</b>
$e_z$ [m]	0.0525	0.0256	51.29	0.0125	<b>76.17</b>	0.0132	74.81
$e_\psi$ [°]	0.9444	0.4566	51.65	0.3426	<b>63.72</b>	0.4934	47.76
$\dot{e}_x$ [m/s]	0.0654	0.0773	-18.23	0.0444	32.15	0.0413	<b>36.81</b>
$\dot{e}_y$ [m/s]	0.0537	0.0413	23.14	0.0351	34.61	0.0249	<b>53.56</b>
$\dot{e}_z$ [m/s]	0.0166	0.0384	-132.04	0.0257	-55.06	0.0115	<b>30.76</b>
$\dot{e}_\psi$ [°/s]	1.0723	0.7651	<b>28.65</b>	1.0781	-0.55	0.8382	21.83
$\mu$			6.22		33.83		<b>48.41</b>

**Table 2**

Mean and standard deviation of the total thrust signal  $F(t)$  obtained by implementing the OLPDC, ANNISM, GRNNC, and proposed OLANNC schemes tracking the eight-shaped path in the experimental tests.

Signal	OLPDC	ANNISM	GRNNC	OLANNC
$\mu_F$	17.715	18.037	17.731	17.833
$\sigma_F$	0.2372	0.3448	0.2765	0.2279

#### 4.4. Performance under external disturbances

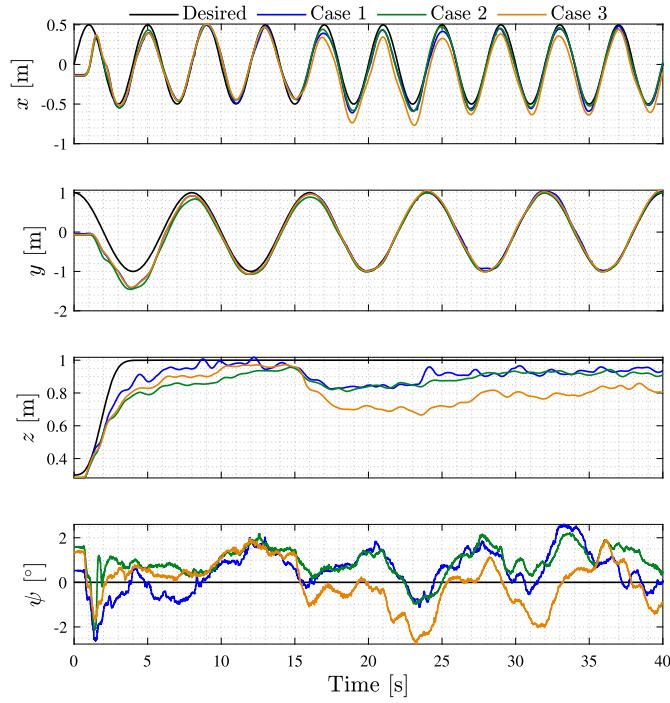
A set of three experiments was performed to show the proposed controller's robustness. In all the experiments, the trajectory described in (44) was tracked in the presence of an actuator fault by implementing the proposed OLANNC scheme using the gains in (43). The actuator fault considered in these tests was the constant partial loss of effectiveness [81–83]. This phenomenon could arise when the propeller suffers any change in its geometry due to physical damage [83]. Therefore, its capability to provide thrust diminishes. Three cases were studied where the actuator presenting the fault is the rear actuator. In the three cases, the fault occurs for all time  $t \geq 15$  [s]. The three cases were:

- **Case 1:** A constant partial loss of effectiveness of 10% in the rear actuator.
- **Case 2:** A constant partial loss of effectiveness of 10% in the rear actuator. Furthermore, a mass of 0.064 [kg] was added at a distance of 0.33 [m] from the vehicle's center of mass (just at the tip of the quadrotor's left arm). Thus, implying an uncertainty on the inertia tensor and the incorporation of unmodeled dynamics. The mass was added before the beginning of the experiment and remained in the same place during the whole experiment.
- **Case 3:** A constant partial loss of effectiveness of 20% in the rear actuator.

