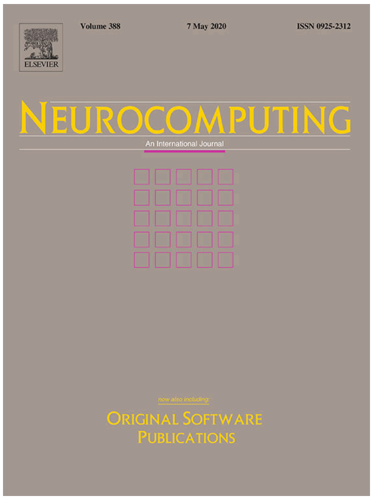


Adaptive Trajectory Tracking Control for Quadrotors with Disturbances by Using Generalized Regression Neural Networks

Ivan Lopez-Sanchez, Francisco Rossomando, Ricardo Pérez-Alcocer, Carlos Soria, Ricardo Carelli, Javier Moreno-Valenzuela

PII: S0925-2312(21)01009-2
DOI: https://doi.org/10.1016/j.neucom.2021.06.079
Reference: NEUCOM 24032

To appear in: *Neurocomputing*

Received Date: 14 October 2020
Accepted Date: 25 June 2021

Adaptive Trajectory Tracking Control for Quadrotors with Disturbances by Using Generalized Regression Neural Networks[⋆]

Ivan Lopez-Sanchez[a], Francisco Rossomando[b], Ricardo Pérez-Alcocer[c], Carlos Soria[b], Ricardo Carelli[b], Javier Moreno-Valenzuela[a,*]

[a] *Instituto Politécnico Nacional-CITEDI, Tijuana, Baja California 22435, México.*
[b] *Universidad Nacional de San Juan-INAUT, San Juan 1109, Argentina.*
[c] *CONACYT-Instituto Politécnico Nacional-CITEDI, Tijuana, Baja California 22435, México.*

Abstract

In this document, the development and experimental validation of a nonlinear controller with an adaptive disturbance compensation system applied on a quadrotor are presented. The introduced scheme relies on a generalized regression neural network (GRNN). The proposed scheme has a structure consisting of an inner control loop inaccessible to the user (*i.e.*, an embedded controller) and an outer control loop which generates commands for the inner control loop. The adaptive GRNN is applied in the outer control loop. The proposed approach lies in the aptitude of the GRNN to estimate the disturbances and unmodeled dynamic effects without requiring accurate knowledge of the quadrotor parameters. The adaptation laws are deduced from a rigorous convergence analysis ensuring asymptotic trajectory tracking. The proposed control scheme is implemented on the QBall 2 quadrotor. Comparisons with respect to a PD-based control, an adaptive model regressor-based scheme, and an adaptive neural-network controller are carried out. The experimental results validate the functionality of the novel control scheme and show a performance improvement since smaller tracking error values are produced.

Keywords: Generalized regression neural network, adaptive control, quadrotor, real-time experiments.

[⋆] This work was supported in part by the Consejo Nacional de Ciencia y Tecnología, CONACYT–Fondo Sectorial de Investigación para la Educación under Project A1-S-24762, and in part by Secretaría de Investigación y Posgrado-Instituto Politécnico Nacional, México. Proyecto Apoyado por el Fondo Sectorial de Investigación para la Educación.

[*] Corresponding author

Email addresses: ilopez@citedi.mx (Ivan Lopez-Sanchez), frosoma@inaut.unsj.edu.ar (Francisco Rossomando), rrperez@citedi.mx (Ricardo Pérez-Alcocer), csoria@inaut.unsj.edu.ar (Carlos Soria), rcarelli@inaut.unsj.edu.ar (Ricardo Carelli), moreno@citedi.mx (Javier Moreno-Valenzuela)

1. Introduction

Since their first appearance in the 1960s, rotary-wing unmanned aerial vehicles have changed considerably, from their geometric configuration, materials, instrumentation to their applications. They were developed for military purposes at the beginning, and with the pass of time, they became useful tools in many different fields [1]. An unmanned aerial vehicle (UAV) is an aircraft with no flight crew, controlled autonomously, or by a pilot from a control station by using pre-programmed flight plans, which in both cases imply using control algorithms. The rotary-wing UAVs have significant advantages over other aerial systems since they allow vertical takeoff and landing, hovering flight, better control of stability when slow trajectories are commanded, simpler design, and easy maintenance [2]. Nevertheless, one of the major challenges of aerial vehicles is to ensure stability and maneuverability under adverse flight conditions [3, 4, 5]. Of the different kinds of rotary-wing UAVs, the most common are those propelled by four rotors and so-called quadrotors. Nowadays, quadrotors are so popular that they appear in many movies and videogames.

In order to maintain accuracy under the desired conditions, the UAV controllers must be robust to the different external disturbances or unmodeled dynamics. In the literature review, there is a significant number of techniques for the suppression of disturbances, among which we can highlight the techniques based on observers [6, 7], on model-based control [8, 9, 10, 11], on adaptive schemes [12, 13, 14] and based on artificial intelligence [15, 16, 17, 18].

On the different approaches of control, adaptive schemes represent a feasible option when dealing with model uncertainties or parameter variations during the platform operation. Besides, for some controllers, the basis of the approach lies in the parameterization of the dynamic model, as can be seen in [14]. With the advance of adaptive control, new approaches raised. For example, [19] introduced an approach based on barrier Lyapunov functions applied to switched nonlinear systems with constraints. With the rise of intelligent control, adaptive schemes with the combination of fuzzy control have been proposed. In [20], an adaptive fuzzy fault-tolerant control based on barrier Lyapunov function was introduced for a switched system. A singularity-free adaptive fuzzy fixed-time control algorithm was developed in [21]. Nevertheless, an advantage of our approach over the mentioned schemes is the capability of the neural network to handle parameter uncertainties and compensate for external disturbances.

A disturbance rejection controller that reduces disturbances affecting the trajectory tracking task was described in [8]. In [10], an active disturbance rejection controller was proposed by combining the nonlinear feedback control and an extended state observer. Another work related to disturbance rejection was presented in [13], where an adaptive sliding mode control for UAVs was developed. In [12], the robust integral of the signum of the error and the immersion and invariance techniques were used to control a quadrotor. Another technique that uses a combination of linear and adaptive control was presented in [17].

Different works aiming to solve the quadrotor control problem by using embedded sensors, vision systems, and on-board cameras have been presented

[2, 14, 22]. However, depending on the quadrotor equipment, only a limited number of data acquisition devices are available. Thus, to supply the need for more sensors or equipment and to provide disturbance rejection control alternatives, some estimation methods are implemented, being the neural networks an alternative to achieve this goal.

Neural networks have been successfully applied in control. This approach is especially useful for nonlinear systems where the advantage of the universal function approximation property of the neural networks is exploited. In [23], the switched system control problem was addressed by means of an online learning radial basis neural network (RBFNN), and the simulation results showed the functionality of the proposed scheme. The fixed-time control problem of nonstrict-feedback nonlinear system subjected to deadzone and output constraint was studied in [24] by using a combination of Barrier Lyapunov function, an online learning RBFNN, and the backstepping scheme. A recurrent neural network fractional-order sliding mode control was proposed in [25], where the performance of the proposed approach was validated in simulation by realizing the current harmonic compensation control for an active power filter.

