Available online at www.sciencedirect.com

ScienceDirect

Journal of the Franklin Institute 360 (2023) 3770–3799

www.elsevier.com/locate/jfranklin

Trajectory tracking double two-loop adaptive neural network control for a Quadrotor☆

Ivan Lopez-Sanchez[a], Ricardo Pérez-Alcocer[b,*], Javier Moreno-Valenzuela[a]

[a] *Instituto Politécnico Nacional-CITEDI, Tijuana, Baja California 22435,México*
[b] *Universidad de Sonora-DIFUS, Hermosillo, Sonora 83000, México*

Received 26 January 2022; received in revised form 21 October 2022; accepted 18 January 2023
Available online 27 January 2023

Abstract

In this paper, the development and experimental validation of a novel double two-loop nonlinear controller based on adaptive neural networks for a quadrotor are presented. The proposed controller has a two-loop structure: an outer loop for position control and an inner loop for attitude control. Similarly, both position and orientation controllers also have a two-loop design with an adaptive neural network in each inner loop. The output weight matrices of the neural networks are updated online through adaptation laws obtained from a rigorous error convergence analysis. Thus, a training stage is unnecessary prior to the neural network implementation. Additionally, an integral action is included in the controller to cope with constant disturbances. The error convergence analysis guarantees the achievement of the trajectory tracking task and the boundedness of the output weight matrix estimation errors. The proposed scheme is designed such that an accurate knowledge of the quadrotor parameters is not needed. A comparison against the proposed controller and two other well-known schemes is presented. The obtained results showed the functionality of the proposed controller and demonstrated robustness to parametric uncertainty.

☆ 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, Proyect number 20230250, México.

* Corresponding author.
E-mail addresses: ilopez@citedi.mx (I. Lopez-Sanchez), ricardoramon.perez@unison.mx (R. Pérez-Alcocer), moreno@citedi.mx (J. Moreno-Valenzuela).

https://doi.org/10.1016/j.jfranklin.2023.01.029

1. Introduction

Nowadays, unmanned aerial vehicles (UAVs) are very popular for military and civilian applications. Among the variety of UAVs, those classified as rotary-wing demonstrated useful advantages over those classified as fixed-wing [1]. These advantages are the capabilities to perform vertical takeoffs and landings, hovering flight, and agile and acrobatic maneuvers in tight spaces. Quadrotors are currently the most popular within the rotary-wing classification of UAVs due to their simple structure and configuration. While helicopters have complex mechanical systems to handle the variable pitch main rotor and the tail rotor, the quadrotors have four fixed-pitch rotors only. In addition, the quadrotors can be smaller, and their maintenance is much easier compared with other aerial vehicles. Because of this, quadrotors became the focus of many research fields, including automatic control.

Different applications have been found for quadrotors recently, like the pipe inspection in industrial sites and refineries [2], the yield estimation and drought monitoring for a crop in a specific area [3], the velocity manipulation of a floating buoy by means of a cable [4], the soil moisture monitoring of large crop fields [5], monitoring and fault detection in overheated power lines [6], and many others for particular purposes can be found in the literature. In order to achieve such applications autonomously, flight controllers must be implemented. The controller is responsible for maintaining the stability of the quadrotor and ensuring the accomplishment of the assigned flight task through the computation of appropriate control input commands from takeoff until landing. Owing to this, different control schemes have been implemented to allow the quadrotors to fly autonomously and perform either, position control or trajectory tracking.

The two-loop control structure has been widely applied to control different systems. Such a structure is useful when controlling systems with coupled dynamics. Besides, separating the whole system dynamics into subsystems simplifies the development of the controller. This allows further analyze the overall system dynamics providing a better and more intuitive understanding of the interactions between each subsystem. In [7], a force-tracking impedance scheme based on a two-loop control structure was presented to control a robot manipulator. A two-loop structure controller for a robot manipulator, which is the result of using variable structure control and adaptive control, was developed in [8]. In [9], a two-loop control scheme was proposed to control the motion of a three degrees of freedom redundant planar arm. A double two-loop structure control scheme was designed in [10] to perform self-balancing and position control of a ball robot with an inverse mouse-ball drive actuated by two independent brushless servomotors. In [11], the formation control and trajectory tracking task of multiple mobile inverted pendulum robots was addressed by means of a two-loop control structure. The two-loop control structure was used in [12] to control a robot manipulator in operational space for warehouse automation applications. The two-loop structure of the position and attitude main loops that constitute the proposed scheme is inspired by the works [13–17] that address the motion control of robot manipulators in operational space. Those approaches consider a two-loop structure where the outer loop is a kinematic control that provides a desired joint velocity to the inner loop which is a velocity controller. The derivation of controllers with such a structure is based on controlling industrial robots, where the main objective is to make the end effector complete a specific task. Similarly, many commercial quadrotors have embedded controllers with input rate commands instead of position commands [18–22]. Hence, the design of an outer control loop is necessary in order to perform position control, whether for trajectory tracking or regulation tasks. Besides, given the quadrotor's underactuated nature,

a two-loop structure can be implemented in order to control the underactuated states of the system through the actuated ones and separate the controller structure into subsystems.

Works addressing the stabilization and control of quadrotors have been presented, some based on control schemes like proportional-integral-derivative (PID) [23,24], linear quadratic regulator [25], H_∞ control [26], sliding modes control (SMC) [27,28], feedback linearization [29], passivity-based control [30], adaptive control [31], quaternion-based control [32,33], backstepping [34,35], adaptive nonsingular fast terminal SMC [36], and even an approximation-free simple controller based scheme [37]. The functionality of the aforementioned schemes has been proved, and their respective advantages and limitations have been identified and discussed, concluding that some schemes are more suitable for certain tasks or work better than others under specific conditions. Nevertheless, with the rise of intelligent systems and the advances in artificial intelligence (AI), proposals based on such schemes are currently the focus of the control community; such control schemes based on AI are known as intelligent control. The benefits of AI in control mainly rely on its capabilities of estimating unmodeled or complex system dynamics and coping with unexpected phenomena, all present in the operation of quadrotors.

Among the different intelligent control strategies, the most used are fuzzy logic and neural networks (NNs), the latter being able to be trained or adaptive. In [38], a reinforcement learning agent was trained with the proximal policy optimization algorithm in order to perform the position control of a quadrotor in simulation. A fuzzy logic scheme based on PID control and iterative learning control was proposed in [39] to address the trajectory tracking problem of a quadrotor. In [40], a robust adaptive neural network-based control was presented to address the trajectory tracking problem of a quadrotor subject to bounded external disturbances. The stabilization problem of a quadrotor under external disturbances was addressed in [41] by employing an enhanced self-adaptive interval type-2 fuzzy control together with an SMC scheme. In [42], a trained NN-based reinforced learning control strategy was implemented to address the position control of a quadrotor under external disturbances. A neuro sliding mode control scheme was proposed in [43] to address the quadrotor suspended payload transport problem. An adaptive disturbance compensation controller based on generalized NNs was developed in [22] to perform trajectory tracking tasks for a quadrotor possessing a closed-architecture embedded controller. The design and implementation of an intelligent control scheme based on trained deep NNs were presented in [44] to control a quadrotor.

Fuzzy and NN-based control schemes have demonstrated good parameter uncertainties handling capabilities. Nevertheless, NNs have an advantage over fuzzy schemes in compensating for parametric uncertainties and unmodeled dynamics due to their function approximation property. In addition, online-learning NN-based schemes represent a better option for dealing with external disturbances than those who need training because the adverse flight conditions and unknown scenarios are difficult to model. This kind of online-learning NN is known as adaptive neural network. Owing to this, the introduced algorithm considers the use of adaptive neural networks (ANNs) to address the parameter uncertainty and unmodeled dynamics issue.

Besides, motivated by the discussion in [45] and the results of subsequent works on the literature [46–50], an integral action is considered in this work in order to handle the steady-state error and constant disturbances. In fact, the last may be due to the error between the real system parameters to the estimated ones when an accurate knowledge of them is not available, especially when the error is considerably significant. Given that the proposed control scheme was developed with the aim to operate with minimal apriori knowledge of the system

parameters, an adaptive neural network used together with an integral action is an excellent option to achieve this goal.

Among the different problems in the quadrotor control field, stabilization, position regulation, and trajectory tracking are the most representative. Specifically, the problem addressed in this document is the trajectory tracking task, which is the most challenging of the aforementioned since the reference is a time-varying function, and the system dynamics and aerodynamic forces have a more significant impact on the vehicle. Its impact on the system's behavior can be easily mitigated by compensating them. Nevertheless, accurate knowledge of the system parameters is required to achieve it successfully. The parameters of the system can be obtained via system identification processes. However, performing such a task is challenging for systems like the quadrotors. Thus, to surpass this issue, an intelligent control strategy based on ANNs is proposed; the proposed controller does not need accurate knowledge of the system parameters. The universal function approximation property of NNs is exploited to estimate and compensate for some terms of the system dynamics without the need for previous knowledge of the system parameters. The overall two-loop structure of the controller allows employing a NN on each loop to stabilize and control the position and attitude dynamics. Besides, the proposed algorithm is an adaptive one; therefore, the NNs do not require training previously for their implementation, which is one of the most time-consuming when using the non-adaptive version of such schemes. Additionally, the proposed controller's two-loop structure helps achieve the control objective since the trajectory tracking problem also implies tracking a variable velocity reference. Therefore, this problem can be addressed more comprehensively by using two loops, one for position control and the other for velocity control. Owing to this, the proposed controller represents an improvement over the other similar works on the trajectory tracking problem of quadrotors. Moreover, this work contributes by providing not only theoretical proof of the proposed controller's functionality as it was done in recent works [40,50–56] but an experimental validation of the implementation of the proposed scheme and a discussion of its performance in contrast with other schemes.

The proposed double two-loop controller is formed by the position and attitude controllers. The position controller handles the position dynamics and generates attitude commands that feed the attitude controller, which handles the attitude dynamics. The position and attitude controllers are two-loop schemes, i.e., each controller has a two-loop structure where the outer loops generate velocity commands to feed the inner loops, being these last ANNs-based schemes that are used to avoid the effects of the parametric uncertainty and unmodeled dynamics. Since the position and attitude controllers form a two-loop scheme and each has a two-loop structure with ANNs, the overall structure results in a double two-loop ANN controller. The position controller computes the total thrust and desired roll and pitch angles. The attitude controller is fed with the desired roll and pitch angles coming from the position controller and computes proper torques. The derivation of the entire control scheme is carried out from the inertial reference frame. The functionality of the proposed scheme is validated through experiments considering a mass different from the nominal one and that the parameters of the system are unknown. In addition, an experimental comparison against an adaptive neural controller and a model-based controller with a similar structure to the proposed scheme is presented. Many works addressing the control of quadrotors by means of neural networks have been presented over the last two decades [22,40,44,50,51,55–66]. Among them, those implementing adaptive neural networks were presented recently [22,40,50,51,55,56,64–66]. Within these works, different activation functions were employed, e.g., in [22,40,51,55,56] radial basis functions were used, in [50,65] sigmoid functions, and in [64,66] hyperbolic tangent

functions. As the literature review indicates, sigmoid functions and hyperbolic tangent functions are not widely reported as activation functions. The proposed controller employs the hyperbolic tangent function since requires fewer parameters to be defined. To the best of the authors' knowledge, the well-known two-loop control structure used, e.g., in motion control of manipulators [67], impedance control [68], and power converters [69], has not been applied simultaneously in the position and attitude subsystems to address the trajectory tracking task for quadrotors by using ANNs. The contributions of this work rely on the following points:

- The inertia tensor is assumed as a symmetric matrix instead of a diagonal matrix; also, the wind-induced drag and external disturbances are considered, aspects that are not taken into account in many other works.
- For the first time, and inspired by the mentioned works, a novel two-loop adaptive neural network-based control structure is used to guarantee position trajectory tracking, and another one is applied to ensure attitude trajectory tracking resulting in a double two-loop adaptive neural network-based controller.
- Accurate knowledge of the quadrotor parameters is not necessary for implementing the proposed scheme.
- The neural networks used in the proposed scheme are online-learning based. Owing to this, a training stage is not necessary for their operation.
- A rigorous analysis of the error's convergence is presented from which the adaptation laws were obtained, guaranteeing convergence to zero of the tracking errors and the boundedness of the neural network output weights estimates.
- The functionality of the proposed scheme is demonstrated experimentally, and its performance is compared with another adaptive neural-based scheme and a model-based scheme, obtaining the best results with the proposed one.

The document is organized as follows: Section 2 describes the quadrotor dynamic model and the control goal. The double two-loop adaptive neural network-based controller design, the error convergence analysis, and the derivation of the adaptation laws are presented in Section 3. Section 4 presents the experimental results. Finally, the conclusions of the work are given in Section 5.

Notation: The notation used in this document is given next. $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|, \forall\ \boldsymbol{x} \in \mathbb{R}^n$ is the l_1 norm. $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}, \forall\ \boldsymbol{x} \in \mathbb{R}^n$ is the Euclidean norm. $\lambda_{\min}\{A\}$ and $\lambda_{\max}\{A\}$ denote the minimum and maximum eigenvalue of matrix $A \in \mathbb{R}^{n\times n}$, respectively. $\mathrm{Tr}\{A\} = \sum_{i=1}^{n} a_{ii}$ denotes the trace operator for matrix $A \in \mathbb{R}^{n\times n}$. Given matrix $A = [a_{ij}]$, the Frobenius norm is defined as $\|A\|_F = \sqrt{\mathrm{Tr}\{A^\top A\}}$. The Frobenius inner product is defined as $\langle A, B\rangle_F = \mathrm{Tr}\{A^\top B\}$ for any A and B matrices of compatible dimensions and according to the Schwarz inequality $|\langle A, B\rangle_F| \leq \|A\|_F \|B\|_F$.