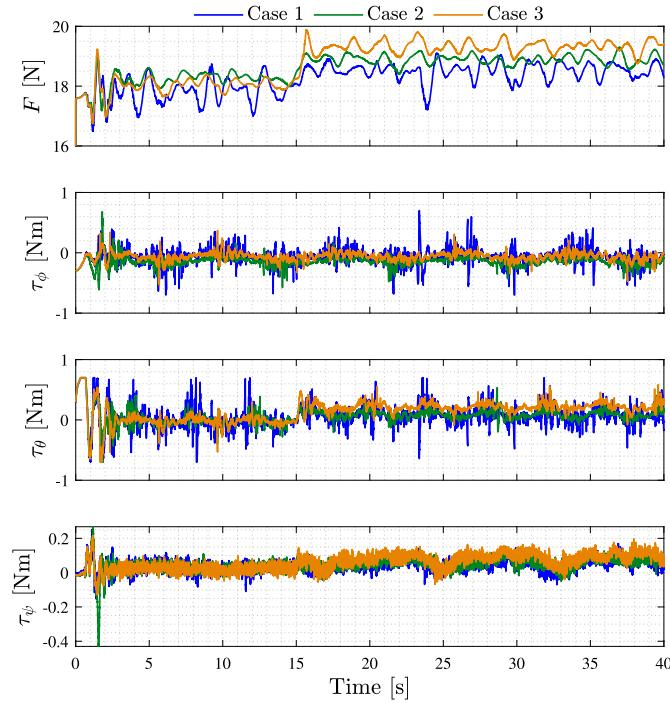
In Fig. 7, the tracking of the desired position and yaw signals for the three cases is depicted. It can be clearly observed how the performance is affected when the actuator fault occurs at  $t \geq 15$  [s]. Though, despite the actuator fault, the task is still being achieved. The most affected state is the position on the z-axis for the three cases, but it can be observed that as time goes by, the desired height is slowly reached. The addition of a mass in Case 2 also incorporates a varying disturbance torque acting over the rolling moment  $\tau_\phi(t)$ . The magnitude of this torque is directly related to the quadrotor's attitude, precisely with the roll angle  $\phi(t)$ . Considering that during Case 2 the roll angle was  $0$  [°]  $\leq |\phi(t)| \leq 12$  [°], the disturbance torque varies from 0.202 [Nm] to 0.207 [Nm]. The behavior in Cases 1 and 2 are similar, but in the y-axis, the oscillations at the beginning of the experiment in Case 2 are due to the disturbance torque. However, after  $t = 10$  [s], the behavior for all the cases is very similar.

Fig. 8 shows the computed control actions by the proposed controller for the three cases. Notice that the total thrust  $F(t)$  must be increased in order to compensate for the partial loss of effectiveness of one actuator. This also has an impact on the computed torques. Notice how the torques related to the pitching and yawing moments  $\tau_\theta(t)$  and  $\tau_\psi(t)$ , respectively, are slightly increased at  $t \geq 15$  [s]. This effect can be easily observed in Case 3. The thrust provided by each actuator is depicted in Fig. 9, where  $F_1$  denotes the front actuator,  $F_2$  the left,  $F_3$  the rear, and  $F_4$  the right. The essential differences in each case lie in the thrust provided by the left and rear actuators. Notice that there are some peeks in the signal corresponding to the left actuator in Case 2 at  $1$  [s]  $\leq t \leq 3$  [s]. Additionally, the thrust in the left actuator in Case 2 is slightly higher than in Cases 1 and 3. This is due to the disturbance torque added in Case 2, which was induced by adding a mass at the tip of the left arm of the quadrotor. The signals for the rear actuator in the three cases are similar before the fault occurs; once the fault is present, the thrust increase to compensate for the partial loss of effectiveness.

The time evolution of the NN output weights corresponding to each case is shown in Fig. 10. The weights are grouped based on their influence on the overall system's output, i.e., all the output weights related to the position in the x-axis dynamics are presented together and so on for the y-axis, z-axis, and yaw angle. Given the structure of the output weight matrix estimates, this is easily achieved by separating them into columns since the first column is directly related to the x-axis dynamics, the second column with the y-axis, the third column with the z-axis, and the fourth column with the yaw angle. Firstly, it can be observed that the weights remain bounded during the operation of the controller, even under external disturbances and actuator fault. Additionally, it is clearly observed when the actuator fault occurs and how the output weights rapidly adapt to overcome this unexpected phenomenon. As aforementioned, the most affected state by the actuator fault is the position on the z-axis. This also can be observed here since the set of output weights that present considerable changes under the actuator fault are those related to the z-axis. Additionally, it is observed that the weights related to the yaw angle are also greatly affected by the actuator fault.



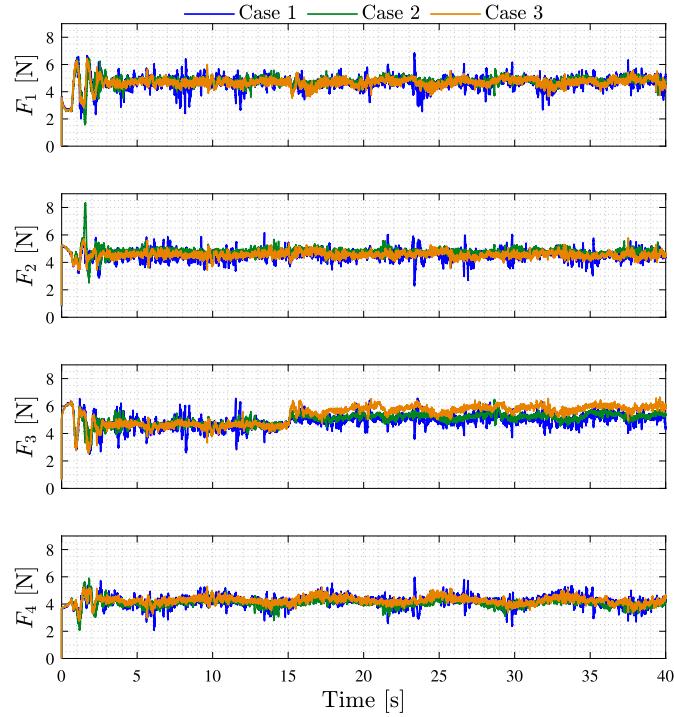
**Fig. 7.** Position and yaw angle signals of the quadrotor during the experimental implementation of the OLANNc in Cases 1, 2, and 3.