Neural networks have been presented in the literature to address the control of UAVs. An intelligent controller based on a pre-trained neural network was developed and implemented in [26] to control the Dragon Flyer 2 quadrotor. In [27], a combination of a state-dependent Riccati equation control scheme and a pre-trained neural network was introduced to control and stabilize a small quadrotor. A leader-follower formation control based on spherical coordinates and neural networks was developed in [28] to control multiple quadrotors. An optimal controller based on the backstepping technique and a neural network was proposed to address the trajectory tracking control of a helicopter UAV in [29]. The results of the numerical simulation were satisfactory. In [30], an indirect adaptive neural controller was developed for a quadrotor to pursue a moving object. A PID controller with a sigma-pi neural network was developed to control a quadrotor in [31]. Simulation and experiments were presented to show the performance improvement by using the neural network. A controller based on the backstepping and sliding mode techniques together with a radial basis function neural network was introduced in [32] to address the position regulation problem. An adaptive control scheme using a radial basis function neural network was proposed in [33] to control a quadrotor while is transporting a payload suspended on a cable. In [34], an online adaptive neural network-based controller was designed to ensure stability and to provide disturbance rejection for a quadrotor that tracks an optimized trajectory.

As described in [35, 36, 37, 38, 39], the generalized regression neural network (GRNN) is a single-pass neural network with a high degree of parallelism structure. It can be used to solve regression or estimation problems where it cannot be assumed that the system is linear. A very useful application for this neural network is to estimate the dynamics of a plant for control purposes [36]. Besides control, different applications for this neural network have also been found in image processing, the estimation of energy consumption, or even fault diagnosis. In [37], a GRNN was implemented with a genetic optimization algorithm to identify tri-dimensional objects from the object bi-dimensional poses and to recognize handwritten digits. An optimization algorithm based on a fruit fly swarm and a GRNN was used to model and forecast the annual energy load of a region in [38]. This neural network has also been successfully implemented for UAV control. In [39], a pre-trained GRNN was used to control the altitude of a flapping wing UAV. The GRNN was implemented with a particle swarm optimization algorithm to diagnose accurately the unbalance fault of the rotor of a UAV helicopter [40].

Many quadrotors are equipped with an autopilot, which takes care of stabilizing the vehicle dynamics. Thus, the commands that the autopilot receives may be interpreted as control inputs for the quadrotor. The dynamics resulting from the quadrotor and the embedded autopilot is a model having four inputs and four outputs, which may be interpreted as a simplified quadrotor model, see the manuscripts [41] and [42]. In order to stabilize the quadrotor, an external control loop should be designed to generate input commands for the autopilot.

This work introduces a robust controller using a GRNN to address the trajectory tracking problem providing effective disturbances rejection. The proposed scheme has a structure consisting of an embedded autopilot on the quadrotor and of an outer control loop that computes commands for the embedded controller. The resulting system has as inputs the commands of normalized inclinations and velocities and as outputs the quadrotor position and yaw angle. The practical viability of the proposed scheme is supported experimentally demonstrating its capabilities and behavior by tracking two different desired trajectories. Besides, comparisons with other control techniques are given. The novelty of this work mainly relies on the following two points:

- The development of a neural network-based controller by using GRNNs on which the output weight matrix as well as the center and standard deviation vectors are online updated.
- An exhaustive real-time experimental study, where the proposed scheme is compared with other controllers.

The proposed control scheme has a structure consisting of an inner control loop, which is assumed to be an embedded controller, and an outer loop, which generates commands for the inner loop. In particular, the outer loop takes advantage of the adaptive GRNNs. Through this interaction, the trajectory tracking task is achieved. The implemented algorithms for the experimental study consist of the embedded controller plus a proportional-derivative (PD) scheme as an outer loop, an adaptive model regressor controller, and an adaptive neural network scheme. The experimental results indicate that the proposed controller presents the best tracking accuracy.

The paper is organized as follows: Section 2 shows the quadrotor dynamics, the embedded autopilot controller, and the resulting closed-loop model, which is a novel input-output representation of the quadrotor. An overview of the GRNN, the proposed adaptive controller, and the adaptation laws are presented in Section 3. Finally, experimental results and conclusions are given in Section 4 and 5, respectively.

2. Quadrotor dynamic model

The six degrees-of-freedom quadrotor dynamic model represented in the inertial reference frame as described in [43, 44] is given by

$$\begin{aligned} m\ddot{\boldsymbol{p}} + mg\boldsymbol{e}_z + D_p(\boldsymbol{\eta})\dot{\boldsymbol{p}} &= R(\boldsymbol{\eta})\boldsymbol{e}_z F, && (1)\\ M(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + C(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + D_\eta(\boldsymbol{\eta})\dot{\boldsymbol{\eta}} &= W(\boldsymbol{\eta})^{-T}\boldsymbol{\tau}, && (2)\end{aligned}$$

where the equation (1) represents the position dynamics and the equation (2) represents the attitude dynamics, $m \in \mathbb{R}$ is the mass of the vehicle, $g \in \mathbb{R}$ is the gravitational acceleration constant, $\boldsymbol{p} = [x\ y\ z]^T \in \mathbb{R}^3$ is the quadrotor position, $\boldsymbol{\eta} = [\phi\ \theta\ \psi]^T \in \mathbb{R}^3$ is the quadrotor attitude, both expressed in the inertial reference frame, $\boldsymbol{e}_z = [0\ 0\ 1]^T \in \mathbb{R}^3$ is a unitary vector along the z-axis in the inertial reference frame, $D_p(\boldsymbol{\eta}) \in \mathbb{R}^{3\times3}$ is the aerodynamic drag matrix, $D_\eta(\boldsymbol{\eta}) \in \mathbb{R}^{3\times3}$ is a positive definite matrix that models an aerodynamic damping effect, $R(\boldsymbol{\eta}) \in \mathbb{R}^{3\times3}$ is a rotation matrix, $M(\boldsymbol{\eta}) \in \mathbb{R}^{3\times3}$ is the inertia matrix, $C(\boldsymbol{\eta},\dot{\boldsymbol{\eta}}) \in \mathbb{R}^{3\times3}$ is the Coriolis matrix, $W(\boldsymbol{\eta}) \in \mathbb{R}^{3\times3}$ is a transformation matrix, $F \in \mathbb{R}$ and $\boldsymbol{\tau} \in \mathbb{R}^3$ are the control inputs.

The quadrotor model in (1) and (2) does not consider the aerodynamic effects which are present during its operation in outdoor environments. Based on the previous works [45, 46, 47, 48], many different aerodynamic effects can be considered, such as the influence of the angle of attack of the blades of the propellers on the provided thrust of the actuators or the wind-induced drag. Aerodynamic effects are more significant in high-speed flights and on acrobatic maneuvering. The aerodynamic effects considered in this work are the force resulting from the wind-induced drag, represented in the left-hand side of equation (1) by the expression $D_p(\boldsymbol{\eta})\dot{\boldsymbol{p}}$, and the the aerodynamic drag torque, represented in the left-hand side of equation (2) by the term $D_\eta(\boldsymbol{\eta})\dot{\boldsymbol{\eta}}$.