2. Quadrotor dynamic model and control goal

2.1. Quadrotor dynamic model

The quadrotor dynamic model in the inertial reference frame is given by [31,70–73]

$$m\ddot{\boldsymbol{p}} + m\boldsymbol{g}_z + D_p(\boldsymbol{\eta})\dot{\boldsymbol{p}} = \boldsymbol{r}_3(\boldsymbol{\eta})F + \boldsymbol{\delta}_p(t), \tag{1}$$

$$M(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + D_{\eta}(\boldsymbol{\eta})\dot{\boldsymbol{\eta}} = \Phi(\boldsymbol{\eta})^{-\mathsf{T}}\boldsymbol{\tau} + \boldsymbol{\delta}_{\eta}(t), \tag{2}$$

where (1) and (2) describe the position and attitude dynamics, respectively, $m \in \mathbb{R}$ is the quadrotor mass, $\boldsymbol{g}_z = [0 \;\; 0 \;\; g]^{\mathsf{T}} \in \mathbb{R}^3$, being $g \in \mathbb{R}$ the acceleration due to gravity, $\boldsymbol{p} = [x \;\; y \;\; z]^{\mathsf{T}} \in \mathbb{R}^3$ and $\boldsymbol{\eta} = [\phi \;\; \theta \;\; \psi]^{\mathsf{T}} \in \mathbb{R}^3$ are the quadrotor position and attitude represented in the inertial reference frame, respectively, $D_p(\boldsymbol{\eta}) \in \mathbb{R}^{3\times3}$ is the aerodynamic drag matrix, $D_{\eta}(\boldsymbol{\eta}) \in \mathbb{R}^{3\times3}$ is a positive definite matrix representing an aerodynamic damping effect, $\boldsymbol{r}_3(\boldsymbol{\eta}) \in \mathbb{R}^3$ is the third column of $R(\boldsymbol{\eta}) \in SO(3)$, which is a rotation matrix defined as

$$R(\boldsymbol{\eta}) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix},$$

with $s_x = \sin(x)$, $c_x = \cos(x)$ and $t_x = \tan(x)$, $\Phi(\boldsymbol{\eta}) \in \mathbb{R}^{3\times3}$ is a transformation matrix that relates the angular velocities in the body reference frame with the change rate of the Euler angles $\dot{\boldsymbol{\eta}}$ and is defined as

$$\Phi(\boldsymbol{\eta}) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix},$$

being its inverse given by

$$\Phi(\boldsymbol{\eta})^{-1} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix},$$

$M(\boldsymbol{\eta}) \in \mathbb{R}^{3\times3}$ is the inertia matrix and $C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) \in \mathbb{R}^{3\times3}$ is the Coriolis matrix, which are explicitly defined as

$$M(\boldsymbol{\eta}) = \Phi(\boldsymbol{\eta})^{-T} I \Phi(\boldsymbol{\eta})^{-1},$$
$$C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) = \Phi(\boldsymbol{\eta})^{-T}\left[\mathcal{S}\left(I\Phi(\boldsymbol{\eta})^{-1}\dot{\boldsymbol{\eta}}\right) - I\Phi(\boldsymbol{\eta})^{-1}\dot{\Phi}(\boldsymbol{\eta})\right]\Phi(\boldsymbol{\eta})^{-1},$$

where

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \in \mathbb{R}^{3\times3}$$

is the inertia tensor considered as a symmetric matrix, $\mathcal{S}(\boldsymbol{a}) \in \mathbb{R}^{3\times3} \;\; \forall \;\; \boldsymbol{a} \in \mathbb{R}^3$ is a skew symmetric matrix defined as

$$\mathcal{S}(\boldsymbol{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix},$$

the time derivative of $\Phi(\boldsymbol{\eta})$ is given by

$$\dot{\Phi}(\boldsymbol{\eta}) = \begin{bmatrix} 0 & c_\phi t_\theta \dot{\phi} + s_\phi \dot{\theta}/c_\theta^2 & c_\phi \dot{\theta}/c_\theta^2 - s_\phi t_\theta \dot{\phi} \\ 0 & -s_\phi \dot{\phi} & -c_\phi \dot{\phi} \\ 0 & (c_\phi c_\theta \dot{\phi} + s_\phi s_\theta \dot{\theta})/c_\theta^2 & (c_\phi s_\theta \dot{\theta} - s_\phi c_\theta \dot{\phi})/c_\theta^2 \end{bmatrix},$$

$F \in \mathbb{R}$ and $\boldsymbol{\tau} = [\tau_\phi \ \tau_\theta \ \tau_\psi]^\mathsf{T} \in \mathbb{R}^3$ are the total thrust and the torque vector, respectively, which represent the four control inputs of the quadrotor, being a six degrees of freedom underactuated system [74]. Notice that $\Phi(\boldsymbol{\eta})$ is singular and $\dot{\Phi}(\boldsymbol{\eta})$ is not well defined at $\theta = \pm\pi/2$, thus, no acrobatic maneuvers are considered in this work, i.e., $|\phi|, |\theta| \leq \pi/2$. The vectors $\boldsymbol{\delta}_p(t)$, $\boldsymbol{\delta}_\eta(t) \in \mathbb{R}^3$ represent the unknown external disturbances and unmodeled dynamics, assumed to be bounded as follows

$$\|\boldsymbol{\delta}_p(t)\|, \ \ \|\boldsymbol{\delta}_\eta(t)\| \leq \delta \ \forall \ t \geq 0.$$

As mentioned previously, the quadrotor has four control inputs related to the four actuators of the quadrotor. The actuator comprises a rotor and a propeller attached to it that produces the thrust and drag forces resulting from its rotation. Thus, the total thrust and the torque vector are related to the thrust of each actuator by the following expression

$$F = \sum_{i=1}^{4} T_i,$$

$$\boldsymbol{\tau} = \begin{bmatrix} l(T_2 - T_4) \\ l(T_3 - T_1) \\ k_\tau \sum_{i=1}^{4} (-1)^{i+1} T_i \end{bmatrix},$$

where $\boldsymbol{T} = [T_1 \ T_2 \ T_3 \ T_4]^\mathsf{T} \in \mathbb{R}^4$ and $T_i \in \mathbb{R}$ is the thrust of the i-th actuator, $l \in \mathbb{R}$ is the distance from the quadrotor center of mass to the actuator rotation axis, and k_τ is a coefficient related to the geometric characteristics of the propeller. The above relationship can be expressed in a matrix form as follows

$$\begin{bmatrix} F \\ \boldsymbol{\tau} \end{bmatrix} = B\boldsymbol{T},$$

being

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & l & 0 & -l \\ -l & 0 & l & 0 \\ k_\tau & -k_\tau & k_\tau & -k_\tau \end{bmatrix}.$$

Further details on the quadrotor dynamic model and its representation in the inertial and body reference frames can be consulted in [31,70–73]. In Fig. 1, the inertial reference frame (solid arrows) and the body reference frame (dashed arrows) are depicted together with the position of the four actuators. In agreement with [75–79], different aerodynamic and physical phenomena can be considered in the quadrotor dynamic model. Besides, these aerodynamic and physical phenomena are significant in acrobatic maneuvering, high-speed flights, and during their operation in outdoor environments. Thus, neglecting them under such operating conditions will produce an inaccurate model representation [79]. Aiming to provide a better representation of the quadrotor dynamics, the wind-induced drag force and the aerodynamic drag torque are considered in the left-hand side of Eqs. (1) and (2), which are represented by the terms $D_p(\boldsymbol{\eta})\dot{\boldsymbol{p}}$ and $D_\eta(\boldsymbol{\eta})\dot{\boldsymbol{\eta}}$, respectively. Also, part of the unmodeled dynamics is contained in vectors $\boldsymbol{\delta}_p(t)$ and $\boldsymbol{\delta}_\eta(t)$. Note that these terms of the quadrotor model are considered in the controller design to provide robustness to our proposed scheme.

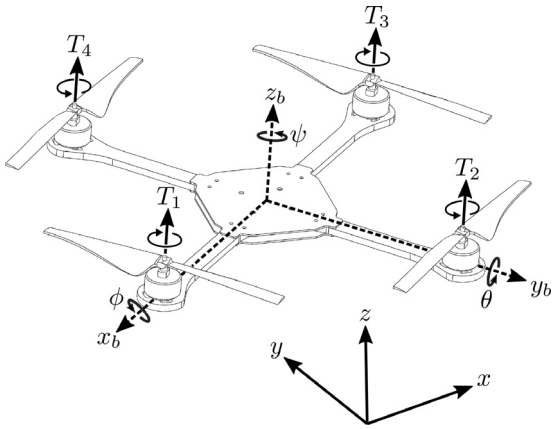

Fig. 1. Graphical representation of a quadrotor and its body reference frame (dashed lines), the inertial reference frame (solid lines), the thrust T_i provided by each actuator, and the propellers' direction of rotation.

2.2. Control goal

The objective is to guarantee the trajectory tracking task in the three-dimensional space of a quadrotor described by the dynamics given in (1) and (2). As mentioned before, the quadrotor is an underactuated system with six degrees of freedom and four control inputs. The altitude and the three Euler angles are fully actuated, while, the position in the horizontal plane is coupled with the total thrust and the attitude, precisely, with the roll and pitch angles. As a result, the tracking of a desired trajectory in the three-dimensional space is achieved via suitable thrust and desired roll and pitch angle commands. Defining the desired position $\boldsymbol{p}_d(t) = [x_d(t)\ \ y_d(t)\ \ z_d(t)]^{\mathsf{T}}$ and the desired attitude $\boldsymbol{\eta}_d(t) = [\phi_d(t)\ \ \theta_d(t)\ \ \psi_d(t)]^{\mathsf{T}}$ where $\boldsymbol{p}_d(t)$ and $\psi_d(t)$ are provided by a navigation system or by the user and $\phi_d(t)$ and $\theta_d(t)$ are computed by the position controller. The desired position $\boldsymbol{p}_d(t)$ and attitude $\boldsymbol{\eta}_d(t)$ are assumed to be at least twice time-differentiable and bounded until at least its second time-derivative. The position and attitude error vectors are defined as

$$\boldsymbol{e}_p(t) = \boldsymbol{p}_d(t) - \boldsymbol{p}(t), \tag{3}$$

$$\boldsymbol{e}_\eta(t) = \boldsymbol{\eta}_d(t) - \boldsymbol{\eta}(t). \tag{4}$$

Then, the control goal is to design control inputs $F(t)$ and $\boldsymbol{\tau}(t)$ and attitude references $\phi_d(t)$ and $\theta_d(t)$ such that

$$\lim_{t\to\infty} \begin{bmatrix} \boldsymbol{e}_p(t) \\ \dot{\boldsymbol{e}}_p(t) \\ \boldsymbol{e}_\eta(t) \\ \dot{\boldsymbol{e}}_\eta(t) \end{bmatrix} = \boldsymbol{0} \tag{5}$$

is guaranteed.

3. Double two-loop adaptive neural network-based controller

This section contains the development of the proposed double two-loop control scheme based on adaptive neural networks introduced to deal with parametric uncertainties, unmodeled dynamics, and external disturbances. The design process of the position and attitude controllers and the corresponding error convergence analysis are presented.

Controller design

The proposed double two-loop controller is formed by the position and attitude controllers, where the position controller handles the position dynamics, and the attitude controller handles the attitude dynamics. The position controller computes the total thrust $F(t)$ and desired roll $\phi_d(t)$ and pitch $\theta_d(t)$ angles that feed the attitude controller to get the torques $\boldsymbol{\tau}(t)$. The position and attitude controllers have each a two-loop structure and includes an adaptive neural network to compensate for parametric uncertainty and unmodeled dynamics. The position controller is comprised of two loops: the position outer loop and the ANN-based linear velocity inner loop. The attitude controller is comprised of two loops: the attitude outer loop and the ANN-based Euler angles rate inner loop. In both cases, the outer loops generate commands to feed the inner loops where the ANNs are housed. Since the proposed controller overall scheme is comprised of a two-loop structure, and each of these loops also has a two-loop structure is said that the proposed controller is a double two-loop controller. With this in mind and given that the position and attitude controllers house an ANN-based scheme each, the name adopted for the proposed algorithm is double two-loop adaptive neural network-based controller. A diagram of the proposed double two-loop adaptive neural network-based controller is given in Fig. 2.

3.1. Position controller

The position controller has a two-loop structure where the outer loop is a position control and the inner loop is an adaptive neural network-based velocity control.

3.1.1. Position outer loop

The position outer loop defines a velocity reference signal $\boldsymbol{U}_p \in \mathbb{R}^3$ given by

$$\boldsymbol{U}_p = \dot{\boldsymbol{p}}_d + K_p \boldsymbol{e}_p, \tag{6}$$

where $K_p \in \mathbb{R}^{3\times 3}$ is a positive definite diagonal gain matrix. Considering the position error in (3), taking its time derivative and substituting the velocity reference signal in (6) the following is obtained

$$\dot{\boldsymbol{e}}_p = -K_p \boldsymbol{e}_p + \boldsymbol{U}_p - \dot{\boldsymbol{p}}, \tag{7}$$

defining the velocity error

$$\tilde{\boldsymbol{U}}_p = \boldsymbol{U}_p - \dot{\boldsymbol{p}}, \tag{8}$$

the position error dynamics (7) results in

$$\dot{\boldsymbol{e}}_p = -K_p \boldsymbol{e}_p + \tilde{\boldsymbol{U}}_p, \tag{9}$$

where it can be observed that as long as $\tilde{\boldsymbol{U}}_p \to 0$ is guaranteed, then, $\boldsymbol{e}_p \to 0$ while $t \to \infty$.

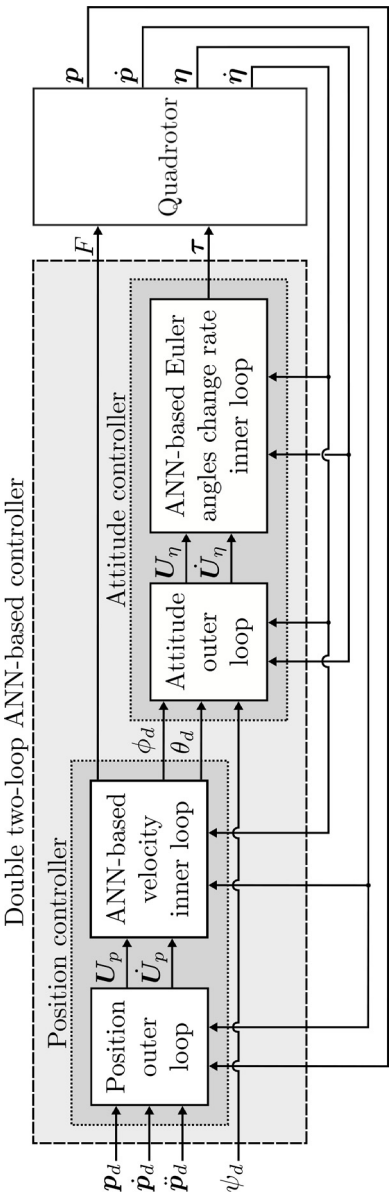

Fig. 2. Diagram of the proposed double two-loop adaptive neural network-based controller (dash-lined rectangle) formed by the position and attitude controller (dot-lined rectangles). The position controller is comprised of the position outer loop and the ANN-based linear velocity inner loop (solid-lined rectangles); meanwhile, the attitude controller is comprised of the attitude outer loop and the ANN-based Euler angles change rate inner loop (solid-lined rectangles).