**Fig. 8.** Control actions provided by the OLANNc scheme during the experimental implementation in Cases 1, 2, and 3.

## 5. Conclusions

In this manuscript, a novel outer loop adaptive neural network-based controller for trajectory tracking tasks was introduced. The proposed algorithm was designed to operate without knowing the parameters of the quadrotor and the inaccessible inner loop controller. The introduced algorithm has a simple structure, and the activation function of the implemented neural network is simpler than a radial basis function. The convergence error analysis of the closed-loop system supported the functionality of the proposed algorithm ensuring convergence to zero of the tracking error and its derivative. The proposed controller was experimentally tested, and its performance was compared with other ANN-based controllers. The introduced algorithm proved to be robust to parameter uncertainties. The experimental results showed that the proposed scheme provided competitive tracking error signals and the best overall performance. An important limitation of the proposed scheme is the need to meet the condition posed in Assumption 1. Therefore, our future work will address the



**Fig. 9.** Thrust of the actuators by implementing the OLANNC scheme during the experimental tests in Cases 1, 2, and 3. \$F\_1\$: front, \$F\_2\$: left, \$F\_3\$: rear, and \$F\_4\$: right.

implications of not meeting such a condition in the error convergence analysis and the development of a robust controller able to operate without meeting Assumption 1.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Data availability

Data will be made available on request.

#### Appendix A. Derivation of the proposed controller and output weight matrix adaptation law

Consider the trajectory tracking error dynamics in (13) and the non-negative function \$L\$ in (22). The time derivative of \$L\$ is given by

$$\dot{L} = \mathbf{e}^\top (K_p + \alpha K_d)\dot{\mathbf{e}} + \alpha \dot{\mathbf{e}}^\top \Lambda \dot{\mathbf{e}} + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \Lambda \ddot{\mathbf{e}} + \xi^\top K_i \dot{\xi} + \text{Tr} \left\{ \tilde{W}^\top N^{-1} \dot{W} \right\}. \quad (45)$$

Since the neural network's output weights estimation error is given as

$$\tilde{W} = W - \hat{W}, \quad (46)$$

its time derivative is \$\dot{\tilde{W}} = -\dot{\hat{W}}\$ because \$W\$ is constant. Using this fact and substituting (13) in (45) yields

$$\dot{L} = \mathbf{e}^\top (K_p + \alpha K_d)\dot{\mathbf{e}} + \alpha \dot{\mathbf{e}}^\top \Lambda \dot{\mathbf{e}} + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top W^\top \sigma + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \epsilon + \xi^\top K_i \dot{\xi} - \text{Tr} \left\{ \tilde{W}^\top N^{-1} \dot{W} \right\} - (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top T(\psi) \mathbf{u}. \quad (47)$$

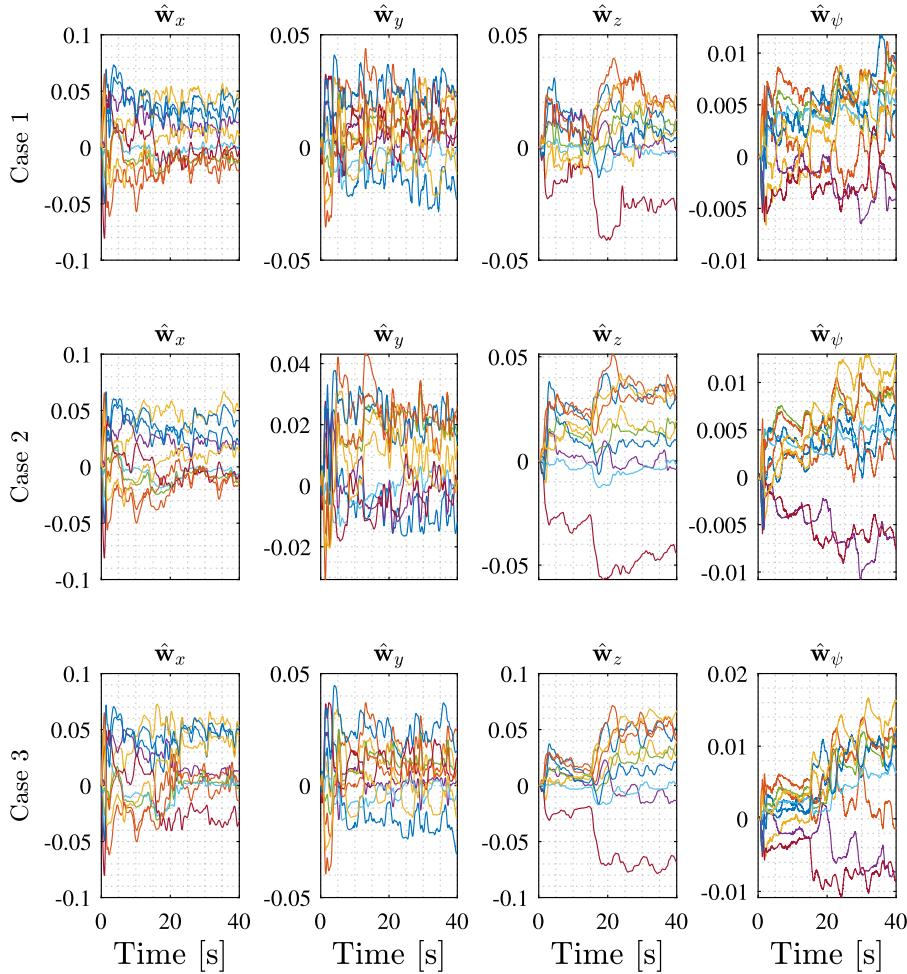
Thus, the controller \$\mathbf{u}\$ in (14) is proposed aiming to make (47) negative semidefinite while taking into account the control goal (10) and the measurable signals. Then, substituting the proposed controller (14) in (47) the following is obtained