Taking into account the ideas discussed in [49, 50, 51, 52], let us assume that there is an inner embedded controller capable of stabilizing the quadrotor in hover flight. In the works [49, 50, 51, 52], the embedded controller is assumed to be given by

$$\begin{aligned} F &= \frac{m}{c_\phi c_\theta}(g + \dot{z}^*), && (3)\\ \boldsymbol{\tau} &= W(\boldsymbol{\eta})^T[M(\boldsymbol{\eta})\tilde{\boldsymbol{\tau}} + C(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}}], && (4)\end{aligned}$$

where the signals

$$\begin{aligned} \dot{z}^* &= \frac{1}{\tau_{\dot{z}}}(\dot{z}_d - \dot{z}), && (5)\\ \begin{bmatrix}\tilde{\tau}_\phi\\ \tilde{\tau}_\theta\\ \tilde{\tau}_\psi\end{bmatrix} &= \begin{bmatrix}\omega_\phi^2(\phi_d - \phi) - 2\xi_\phi\omega_\phi\dot{\phi}\\ \omega_\theta^2(\theta_d - \theta) - 2\xi_\theta\omega_\theta\dot{\theta}\\ \frac{1}{\tau_{\dot{\psi}}}(\dot{\psi}_d - \dot{\psi})\end{bmatrix}, && (6)\end{aligned}$$

are related to first and second order linear systems as will be seen later, where ω_ϕ and ω_θ are the natural frequencies, ξ_ϕ and ξ_θ are the damping constants, and $\tau_{\dot{z}}$ and $\tau_{\dot{\psi}}$ are time constants for each system.

Notice that the reference signals $\dot{z}_d$, ϕ_d, θ_d, and $\dot{\psi}_d$ are the input commands for the inner embedded controller. Let us consider that input commands for the embedded controller satisfy $|\dot{z}_d| \leq \dot{z}_{\max}$, $|\theta_d| \leq \theta_{\max}$, $|\phi_d| \leq \phi_{\max}$, $|\dot{\psi}_d| \leq \dot{\psi}_{\max}$. Then, the following relationships are established

$$\begin{aligned}
\dot{z}_d &= \dot{z}_{\max} u_{\dot{z}}, &(7)\\
\theta_d &= \theta_{\max} u_\theta, &(8)\\
\phi_d &= \phi_{\max} u_\phi, &(9)\\
\dot{\psi}_d &= \dot{\psi}_{\max} u_{\dot{\psi}}, &(10)
\end{aligned}$$

where $\boldsymbol{u} = [u_\theta \ u_\phi \ u_{\dot{z}} \ u_{\dot{\psi}}]^T \in \mathbb{R}^4$ is the dimensionless and normalized control input vector, being u_θ an angular position control input related to the displacement along the x-axis, u_ϕ an angular position control input related to the displacement along the y-axis, $u_{\dot{z}}$ a velocity control input related to the displacement along the z-axis, and $u_{\dot{\psi}}$ an angular velocity control input related to the rotation around the z-axis, all in the inertial reference frame.

By replacing the expressions (7)-(10) into the equations (5) and (6), the following is obtained:

$$\begin{aligned}
\dot{z}^* &= \frac{\dot{z}_{\max}}{\tau_{\dot{z}}} u_{\dot{z}} - \frac{1}{\tau_{\dot{z}}}\dot{z},\\
\begin{bmatrix} \tilde{\tau}_\phi \\ \tilde{\tau}_\theta \\ \tilde{\tau}_\psi \end{bmatrix} &= \begin{bmatrix} \omega_\phi^2 \phi_{\max} u_\phi - 2\xi_\phi \omega_\phi \dot{\phi} - \omega_\phi^2 \phi, \\ \omega_\theta^2 \theta_{\max} u_\theta - 2\xi_\theta \omega_\theta \dot{\theta} - \omega_\theta^2 \theta, \\ \frac{\dot{\psi}_{\max}}{\tau_{\dot{\psi}}} u_{\dot{\psi}} - \frac{1}{\tau_{\dot{\psi}}}\dot{\psi} \end{bmatrix}.
\end{aligned}$$

Thus, by replacing the equations (3)–(10) into the equations (1) and (2), the closed-loop system resulting from the quadrotor dynamics and embedded controller (also called inner control loop) is

$$\begin{aligned}
\ddot{x} &= \frac{F}{m}(s_\psi s_\phi + c_\psi c_\phi s_\theta) - d_x \dot{x}, &(11)\\
\ddot{y} &= \frac{F}{m}(-c_\psi s_\phi + s_\psi c_\phi s_\theta) - d_y \dot{y}, &(12)\\
\ddot{z} &= \frac{\dot{z}_{\max}}{\tau_{\dot{z}}} u_{\dot{z}} - \left(\frac{1}{\tau_{\dot{z}}} + d_z\right)\dot{z},\\
\ddot{\phi} &= \omega_\phi^2 \phi_{\max} u_\phi - (2\xi_\phi \omega_\phi + d_\phi)\dot{\phi} - \omega_\phi^2 \phi,\\
\ddot{\theta} &= \omega_\theta^2 \theta_{\max} u_\theta - (2\xi_\theta \omega_\theta + d_\theta)\dot{\theta} - \omega_\theta^2 \theta,\\
\ddot{\psi} &= \frac{\dot{\psi}_{\max}}{\tau_{\dot{\psi}}} u_{\dot{\psi}} - \left(\frac{1}{\tau_{\dot{\psi}}} + d_\psi\right)\dot{\psi},
\end{aligned}$$

where the assumptions $\frac{1}{m}D_p(\boldsymbol{\eta}) \approx \operatorname{diag}\{d_x,\ d_y,\ d_z\}$ and $M(\boldsymbol{\eta})^{-1}D_\eta(\boldsymbol{\eta}) \approx$

diag $\{d_\phi,\ d_\theta,\ d_\psi\}$ where used.