3.1.2. Adaptive neural network-based velocity inner loop

Taking the time derivative of the velocity error (8), substituting the position dynamics in (1) on it, and multiplying both sides of the equation by m, the velocity error dynamics is obtained

$$m\dot{\tilde{\boldsymbol{U}}}_p = m\dot{\boldsymbol{U}}_p + m\boldsymbol{g}_z + D_p(\boldsymbol{\eta})\dot{\boldsymbol{p}} - \boldsymbol{\delta}_p(t) - \boldsymbol{r}_3(\boldsymbol{\eta})F. \tag{10}$$

Defining the mass estimation error as $\tilde{m} = m - \hat{m}$, with $\hat{m}$ denoting the mass estimation, one gets

$$m\dot{\tilde{\boldsymbol{U}}}_p = \hat{m}(\dot{\boldsymbol{U}}_p + \boldsymbol{g}_z) + \tilde{m}(\dot{\boldsymbol{U}}_p + \boldsymbol{g}_z) + D_p(\boldsymbol{\eta})\dot{\boldsymbol{p}} - \boldsymbol{\delta}_p(t) - \boldsymbol{r}_3(\boldsymbol{\eta})F,$$

based in the attitude error definition in (4) the term $\boldsymbol{r}_3(\boldsymbol{\eta}_d)F$ is added and subtracted to yield

$$m\dot{\tilde{\boldsymbol{U}}}_p = \hat{m}(\dot{\boldsymbol{U}}_p + \boldsymbol{g}_z) + \tilde{m}(\dot{\boldsymbol{U}}_p + \boldsymbol{g}_z) + D_p(\boldsymbol{\eta})\dot{\boldsymbol{p}} - \boldsymbol{\delta}_p(t) + [\boldsymbol{r}_3(\boldsymbol{e}_\eta + \boldsymbol{\eta}) - \boldsymbol{r}_3(\boldsymbol{\eta})]F - \boldsymbol{r}_3(\boldsymbol{\eta}_d)F.$$

Assuming that the attitude dynamics are faster than the position dynamics [80–83], the attitude error converges to zero first than the position error, and the term $[\boldsymbol{r}_3(\boldsymbol{e}_\eta + \boldsymbol{\eta}) - \boldsymbol{r}_3(\boldsymbol{\eta})]F = 0$, leading to

$$m\dot{\tilde{\boldsymbol{U}}}_p = \hat{m}(\dot{\boldsymbol{U}}_p + \boldsymbol{g}_z) + \tilde{m}(\dot{\boldsymbol{U}}_p + \boldsymbol{g}_z) + D_p(\boldsymbol{\eta})\dot{\boldsymbol{p}} - \boldsymbol{\delta}_p(t) - \boldsymbol{r}_3(\boldsymbol{\eta}_d)F,$$

which, considering the time derivative of (6), can be expressed as

$$m\dot{\tilde{\boldsymbol{U}}}_p = \hat{m}(\ddot{\boldsymbol{p}}_d + \boldsymbol{g}_z) + \hat{m}K_p\dot{\boldsymbol{e}}_p + \tilde{m}(\dot{\boldsymbol{U}}_p + \boldsymbol{g}_z) + D_p(\boldsymbol{\eta})\dot{\boldsymbol{p}} - \boldsymbol{\delta}_p(t) - \boldsymbol{r}_3(\boldsymbol{\eta}_d)F,$$

that can be rewritten as

$$m\dot{\tilde{\boldsymbol{U}}}_p = \hat{m}(\ddot{\boldsymbol{p}}_d + \boldsymbol{g}_z) + \boldsymbol{f}_{W_p}(\dot{\boldsymbol{p}},\ \boldsymbol{\eta},\ \dot{\boldsymbol{U}}_p) - \boldsymbol{r}_3(\boldsymbol{\eta}_d)F, \tag{11}$$

with $\boldsymbol{f}_{W_p}(\dot{\boldsymbol{p}},\ \boldsymbol{\eta},\ \dot{\boldsymbol{U}}_p) = \hat{m}K_p\dot{\boldsymbol{e}}_p + \tilde{m}(\dot{\boldsymbol{U}}_p + \boldsymbol{g}_z) + D_p(\boldsymbol{\eta})\dot{\boldsymbol{p}}$. Exploiting the continuous function approximation property of neural networks [84–86], a neural network can be used to estimate $\boldsymbol{f}_{W_p}(\dot{\boldsymbol{p}},\ \boldsymbol{\eta},\ \dot{\boldsymbol{U}}_p)$ as follows

$$\boldsymbol{f}_{W_p}(\boldsymbol{\gamma}_p) = W_p^{\mathsf{T}}\sigma(V_p^{\mathsf{T}}\boldsymbol{\gamma}_p) + \boldsymbol{\epsilon}_p,$$

where $\boldsymbol{\gamma}_p = [1\ \dot{\boldsymbol{p}}^{\mathsf{T}}\ \boldsymbol{\eta}^{\mathsf{T}}\ \dot{\boldsymbol{U}}_p^{\mathsf{T}}]^{\mathsf{T}} \in \mathbb{R}^{10}$ is the neural network extended input vector, $V_p \in \mathbb{R}^{10\times L_p}$ and $W_p \in \mathbb{R}^{L_p\times 3}$ are the ideal weights input and output matrices, respectively, that are bounded as $\|V_p\|_F \leq V_P$ and $\|W_p\|_F \leq W_P$, respectively, $\boldsymbol{\epsilon}_p \in \mathbb{R}^3$ is the neural network approximation error, bounded as $0 < \|\boldsymbol{\epsilon}_p\| \leq \epsilon_P$, $L_p \in \mathbb{N}$ is the neurons quantity, and $\sigma(\boldsymbol{x}), \forall\ \boldsymbol{x} \in \mathbb{R}^n$ represents the neural network activation function. Considering $\boldsymbol{v}_{\gamma_p} = V_p^{\mathsf{T}}\boldsymbol{\gamma}_p \in \mathbb{R}^{L_p}$, then, $\sigma(\boldsymbol{v}_{\gamma_p}) = [\tanh(v_{\gamma_p 1})\cdots\tanh(v_{\gamma_p L_p})]^{\mathsf{T}} \in \mathbb{R}^{L_p}$ which henceforth be denoted as $\boldsymbol{\sigma}_p \in \mathbb{R}^{L_p}$. Therefore, Eq. (11) can be written as

$$m\dot{\tilde{\boldsymbol{U}}}_p = \hat{m}(\ddot{\boldsymbol{p}}_d + \boldsymbol{g}_z) + W_p^{\mathsf{T}}\boldsymbol{\sigma}_p + \boldsymbol{\varepsilon}_p - \boldsymbol{r}_3(\boldsymbol{\eta}_d)F, \tag{12}$$

where $\boldsymbol{\varepsilon}_p = \boldsymbol{\epsilon}_p - \boldsymbol{\delta}_p(t)$ is bounded as $0 < \|\boldsymbol{\varepsilon}_p\| \leq \varepsilon_P$. Thus, the following definition for $\boldsymbol{r}_3(\boldsymbol{\eta}_d)F = [f_x\ f_y\ f_z]^{\mathsf{T}}$ is proposed

$$\boldsymbol{r}_3(\boldsymbol{\eta}_d)F = K_{U_p}\tilde{\boldsymbol{U}}_p + K_{ip}\boldsymbol{\xi}_p + \hat{m}(\ddot{\boldsymbol{p}}_d + \boldsymbol{g}_z) + \hat{W}_p^{\mathsf{T}}\boldsymbol{\sigma}_p + \alpha_p\mathrm{sign}(\tilde{\boldsymbol{U}}_p), \tag{13}$$

$$\dot{\boldsymbol{\xi}}_p = \tilde{\boldsymbol{U}}_p, \tag{14}$$

with the output weights adaptation law

$$\dot{\hat{W}}_p = -N_p\Big(\boldsymbol{\sigma}_p\tilde{\boldsymbol{U}}_p^{\mathsf{T}} - \kappa_p\|\tilde{\boldsymbol{U}}_p\|\hat{W}_p\Big), \tag{15}$$

where the output weight matrix estimation error was defied as

$$\tilde{W}_p = W_p - \hat{W}_p, \tag{16}$$

and $\text{sign}(\tilde{\boldsymbol{U}}_p) = [\text{sign}(\tilde{U}_{p1})\ \text{sign}(\tilde{U}_{p2})\ \text{sign}(\tilde{U}_{p3})]^{\mathsf{T}} \in \mathbb{R}^3$, being

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

$\hat{W}_p$ is an estimation of the output weight matrix W_p, α_p, $\kappa_p \in \mathbb{R}$ are strictly positive constants, K_{U_p}, $K_{ip} \in \mathbb{R}^{3\times3}$ are positive definite diagonal gain matrices, $N_p \in \mathbb{R}^{3\times3}$ is a positive definite diagonal matrix with the neural network adaptation gains. It can be observed that (13), is a nonlinear algebraic equation system, and can be solved in order to obtain $F(t)$, $\phi_d(t)$ and $\theta_d(t)$ being $\psi_d(t)$ an arbitrary value. Thus, by using (13) the total thrust and the desired roll and pitch angles can be obtained and the control inputs are given by

$$F(t) = \frac{f_z}{\cos(\phi)\cos(\theta)}, \tag{17}$$

$$\phi_d(t) = \tan^{-1}\left[\frac{\cos(\theta_d)}{f_z}(f_x\sin(\psi_d) - f_y\cos(\psi_d))\right], \tag{18}$$

$$\theta_d(t) = \tan^{-1}\left[\frac{1}{f_z}(f_x\cos(\psi_d) + f_y\sin(\psi_d))\right]. \tag{19}$$

The velocity error dynamics (12) with the controller (13)-(19) results in the following closed-loop system

$$m\dot{\tilde{\boldsymbol{U}}}_p = -K_{U_p}\tilde{\boldsymbol{U}}_p - K_{ip}\boldsymbol{\xi}_p + \tilde{W}_p^{\mathsf{T}}\boldsymbol{\sigma}_p + \boldsymbol{\varepsilon}_p - \alpha_p\text{sign}(\tilde{\boldsymbol{U}}_p). \tag{20}$$

Proposition 1. *Assume gain matrices K_p, K_{U_p} and N_p to be positive definite diagonal matrices and positive constants α_p and κ_p satisfying the condition*

$$\alpha_p \geq \frac{\kappa_p W_P^2}{4} + \varepsilon_P. \tag{21}$$

Then, for all initial conditions starting as some compact set, the solutions $\boldsymbol{e}_p(t)$, $\dot{\boldsymbol{e}}_p(t)$ and $\tilde{\boldsymbol{U}}_p(t)$ of the overall closed-loop system in (9), (14), (15), and (20) converge to zero as time goes to infinity, i.e., $\boldsymbol{e}_p(t)$, $\dot{\boldsymbol{e}}_p(t)$, $\tilde{\boldsymbol{U}}_p(t) \to 0$ while $t \to \infty$. Furthermore, the state variable $\boldsymbol{\xi}_p(t)$ and the output weight matrix estimation error $\tilde{W}_p(t)$ remains bounded for all time $t \geq 0$.

Proof. The following non-negative function is defined

$$V_p = \frac{\beta_p}{2}\boldsymbol{e}_p^{\mathsf{T}}\boldsymbol{e}_p + \frac{m}{2}\tilde{\boldsymbol{U}}_p^{\mathsf{T}}\tilde{\boldsymbol{U}}_p + \frac{1}{2}\boldsymbol{\xi}_p^{\mathsf{T}}K_{ip}\boldsymbol{\xi}_p + \frac{1}{2}\text{Tr}\Big\{\tilde{W}_p^{\mathsf{T}}N_p^{-1}\tilde{W}_p\Big\}.$$

The time derivative along the trajectories of the closed-loop system is given by

$$\dot{V}_p = -\beta_p \boldsymbol{e}_p^{\mathsf{T}} K_p \boldsymbol{e}_p + \beta_p \boldsymbol{e}_p^{\mathsf{T}} \tilde{\boldsymbol{U}}_p - \tilde{\boldsymbol{U}}_p^{\mathsf{T}} K_{U_p} \tilde{\boldsymbol{U}}_p - \alpha_p \tilde{\boldsymbol{U}}_p^{\mathsf{T}} \operatorname{sign}(\tilde{\boldsymbol{U}}_p) + \tilde{\boldsymbol{U}}_p^{\mathsf{T}} \tilde{W}_p^{\mathsf{T}} \boldsymbol{\sigma}_p + \tilde{\boldsymbol{U}}_p^{\mathsf{T}} \boldsymbol{\varepsilon}_p + \operatorname{Tr}\left\{ \tilde{W}_p^{\mathsf{T}} N_p^{-1} \dot{\tilde{W}}_p \right\}.$$

Using that by the definition of the output weight matrix estimation error in (16) $\dot{\tilde{W}}_p = -\dot{\hat{W}}_p$, substituing the adaptation law (15) and using the facts $\tilde{\boldsymbol{U}}_p^{\mathsf{T}} \tilde{W}_p^{\mathsf{T}} \boldsymbol{\sigma}_p = \operatorname{Tr}\left\{ \tilde{W}_p^{\mathsf{T}} \boldsymbol{\sigma}_p \tilde{\boldsymbol{U}}_p^{\mathsf{T}} \right\}$ and $\alpha_p \tilde{\boldsymbol{U}}_p^{\mathsf{T}} \operatorname{sign}(\tilde{\boldsymbol{U}}_p) = \alpha_p \|\tilde{\boldsymbol{U}}_p\|_1$ is obtained

$$\dot{V}_p = -\beta_p \boldsymbol{e}_p^{\mathsf{T}} K_p \boldsymbol{e}_p + \beta_p \boldsymbol{e}_p^{\mathsf{T}} \tilde{\boldsymbol{U}}_p - \tilde{\boldsymbol{U}}_p^{\mathsf{T}} K_{U_p} \tilde{\boldsymbol{U}}_p - \alpha_p \|\tilde{\boldsymbol{U}}_p\|_1 + \tilde{\boldsymbol{U}}_p^{\mathsf{T}} \boldsymbol{\varepsilon}_p + \kappa_p \|\tilde{\boldsymbol{U}}_p\| \operatorname{Tr}\left\{ \tilde{W}_p^{\mathsf{T}} (W_p - \tilde{W}_p) \right\},$$

which can be upper bounded as follows

$$\dot{V}_p \leq -\boldsymbol{\chi}_p^{\mathsf{T}} Q_p \boldsymbol{\chi}_p - \alpha_p \|\tilde{\boldsymbol{U}}_p\|_1 + \|\boldsymbol{\varepsilon}_p\| \|\tilde{\boldsymbol{U}}_p\| - \kappa_p \|\tilde{\boldsymbol{U}}_p\| \left[\|\tilde{W}_p\|_F^2 - \|\tilde{W}_p\|_F \|W_p\|_F \right], \tag{22}$$

where $\boldsymbol{\chi}_p = [\|\boldsymbol{e}_p\| \ \ ||\tilde{\boldsymbol{U}}_p||]^{\mathsf{T}} \in \mathbb{R}^2$,

$$Q_p = \begin{bmatrix} \beta_p \lambda_{\min}\{K_p\} & -\frac{\beta_p}{2} \\ -\frac{\beta_p}{2} & \lambda_{\min}\{K_{U_p}\} \end{bmatrix},$$

and the Frobenius norm [85]

$$\operatorname{Tr}\left\{ \tilde{W}_p^{\mathsf{T}} (W_p - \tilde{W}_p) \right\} = \left\langle \tilde{W}_p, \ W_p \right\rangle_F - \|\tilde{W}_p\|_F^2, \leq \|\tilde{W}_p\|_F \|W_p\|_F - \|\tilde{W}_p\|_F^2,$$

was used. Notice that the symmetric matrix Q_p is positive definite while $\beta_p < 4\lambda_{\min}\{K_p\}\lambda_{\min}\{K_{Up}\}$ is satisfied. Thus, completing the square in the last term of (22) and considering the bounds $-\boldsymbol{\chi}_p^{\mathsf{T}} Q_p \boldsymbol{\chi}_p \leq -\lambda_{\min}\{Q_p\}\|\boldsymbol{\chi}_p\|^2$, $\|\boldsymbol{x}\| \leq \|\boldsymbol{x}\|_1, \forall \ \boldsymbol{x} \in \mathbb{R}^n$ and recalling that $\|\boldsymbol{\varepsilon}_p\| \leq \varepsilon_P$ and $\|W_p\|_F \leq W_P$, the following upper bound is obtained