$$\begin{aligned} \dot{L} &= -\alpha \mathbf{e}^\top K_p \mathbf{e} - \dot{\mathbf{e}}^\top (K_d - \alpha \Lambda) \dot{\mathbf{e}} + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top W^\top \sigma + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \epsilon + \xi^\top K_i \dot{\xi} - \text{Tr} \left\{ \tilde{W}^\top N^{-1} \dot{W} \right\} \\ &\quad - (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top K_i \dot{\xi} - (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \hat{W}^\top \sigma - (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \Delta \text{sign}(\alpha \mathbf{e} + \dot{\mathbf{e}}). \end{aligned} \quad (48)$$

Now, based on (48), one can observe that by selecting \$\dot{\xi}\$ as in (15), the terms containing \$\dot{\xi}\$ are canceled out, resulting in

$$\begin{aligned} \dot{L} &= -\alpha \mathbf{e}^\top K_p \mathbf{e} - \dot{\mathbf{e}}^\top (K_d - \alpha \Lambda) \dot{\mathbf{e}} + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top W^\top \sigma + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \epsilon - \text{Tr} \left\{ \tilde{W}^\top N^{-1} \dot{W} \right\} \\ &\quad - (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \hat{W}^\top \sigma - (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \Delta \text{sign}(\alpha \mathbf{e} + \dot{\mathbf{e}}). \end{aligned} \quad (49)$$

In agreement with (46) and using the fact \$(\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \Delta \text{sign}(\alpha \mathbf{e} + \dot{\mathbf{e}}) = \|\Delta(\alpha \mathbf{e} + \dot{\mathbf{e}})\|\_1\$, (49) can be rewritten as



**Fig. 10.** ANN output weight estimates  $\hat{W}$  obtained with the OLANNC scheme during the experimental tests in Cases 1, 2, and 3.  $\hat{w}_x$ : first column of  $\hat{W}$ ,  $\hat{w}_y$ : second column of  $\hat{W}$ ,  $\hat{w}_z$ : third column of  $\hat{W}$ , and  $\hat{w}_\psi$ : fourth column of  $\hat{W}$ .

$$\dot{L} = -\alpha \mathbf{e}^\top K_p \mathbf{e} - \dot{\mathbf{e}}^\top (K_d - \alpha \Lambda) \dot{\mathbf{e}} - \|\Delta(\alpha \mathbf{e} + \dot{\mathbf{e}})\|_1 + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \tilde{W}^\top \boldsymbol{\sigma} + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \boldsymbol{\epsilon} - \text{Tr}\{\tilde{W}^\top N^{-1} \dot{W}\}. \quad (50)$$

Then, based on (50), the adaptation law  $\dot{\tilde{W}}$  is proposed as (16) with the aim of canceling out the remaining positive terms. Notice that  $(\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \boldsymbol{\epsilon}$  can not be canceled out because it contains the approximation error of the neural network  $\mathbf{e}$  and the vector of external disturbances and unmodeled dynamics  $\delta$ . Although  $\mathbf{e}$  and  $\delta$  are not measurable, they are bounded. Therefore, the term  $(\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \boldsymbol{\epsilon}$  can be dominated.

After substituting the adaptation law (16) in (50), yields

$$\begin{aligned} \dot{L} &= -\alpha \mathbf{e}^\top K_p \mathbf{e} - \dot{\mathbf{e}}^\top (K_d - \alpha \Lambda) \dot{\mathbf{e}} - \|\Delta(\alpha \mathbf{e} + \dot{\mathbf{e}})\|_1 + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \boldsymbol{\epsilon} \\ &\quad + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \tilde{W}^\top \boldsymbol{\sigma} - \text{Tr}\{\tilde{W}^\top \boldsymbol{\sigma} (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top\} + \kappa \|(\alpha \mathbf{e} + \dot{\mathbf{e}})\| \text{Tr}\{\tilde{W}^\top \dot{W}\}, \end{aligned}$$

where the properties of the trace operator  $\text{Tr}\{A + B\} = \text{Tr}\{A\} + \text{Tr}\{B\}$  and  $\text{Tr}\{cA\} = c\text{Tr}\{A\}$  were used. Furthermore, using the fact  $(\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \tilde{W}^\top \boldsymbol{\sigma} = \text{Tr}\{\tilde{W}^\top \boldsymbol{\sigma} (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top\}$  the following is obtained

$$\dot{L} = -\mathbf{e}^\top \alpha K_p \mathbf{e} - \dot{\mathbf{e}}^\top (K_d - \alpha \Lambda) \dot{\mathbf{e}} - \|\Delta(\alpha \mathbf{e} + \dot{\mathbf{e}})\|_1 + \kappa \|\alpha \mathbf{e} + \dot{\mathbf{e}}\| \text{Tr}\{\tilde{W}^\top (W - \tilde{W})\} + (\alpha \mathbf{e} + \dot{\mathbf{e}})^\top \boldsymbol{\epsilon}. \quad (51)$$

Notice that (51) corresponds to equation (23), which has a useful structure for the error convergence analysis.