Linearizing the equations (11) and (12) around the operation point $\psi =$ constant, $\theta = \phi = 0$ and $F = mg$ corresponding to the hovering flight, leads to

$$\begin{aligned} \ddot{x} &= g(c_\psi \theta + s_\psi \phi) - d_x \dot{x}, && (13) \\ \ddot{y} &= g(s_\psi \theta - c_\psi \phi) - d_y \dot{y}. && (14) \end{aligned}$$

Expressions (13) and (14) denote the relation of the position dynamics in the horizontal plane with the attitude. Adding and subtracting $g(c_\psi \theta_d + s_\psi \phi_d)$ and $g(s_\psi \theta_d - c_\psi \phi_d)$ to the equations (13) and (14), respectively, and defining the attitude error for the pitch and the roll angles as

$$\begin{aligned} \tilde{\theta} &= \theta_d - \theta, \\ \tilde{\phi} &= \phi_d - \phi, \end{aligned}$$

and realizing some algebraic manipulations, the dynamics of the quadrotor under the embedded controller (3)-(4) and (7)-(10) are expressed as

$$\begin{aligned} \ddot{x} &= c_\psi g \theta_{\max} u_\theta + s_\psi g \phi_{\max} u_\phi - d_x \dot{x} + g(c_\psi \tilde{\theta} + s_\psi \tilde{\phi}), && (15) \\ \ddot{y} &= s_\psi g \theta_{\max} u_\theta - c_\psi g \phi_{\max} u_\phi - d_y \dot{y} + g(s_\psi \tilde{\theta} - c_\psi \tilde{\phi}), && (16) \\ \ddot{z} &= \frac{\dot{z}_{\max}}{\tau_{\dot{z}}} u_{\dot{z}} - \left(\frac{1}{\tau_{\dot{z}}} + d_z \right) \dot{z}, && (17) \\ \ddot{\theta} &= -(2\xi_\theta \omega_\theta + d_\theta)\dot{\theta} + \omega_\theta^2 \tilde{\theta}, && (18) \\ \ddot{\phi} &= -(2\xi_\phi \omega_\phi + d_\phi)\dot{\phi} + \omega_\phi^2 \tilde{\phi}, && (19) \\ \ddot{\psi} &= \frac{\dot{\psi}_{\max}}{\tau_{\dot{\psi}}} u_{\dot{\psi}} - \left(\frac{1}{\tau_{\dot{\psi}}} + d_\psi \right) \dot{\psi}. && (20) \end{aligned}$$

Under the assumption that the inner embedded controller (4) stabilizes the quadrotor such that $\tilde{\theta}(t) \approx 0$ and $\tilde{\phi}(t) \approx 0$ for all $t \geq 0$, the system in (15)-(17), (20) can be written in matrix form as in [2, 22, 41, 42, 53, 54, 55, 56, 57]

$$\ddot{\boldsymbol{x}}^w = T(\psi) K_u \boldsymbol{u} - K_v \dot{\boldsymbol{x}}^w, \qquad (21)$$

where $\boldsymbol{x}^w = [x\ y\ z\ \psi]^T \in \mathbb{R}^4$ is the vector containing the position (x, y, z) and the yaw angle ψ with respect the inertial reference frame (the super-index w indicates the relation with the inertial reference frame), $K_u \in \mathbb{R}^{4\times4}$ and $K_v \in \mathbb{R}^{4\times4}$ are positive definite diagonal matrices related to the parameters of the vehicle and the embedded controller explicitly given as

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

and

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

The matrix $T(\psi) \;\in \mathbb{R}^4$ is a transformation matrix given by

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

The vector

$$\boldsymbol{u} = [u_\theta \; u_\phi \; u_{\dot{z}} \; u_{\dot{\psi}}]^T \tag{22}$$

is the control input vector for the inner control loop where:

- u_θ is the angular position control input related to the displacement along the x-axis.
- u_ϕ is the angular position control input related to the displacement along the y-axis.
- $u_{\dot{z}}$ is the velocity control input related to the displacement along the z-axis.
- $u_{\dot{\psi}}$ is the angular velocity control input related to the rotation around the z-axis.

As mentioned earlier, the nonlinear equation system expressed in (21) is part of the closed-loop system resulting from the quadrotor dynamics (1)-(2) and the inaccessible-to-the-user embedded controller (3)-(4). However, the system in (21) can be assumed as the quadrotor model since the parameters of the embedded controller cannot be modified, which motivates the design of an outer control loop. In some quadrotor applications, the model parameters may change as a consequence of weather conditions (wind gusts, rain, changes in air density, etc.), or the task to be performed, such as transporting payload suspended on a cable. Due to the aforementioned, the system in (21) can be expressed similarly to [58, 59], resulting in the following input-output representation

$$\ddot{\boldsymbol{x}}^w = T(\psi)K_u(\boldsymbol{x}^w, \dot{\boldsymbol{x}}^w)\boldsymbol{u} + \boldsymbol{\delta}(t), \tag{23}$$

where the term $-K_v\dot{\boldsymbol{x}}^w$ of the model in (21) is contained into $\boldsymbol{\delta}(t) \;\in \mathbb{R}^4$, which represents the vector of disturbances bounded as

$$||\boldsymbol{\delta}(t)|| \leq \delta_0, \; \forall \begin{bmatrix} \boldsymbol{x}^w \\ \dot{\boldsymbol{x}}^w \end{bmatrix} \in \Omega,$$

where δ_0 is a strictly positive constant and Ω is a compact set.

In order to ensure trajectory tracking control of the system (23), an outer scheme $\boldsymbol{u}$ to supply commands of position and velocity to the inner controller should be designed. Hence one of the purposes of this manuscript is to introduce an adaptive GRNN outer controller $\boldsymbol{u}$ for the system (23). Since the definition of $\boldsymbol{u}$ in (22), one can think that actually the outer control loop is a real-time trajectory planning stage.

3. Disturbance rejection controller (outer control loop)

The proposed controller is composed by an adaptive GRNN, a small-gain discontinuous term used to eliminate the approximation error of the neural network, and a continuous nonlinear term which may improve the convergence rate of the tracking error. Considering $\boldsymbol{x}_d^w = [x_d \; y_d \; z_d \; \psi_d]^T \in \mathbb{R}^4$ as the desired position and yaw angle vector, the generalized tracking error is defined as

$$\boldsymbol{e} = \boldsymbol{x}^w - \boldsymbol{x}_d^w. \tag{24}$$

Similar to [60], a sliding surface for a MIMO system is proposed as follows

$$\boldsymbol{r} = \dot{\boldsymbol{e}} + \alpha \boldsymbol{e}, \tag{25}$$

where $\alpha \in \mathbb{R}^{4\times4}$ is a diagonal positive definite gain matrix. Differentiating the equation (25) with respect to time, it leads to

$$\dot{\boldsymbol{r}} = \ddot{\boldsymbol{e}} + \alpha \dot{\boldsymbol{e}}. \tag{26}$$

By replacing the tracking error (24) into (26) we obtain

$$\ddot{\boldsymbol{x}}^w - \ddot{\boldsymbol{x}}_d^w = \dot{\boldsymbol{r}} - \alpha \dot{\boldsymbol{e}}. \tag{27}$$

Now, replacing the equation (25) into (27) and rearranging, we get

$$\ddot{\boldsymbol{x}}^w = \dot{\boldsymbol{r}} - \alpha(\boldsymbol{r} - \alpha \boldsymbol{e}) + \ddot{\boldsymbol{x}}_d^w = \dot{\boldsymbol{r}} - \alpha \boldsymbol{r} + \alpha^2 \boldsymbol{e} + \ddot{\boldsymbol{x}}_d^w. \tag{28}$$

Then, substituting the equation (28) into the quadrotor dynamic model in (23) and clearing $\dot{\boldsymbol{r}}$, we have

$$\dot{\boldsymbol{r}} = T(\psi) K_u \boldsymbol{u} + \boldsymbol{\delta}(t) + \alpha \boldsymbol{r} - \alpha^2 \boldsymbol{e} - \ddot{\boldsymbol{x}}_d^w. \tag{29}$$