$$\dot{V}_p \leq -\lambda_{\min}\{Q_p\}\|\boldsymbol{\chi}_p\|^2 - \kappa_p \|\tilde{\boldsymbol{U}}_p\| \left(\|\tilde{W}_p\|_F - \frac{W_P}{2} \right)^2 - \|\tilde{\boldsymbol{U}}_p\|_1 \left[\alpha_p - \left(\frac{\kappa_p W_P^2}{4} + \varepsilon_P \right) \right], \tag{23}$$

and based on the upper bound (23) is observed that $\dot{V}_p$ is negative semidefinite while condition (21) is satisfied, which implies that $\boldsymbol{e}_p(t)$, $\tilde{\boldsymbol{U}}_p(t)$, $\boldsymbol{\xi}_p(t)$ and $\tilde{W}_p(t)$ are bounded, i.e., $\boldsymbol{e}_p(t)$, $\tilde{\boldsymbol{U}}_p(t)$, $\boldsymbol{\xi}_p(t) \in \mathcal{L}_\infty^3$ and $\tilde{W}_p \in \mathcal{L}_\infty^{L_N \times 3}$. An upper bound of (23) is given by

$$\dot{V}_p \leq -\lambda_{\min}\{Q_p\}\|\boldsymbol{\chi}_p\|^2.$$

Integrating both sides of the inequality yields

$$\frac{V_p(0)}{\lambda_{\min}\{Q_p\}} \geq \int_0^t \|\boldsymbol{\chi}_p\|^2 \mathrm{d}t,$$

since the integral exists and it is bounded, $\boldsymbol{\chi}_p(t)$ is a square-integrable function, i.e., $\boldsymbol{\chi}_p(t) \in \mathcal{L}_2^2 \Rightarrow \boldsymbol{e}_p(t)$, $\tilde{\boldsymbol{U}}_p(t) \in \mathcal{L}_2^3$. Previously by (23), it was proved that $\boldsymbol{e}_p(t)$, $\tilde{\boldsymbol{U}}_p(t)$, $\boldsymbol{\xi}_p(t) \in \mathcal{L}_\infty^3$ and $\tilde{W}_p \in \mathcal{L}_\infty^{10\times 3}$. Thus, using (9), (14), (15) and (20), can be demonstrated that $\dot{\boldsymbol{e}}_p(t)$, $\dot{\tilde{\boldsymbol{U}}}_p(t)$, $\dot{\boldsymbol{\xi}}_p(t) \in \mathcal{L}_\infty^3$ and $\dot{\tilde{W}}_p \in \mathcal{L}_\infty^{10\times 3}$. Besides, deriving (9) and using the previous facts

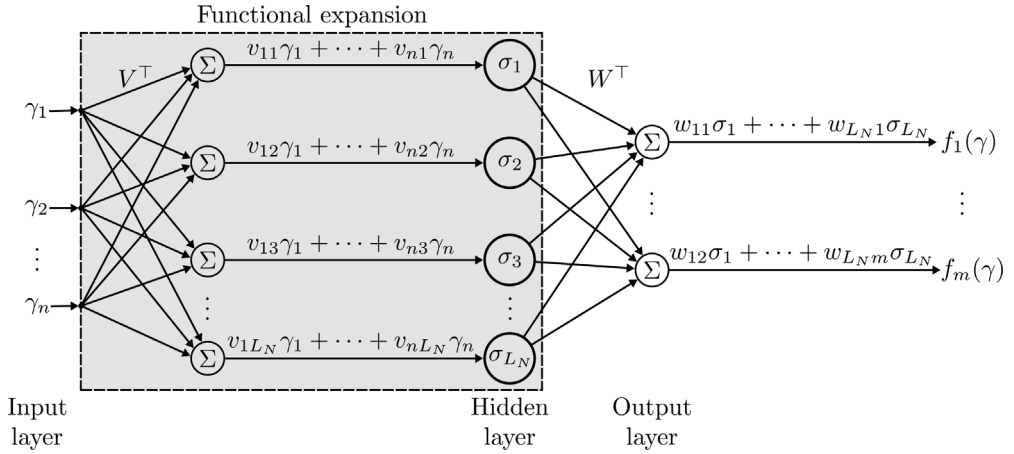

Fig. 3. Diagram of a FLNN where $\boldsymbol{\gamma} \in \mathbb{R}^n$ is the NN's extended input vector, $V \in \mathbb{R}^{n \times L_N}$ and $W \in \mathbb{R}^{L_N \times m}$ are the input and output weights matrices, respectively, $\boldsymbol{\sigma} \in \mathbb{R}^{L_N}$ is the vector of activation functions, $\boldsymbol{f}(\boldsymbol{\gamma}) \in \mathbb{R}^m$ NN's output vector, and $L_N \in \mathbb{R}$ is the number of neurons in the hidden layer.

it can be concluded that $\ddot{\boldsymbol{e}}_p(t) \in \mathcal{L}_\infty^3$. Sufficient conditions have been met to invoke Barbalat's lemma [87], leading to conclude that $\boldsymbol{e}_p(t)$, $\tilde{\boldsymbol{U}}_p(t) \to 0$ as $t \to \infty$ implying that $\dot{\boldsymbol{e}}_p(t) \to 0$ as $t \to \infty$, which proofs Proposition 1. □

The neural networks adopted in this work are single-hidden-layer neural networks, which are linear-in-the-parameters (LIP) of the output weights [85]. This specific NN is known as functional-link neural network (FLNN) [85,86]. The general structure of an FLNN is formed by three layers: input layer, hidden layer, and output layer. The hidden layer is contained in a block known as "functional expansion" that comprises the product of the NN's input vector and the input weights and its processing through the activation functions. The FLNNs are simpler than the two-layer NNs since their LIP property. In the FLNN, only the output weights are tuned. Thus, the error convergence analysis is more straightforward, and its implementation is more accessible and computationally less demanding. See Fig. 3 for a diagram of the FLNN's structure.

3.2. Attitude controller

The attitude controller has a two-loop structure where the outer loop is an attitude control and the inner loop is an adaptive neural network-based Euler angles change rate control.

3.2.1. Attitude outer loop

The attitude outer loop defines an Euler angles change rate reference signal $\boldsymbol{U}_\eta \in \mathbb{R}^3$ given by

$$\boldsymbol{U}_\eta = \dot{\boldsymbol{\eta}}_d + K_\eta \boldsymbol{e}_\eta, \tag{24}$$

where $K_\eta \in \mathbb{R}^{3\times3}$ is a positive definite diagonal gain matrix. Considering the attitude error in (4), taking its time derivative and substituting the Euler angles change rate reference signal in (24) the following is obtained

$$\dot{\boldsymbol{e}}_\eta = -K_\eta \boldsymbol{e}_\eta + \boldsymbol{U}_\eta - \dot{\boldsymbol{\eta}}, \tag{25}$$

defining the Euler angles change rate error

$$\tilde{\boldsymbol{U}}_\eta = \boldsymbol{U}_\eta - \dot{\boldsymbol{\eta}}, \tag{26}$$

the attitude error dynamics (25) results in

$$\dot{\boldsymbol{e}}_\eta = -K_\eta \boldsymbol{e}_\eta + \tilde{\boldsymbol{U}}_\eta, \tag{27}$$

where it can be observed that as long as $\tilde{\boldsymbol{U}}_\eta \to 0$ is guaranteed, then, $\boldsymbol{e}_\eta \to 0$ while $t \to \infty$.

3.2.2. Adaptive neural network-based Euler angles change rate inner loop

Taking the time derivative of (26) and substituting the attitude dynamics (2) on it, the resulting Euler angles change rate error dynamics is defined as follows

$$\dot{\tilde{\boldsymbol{U}}}_\eta = \dot{\boldsymbol{U}}_\eta - M(\boldsymbol{\eta})^{-1}\left[\Phi(\boldsymbol{\eta})^{-\mathsf{T}}\boldsymbol{\tau} + \boldsymbol{\delta}_\eta(t) - \left(C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) + D_\eta(\boldsymbol{\eta})\right)\dot{\boldsymbol{\eta}}\right]. \tag{28}$$

Based on (28), assume the existence of $\boldsymbol{\tau}_{MB}(t)$, an external disturbances and exact model compensator given by

$$\boldsymbol{\tau}_{MB} = \Phi(\boldsymbol{\eta})^{\mathsf{T}}\left[M(\boldsymbol{\eta})\dot{\boldsymbol{U}}_\eta + \left(C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) + D_\eta(\boldsymbol{\eta})\right)\dot{\boldsymbol{\eta}} - \boldsymbol{\delta}_\eta(t)\right], \tag{29}$$

which in closed-loop, ideally will reject external disturbances and compensate the whole system dynamics. It is worth mentioning that such a compensator implies an accurate knowledge of the system parameters and precise measurement and estimation of the external disturbances, which is very difficult to achieve in real applications. However, the capabilities of the neural network for the approximation of continuous functions [84–86] can be used to obtain a said model compensator, represented as follows

$$\boldsymbol{\tau}_{MB} = \Phi(\boldsymbol{\eta})^{\mathsf{T}}\left[\boldsymbol{f}_\eta(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \dot{\boldsymbol{U}}_\eta) - \boldsymbol{\delta}_\eta(t)\right],$$

where $\boldsymbol{f}_\eta(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \dot{\boldsymbol{U}}_\eta) = M(\boldsymbol{\eta})\dot{\boldsymbol{U}}_\eta + \left(C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) + D_\eta(\boldsymbol{\eta})\right)\dot{\boldsymbol{\eta}}$. Finally, the model compensator can be approximated with a neural network as follows

$$\boldsymbol{\tau}_{MB} = \Phi(\boldsymbol{\eta})^{\mathsf{T}}\left[\boldsymbol{f}_{W_\eta}(\boldsymbol{\gamma}_\eta) - \boldsymbol{\delta}_\eta(t)\right] = \Phi(\boldsymbol{\eta})^{\mathsf{T}}\left[W_\eta^{\mathsf{T}}\boldsymbol{\sigma}_\eta\left(V_\eta^{\mathsf{T}}\boldsymbol{\gamma}_\eta\right) + \boldsymbol{\epsilon}_\eta - \boldsymbol{\delta}_\eta(t)\right], \tag{30}$$

where $\boldsymbol{\gamma}_\eta = [1\ \boldsymbol{\eta}^{\mathsf{T}}\ \dot{\boldsymbol{\eta}}^{\mathsf{T}}\ \dot{\boldsymbol{U}}_\eta^{\mathsf{T}}]^{\mathsf{T}} \in \mathbb{R}^{10}$ is the neural network extended input vector, $V_\eta \in \mathbb{R}^{10\times L_\eta}$ and $W_\eta \in \mathbb{R}^{L_\eta\times 3}$ are the ideal weight input and output matrices, respectively, that are bounded as $\|V_\eta\|_F \leq V_\Omega$ and $\|W_\eta\|_F \leq W_\Omega$, respectively, $\boldsymbol{\epsilon}_\eta \in \mathbb{R}^3$ is the neural network approximation error, bounded as $0 < \|\boldsymbol{\epsilon}_\eta\| \leq \epsilon_\Omega$, $L_\eta \in \mathbb{N}$ is the neurons quantity, and $\sigma(\boldsymbol{x})\ \forall\ \boldsymbol{x} \in \mathbb{R}^n$ represents the neural network activation function. Considering $\boldsymbol{v}_{\gamma_\eta} = V_\eta^{\mathsf{T}}\boldsymbol{\gamma}_\eta \in \mathbb{R}^{L_\eta}$, then, $\sigma(\boldsymbol{v}_{\gamma_\eta}) = [\tanh(v_{\gamma_\eta 1})\cdots\tanh(v_{\gamma_\eta L_\eta})]^{\mathsf{T}} \in \mathbb{R}^{L_\eta}$ which henceforth be denoted as $\boldsymbol{\sigma}_\eta \in \mathbb{R}^{L_\eta}$. Thus, based in Eqs. (28)–(30) the following neural network-based controller is proposed

$$\boldsymbol{\tau} = \Phi(\boldsymbol{\eta})^{\mathsf{T}}\left[K_{U_\eta}\tilde{\boldsymbol{U}}_\eta + K_{i\eta}\boldsymbol{\xi}_\eta + \hat{W}_\eta^{\mathsf{T}}\boldsymbol{\sigma}_\eta + \alpha_\eta \mathrm{sign}(\tilde{\boldsymbol{U}}_\eta)\right], \tag{31}$$

$$\dot{\boldsymbol{\xi}}_\eta = \tilde{\boldsymbol{U}}_\eta, \tag{32}$$

where K_{U_η}, $K_{i\eta} \in \mathbb{R}^{3\times 3}$ are positive definite diagonal gain matrices, $\alpha_\eta \in \mathbb{R}$ is a strictly positive constant and $\hat{W}_\eta$ is an estimation of the output weight matrix W_η obtained through the following adaptation law

$$\dot{\hat{W}}_\eta = -N_\eta\left(\boldsymbol{\sigma}_\eta\tilde{\boldsymbol{U}}_\eta^{\mathsf{T}} - \kappa_\eta\|\tilde{\boldsymbol{U}}_\eta\|\hat{W}_\eta\right), \tag{33}$$

where the output weight matrix estimation error was defied as

$$\tilde{W}_\eta = W_\eta - \hat{W}_\eta, \tag{34}$$

$N_\eta \in \mathbb{R}^{3\times 3}$ is a positive definite diagonal matrix with the neural network adaptation gains and $\kappa_\eta \in \mathbb{R}$ is a positive constant. Based on Eqs. (29) and (30), the controller (31) can be rewritten as

$$\boldsymbol{\tau} = \Phi(\boldsymbol{\eta})^\top \Big[K_{U_\eta}\tilde{\boldsymbol{U}}_\eta + K_{i\eta}\boldsymbol{\xi}_\eta + M(\boldsymbol{\eta})\dot{\boldsymbol{U}}_\eta + \big(C(\boldsymbol{\eta},\dot{\boldsymbol{\eta}}) + D_\eta(\boldsymbol{\eta})\big)\dot{\boldsymbol{\eta}} - \boldsymbol{\delta}_\eta(t) - \boldsymbol{\varepsilon}_\eta - \tilde{W}_\eta^\top \boldsymbol{\sigma}_\eta + \alpha_\eta \text{sign}(\tilde{\boldsymbol{U}}_\eta)\Big],$$

which in closed-loop with the angular velocity error dynamics (28) results in the following

$$M(\boldsymbol{\eta})\dot{\tilde{\boldsymbol{U}}}_\eta = -K_{U_\eta}\tilde{\boldsymbol{U}}_\eta - K_{i\eta}\boldsymbol{\xi}_\eta + \tilde{W}_\eta^\top \boldsymbol{\sigma}_\eta + \boldsymbol{\varepsilon}_\eta - \alpha_\eta \text{sign}(\tilde{\boldsymbol{U}}_\eta), \tag{35}$$

where $\boldsymbol{\varepsilon}_\eta = \boldsymbol{\epsilon}_\eta - \boldsymbol{\delta}_\eta(t)$ is bounded as $0 < \|\boldsymbol{\varepsilon}_\eta\| \leq \varepsilon_\Omega$.