## References

- [1] K.P. Valavanis, G.J. Vachtsevanos, *Handbook of Unmanned Aerial Vehicles*, Springer Cham, Cham, Switzerland, 2015.
- [2] M. Idrissi, M. Salami, F. Annaz, A review of quadrotor unmanned aerial vehicles: applications, architectural design and control algorithms, *J. Intell. Robot. Syst.* 104 (2) (2022) 1–33.
- [3] T. Elmokadem, Distributed coverage control of quadrotor multi-uav systems for precision agriculture, *IFAC-PapersOnLine* 52 (30) (2019) 251–256.
- [4] M.F. Aslan, A. Durdu, K. Sabancı, E. Ropelewski, S.S. Gültæk, A comprehensive survey of the recent studies with UAV for precision agriculture in open fields and greenhouses, *Appl. Sci.* 12 (3(1047)) (2022) 1–29.
- [5] E. Leslie, S. Flint, H.C. Briggs, M.L. Anderson, G. Appel, S. Konig, N. Bowen, C. Foster, P. Ganesh, H. Ramos, An unmanned system for persistent surveillance in GPS-denied environments, in: Proc. AIAA SCITECH 2022 Forum, No. 0118, San Diego, CA, USA, January 2022, pp. 3–7.

- [6] A.Y. Chen, Y.-N. Huang, J.-Y. Han, S.-C.J. Kang, A review of rotorcraft unmanned aerial vehicle (UAV) developments and applications in civil engineering, *Smart Struct. Syst.* 13 (6) (2014) 1065–1094.
- [7] Q.F. Dupont, D.K. Chua, A. Tashrif, E.L. Abbott, Potential applications of UAV along the construction's value chain, *Proc. Eng.* 182 (2017) 165–173.
- [8] N. Wang, C.K. Ahn, Coordinated trajectory-tracking control of a marine aerial-surface heterogeneous system, *IEEE/ASME Trans. Mechatron.* 26 (6) (2021) 3198–3210.
- [9] A. Kourani, N. Daher, Marine locomotion: a tethered UAV-buoy system with surge velocity control, *Robot. Auton. Syst.* 145 (103858) (2021) 1–17.
- [10] R. Amin, L. Ajun, S. Shamshirband, A review of quadrotor UAV: control methodologies and performance evaluation, *Int. J. Autom. Control* 10 (2) (2016) 87–103.
- [11] H. Lee, H.J. Kim, Trajectory tracking control of multirotors from modelling to experiments: a survey, *Int. J. Control Autom. Syst.* 15 (1) (2017) 281–292.
- [12] H. Shraim, A. Awada, R. Youness, A survey on quadrotors: configurations, modeling and identification, control, collision avoidance, fault diagnosis and tolerant control, *IEEE Aerosp. Electron. Syst. Mag.* 33 (7) (2018) 14–33.
- [13] H. Mo, G. Farid, Nonlinear and adaptive intelligent control techniques for quadrotor uav—a survey, *Asian J. Control* 21 (2) (2019) 989–1008.
- [14] J. Kim, S.A. Gadsden, S.A. Wilkerson, A comprehensive survey of control strategies for autonomous quadrotors, *Can. J. Electr. Comput. Eng.* 43 (1) (2019) 3–16.
- [15] T.P. Nascimento, M. Saska, Position and attitude control of multi-rotor aerial vehicles: a survey, *Annu. Rev. Control* 48 (2019) 129–146.
- [16] A. Tahir, J. Böling, M.-H. Haghbayan, H.T. Toivonen, J. Plosila, Swarms of unmanned aerial vehicles—a survey, *J. Ind. Inf. Integr.* 16 (100106) (2019).
- [17] H.T. Nguyen, T.V. Quyen, C.V. Nguyen, A.M. Le, H.T. Tran, M.T. Nguyen, Control algorithms for UAVs: a comprehensive survey, *EAI Endorsed Trans. Ind. Netw. Intell. Syst.* 7 (23) (2020).
- [18] T.E. Lechekhab, S. Manojlovic, M. Stankovic, R. Madonski, S. Simic, Robust error-based active disturbance rejection control of a quadrotor, *Aircr. Eng. Aerosp. Technol.* 93 (1) (2020) 89–104.
- [19] H. Nguyen, M. Kamel, K. Alexis, R. Siegwart, Model predictive control for micro aerial vehicles: a survey, in: *Proc. European Control Conference*, Delft, Netherlands, 2021, pp. 1556–1563.
- [20] J. Pliego-Jiménez, Quaternion-based adaptive control for trajectory tracking of quadrotor unmanned aerial vehicles, *Int. J. Adapt. Control Signal Process.* 35 (5) (2021) 628–641.