Considering the equation (29), the following controller is proposed

$$\boldsymbol{u} = (T(\psi)K_u)^{-1}(\ddot{\boldsymbol{x}}_d^w - b_1 \tanh(b_2 \boldsymbol{r}) - \hat{\boldsymbol{\delta}}(t) - K_r \mathrm{sign}(\boldsymbol{r})), \tag{30}$$

where $\mathrm{sign}(\boldsymbol{r}) = [\mathrm{sign}(r_1) \; \mathrm{sign}(r_2) \; \mathrm{sign}(r_3) \; \mathrm{sign}(r_4)]^T \in \mathbb{R}^4$, being

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

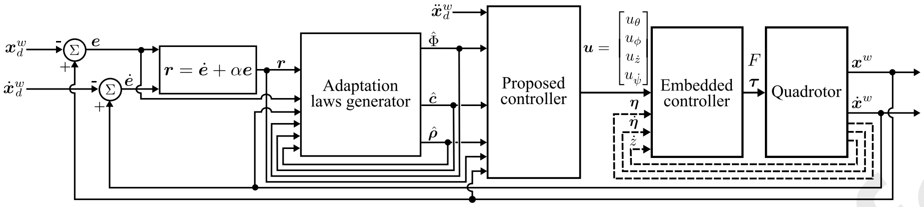

$\boldsymbol{x}_d^w$
-
Σ
+
$\boldsymbol{e}$
$\dot{\boldsymbol{x}}_d^w$
-
Σ
+
$\dot{\boldsymbol{e}}$
$\boldsymbol{r} = \dot{\boldsymbol{e}} + \alpha\boldsymbol{e}$
$\boldsymbol{r}$
Adaptation laws generator
$\hat{\Phi}$
$\hat{\boldsymbol{c}}$
$\hat{\boldsymbol{\rho}}$
$\ddot{\boldsymbol{x}}_d^w$
Proposed controller
$\boldsymbol{u} = \begin{bmatrix} u_\theta \\ u_\phi \\ u_{\dot{z}} \\ u_{\dot{\psi}} \end{bmatrix}$
$\boldsymbol{\eta}$
$\dot{\boldsymbol{\eta}}$
$\dot{z}$
Embedded controller
F
$\boldsymbol{\tau}$
Quadrotor
$\boldsymbol{x}^w$
$\dot{\boldsymbol{x}}^w$

Figure 1: Block diagram of the GRNN control scheme in (30).

$\tanh(b_2\boldsymbol{r}) = [\tanh(b_2r_1)\ \tanh(b_2r_2)\ \tanh(b_2r_3)\ \tanh(b_2r_4)]^T \in \mathbb{R}^4$, b_1 and b_2 are strictly positive constants, $K_r \in \mathbb{R}^{4\times4}$ is a diagonal positive definite gain matrix and $\hat{\boldsymbol{\delta}}(t) \in \mathbb{R}^4$ is an estimation of the disturbance $\boldsymbol{\delta}(t) \in \mathbb{R}^4$ produced by environmental conditions, variation of the payload and the unmodeled dynamics acting on the system. The term $b_1\tanh(b_2\boldsymbol{r})$ is used as a soft saturation function to bound the control action avoiding high values that could unstabilize the quadrotor. This strategy is useful especially for the experimental implementation and during the tuning process. The term $b_1\tanh(b_2\boldsymbol{r})$ was used similarly in [60]. It is noteworthy that the term $b_1\tanh(b_2\boldsymbol{r})$ could be replaced by $b_1\boldsymbol{r}$ and the control goal will be still ensured by using an appropriate positive definite function V and the corresponding conditions. The block diagram of the closed-loop system with the proposed control scheme and the embedded controller is presented in Figure 1.

Notice that the selection of the sliding surface involved in the discontinuous terms of the controller (30) corresponds to a linear combination of the position and velocity errors, $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$, respectively, as done in many other designs in the literature.

Now, the closed-loop system obtained by replacing (30) in (29) is given by

$$\dot{\boldsymbol{r}} = -b_1\tanh(b_2\boldsymbol{r}) + (\boldsymbol{\delta}(t) - \hat{\boldsymbol{\delta}}(t)) + \alpha\boldsymbol{r} - \alpha^2\boldsymbol{e} - K_r\text{sign}(\boldsymbol{r}). \tag{31}$$

It should be noticed that the signal $\boldsymbol{\delta}(t)$ can be expressed by a GRNN, which in agreement with the universal approximation theorem [61, 62], is used to approximate any continuous function as follows

$$||\boldsymbol{\delta}(t) - \Phi^T\boldsymbol{H}(\bar{\boldsymbol{r}}, \boldsymbol{c}, \boldsymbol{\rho})|| \leq ||\boldsymbol{\epsilon}|| < \infty,$$

or

$$\boldsymbol{\delta}(t) = \Phi^T\boldsymbol{H}(\bar{\boldsymbol{r}}, \boldsymbol{c}, \boldsymbol{\rho}) + \boldsymbol{\epsilon}, \tag{32}$$

where $\bar{\boldsymbol{r}} = [1\ \boldsymbol{r}^T\ \boldsymbol{e}^T\ \dot{\boldsymbol{x}}^{wT}]^T \in \mathbb{R}^{13}$ is the extended input vector for the GRNN, $\boldsymbol{H}(\bar{\boldsymbol{r}}, \boldsymbol{c}, \boldsymbol{\rho}) \in \mathbb{R}^m$ is a vector of radial basis activation functions defined as

$$H_i(\bar{\boldsymbol{r}}, \boldsymbol{c}_i, \boldsymbol{\rho}_i) = \frac{e^{\frac{||\bar{\boldsymbol{r}} - \boldsymbol{c}_i||}{2\rho_i}}}{\sum_{j=1}^{m} e^{\frac{||\bar{\boldsymbol{r}} - \boldsymbol{c}_j||}{2\rho_j}}}, \tag{33}$$

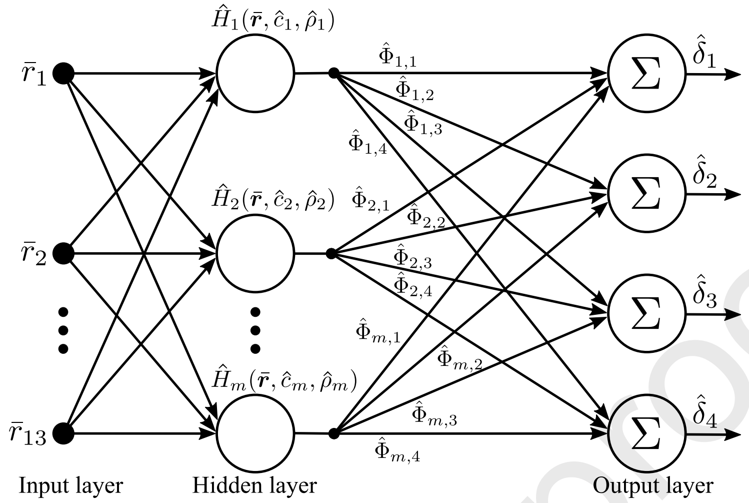


Figure 2: Diagram of the generalized regression neural network used to obtain the estimate $\hat{\boldsymbol{\delta}}$.