Proposition 2. *Assume gain matrices K_η, K_{U_η} and N_η to be positive definite diagonal matrices and positive constants α_η and κ_η satisfying the condition*

$$\alpha_\eta \geq \frac{\kappa_p W_\Omega^2}{4} + \varepsilon_\Omega. \tag{36}$$

Then, for all initial conditions starting as some compact set, the solutions $\boldsymbol{e}_\eta(t)$, $\dot{\boldsymbol{e}}_\eta(t)$ and $\tilde{\boldsymbol{U}}_\eta(t)$ of the overall closed-loop system in (27), (32), (33), and (35) converge to zero as time goes to infinity, i.e., $\boldsymbol{e}_\eta(t)$, $\dot{\boldsymbol{e}}_\eta(t)$, $\tilde{\boldsymbol{U}}_\eta(t) \to 0$ while $t \to \infty$. Furthermore, the state variable $\boldsymbol{\xi}_\eta(t)$ and the output weight matrix estimation error $\tilde{W}_\eta(t)$ remains bounded for all time $t \geq 0$.

Proof. The following non-negative function is proposed

$$V_\eta = \frac{\beta_\eta 1}{2}\boldsymbol{e}_\eta^\top \boldsymbol{e}_\eta + \frac{1}{2}\tilde{\boldsymbol{U}}_\eta^\top M(\boldsymbol{\eta})\tilde{\boldsymbol{U}}_\eta + \frac{1}{2}\boldsymbol{\xi}_\eta^\top K_{i\eta}\boldsymbol{\xi}_\eta + \frac{1}{2}\text{Tr}\Big\{\tilde{W}_\eta^\top N_\eta^{-1}\tilde{W}_\eta\Big\}.$$

The time derivative along the trajectories of the closed-loop system is given by

$$\dot{V}_\eta = -\beta_\eta 1 \boldsymbol{e}_\eta^\top K_\eta \boldsymbol{e}_\eta + \beta_\eta 1 \boldsymbol{e}_\eta^\top \tilde{\boldsymbol{U}}_\eta - \tilde{\boldsymbol{U}}_\eta^\top K_{U_\eta}\tilde{\boldsymbol{U}}_\eta - \alpha_\eta \tilde{\boldsymbol{U}}_\eta^\top \text{sign}(\tilde{\boldsymbol{U}}_\eta) + \tilde{\boldsymbol{U}}_\eta^\top \tilde{W}_\eta^\top \boldsymbol{\sigma}_\eta + \tilde{\boldsymbol{U}}_\eta^\top \boldsymbol{\varepsilon}_\eta + \text{Tr}\Big\{\tilde{W}_\eta^\top N_\eta^{-1}\dot{\tilde{W}}_\eta\Big\}.$$

Using that by the definition of the output weight matrix estimation error in (34) $\dot{\tilde{W}}_\eta = -\dot{\hat{W}}_\eta$, substituting the adaptation law (33) and using the facts $\tilde{\boldsymbol{U}}_\eta^\top \tilde{W}_\eta^\top \boldsymbol{\sigma}_\eta = \text{Tr}\Big\{\tilde{W}_\eta^\top \boldsymbol{\sigma}_\eta \tilde{\boldsymbol{U}}_\eta^\top\Big\}$ and $\alpha_\eta \tilde{\boldsymbol{U}}_\eta^\top \text{sign}(\tilde{\boldsymbol{U}}_\eta) = \alpha_\eta \|\tilde{\boldsymbol{U}}_\eta\|_1$ is obtained

$$\dot{V}_\eta = -\beta_\eta 1 \boldsymbol{e}_\eta^\top K_\eta \boldsymbol{e}_\eta + \beta_\eta 1 \boldsymbol{e}_\eta^\top \tilde{\boldsymbol{U}}_\eta - \tilde{\boldsymbol{U}}_\eta^\top K_{U_\eta}\tilde{\boldsymbol{U}}_\eta - \alpha_\eta \|\tilde{\boldsymbol{U}}_\eta\|_1 + \tilde{\boldsymbol{U}}_\eta^\top \boldsymbol{\varepsilon}_\eta + \kappa_\eta \|\tilde{\boldsymbol{U}}_\eta\| \text{Tr}\Big\{\tilde{W}_\eta^\top \Big(W_\eta - \tilde{W}_\eta\Big)\Big\},$$

which can be upper bounded as follows

$$\dot{V}_\eta \leq -\boldsymbol{\chi}_\eta^\top Q_\eta \boldsymbol{\chi}_\eta - \alpha_\eta \|\tilde{\boldsymbol{U}}_\eta\|_1 + \|\boldsymbol{\varepsilon}_\eta\| \|\tilde{\boldsymbol{U}}_\eta\| - \kappa_\eta \|\tilde{\boldsymbol{U}}_\eta\| \Big[\|\tilde{W}_\eta\|_F^2 - \|\tilde{W}_\eta\|_F \|W_\eta\|_F \Big], \tag{37}$$

where $\boldsymbol{\chi}_\eta = [\|\boldsymbol{e}_\eta\| \;\; ||\tilde{\boldsymbol{U}}_\eta||]^\top \in \mathbb{R}^2$,

$$Q_\eta = \begin{bmatrix} \beta_\eta 1 \lambda_{\min}\{K_\eta\} & -\frac{\beta_\eta 1}{2} \\ -\frac{\beta_\eta 1}{2} & \lambda_{\min}\{K_{U_\eta}\} \end{bmatrix},$$

and the Frobenius norm [85]

$$\begin{aligned}\mathrm{Tr}\left\{\tilde{W}_\eta^{\mathrm{T}}(W_\eta-\tilde{W}_\eta)\right\} &= \left\langle \tilde{W}_\eta,\ W_\eta\right\rangle_F - \|\tilde{W}_\eta\|_F^2,\\ &\leq \|\tilde{W}_\eta\|_F\|W_\eta\|_F - \|\tilde{W}_\eta\|_F^2,\end{aligned}$$

was used. Notice that the symmetric matrix Q_η is positive definite while $\beta_\eta 1 < 4\lambda_{\min}\{K_\eta\}\lambda_{\min}\{K_{U\eta}\}$ is satisfied. Thus, completing the square in the last term of (37) and considering the bounds $-\boldsymbol{\chi}_\eta^{\mathrm{T}}Q_\eta\boldsymbol{\chi}_\eta \leq -\lambda_{\min}\{Q_\eta\}||\boldsymbol{\chi}_\eta||^2$, $\|\boldsymbol{x}\| \leq \|\boldsymbol{x}\|_1, \forall\ \boldsymbol{x}\in\mathbb{R}^n$ and recalling that $\|\boldsymbol{\varepsilon}_\eta\| \leq \varepsilon_\Omega$ and $\|W_\eta\|_F \leq W_\Omega$, the following upper bound is obtained

$$\dot{V}_\eta \leq -\lambda_{\min}\{Q_\eta\}\|\boldsymbol{\chi}_\eta\|^2 - \kappa_\eta\|\tilde{\boldsymbol{U}}_\eta\|\left(\|\tilde{W}_\eta\|_F - \frac{W_\Omega}{2}\right)^2 - \|\tilde{\boldsymbol{U}}_\eta\|_1\left[\alpha_\eta - \left(\frac{\kappa_\eta W_\Omega^2}{4}+\varepsilon_\Omega\right)\right], \tag{38}$$

and based on the upper bound (38) is observed that $\dot{V}_\eta$ is negative semidefinite while condition (36) is satisfied, which implies that $\boldsymbol{e}_\eta(t)$, $\tilde{\boldsymbol{U}}_\eta(t)$, $\boldsymbol{\xi}_\eta(t)$ and $\tilde{W}_\eta(t)$ are bounded, i.e., $\boldsymbol{e}_\eta(t),\ \tilde{\boldsymbol{U}}_\eta(t),\ \boldsymbol{\xi}_\eta(t)\in\mathcal{L}_\infty^3$ and $\tilde{W}_\eta\in\mathcal{L}_\infty^{L_N\times 3}$. An upper bound of (38) is given by

$$\dot{V}_\eta \leq -\lambda_{\min}\{Q_\eta\}\|\boldsymbol{\chi}_\eta\|^2.$$

Integrating both sides of the inequality yields

$$\frac{V_\eta(0)}{\lambda_{\min}\{Q_\eta\}} \geq \int_0^t \|\boldsymbol{\chi}_\eta\|^2\mathrm{d}t,$$

since the integral exists and it is bounded, $\boldsymbol{\chi}_\eta(t)$ is a square-integrable function, i.e., $\boldsymbol{\chi}_\eta(t)\in\mathcal{L}_2^2 \Rightarrow \boldsymbol{e}_\eta(t),\ \tilde{\boldsymbol{U}}_\eta(t)\in\mathcal{L}_2^3$. Previously by (38), it was proved that $\boldsymbol{e}_\eta(t),\ \tilde{\boldsymbol{U}}_\eta(t),\ \boldsymbol{\xi}_\eta(t)\in\mathcal{L}_\infty^3$ and $\tilde{W}_\eta\in\mathcal{L}_\infty^{10\times 3}$. Thus, using (27), (32), (33) and (35), can be demonstrated that $\dot{\boldsymbol{e}}_\eta(t),\ \dot{\tilde{\boldsymbol{U}}}_\eta(t),\ \dot{\boldsymbol{\xi}}_\eta(t)\in\mathcal{L}_\infty^3$ and $\dot{\tilde{W}}_\eta\in\mathcal{L}_\infty^{10\times 3}$. Besides, deriving (27) and using the previous facts it can be concluded that $\ddot{\boldsymbol{e}}_\eta(t)\in\mathcal{L}_\infty^3$. Sufficient conditions have been met to invoke Barbalat's lemma [87], leading to conclude that $\boldsymbol{e}_\eta(t),\ \tilde{\boldsymbol{U}}_\eta(t)\to 0$ as $t\to\infty$ implying that $\dot{\boldsymbol{e}}_\eta(t)\to 0$ as $t\to\infty$, which proofs Proposition 2. □

Therefore, with the fulfillment of Propositions 1 and 2, the control goal (5) is satisfied.

4. Experimental results

Aiming to prove the functionality of the proposed controller, an experimental comparison against an adaptive neural controller and a model-based controller was carried out using the QBall 2 quadrotor from Quanser®, which has been widely used in the quadrotor control research field [88–93]. The experimental setup used in this work is depicted in Fig. 4 and is formed by the motion capture system OptiTrack® together with the QBall 2 quadrotor. The OptiTrack® system is constituted by an array of six Flex 3 cameras synchronized and connected to a PC running the optical motion capture software Motive™. This software handles the data acquired by the cameras and computes the position $\boldsymbol{p}(t)$ and yaw angle $\psi(t)$ of the quadrotor in real-time. The quadrotor has an inertial measurement unit (IMU) and a sonar from which the angular velocity, attitude, and height are estimated. In this work, the IMU is in charge of acquiring the roll $\phi(t)$ and pitch $\theta(t)$ angles only. The controller is implemented in MATLAB-Simulink® using the real-time control software Quarc™. The model is compiled and transformed into an executable code which is downloaded via Wi-Fi

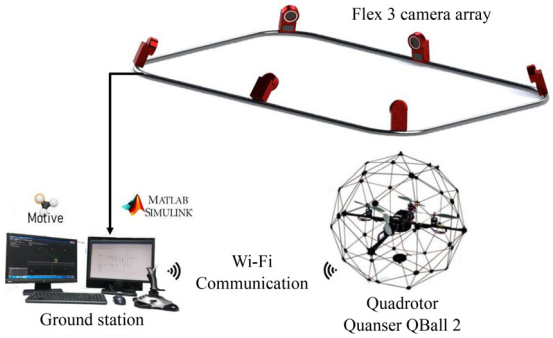

Fig. 4. Experimental setup formed by the motion capture system OptiTrack® and the quadrotor QBall 2 from Quanser®.

Table 1
QBall 2 quadrotor parameters.

Parameter	Description	Value	Units
g	Gravitational acceleration constant	9.81	m/s^2
m	Quadrotor mass	1.79	kg
I_{xx}	Inertia moment with respect to the axis x	0.03	kg m^2
I_{yy}	Inertia moment with respect to the axis y	0.03	kg m^2
I_{zz}	Inertia moment with respect to the axis z	0.04	kg m^2
$D_p(\boldsymbol{\eta})$	Aerodynamic drag coefficient matrix	diag{0.002, 0.002, 0.004}	kg/s
$D_\eta(\boldsymbol{\eta})$	Aerodynamic damping coefficient matrix	diag{0.002, 0.002, 0.004}	kg m^2/s

on the quadrotor to be executed in real-time on the Gumstix® DuoVero onboard computer operating at a maximum frequency of 1 [GHz]. The experimental tests were performed with a sampling frequency of 2 [kHz] for the IMU and 30 [kHz] for the motion capture system. The parameters of the QBall 2 quadrotor used during the experimental tests correspond to the given by the manufacturer and are listed in Table 1. Further details about the QBall 2 quadrotor can be consulted in the user manual [94]. Note that the implementation of the controllers was carried on using a mass estimate corresponding to 70% of the nominal value. This was set to evaluate the performance of the controllers under parametric uncertainties.

Two-loop model based controller for comparison

The model based controller used for the comparison was obtained based on the error dynamics (10) and (28) and is explicitly given by

$$F = \frac{f_{p3}}{\cos(\phi)\cos(\theta)}, \tag{39}$$

$$\phi_d = \tan^{-1}\left[\frac{\cos(\theta_d)}{f_{p3}}\left(f_{p1}\sin(\psi_d) - f_{p2}\cos(\psi_d)\right)\right], \tag{40}$$

$$\theta_d = \tan^{-1}\left[\frac{1}{f_{p3}}\left(f_{p1}\cos(\psi_d) + f_{p2}\sin(\psi_d)\right)\right], \tag{41}$$

$$\boldsymbol{\tau} = \Phi(\boldsymbol{\eta})^{\mathsf{T}}\Big[K_{1\eta}\tilde{\boldsymbol{U}}_{\eta} + K_{2\eta}\boldsymbol{\xi}_{\eta} + M(\boldsymbol{\eta})\dot{\boldsymbol{U}}_{\eta} + \big(C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) + D_{\eta}(\boldsymbol{\eta})\big)\dot{\boldsymbol{\eta}}\Big], \tag{42}$$

$$\dot{\boldsymbol{\xi}}_{\eta} = \tilde{\boldsymbol{U}}_{\eta}, \tag{43}$$

where $F_p = [f_{p1}\ \ f_{p2}\ \ f_{p3}]^{\mathsf{T}}$ is defined as

$$\boldsymbol{F}_p = K_{1p}\tilde{\boldsymbol{U}}_p + K_{2p}\boldsymbol{\xi}_p + m(\dot{\boldsymbol{U}}_p + \boldsymbol{g}_z) + D_p(\eta)\dot{\boldsymbol{p}}, \tag{44}$$

$$\dot{\boldsymbol{\xi}}_p = \tilde{\boldsymbol{U}}_p, \tag{45}$$

K_{1p}, K_{2p}, K_{3p}, $K_{1\eta}$, $K_{2\eta}$ and $K_{3\eta} \in \mathbb{R}^{3\times 3}$ are positive definite diagonal gain matrices. $\tilde{\boldsymbol{U}}_p$, $\boldsymbol{U}_p$, $\tilde{\boldsymbol{U}}_{\eta}$, and $\boldsymbol{U}_{\eta}$ are defined as

$$\tilde{\boldsymbol{U}}_p = \boldsymbol{U}_p - \dot{\boldsymbol{p}}, \tag{46}$$

$$\boldsymbol{U}_p = \dot{\boldsymbol{p}}_d + K_{3p}\boldsymbol{e}_p, \tag{47}$$

$$\tilde{\boldsymbol{U}}_{\eta} = \boldsymbol{U}_{\eta} - \dot{\boldsymbol{\eta}}, \tag{48}$$

$$\boldsymbol{U}_{\eta} = \dot{\boldsymbol{\eta}}_d + K_{3\eta}\boldsymbol{e}_{\eta}. \tag{49}$$

Hereafter, the controller (39)-(49) will be denoted as TLMBC (two-loop model based controller). The gains for the TLMBC were obtained through a trial and error process resulting in the following values

$K_{1p} = \text{diag}\{2.25,\ 2.25,\ 2.25\}$, $K_{2p} = \text{diag}\{0.004,\ 0.004,\ 1.0\}$,
$K_{3p} = \text{diag}\{1.7957,\ 1.7957,\ 1.7957\}$,
$K_{1\eta} = \text{diag}\{0.475,\ 0.475,\ 0.9\}$, $K_{2\eta} = \text{diag}\{0.001,\ 0.001,\ 0.005\}$,
$K_{3\eta} = \text{diag}\{4.7368,\ 4.7368,\ 7.7778\}$.