- [21] T. Madani, A. Benallegue, Backstepping control for a quadrotor helicopter, in: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Beijing, China, Oct. 9–15, 2006, pp. 3255–3260.
- [22] X. Shao, J. Liu, H. Cao, C. Shen, H. Wang, Robust dynamic surface trajectory tracking control for a quadrotor UAV via extended state observer, *Int. J. Robust Nonlinear Control* 28 (7) (2018) 2700–2719.
- [23] N. Wang, S.-F. Su, M. Han, W.-H. Chen, Backpropagating constraints-based trajectory tracking control of a quadrotor with constrained actuator dynamics and complex unknowns, *IEEE Trans. Syst. Man Cybern. Syst.* 49 (7) (2018) 1322–1337.
- [24] H. Liu, W. Zhao, S. Hong, F.L. Lewis, Y. Yu, Robust backstepping-based trajectory tracking control for quadrotors with time delays, *IET Control Theory Appl.* 13 (12) (2019) 1945–1954.
- [25] P.J. Antsaklis, Intelligent Control, *Encyclopedia of Electrical and Electronics Engineering*, vol. 10, 1997, pp. 493–503.
- [26] S.-s. Chen, M. Zhang, Adaptive (neural network) control in computer-integrated-manufacturing, in: *Proc. Applications of Artificial Intelligence VI*, Orlando, FL, USA, Apr. 4–8, vol. 937, 1988, pp. 470–473.
- [27] W. Li, J.-E. Slotine, Neural network control of unknown nonlinear systems, in: *Proc. American Control Conference*, Pittsburgh, PA, USA, June 21–23, 1989, pp. 1136–1141.
- [28] S.N. Kumpati, P. Kannan, Identification and control of dynamical systems using neural networks, *IEEE Trans. Neural Netw.* 1 (1) (1990) 4–27.
- [29] T. Chen, H. Chen, Approximations of continuous functionals by neural networks with application to dynamic systems, *IEEE Trans. Neural Netw.* 4 (6) (1993) 910–918.
- [30] M.T. Hagan, H.B. Demuth, Neural Networks for Control, *Proc. American Control Conference*, vol. 3, IEEE, 1999, pp. 1642–1656.
- [31] S.S. Ge, C.C. Hang, T.H. Lee, T. Zhang, *Stable Adaptive Neural Network Control*, The International Series on Asian Studies in Computer and Information Science, vol. 13, Springer New York, NY, USA, 2002.
- [32] P. Melin, O. Castillo, Adaptive intelligent control of aircraft systems with a hybrid approach combining neural networks, fuzzy logic and fractal theory, *Appl. Soft Comput.* 3 (4) (2003) 353–362.
- [33] J.S. Litt, D.L. Simon, S. Garg, T.-H. Guo, C. Mercer, R. Millar, A. Behbahani, A. Bajwa, D.T. Jensen, A survey of intelligent control and health management technologies for aircraft propulsion systems, *J. Aerosp. Comput. Inf. Commun.* 1 (12) (2004) 543–563.
- [34] N. Gandhi, A. Jha, J. Monaco, T. Seigler, D. Ward, D. Inman, Intelligent control of a morphing aircraft, in: *Proc. 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, Hawaii, USA, April 23–26, 2007, p. 1716.
- [35] M. Jafari, A.M. Shahri, S.B. Shouraki, Attitude control of a quadrotor using brain emotional learning based intelligent controller, in: *Proc. 13th Iranian Conference on Fuzzy Systems*, Aug, Qazvin, Iran, Aug. 27–29, IEEE, 2013, pp. 1–5.
- [36] M. Jafari, H. Xu, Intelligent control for unmanned aerial systems with system uncertainties and disturbances using artificial neural network, *Drones* 2 (3(30)) (2018).
- [37] Z. He, W. Gao, X. He, M. Wang, Y. Liu, Y. Song, Z. An, Fuzzy intelligent control method for improving flight attitude stability of plant protection quadrotor UAV, *Int. J. Agric. Biol. Eng.* 12 (6) (2019) 110–115.
- [38] M. Jafari, H. Xu, L.R. Garcia Carrillo, A neurobiologically-inspired intelligent trajectory tracking control for unmanned aircraft systems with uncertain system dynamics and disturbance, *Trans. Inst. Meas. Control* 41 (2) (2019) 417–432.