where m is the number of neurons, $\boldsymbol{c}_i \in \mathbb{R}^{13}$ is the vector of centers for each element of $\boldsymbol{H}(\bar{\boldsymbol{r}}, \boldsymbol{c}, \boldsymbol{\rho})$, the vector $\boldsymbol{c} \in \mathbb{R}^{13m}$ is a vertical concatenation of the vectors $\boldsymbol{c}_i$ so that $\boldsymbol{c} = [\boldsymbol{c}_1^T \ \boldsymbol{c}_2^T \ ... \ \boldsymbol{c}_{13}^T]^T$, whereas $\rho_i \in \mathbb{R}$ is the standard deviation for each radial basis function $H_i(\bar{\boldsymbol{r}}, \boldsymbol{c}_i, \rho_i)$ in $\boldsymbol{H}(\bar{\boldsymbol{r}}, \boldsymbol{c}, \boldsymbol{\rho})$. Thus, $\boldsymbol{\rho} \in \mathbb{R}^m$ is the standard deviation vector, $\Phi \in \mathbb{R}^{m \times n}$ is the optimal parameter matrix, which is constant and unknown, and $n = 4$ is the number of outputs. The vector $\boldsymbol{\epsilon} \in \mathbb{R}^4$ is the approximation error which is bounded as $|\epsilon_i| \leq k_{ri}$. The matrix $K_r = \text{diag}\{k_{ri}\}$ in the control law (30) helps to compensate the approximation errors. Based on the discussions in [35, 36, 37, 38, 39, 63], the GRNN is a variation of the radial basis function neural network with a multilayer structure. It is mainly composed by four layers: an input layer, a hidden layer, a summation layer, and an output layer. The radial basis functions are contained in the hidden layer. Specifically, the GRNN is a variation of the RBFNN because it uses a Gaussian function as an activation function. But, in contrast with the classical RBFNN, the activation function of the GRNN is normalized as can be observed in equation (33).

Then, the estimation $\hat{\boldsymbol{\delta}}$ is given by a generalized regression neural network, described as

$$\hat{\boldsymbol{\delta}}(t) = \hat{\Phi}^T \hat{\boldsymbol{H}}(\bar{\boldsymbol{r}}, \hat{\boldsymbol{c}}, \hat{\boldsymbol{\rho}}). \tag{34}$$

In Figure 2, a diagram of the GRNN structure in (34) is depicted. The parameter estimation errors are defined as

$$\tilde{\Phi} = \Phi - \hat{\Phi}, \tag{35}$$

$$\tilde{\boldsymbol{c}} = \boldsymbol{c} - \hat{\boldsymbol{c}}, \tag{36}$$

$$\tilde{\boldsymbol{\rho}} = \boldsymbol{\rho} - \hat{\boldsymbol{\rho}}. \tag{37}$$

From equation (32), the vector of external disturbance can be represented as a function of the parameter estimation errors in (35), (36) and (37) as

$$\begin{aligned}\boldsymbol{\delta}(t) &= \left[\tilde{\Phi}(t)+\hat{\Phi}(t)\right]^T\left[\hat{\boldsymbol{H}}(\bar{\boldsymbol{r}},\hat{\boldsymbol{c}},\hat{\boldsymbol{\rho}})+\tilde{\boldsymbol{H}}(\bar{\boldsymbol{r}},\tilde{\boldsymbol{c}},\tilde{\boldsymbol{\rho}})\right]+\boldsymbol{\epsilon},\\ &= \tilde{\Phi}(t)^T\hat{\boldsymbol{H}}(\bar{\boldsymbol{r}},\hat{\boldsymbol{c}},\hat{\boldsymbol{\rho}})+\tilde{\Phi}(t)^T\tilde{\boldsymbol{H}}(\bar{\boldsymbol{r}},\tilde{\boldsymbol{c}},\tilde{\boldsymbol{\rho}})+\hat{\Phi}(t)^T\hat{\boldsymbol{H}}(\bar{\boldsymbol{r}},\hat{\boldsymbol{c}},\hat{\boldsymbol{\rho}})\\ &\quad+\hat{\Phi}(t)^T\tilde{\boldsymbol{H}}(\bar{\boldsymbol{r}},\tilde{\boldsymbol{c}},\tilde{\boldsymbol{\rho}})+\boldsymbol{\epsilon}. \end{aligned}\tag{38}$$

By replacing the estimation function (34) into the equation (38) we obtain

$$\boldsymbol{\delta}(t)=\hat{\boldsymbol{\delta}}(t)+\tilde{\Phi}(t)^T\hat{\boldsymbol{H}}(\bar{\boldsymbol{r}},\hat{\boldsymbol{c}},\hat{\boldsymbol{\rho}})+\tilde{\Phi}(t)^T\tilde{\boldsymbol{H}}(\bar{\boldsymbol{r}},\tilde{\boldsymbol{c}},\tilde{\boldsymbol{\rho}})+\hat{\Phi}(t)^T\tilde{\boldsymbol{H}}(\bar{\boldsymbol{r}},\tilde{\boldsymbol{c}},\tilde{\boldsymbol{\rho}})+\boldsymbol{\epsilon}. \tag{39}$$

Defining the error of the estimation function (34) as $\tilde{\boldsymbol{\delta}}(t)=\boldsymbol{\delta}(t)-\hat{\boldsymbol{\delta}}(t)$, the equation (39) can be written as

$$\tilde{\boldsymbol{\delta}}(t)=\tilde{\Phi}(t)^T\hat{\boldsymbol{H}}(\bar{\boldsymbol{r}},\hat{\boldsymbol{c}},\hat{\boldsymbol{\rho}})+\tilde{\Phi}(t)^T\tilde{\boldsymbol{H}}(\bar{\boldsymbol{r}},\tilde{\boldsymbol{c}},\tilde{\boldsymbol{\rho}})+\hat{\Phi}(t)^T\tilde{\boldsymbol{H}}(\bar{\boldsymbol{r}},\tilde{\boldsymbol{c}},\tilde{\boldsymbol{\rho}})+\boldsymbol{\epsilon}. \tag{40}$$