Adaptive neural network control for comparison

The adaptive neural scheme chosen for the comparison was reported in [95]. The scheme also possesses a two-loop structure: an adaptive neural network-based position controller and an integral sliding mode attitude controller. The position controller is given by

$$F = u_{A3}, \tag{50}$$

$$\phi_d(t) = \sin^{-1}\left[u_{A1}\sin(\psi) - u_{A2}\cos(\psi)\right], \tag{51}$$

$$\theta_d(t) = \sin^{-1}\left[(u_{A1}\cos(\psi) + u_{A2}\sin(\psi))/\cos(\phi_d)\right], \tag{52}$$

with $\boldsymbol{u}_A = [\bar{u}_1 F,\ \bar{u}_2 F,\ F]^{\mathsf{T}}$ defined as

$$\boldsymbol{u}_A = B(\boldsymbol{\eta})^{-1}[\boldsymbol{g}_z + \hat{\boldsymbol{f}}(\boldsymbol{X}_{\text{in}}) + k_v\boldsymbol{\gamma} + \ddot{\boldsymbol{p}}_d], \tag{53}$$

being $\bar{u}_1 = \cos(\psi)\sin(\theta)\cos(\phi) + \sin(\psi)\sin(\phi)$ and $\bar{u}_2 = \sin(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi)$. The matrix $B(\boldsymbol{\eta}) = 1/m\ \text{diag}\{1,\ 1,\ \cos(\phi)\cos(\theta)\} \in \mathbb{R}^{3\times 3}$, $\boldsymbol{\gamma} = \dot{\boldsymbol{e}}_p + \Lambda \boldsymbol{e}_p \in \mathbb{R}^3$ is an auxiliary state vector, being $\Lambda = \Lambda^{\mathsf{T}} > 0$, $\boldsymbol{e}_p \in \mathbb{R}^3$ is the position error defined in (3), and $k_v \in \mathbb{R}$ is a strictly positive constant. The term $\ddot{\boldsymbol{p}}_d = [\ddot{x}_d,\ \ddot{y}_d,\ \ddot{z}_d]^{\mathsf{T}}$ was added by us to improve the performance of the controller. The vector $\hat{\boldsymbol{f}}(\boldsymbol{X}_{\text{in}}) = \hat{W}^{\mathsf{T}}\boldsymbol{P}(\boldsymbol{X}_{\text{in}})$ represents the output of the radial basis function neural network, being $\boldsymbol{X}_{\text{in}}$ the input vector of the neural network with an activation function defined by

$$P_i(\boldsymbol{X}_{\text{in}}) = e^{\left(-\frac{||X_{\text{in}}-c_i||^2}{\sigma_i^2}\right)},$$

where c_i is the center and σ_i is the width of the Gaussian function. The matrix $\hat{W}$ is the output weight matrix estimate, obtained as

$$\dot{\hat{W}} = A\boldsymbol{P}(\boldsymbol{X}_{\text{in}})\boldsymbol{\gamma}^{\mathsf{T}}, \tag{54}$$

being A a symmetric positive definite gain matrix. The attitude controller is given by

$$\boldsymbol{\tau} = M(\boldsymbol{\eta})[\dot{\boldsymbol{\nu}}_\eta + k_\eta \tilde{\boldsymbol{\nu}}] + C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \rho_\eta \text{sign}(\boldsymbol{s}_\eta), \tag{55}$$

where ρ_η and k_η are strictly positive constants, $\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu}_\eta - \dot{\boldsymbol{\eta}}$ is the angular velocity error, $\boldsymbol{\nu}_\eta = \dot{\boldsymbol{\eta}}_d + k_w \boldsymbol{e}_\eta + \rho_w \text{sign}(\boldsymbol{s}_w)$, the attitude error $\boldsymbol{e}_\eta \in \mathbb{R}^3$ was defined in (4), and ρ_w and k_w are positive constants. The sliding surfaces are defined as $\boldsymbol{s}_w = \boldsymbol{e}_\eta + k_w \in 0^t \boldsymbol{e}_\eta$ and $\boldsymbol{s}_\eta = \tilde{\boldsymbol{\nu}} + k_\eta \in 0^t \tilde{\boldsymbol{\nu}}$. More details about this control scheme can be obtained in [95]. The adaptive neural controller was implemented using the following values

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

The controller (50)-(55) will be referenced from now on as NNSMC (neural network sliding modes control).

Proposed controller

The proposed controller (13)-(14), (17)-(19) and (31)-(32) will be referred to as DTLANNC (double two-loop adaptive neural network-based controller) in the remainder of the document. The DTLANNC gains were obtained through a trial and error process resulting in the following values

$$\begin{aligned}
&K_{U_p} = \text{diag}\{4.0,\ 4.0,\ 4.5\}, && K_{U_\eta} = \text{diag}\{0.475,\ 0.475,\ 0.9\},\\
&K_{ip} = \text{diag}\{0.004,\ 0.004,\ 1.0\}, && K_{i\eta} = \text{diag}\{0.001,\ 0.001,\ 0.005\},\\
&K_p = \text{diag}\{1.5375,\ 1.5375,\ 2.3\}, && K_\eta = \text{diag}\{4.7368,\ 4.7368,\ 7.7778\},\\
&N_p = \text{diag}_{10}\{0.35\}, && N_\eta = \text{diag}_{10}\{0.001\},\\
&\kappa_p = 0.075, && \kappa_\eta = 0.002,\\
&\alpha_p = 0.001, && \alpha_\eta = 0.001.
\end{aligned}$$

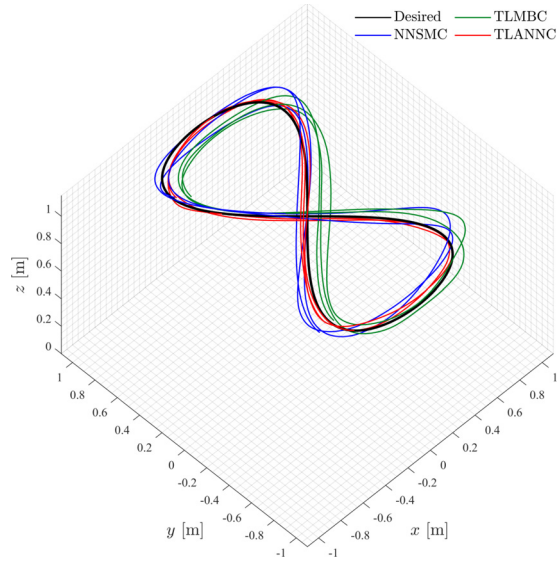

Fig. 5. Paths drawn by the quadrotor during the experimental tests implementing the TLMBC, NNSMC, and DT-LANNC schemes by tracking the trajectory in (56) in the interval 15 [s] $\leq t \leq$ 35 [s].

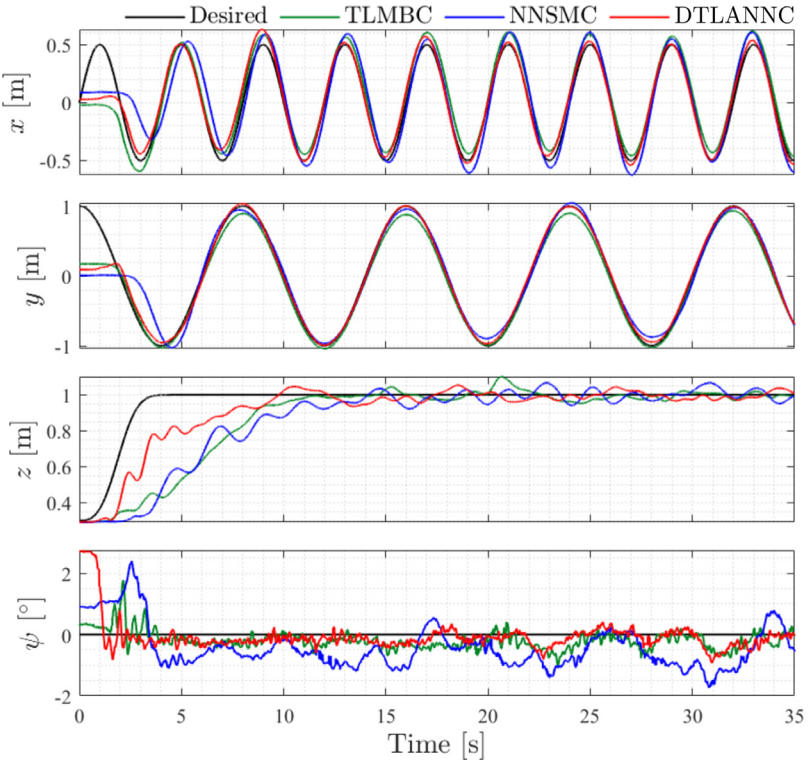

Fig. 6. Time evolution of the position $\boldsymbol{p}(t) = [x(t)\ y(t)\ z(t)]^{\mathsf{T}}$ and the yaw angle $\psi(t)$ by implementing the TLMBC, NNSMC, and DTLANNC schemes in real-time experiments performing the trajectory tracking task defined in (56).

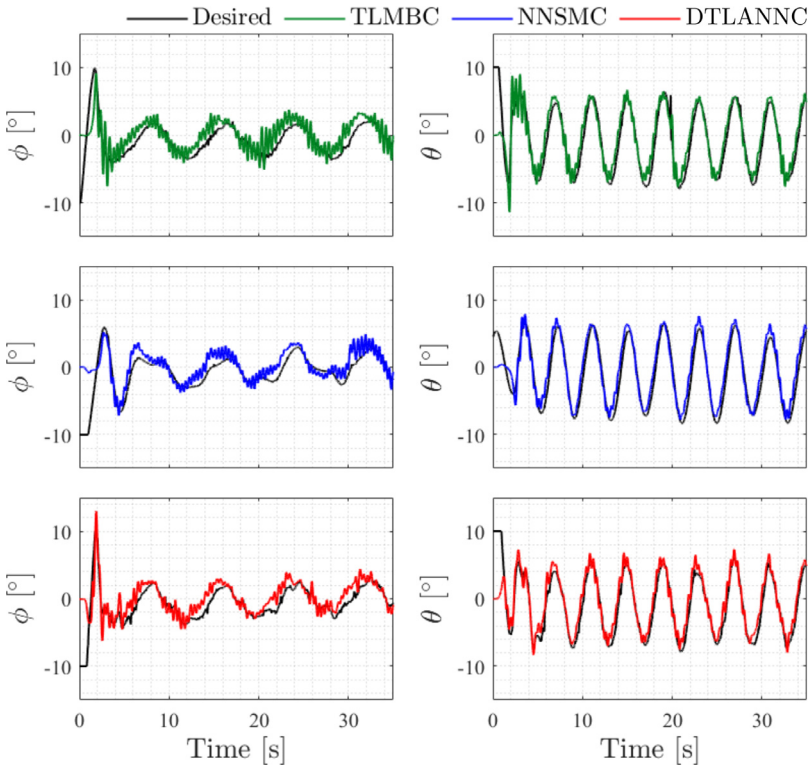

Fig. 7. Time evolution of the computed desired roll $\phi_d(t)$ and pitch $\theta_d(t)$ angles and the actual roll $\phi(t)$ and pitch $\theta(t)$ angles by implementing the TLMBC, NNSMC, and DTLANNC schemes in real-time experiments performing the trajectory tracking task defined in (56).

The output weight matrix estimate $\hat{W}_p$ was initialized with zeros and the output weight matrix estimate $\hat{W}_\eta$ was initialized as 0.01 for all its elements while the input weight matrices V_p and V_η were initialized with values into the set $[-1,\ 1]$. The number of neurons used in the experiment was $L_p = L_\eta = 10$. Notice that the experimental platform provides sufficient computational capabilities to implement the adaptive neural network considering the specifications previously mentioned. In order to reduce the chattering phenomenon resulting from the discontinuous term in (13) and (31), the gains related to these terms were chosen small.