- [39] A.H. Ginting, S.Y. Doo, D.E. Pollo, H.J. Djahi, E.R. Mauboy, Attitude control of a quadrotor with fuzzy logic controller on SO (3), *J. Robot. Control* 3 (1) (2022) 101–106.
- [40] A.J. Abougarair, M.K. Aburakhis, M.M. Edardar, Adaptive neural networks based robust output feedback controllers for nonlinear systems, *Int. J. Robot. Control Syst.* 2 (1) (2022) 37–56.
- [41] Y. Yang, Y. Li, X. Liu, D. Huang, Adaptive neural network control for a hydraulic knee exoskeleton with valve deadband and output constraint based on nonlinear disturbance observer, *Neurocomputing* 473 (2022) 14–23.
- [42] Z. Qiu, Z. Wu, Adaptive neural network control for image-based visual servoing of robot manipulators, *IET Control Theory Appl.* 16 (4) (2022) 443–453.
- [43] A.M. Al Aela, J.-P. Kenne, H.A. Mintsa, Adaptive neural network and nonlinear electrohydraulic active suspension control system, *J. Vib. Control* 28 (3–4) (2022) 243–259.
- [44] J. Montoya-Cháirez, J. Moreno-Valenzuela, V. Santibáñez, R. Carelli, F.G. Rossomando, R. Pérez-Alcocer, Combined adaptive neural network and regressor-based trajectory tracking control of flexible joint robots, *IET Control Theory Appl.* 16 (1) (2022) 31–50.
- [45] F.-x. Qiao, J.-p. Shi, X.-b. Qu, Y.-x. Lyu, Hardware-in-loop adaptive neural control for a tiltable V-tail morphing aircraft, *Def. Technol.* (2022), available online at <https://doi.org/10.1016/j.dt.2021.12.012>.
- [46] C. Rosales, F. Rossomando, C. Soria, R. Carelli, Neural control of a quadrotor: a state-observer based approach, in: *Proc. Int. Conf. on Unmanned Aircraft Systems*, Dallas, TX, USA, 2018, pp. 647–653.
- [47] H. Razmi, S. Afshinifar, Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV, *Aerosp. Sci. Technol.* 91 (2019) 12–27.
- [48] O. Doukhi, D.J. Lee, Neural network-based robust adaptive certainty equivalent controller for quadrotor UAV with unknown disturbances, *Int. J. Control. Autom. Syst.* 17 (9) (2019) 2365–2374.
- [49] V.P. Tran, F. Santoso, M.A. Garrat, S.G. Anavatti, Neural network-based self-learning of an adaptive strictly negative imaginary tracking controller for a quadrotor transporting a cable-suspended payload with minimum swing, *IEEE Trans. Ind. Electron.* 68 (10) (2020) 258–10 268.
- [50] Ö. Bingöl, H. Güzey, Neuro sliding mode control of quadrotor UAVs carrying suspended payload, *Adv. Robot.* (2021) 1–12.
- [51] S.-E.-I. Hasseni, L. Abdou, Adaptive nonlinear robust control of an underactuated micro UAV, *Int. J. Dyn. Control* (2021) 1–23.
- [52] K. T. V. Vonásek, D. Fišer, J. Faigl, AR-Drone as a robotic platform for research and education, in: *Proc. International Conference on Research and Education in Robotics*, Prague, Czech Republic, June 15–17, 2011, pp. 1–15.
- [53] J. Engel, J. Sturm, D. Cremers, Camera-based navigation of a low-cost quadrocopter, in: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, October 7–12, 2012, pp. 2815–2821.
- [54] A. Hernandez, C. Copot, R. De Keyser, T. Vlas, I. Nasu, Identification and path following control of an AR.Drone quadrotor, in: *Proc. 17th International Conference on System Theory, Control and Computing*, Sinaia, Romania, October 11–13, 2013, pp. 583–588.
- [55] A. Chakrabarty, R. Morris, X. Bouyssounouse, R. Hunt, Autonomous indoor object tracking with the parrot AR.Drone, in: *Proc. International Conference on Unmanned Aircraft Systems*, IEEE, Arlington, VA, USA, June 7–10 2016, pp. 25–30.