Now, linearizing the function $\Phi^T\boldsymbol{H}(\bar{\boldsymbol{r}},\boldsymbol{c},\boldsymbol{\rho})$ around the operation point $\boldsymbol{c}=\hat{\boldsymbol{c}}$ and $\boldsymbol{\rho}=\hat{\boldsymbol{\rho}}$ as

$$\begin{aligned}\Phi^T\boldsymbol{H}(\bar{\boldsymbol{r}},\boldsymbol{c},\boldsymbol{\rho}) &= \Phi^T\hat{\boldsymbol{H}}(\bar{\boldsymbol{r}},\hat{\boldsymbol{c}},\hat{\boldsymbol{\rho}})+\left.\frac{\partial[\Phi^T\boldsymbol{H}(\bar{\boldsymbol{r}},\boldsymbol{c},\boldsymbol{\rho})]}{\partial\boldsymbol{c}}\right|_{\substack{\boldsymbol{\rho}=\hat{\boldsymbol{\rho}}\\ \boldsymbol{c}=\hat{\boldsymbol{c}}}}(\boldsymbol{c}-\hat{\boldsymbol{c}})\\ &\quad+\left.\frac{\partial[\Phi^T\boldsymbol{H}(\bar{\boldsymbol{r}},\boldsymbol{c},\boldsymbol{\rho})]}{\partial\boldsymbol{\rho}}\right|_{\substack{\boldsymbol{\rho}=\hat{\boldsymbol{\rho}}\\ \boldsymbol{c}=\hat{\boldsymbol{c}}}}(\boldsymbol{\rho}-\hat{\boldsymbol{\rho}})\\ &= \Phi^T\hat{\boldsymbol{H}}(\bar{\boldsymbol{r}},\hat{\boldsymbol{c}},\hat{\boldsymbol{\rho}})+\Phi^T\left.\frac{\partial\boldsymbol{H}(\bar{\boldsymbol{r}},\boldsymbol{c},\boldsymbol{\rho})}{\partial\boldsymbol{c}}\right|_{\substack{\boldsymbol{\rho}=\hat{\boldsymbol{\rho}}\\ \boldsymbol{c}=\hat{\boldsymbol{c}}}}(\boldsymbol{c}-\hat{\boldsymbol{c}})\\ &\quad+\Phi^T\left.\frac{\partial\boldsymbol{H}(\bar{\boldsymbol{r}},\boldsymbol{c},\boldsymbol{\rho})}{\partial\boldsymbol{\rho}}\right|_{\substack{\boldsymbol{\rho}=\hat{\boldsymbol{\rho}}\\ \boldsymbol{c}=\hat{\boldsymbol{c}}}}(\boldsymbol{\rho}-\hat{\boldsymbol{\rho}}), \end{aligned}\tag{41}$$

which will be useful. The partial derivatives of the activation function in (33) with respect to the center and standard deviation vectors in the equation (41) are redefined for simple notation as

$$\Delta\hat{H}_c = \left.\frac{\partial\boldsymbol{H}(\bar{\boldsymbol{r}},\boldsymbol{c},\boldsymbol{\rho})}{\partial\boldsymbol{c}}\right|_{\substack{\boldsymbol{\rho}=\hat{\boldsymbol{\rho}}\\ \boldsymbol{c}=\hat{\boldsymbol{c}}}}, \tag{42}$$

$$\Delta\hat{H}_\rho = \left.\frac{\partial\boldsymbol{H}(\bar{\boldsymbol{r}},\boldsymbol{c},\boldsymbol{\rho})}{\partial\boldsymbol{\rho}}\right|_{\substack{\boldsymbol{\rho}=\hat{\boldsymbol{\rho}}\\ \boldsymbol{c}=\hat{\boldsymbol{c}}}}, \tag{43}$$

where $\Delta\hat{H}_c \in \mathbb{R}^{m\times[m\mathrm{dim}(\bar{\boldsymbol{r}})]}$ and $\Delta\hat{H}_\rho \in \mathbb{R}^{m\times m}$ are the Jacobians of the activation function (33) with respect the center vector $\boldsymbol{c}$ and the standard deviation vector $\boldsymbol{\rho}$, respectively, with $m\mathrm{dim}(\bar{\boldsymbol{r}})$ meaning m times by the dimension of the vector $\bar{\boldsymbol{r}}$. Then, rewriting and rearranging the equation (41) with the expressions (42) and (43) we obtain

$$\Phi^T\boldsymbol{H}(\bar{\boldsymbol{r}},\boldsymbol{c},\boldsymbol{\rho})-\Phi^T\hat{\boldsymbol{H}}(\bar{\boldsymbol{r}},\hat{\boldsymbol{c}},\hat{\boldsymbol{\rho}})=\Phi^T\Delta\hat{H}_c(\boldsymbol{c}-\hat{\boldsymbol{c}})+\Phi^T\Delta\hat{H}_\rho(\boldsymbol{\rho}-\hat{\boldsymbol{\rho}}). \tag{44}$$

Thus, the estimation error of the activation function (44) is given by

$$\tilde{\boldsymbol{H}}(\bar{\boldsymbol{r}}, \tilde{\boldsymbol{c}}, \tilde{\boldsymbol{\rho}}) = \Delta\hat{H}_c\tilde{\boldsymbol{c}} + \Delta\hat{H}_\rho\tilde{\boldsymbol{\rho}}. \tag{45}$$

Replacing the equation (45) into (40), and defining $\bar{\boldsymbol{\epsilon}} = \tilde{\Phi}^T\tilde{\boldsymbol{H}}(\bar{\boldsymbol{r}}, \tilde{\boldsymbol{c}}, \tilde{\boldsymbol{\rho}}) + \boldsymbol{\epsilon}$, the disturbance estimation error can be written as

$$\tilde{\boldsymbol{\delta}}(t) = \tilde{\Phi}^T\hat{\boldsymbol{H}}(\bar{\boldsymbol{r}}, \hat{\boldsymbol{c}}, \hat{\boldsymbol{\rho}}) + \hat{\Phi}^T(\Delta\hat{H}_c\tilde{\boldsymbol{c}} + \Delta\hat{H}_\rho\tilde{\boldsymbol{\rho}}) + \bar{\boldsymbol{\epsilon}}. \tag{46}$$

By replacing (46) into the equation (31), the closed-loop system is rewritten as

$$\dot{\boldsymbol{r}} = -b_1\tanh(b_2\boldsymbol{r}) + \tilde{\Phi}^T\hat{\boldsymbol{H}}(\bar{\boldsymbol{r}}, \hat{\boldsymbol{c}}, \hat{\boldsymbol{\rho}}) + \hat{\Phi}^T(\Delta\hat{H}_c\tilde{\boldsymbol{c}} + \Delta\hat{H}_\rho\tilde{\boldsymbol{\rho}}) + \bar{\boldsymbol{\epsilon}} + \alpha\boldsymbol{r} - \alpha^2\boldsymbol{e} - K_r\mathrm{sign}(\boldsymbol{r}). \tag{47}$$

Besides, the adaptation laws are defined as follows:

$$\begin{aligned}
\frac{d}{dt}\tilde{\Phi}_i = -\frac{d}{dt}\hat{\Phi}_i &= -b_3\hat{\boldsymbol{H}}(\bar{\boldsymbol{r}}, \hat{\boldsymbol{c}}, \hat{\boldsymbol{\rho}})\tanh(b_2 r_i), && (48)\\
\frac{d}{dt}\tilde{\boldsymbol{c}} = -\frac{d}{dt}\hat{\boldsymbol{c}} &= -b_4\sum_{i=1}^{n}[\tanh(b_2 r_i)(\Delta\hat{H}_c^T\hat{\Phi}_i(t))], && (49)\\
\frac{d}{dt}\tilde{\boldsymbol{\rho}} = -\frac{d}{dt}\hat{\boldsymbol{\rho}} &= -b_5\sum_{i=1}^{n}[\tanh(b_2 r_i)(\Delta\hat{H}_\rho^T\hat{\Phi}_i(t))], && (50)
\end{aligned}$$

where $\tilde{\Phi}_i \in \mathbb{R}^m$ denotes the ith column of matrix $\tilde{\Phi} \in \mathbb{R}^{m\times n}$. Finally, the overall closed-loop system is expressed by the equations (26), (47), (48), (49), and (50).