4.1. Lemniscate path tracking

A trajectory tracking task was used to validate the performance of the proposed controller. The tracking lemniscate path was defined with the following desired signals

$$x_d(t) = 0.5 \sin\left(\frac{2\pi}{4}t\right) \text{ [m]},$$

$$y_d(t) = \cos\left(\frac{2\pi}{8}t\right) \text{ [m]},$$

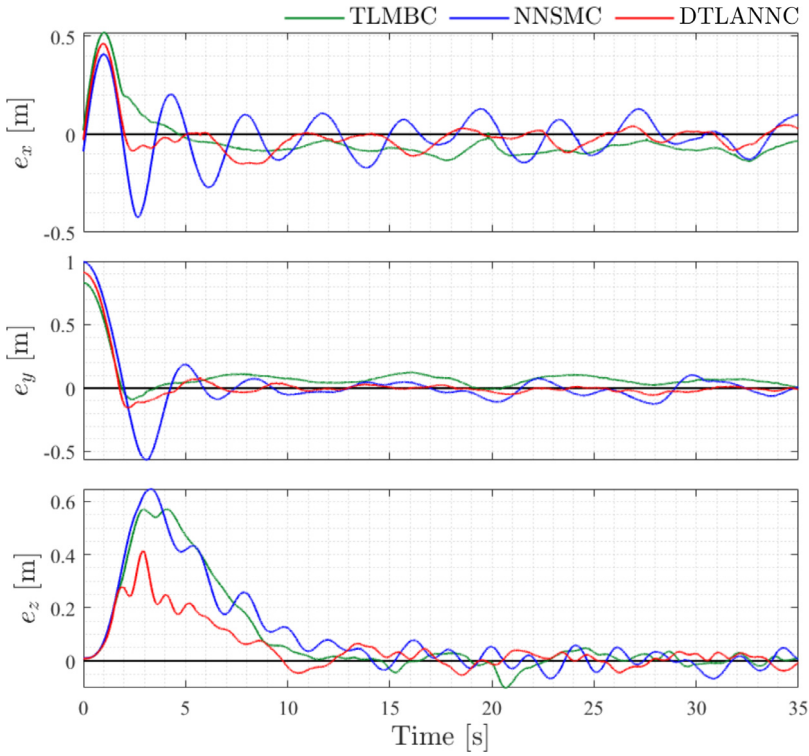

Fig. 8. Time evolution of the position error $\boldsymbol{e}_p(t) = [e_x(t)\ e_y(t)\ e_z(t)]^{\mathsf{T}}$ by implementing the TLMBC, NNSMC, and DTLANNC schemes in real-time experiments performing the trajectory tracking task defined in (56).

$$z_d(t) = \begin{cases} 1 - 0.7e^{-0.1t^3}\ [\text{m}], & t \leq 5, \\ 1\ [\text{m}], & t > 5, \end{cases}$$
$$\psi_d(t) = 0.0\ [^\circ]. \tag{56}$$

The results of the trajectory tracking of (56) are shown in a tridimensional fashion in Fig. 5. It is worth mentioning that the TLMBC, NNSMC, and DTLANNC schemes were implemented with $m = \hat{m} = 1.253$ [kg], which represents 70% of the quadrotor nominal mass presented in Table 1. The reason for this is to evaluate the capabilities of the tested controllers to overcome the uncertainty of such a crucial parameter for the overall performance as mass is. Notice that the value of the mass plays a vital role since, to takeoff, a force greater than the vehicle's weight is needed, and to hovering, a force equal to its weight is needed; given that the force to perform both comes from the total thrust which is the summation of the individual thrust of each actuator, a mismatch in this parameter will affect the overall behavior of the quadrotor significantly. As can be seen, all the controllers achieve the specified trajectory tracking task. Nonetheless, it can be readily appreciated that the closest to the desired trajectory (black line) path was obtained by the DTLANNC (red line). In Figs. 6 and 7, the tracking of the position and attitude signals obtained with the TLMBC, NNSMC and DTLANNC schemes are depicted. As can be appreciated, the response obtained by using the DTLANNC scheme is the fastest reaching the desired signals, especially in the z-axis. That is due to the incorporation

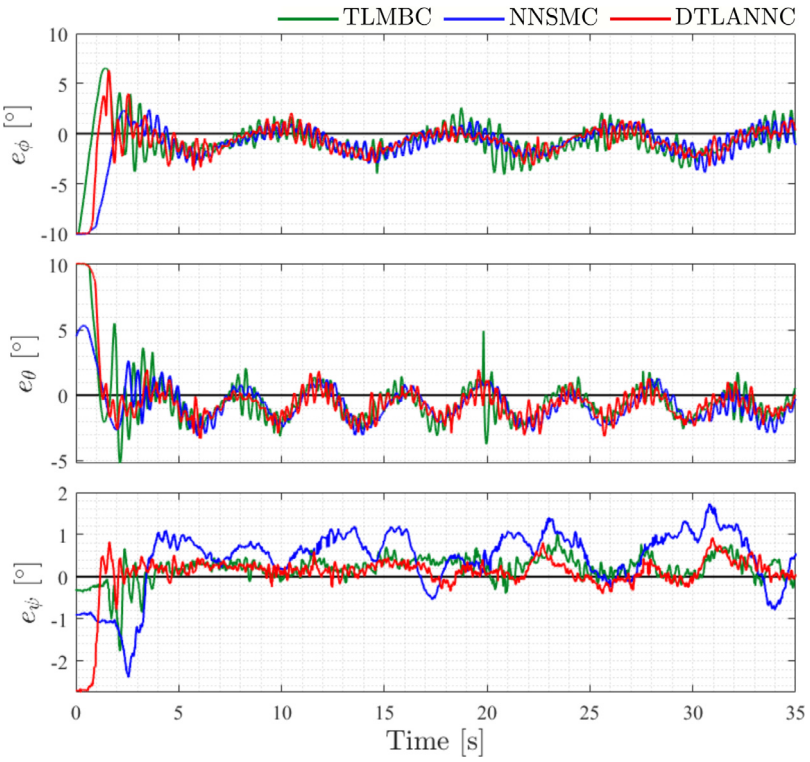

Fig. 9. Time evolution of the attitude error $\boldsymbol{e}_\eta(t) = [e_\phi(t)\ e_\theta(t)\ e_\psi(t)]^{\mathsf{T}}$ by implementing the TLMBC, NNSMC, and DTLANNC schemes in real-time experiments performing the trajectory tracking task defined in (56).

of an estimated value of the quadrotor's mass explicitly in the structure of the DTLANNC and also, the estimation and compensation of the parametric uncertainties and unmodeled dynamics made by the ANNs. Fig. 8 depicts position errors, showing which scheme remains closer to zero, being the performance using the DTLANNC scheme the best. Besides, note that the position errors converge to zero the fastest with the DTLANNC, resulting from the ANN's operation in the position and attitude dynamics.

Fig. 9 presents the time evolution of the attitude errors. As can be seen, all the schemes produce oscillations around zero. In general, the closest oscillations to zero are obtained with the DTLANNC scheme. The total thrust $F(t)$ and the torque vector $\boldsymbol{\tau}(t)$, which corresponds to the control inputs, are depicted in Fig. 10. All the signals are very similar to each other during the trajectory tracking task, except for the total thrust $F(t)$ provided by the DTLANNC at the first 5 seconds, the period when the neural network starts to adapt the output weights. Furthermore, the amplitudes of the oscillations of the torques provided by the DTLANNC are smaller than those provided by the TLMBC and quite similar to those from the NNSMC; indicating a more smooth and stable flight than the obtained with the TLMBC. Additionally, the RMS (root mean square) value of the position and attitude errors were computed in order to provide a quantitative performance index. The RMS values were computed in the time interval 15 [s] $\leq t \leq$ 35 [s], when the steady-state was reached. These results are presented

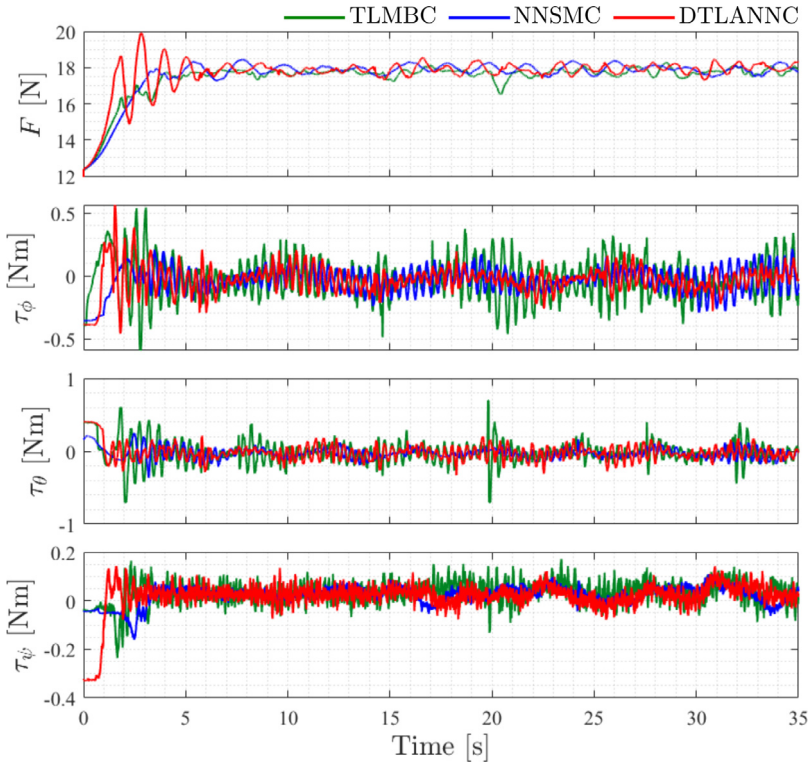

Fig. 10. Time evolution of the control actions $F(t)$ and $\boldsymbol{\tau}(t) = [\tau_\phi(t)\ \tau_\theta(t)\ \tau_\psi(t)]^{\mathsf{T}}$ computed by the TLMBC, NNSMC, and DTLANNC schemes in real-time experiments performing the trajectory tracking task defined in (56).

Table 2
RMS values of position and attitude errors $\boldsymbol{e}_p(t)$ and $\boldsymbol{e}_\eta(t)$ obtained by implementing the TLMBC, NNSMC and DTLANNC schemes during the trajectory tracking task in the experimental tests.

Signal	TLMBC	NNSMC	P_{imp}%	DTLANNC	P_{imp}%
e_x [m]	0.0842	0.0731	13.18	0.0448	46.79
e_y [m]	0.0687	0.0580	15.57	0.0216	68.56
e_z [m]	0.0282	0.0342	-21.28	0.0238	15.60
e_ϕ [°]	1.4972	1.2835	14.27	1.1505	23.16
e_θ [°]	1.2705	1.3375	-5.27	1.1638	8.40
e_ψ [°]	0.3655	0.7915	-116.55	0.2834	22.46

in Table 2 accompanied by their respective relative percentage improvement computed as

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

where ζ represents the RMS values obtained with the NNSMC or the DTLANNC schemes as corresponds. From Table 2 can be observed that both neural network-based schemes provided an improvement with respect to the TLMBC in the (x, y) plane. The performance of the NNSMC in the z-axis is the worst and can be attributed to the lack of integral action in its structure. In general, the smallest RMS error values were obtained with the proposed

scheme providing an improvement of 8.40% up to 68.56% with respect to the TLMBC scheme. Remember that the performance of the TLMBC is strongly coupled with an accurate knowledge of the system parameters and the same for the NNSMC scheme but in a minor way. Our proposed scheme is specially designed to work without any knowledge of inertial parameters and with a considerable error of the nominal mass value. Based on these results, the robustness to parameter uncertainty of the DTLANNC scheme is demonstrated, and its effectiveness for trajectory tracking tasks is validated.

5. Conclusions

This paper presented the development and experimental validation of a novel double two-loop nonlinear controller based on adaptive neural networks (ANNs) for a quadrotor. The introduced controller was formed by a position and an attitude controller, both two-loop schemes, i.e., each controller has a two-loop structure. In such a two-loop structure, the outer loops generate velocity commands to feed the inner loops, being these last ANNs-based schemes. Since the position and attitude controllers form a two-loop scheme on their own and each controller has a two-loop structure with ANNs, the overall structure resulted in a double two-loop ANN controller. The design of the proposed controller took into account the aerodynamic drag and the presence of disturbances in the quadrotor dynamics model. The position and attitude error convergence analysis of the closed-loop system was presented and the control goal was theoretically guaranteed. An experimental comparison against an adaptive neural network-based controller and a model-based controller obtained from the same error dynamics and a similar structure than the proposed scheme was presented. The obtained results demonstrated the functionality and robustness of the proposed scheme. The capabilities of the proposed scheme to deal with parameter uncertainties were probed, and it turned to be effective for trajectory tracking tasks without accurate knowledge of the system parameters. In future work, the effect of input and output weights update for a double two-loop ANN-based controller can be analyzed. Besides, the application of the proposed controller can be studied when payloads are included and actuator faults are presented.

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.

References

[1] R. Austin, Unmanned Aircraft Systems: UAVS Design, in: development and deployment, volume 54, John Wiley & Sons, 2011.

[2] S.R. Nekoo, J.A. Acosta, G. Heredia, A. Ollero, A benchmark mechatronics platform to assess the inspection around pipes with variable pitch quadrotor for industrial sites, Mechatronics 79 (102641) (2021) 1–9.

[3] S.A.S. Rasul, J.S. Narendra, D.Y. Sakhare, Yield Estimation and Drought Monitoring through Image Processing Using MATLAB, in: S. Satapathy, V. Bhateja, M. Favorskaya, T. Adilakshmi (Eds.), Vol. 225 of Smart Innovation, Systems and Technologies, Springer, Singapore, 2021, pp. 525–536.

[4] A. Kourani, N. Daher, A Tethered Quadrotor UAV−Buoy System for Marine Locomotion, in: Proc. IEEE Int. Conf. on Robotics and Automation, 2021, pp. 59–65. Xi'an, China

[5] M.-G. Seo, H.-S. Shin, A. Tsourdos, Soil moisture retrieval model design with multispectral and infrared images from unmanned aerial vehicles using convolutional neural network, Agronomy 11 (2) (2021) 398–412.

[6] H.A. Foudeh, P.C.-K. Luk, J.F. Whidborne, An advanced unmanned aerial vehicle (UAV) approach via learning-based control for overhead power line monitoring: acomprehensive review, IEEE Access 9 (2021) 130410–130433.
[7] T.A. Lasky, T.C. Hsia, On Force-tracking Impedance Control of Robot Manipulators, in: Proc. IEEE Int. Conf. on Robotics and Automation, 1991, pp. 274–275. Sacramento, CA, USA
[8] B. Yao, S. Chan, D. Wang, Variable structure adaptive motion and force control of robot manipulators, Automatica 30 (9) (1994) 1473–1477.
[9] I. Soto, R. Campa, Two-loop control of redundant manipulators: analysis and experiments on a 3-DOF planar arm, Int. J. Adv. Robot. Syst. 10 (1) (2013) 85–92.
[10] C.-C. Tsai, M.-H. Juang, C. KChan, C.-W. Liao, S.J. Chan, Self-balancing and Position Control Using Multi-loop Approach for Ball Robots, in: Proc. Int. Conf. on System Science and Engineering, 2010, pp. 251–256. Yichang, Hubei, China
[11] A. Saradagi, V. Muralidharan, V. Krishnan, S. Menta, A.D. Mahindrakar, Formation control and trajectory tracking of nonholonomic mobile robots, IEEE Trans. Control Syst. Technol. 26 (6) (2017) 2250–2258.
[12] R. Prakash, L. Behera, S. Mohan, S. Jagannathan, Dual-loop optimal control of a robot manipulator and its application in warehouse automation, IEEE Trans. Autom. Sci. Eng. 19 (1) (2022) 262–279.
[13] M. Aicardi, A. Caiti, G. Cannata, G. Casalino, Stability and Robustness Analysis of a Two Layered Hierarchical Architecture for the Closed Loop Control of Robots in the Operational Space, in: Proc. IEEE Int. Conf. on Robotics and Automation, volume 3, 1995, pp. 2771–2778. Nagoya, Japan
[14] R. Kelly, J. Moreno-Valenzuela, F. Pérez, Manipulator Motion Control in Operational Space Using Joint Velocity Inner Loops, in: Proc. 40th IEEE Conf. on Decision and Control, volume 3, 2001, pp. 2416–2417. Orlando, FL, USA
[15] R. Kelly, J. Moreno-Valenzuela, Manipulator motion control in operational space using joint velocity inner loops, Automatica 41 (8) (2005) 1423–1432.
[16] L. Wang, T. Chai, Z. Fang, Neural-network-based two-loop control of robotic manipulators including actuator dynamics in task space, J. Control Theory Appl. 7 (2) (2009) 112–118.
[17] J. Moreno-Valenzuela, L. González-Hernández, Operational space trajectory tracking control of robot manipulators endowed with a primary controller of synthetic joint velocity, ISA Trans. 50 (1) (2011) 131–140.
[18] L.V. Santana, A.S. Brandão, M. Sarcinelli-Filho, Navigation and cooperative control using the A R. Drone quadrotor, J. Intell. Robot. Syst. 84 (1) (2016) 327–350.
[19] M.C. Santos, M. Sarcinelli-Filho, R. Carelli, Trajectory Tracking for UAV with Saturation of Velocities, in: Proc. Int. Conf. on Unmanned Aircraft Systems, 2016, pp. 643–648.
[20] C.D. Rosales, S.R. Tosetti, C.M. Soria, F.G. Rossomando, Neural adaptive PID control of a quadrotor using EFK, IEEE Lat. Am. Trans. 16 (11) (2018) 2722–2730.
[21] J.A. Sarapura, F. Roberti, J.M. Toibero, J.M. Sebastián, R. Carelli, Visual Servo Controllers for an UAV Tracking Vegetal Paths, in: Machine Vision and Navigation, Springer, Cham, Switzerland, 2020, pp. 597–625.
[22] I. Lopez-Sanchez, 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.
[23] A. Noordin, M.M. Basri, Z. Mohamed, A.Z. Abidin, Modelling and PSO fine-tuned PID control of quadrotor UAV, int. journal on advanced science, Eng. Inf. Technol. 7 (4) (2017) 1367–1373.
[24] A. Kaba, Improved PID rate control of a quadrotor with a convexity-based surrogated model, Aircraft Eng. Aerospace Technol. 93 (8) (2021) 1287–1301.
[25] A.A. Ghaffar, T. Richardson, Model reference adaptive control and LQR control for quadrotor with parametric uncertainties, Int. J. Mech. Mechatron. Eng. 9 (2) (2015) 244–250.
[26] A. Noormohammadi-Asl, O. Esrafilian, M.A. Arzati, H.D. Taghirad, System identification and h∞-based control of quadrotor attitude, Mech. Syst. Signal Process. 135 (106358) (2020) 1–16.
[27] D. Ma, Y. Xia, G. Shen, Z. Jia, T. Li, Flatness-based adaptive sliding mode tracking control for a quadrotor with disturbances, J. Franklin Inst. 355 (14) (2018) 6300–6322.
[28] H. Maqsood, Y. Qu, Nonlinear disturbance observer based sliding mode control of quadrotor helicopter, J. Electric. Eng. Technol. 15 (3) (2020) 1453–1461.
[29] L. Martins, C. Cardeira, P. Oliveira, Feedback linearization with zero dynamics stabilization for quadrotor control, J. Intell. Robotic Syst. 101 (1) (2021) 1–17.
[30] J. Song, D.E. Chang, Y. Eun, Passivity-based Adaptive Control of Quadrotors with Mass and Moment of Inertia Uncertainties, in: Proc. IEEE 58th Conf. on Decision and Control, Nice, France, 2019, pp. 90–95.