- [56] A. Rohan, M. Rabah, S.-H. Kim, Convolutional neural network-based real-time object detection and tracking for parrot AR drone 2, *IEEE Access* 7 (2019) 69575–69584.
- [57] L.V. Santana, A.S. Brandão, M. Sarcinelli-Filho, An open-source testbed for outdoor navigation with the AR.Drone quadrotor, *IEEE Syst. J.* 15 (3) (2021) 3597–3608.
- [58] L.V. Santana, A.S. Brandao, M. Sarcinelli-Filho, R. Carelli, A trajectory tracking and 3D positioning controller for the AR.Drone quadrotor, in: Proc. International Conference on Unmanned Aircraft Systems, Orlando, FL, USA, May 27–30, 2014, pp. 756–767.
- [59] M.C.P. Santos, C.D. Rosales, M. Sarcinelli-Filho, R. Carelli, A novel null-space-based UAV trajectory tracking controller with collision avoidance, *IEEE/ASME Trans. Mechatron.* 22 (6) (2017) 2543–2553.
- [60] L.V. Santana, A.S. Brandão, M. Sarcinelli-Filho, On the design of outdoor leader-follower uav-formation controllers from a practical point of view, *IEEE Access* 9 (2021) 107493–107501.
- [61] I. Lopez-Sánchez, F. Rossomando, R. Pérez-Alcocer, C. Soria, R. Carelli, J. Moreno-Valenzuela, Adaptive trajectory tracking control for quadrotors with disturbances by using generalized regression neural networks, *Neurocomputing* 460 (2021) 243–255.
- [62] M.C. Santos, L.V. Santana, M.M. Martins, A.S. Brandão, M. Sarcinelli-Filho, Estimating and controlling UAV position using RGB-D/IMU data fusion with decentralized information/Kalman filter, in: Proc. IEEE International Conference on Industrial Technology, Seville, Spain, March 17–19, 2015, pp. 232–239.
- [63] M.C. Santos, M. Sarcinelli-Filho, R. Carelli, Trajectory tracking for UAV with saturation of velocities, in: Proc. International Conference on Unmanned Aircraft Systems, Arlington, VA, USA, June 7–10, 2016, pp. 643–648.
- [64] M.C. Santos, L.V. Santana, A.S. Brandão, M. Sarcinelli-Filho, R. Carelli, Indoor low-cost localization system for controlling aerial robots, *Control Eng. Pract.* 61 (2017) 93–111.
- [65] C. Rosales, C.M. Soria, F.G. Rossomando, Identification and adaptive PID control of a hexacopter UAV based on neural networks, *Int. J. Adapt. Control Signal Process.* 33 (1) (2019) 74–91.
- [66] R. Pérez-Alcocer, J. Moreno-Valenzuela, R. Miranda-Colorado, A robust approach for trajectory tracking control of a quadrotor with experimental validation, *ISA Trans.* 65 (2016) 262–274.
- [67] J. Moreno-Valenzuela, R. Pérez-Alcocer, M. Guerrero-Medina, A. Dzul, Nonlinear PID-type controller for quadrotor trajectory tracking, *IEEE/ASME Trans. Mechatron.* 23 (5) (2018) 2436–2447.
- [68] R. Pérez-Alcocer, J. Moreno-Valenzuela, Adaptive control for quadrotor trajectory tracking with accurate parametrization, *IEEE Access* 7 (2019) 53236–53247.
- [69] M.C.P. Santos, C.D. Rosales, J.A. Sarapura, M. Sarcinelli-Filho, R. Carelli, An adaptive dynamic controller for quadrotor to perform trajectory tracking tasks, *J. Intell. Robot. Syst.* 93 (1) (2019) 5–16.
- [70] S.S. Ge, T.H. Lee, C.J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*, World Scientific Series in Robotics and Intelligent Systems, vol. 19, World Scientific, Singapore, 1998.
- [71] F.L. Lewis, S. Jagannathan, A. Yesildirak, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis Inc., Philadelphia, PA, USA, 1999.
- [72] F.L. Lewis, J. Campos, R. Selmic, *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.
- [73] R. Kelly, V.S. Davila, J.A.L. Perez, *Control of Robot Manipulators in Joint Space*, 1st ed., Advanced Textbooks in Control and Signal Processing, Springer-Verlag London Limited, Leipzig, Germany, 2005.
- [74] A. Kourani, N. Daher, Leveraging PID gain selection towards adaptive backstepping control for a class of second-order systems, in: Proc. American Control Conference, New Orleans, LA, USA, May 25–28, 2021, pp. 1174–1179.
- [75] P. Yang, Z. Liu, Y. Wang, D. Li, Adaptive sliding mode fault-tolerant control for uncertain systems with time delay, *Int. J. Autom. Technol.* 14 (2) (2020) 337–345.
- [76] B. Wang, X. Yu, L. Mu, Y. Zhang, A dual adaptive fault-tolerant control for a quadrotor helicopter against actuator faults and model uncertainties without overestimation, *Aerospace Sci. Technol.* 99 (105744) (2020).
- [77] M. Dhaybi, N. Daher, Accurate real-time estimation of the inertia tensor of package delivery quadrotors, in: Proc. American Control Conference, Denver, CO, USA, July, 2020.
- [78] B. Wang, Y. Shen, Y. Zhang, Active fault-tolerant control for a quadrotor helicopter against actuator faults and model uncertainties, *Aerospace Sci. Technol.* 99 (105745) (2020).
- [79] I. Lopez-Sánchez, J. Montoya-Cháirez, R. Pérez-Alcocer, J. Moreno-Valenzuela, Experimental parameter identifications of a quadrotor by using an optimized trajectory, *IEEE Access* 8 (2020) 167355–167370.
- [80] S. Li, Y. Wang, J. Tan, Y. Zheng, Adaptive RBFNNs/integral sliding mode control for a quadrotor aircraft, *Neurocomputing* 216 (2016) 126–134.
- [81] M.H. Amoozgar, A. Chamseddine, Y. Zhang, Experimental test of a two-stage kalman filter for actuator fault detection and diagnosis of an unmanned quadrotor helicopter, *J. Intell. Robot. Syst.* 70 (1) (2013) 107–117.
- [82] B. Yu, Y. Zhang, I. Minchala, Y. Qu, Fault-tolerant control with linear quadratic and model predictive control techniques against actuator faults in a quadrotor UAV, in: Conference on Control and Fault-Tolerant Systems, Oct. 9–11, IEEE, Nice, France, 2013, pp. 661–666.
- [83] R.C. Avram, X. Zhang, J. Muse, Nonlinear adaptive fault-tolerant quadrotor altitude and attitude tracking with multiple actuator faults, *IEEE Trans. Control Syst. Technol.* 26 (2) (2017) 701–707.