[31] R. Pérez-Alcocer, J. Moreno-Valenzuela, Adaptive control for quadrotor trajectory tracking with accurate parametrization, IEEE Access 7 (2019) 53236–53247.
[32] 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.
[33] J.F. Guerrero-Castellanos, S. Durand, G.A. Munoz-Hernandez, N. Marchand, L.L.G. Romeo, J. Linares-Flores, G.M. Aguilar, W.F. Guerrero-Sánchez, Bounded attitude control with active disturbance rejection capabilities for multirotor UAVs, Appl. Sci. 11 (13) (2021) 5960–5980.
[34] M. Labbadi, M. Cherkaoui, Robust adaptive backstepping fast terminal sliding mode controller for uncertain quadrotor UAV, Aerosp. Sci. Technol. 93 (2019) 105306.
[35] K. Liu, R. Wang, Antisaturation command filtered backstepping control based disturbance rejection for a quadarotor UAV, IEEE Trans. Circuits Syst. II: Express Briefs 68 (12) (2021) 3577–3581.
[36] M. Labbadi, M. Cherkaoui, Robust adaptive nonsingular fast terminal sliding-mode tracking control for an uncertain quadrotor UAV subjected to disturbances, ISA Trans. 99 (2020) 290–304.
[37] G. Wang, W. Yang, N. Zhao, Y. Shen, C. Wang, An approximation-free simple controller for uncertain quadrotor systems in the presence of thrust saturation, Mechatronics 72 (102450) (2020) 1–9.
[38] G.C. Lopes, M. Ferreira, A.d. S. Simões, E.L. Colombini, Intelligent Control of a Quadrotor with Proximal Policy Optimization Reinforcement Learning, in: 2018 Latin American Robotic Symp., 2018, pp. 503–508.
[39] J. Dong, B. He, Novel fuzzy PID-type iterative learning control for quadrotor UAV, Sensors 19 (1 (24)) (2019) 1–11.
[40] X.-Z. Jin, T. He, X.-M. Wu, H. Wang, J. Chi, Robust adaptive neural network-based compensation control of a class of quadrotor aircrafts, J. Franklin Inst. 357 (17) (2020) 12241–12263.
[41] A. Al-Mahturi, F. Santoso, M.A. Garratt, S.G. Anavatti, Self-learning in aerial robotics using type-2 fuzzy systems: case study in hovering quadrotor flight control, IEEE Access 9 (2021) 119520–119532.
[42] C.-H. Pi, W.-Y. Ye, S. Cheng, Robust quadrotor control through reinforcement learning with disturbance compensation, Appl. Sci. 11 (7 (3257)) (2021) 1–13.
[43] O. Bingöl, H. Güzey, Neuro sliding mode control of quadrotor UAVs carrying suspended payload, Adv. Rob. 35 (3–4) (2021) 255–266.
[44] X. Guan, S. Lou, H. Li, T. Tang, Intelligent control of quad-rotor aircrafts with a STM32 microcontroller using deep neural networks, Ind. Robot: Int. J. Robot. Res. Appl. 48 (5) (2021) 700–709.
[45] H. Lim, T. Lee, T. Low, An investigation into neural net control systems with integral action, 1991, Singapore. Proc. IEEE Int. Joint Conf. on Neural Networks, 1053–1058
[46] I.S. Baruch, R. Garrido, A direct adaptive neural control scheme with integral terms, Int. J. Intell. Syst. 20 (2) (2005) 213–224.
[47] C.-H. Lu, C.C. Tsai, Adaptive predictive control with recurrent neural network for industrial processes: an application to temperature control of a variable-frequency oil-cooling machine, IEEE Trans. Ind. Electron. 55 (3) (2008) 1366–1375.
[48] M. Bozic, P. Maric, I. Krcmar, An Adaptive Internal Model-based Neural Controller with Embedded Integral Action, in: Proc. ETRAN Int. Conf. on Electrical, Electronics and Computing Engineering, 2017, pp. 1–6.
[49] X. Yin, Z. Jiang, L. Pan, Recurrent neural network based adaptive integral sliding mode power maximization control for wind power systems, Renew. Energy 145 (2020) 1149–1157.
[50] S.C. Yogi, V.K. Tripathi, L. Behera, Adaptive integral sliding mode control using fully connected recurrent neural network for position and attitude control of quadrotor, IEEE Trans. Neural Netw. Learn. Syst. 32 (12) (2021) 5595–5609.
[51] L. Guettal, A. Chelihi, R. Ajgou, M.M. Touba, Robust tracking control for quadrotor with unknown nonlinear dynamics using adaptive neural network based fractional-order backstepping control, J. Franklin Inst. 359 (14) (2022) 7337–7364.
[52] J. Wang, P. Wang, B. Tian, Hyperbolic tangent function-based fixed-time event-triggered control for quadrotor aircraft with prescribed performance, J. Franklin Inst. 359 (12) (2022) 6267–6285.
[53] M. Alfayizi, Y. Shtessel, C. Edwards, Quad-rotor adaptive sliding mode control using only position and yaw sensors: generalized relative degree approach, J. Franklin Inst. 359 (1) (2022) 492–519.
[54] C. Cao, C. Wei, Y. Liao, Y. Zhang, J. Li, On novel trajectory tracking control of quadrotor UAV: a finite-time guaranteed performance approach, J. Franklin Inst. (2022).
[55] P. Lu, M. Liu, X. Zhang, G. Zhu, Z. Li, C.Y. Su, Neural network based adaptive event-triggered control for quadrotor unmanned aircraft robotics, Machines 10 (8) (2022) 617.
[56] S. Huang, Y. Yang, Adaptive neural-network-based nonsingular fast terminal sliding mode control for a quadrotor with dynamic uncertainty, Drones 6 (8) (2022) 206.

[57] J. Dunfied, M. Tarbouchi, G. Labonte, Neural Network Based Control of a Four Rotor Helicopter, in: Proc. 2004 IEEE Int. Conf. on Industrial Technology, 3, pp. 1543–1548.
[58] H. Voos, Nonlinear and Neural Network-based Control of a Small Four-rotor Aerial Robot, in: 2007 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, pp. 1–6.
[59] T. Dierks, S. Jagannathan, Neural Network Control of Quadrotor UAV Formations, in: 2009 American Control Conf., 2009, pp. 2990–2996.
[60] M.O. Efe, Neural network assisted computationally simple PID control of a quadrotor uav, IEEE Trans. Ind. Inf. 7 (2) (2011) 354–361.
[61] D. Nodland, H. Zargarzadeh, S. Jagannathan, Neural network-based optimal adaptive output feedback control of a helicopter UAV, IEEE Trans. Neural Netw. Learn. Syst. 24 (7) (2013) 1061–1073.
[62] Y.-m. Chen, Y.-l. He, M.f. Zhou, Decentralized PID neural network control for a quadrotor helicopter subjected to wind disturbance, J. Central South Univ. 22 (1) (2015) 168–179.
[63] T. Matassini, H.-S. Shin, A. Tsourdos, M. Innocenti, Adaptive control with neural networks-based disturbance observer for a spherical UAV, IFAC-PapersOnLine 49 (17) (2016) 308–313.
[64] 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.
[65] R.-J. Wai, A.S. Prasetia, Adaptive neural network control and optimal path planning of UAV surveillance system with energy consumption prediction, IEEE Access 7 (2019) 126137–126153.
[66] I. Lopez-Sanchez, J. Moyrón, J. Moreno-Valenzuela, Adaptive neural network-based trajectory tracking outer loop control for a quadrotor, Aerosp. Sci. Technol. 129 (107847) (2022) 1–18.
[67] X. Jing, H. Gao, Y. Wang, Z. Chen, Cooperative compliance control of the dual-arm manipulators with elastic joints, J. Mech. Sci. Technol. 35 (12) (2021) 5689–5697.
[68] Y. Lin, Z.A.F.T. Chen, B. Yao, Unified motion/force/impedance control for manipulators in unknown contact environments based on robust model-reaching approach, IEEE/ASME Trans. Mechatron. 26 (4) (2021) 1905–1913.
[69] J. Alvarez-Ramirez, I. Cervantes, G. Espinosa-Perez, P. Maya, A. Morales, A stable design of PI control for DC-DC converters with an RHS zero, IEEE Trans. Circuit. Syst. I: Fund. Theory Appl. 48 (1) (2001) 103–106.
[70] F. Kendoul, Z. Yu, K. Nonami, Guidance and nonlinear control system for autonomous flight of minirotorcraft unmanned aerial vehicles, J. Field Rob. 27 (3) (2010) 311–334.
[71] L.R.G. Carrillo, A.E.D. López, R. Lozano, C. Pégard, Quad rotorcraft control: Vision-based hovering and navigation, Springer-Verlag, London, UK, 2012.
[72] 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.
[73] 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.
[74] J. Moreno-Valenzuela, C. Aguilar-Avelar, Motion Control of Underactuated Mechanical Systems, in: Vol. 88 of Intelligent Systems, Control and Automation: Science and Engineering, Springer International Publishing AG, Cham, Switzerland, 2018.
[75] H. Bouadi, M. Tadjine, Nonlinear observer design and sliding mode control of four rotors helicopter, Int. J. Mech. Aerospace Ind. Mechatron. Manuf. Eng. 1 (7) (2007) 354–359.
[76] H. Bouadi, M. Bouchoucha, M. Tadjine, Sliding mode control based on backstepping approach for an UAV type-quadrotor, World Acad. Sci. Eng. Technol. 26 (5) (2007) 22–27.
[77] T. Luukkonen, Modelling and control of quadcopter (aalto university, 2011, Espoo, Finland, Int. Rep. 22
[78] M. Bangura, R. Mahony, Nonlinear dynamic modeling for high performance control of a quadrotor, Proc. Australasian Conf. Robot. Automat. (2012) 115–124.
[79] L. Bauersfeld, E. Kaufmann, P. Foehn, S. Sun, D. Scaramuzza, NeuroBEM: Hybrid Aerodynamic Quadrotor Model, in: Proc. XVII Robotics: Science and Systems, 2021.
[80] G. Antonelli, F. Arrichiello, S. Chiaverini, P.R. Giordano, Adaptive Trajectory Tracking for Quadrotor MAVSin Presence of Parameter Uncertainties and External Disturbances, in: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Wollongong, NSW, Australia, 2013, pp. 1337–1342.
[81] X. Liang, Y. Fang, N. Sun, A Hierarchical Controller for Quadrotor Unmanned Aerial Vehicle Transportation Systems, in: 35th Chinese Control Conference, 2016, pp. 6148–6153. Chengdu, China
[82] G. Antonelli, E. Cataldi, F. Arrichiello, P.R. Giordano, S. Chiaverini, A. Franchi, Adaptive trajectory tracking for quadrotor MAVs in presence of parameter uncertainties and external disturbances, IEEE Trans. Control Syst. Technol. 26 (1) (2017) 248–254.
[83] C. Trapiello, V. Puig, B. Morcego, Position-heading quadrotor control using LPV techniques, IET Control Theory Appl. 13 (6) (2019) 783–794.

[84] 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.
[85] F.L. Lewis, S. Jagannathan, A. Yesildirak, Neural network control of robot manipulators and non-linear systems, CRC press, Philadelphia, PA, USA, 1999.
[86] F.L. Lewis, J. Campos, R. Selmic, Neuro-fuzzy Control of Industrial Systems with Actuator Nonlinearities, in: Society for Industrial and Applied Mathematics, 2002. Philadelphia, PA, USA
[87] H.K. Khalil, Nonlinear systems, 3, Prentice Hall, Upper Saddle River, NJ, USA, 2002.
[88] Y. Fan, Y. Cao, T. Li, Adaptive Integral Backstepping Control for Trajectory Tracking of a Quadrotor, in: Proc. 4th Int. Conf. on Information, Cybernetics and Computational Social Systems, 2017, pp. 619–624.
[89] F. Jurado, S. Lopez, A wavelet neural control scheme for a quadrotor unmanned aerial vehicle, Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci. 376 (2126) (2018) 1–21.
[90] C. Hua, J. Chen, X. Guan, Fractional-order sliding mode control of uncertain QUAVs with time-varying state constraints, Nonlinear Dyn. 95 (2) (2019) 1347–1360.
[91] M. Chen, S. Xiong, Q. Wu, Tracking flight control of quadrotor based on disturbance observer, IEEE Trans. Syst. Man Cybern. 51 (3) (2019) 1414–1423.
[92] I. Lopez-Sanchez, 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.
[93] A. Kourani, N. Daher, Leveraging PID Gain Selection towards Adaptive Backstepping Control for a Class of Second-order Systems, in: Proc. American Control Conf., 2021, pp. 1174–1179.
[94] I. Quanser, Markham, 2015, Canada, User Manual Qball 2 for QUARC, Set Up and Configuration.
[95] S. Li, Y. Wang, J. Tan, Y. Zheng, Adaptive RBFNNs/integral sliding mode control for a quadrotor aircraft, Neurocomputing 216 (2016) 126–134.