It is worthwhile to notice that the estimated parameters of the neural network $\hat{\Phi}_i$, $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{\rho}}$ are obtained by the adaptation laws (48), (49) and (50), respectively, and they are designed to match the convergence analysis shown in the coming analysis. There is no optimization algorithm nor optimal criterion to compute the mentioned parameters.

Proposition 1. *Assume gain matrices α and K_r to be positive definite matrices. In addition, consider that b_1, b_2, b_3, b_4, $b_5 > 0$ and the condition*

$$\alpha_i(2b_1 - \alpha_i^3) - \frac{1}{b_2^2} > 0 \tag{51}$$

is satisfied. Then, for all initial conditions starting at some compact set, the solutions $\boldsymbol{e}(t)$ and $\boldsymbol{r}(t)$ converge to zero as time t increases. In addition, the adaptation errors $\tilde{\Phi}_i(t)$, $\tilde{\boldsymbol{c}}(t)$ and $\tilde{\boldsymbol{\rho}}(t)$ remain bounded for all time $t \geq 0$.

Proof. First, the positive definite function

$$V = \sum_{i=1}^{n}\left[\frac{1}{b_2}\ln(\cosh(b_2 r_i)) + \frac{1}{2}e_i^2 + \frac{1}{2b_3}\tilde{\Phi}_i^T\tilde{\Phi}_i\right] + \frac{1}{2b_4}\tilde{\boldsymbol{c}}^T\tilde{\boldsymbol{c}} + \frac{1}{2b_5}\tilde{\boldsymbol{\rho}}^T\tilde{\boldsymbol{\rho}} \tag{52}$$

is defined, where b_1, b_2, b_3, b_4, and b_5 are strictly positive constants used in the control and the adaptation laws. Taking the time derivative of (52) along of the closed-loop equations (26) and (47) we have

$$\dot{V} = \sum_{i=1}^{n} \left[\tanh(b_2 r_i)\dot{r}_i + e_i \dot{e}_i + \frac{1}{b_3}\tilde{\Phi}_i^T \dot{\tilde{\Phi}}_i \right] + \frac{1}{b_4}\tilde{\boldsymbol{c}}^T \dot{\tilde{\boldsymbol{c}}} + \frac{1}{b_5}\tilde{\boldsymbol{\rho}}^T \dot{\tilde{\boldsymbol{\rho}}}. \tag{53}$$

By replacing the equation (25) and (47) into equation (53) and performing the appropriate algebraic manipulations we obtain

$$\begin{aligned} \dot{V} \;\; = \;\; & \sum_{i=1}^{n} \Big[-b_1 \tanh(b_2 r_i)^2 + \tanh(b_2 r_i)\tilde{\Phi}_i^T \hat{\boldsymbol{H}}(\bar{\boldsymbol{r}}, \hat{\boldsymbol{c}}, \hat{\boldsymbol{\rho}}) + \tanh(b_2 r_i)\hat{\Phi}_i^T \Delta \hat{H}_c \tilde{\boldsymbol{c}} \\ & + \tanh(b_2 r_i)\hat{\Phi}_i^T \Delta \hat{H}_\rho \tilde{\boldsymbol{\rho}} + \tanh(b_2 r_i)\bar{\epsilon}_i + \tanh(b_2 r_i)(\alpha r_i - \alpha^2 e_i) \\ & - \tanh(b_2 r_i) k_{ri} \mathrm{sign}(r_i) + e_i(r_i - \alpha e_i) + \frac{1}{b_3}\tilde{\Phi}_i^T \dot{\tilde{\Phi}}_i \Big] + \frac{1}{b_4}\tilde{\boldsymbol{c}}^T \dot{\tilde{\boldsymbol{c}}} \\ & + \frac{1}{b_5}\tilde{\boldsymbol{\rho}}^T \dot{\tilde{\boldsymbol{\rho}}}. \end{aligned} \tag{54}$$

In order to simplify the equation (54), the following products are reordered by using the property $\boldsymbol{x}^T A \boldsymbol{y} = \boldsymbol{y}^T A^T \boldsymbol{x}$, with matching dimensions of $\boldsymbol{x}$, A and $\boldsymbol{y}$

$$\begin{aligned} \hat{\Phi}_i^T \Delta \hat{H}_c \tilde{\boldsymbol{c}} &= \tilde{\boldsymbol{c}}^T \Delta \hat{H}_c^T \hat{\Phi}_i, \\ \hat{\Phi}_i^T \Delta \hat{H}_\rho \tilde{\boldsymbol{\rho}} &= \tilde{\boldsymbol{\rho}}^T \Delta \hat{H}_\rho^T \hat{\Phi}_i. \end{aligned}$$

Then, the equation (54) can be rewritten after grouping common terms as

$$\begin{aligned} \dot{V} \;\; = \;\; & \sum_{i=1}^{n} \left[-b_1 \tanh(b_2 r_i)^2 + \tanh(b_2 r_i)\bar{\epsilon}_i + \tanh(b_2 r_i)(\alpha r_i - \alpha^2 e_i) \right. \\ & \left. - k_{ri} |\tanh(b_2 r_i)| + e_i(r_i - \alpha e_i) \right] \\ & + \tilde{\Phi}_i^T \left(\sum_{i=1}^{n} \left[\hat{\boldsymbol{H}}(\bar{\boldsymbol{r}}, \hat{\boldsymbol{c}}, \hat{\boldsymbol{\rho}}) \tanh(b_2 r_i) + \frac{1}{b_3} \dot{\tilde{\Phi}}_i \right] \right) \\ & + \tilde{\boldsymbol{c}}^T \left(\sum_{i=1}^{n} \left[\tanh(b_2 r_i) \Delta \hat{H}_c^T \hat{\Phi}_i \right] + \frac{1}{b_4} \dot{\tilde{\boldsymbol{c}}} \right) \\ & + \tilde{\boldsymbol{\rho}}^T \left(\sum_{i=1}^{n} \left[\tanh(b_2 r_i) \Delta \hat{H}_\rho^T \hat{\Phi}_i \right] + \frac{1}{b_5} \dot{\tilde{\boldsymbol{\rho}}} \right). \end{aligned} \tag{55}$$

It is clear that the adaptation laws (48), (49) and (50) are suggested from the last three terms in (55). Specifically, substituting the adaptation laws (48), (49) and (50) into the equation (55) leads to

$$\dot{V} = \sum_{i=1}^{n} -b_1 \tanh(b_2 r_i)^2 + \tanh(b_2 r_i)(\alpha_i r_i - \alpha_i^2 e_i) + e_i(r_i - \alpha_i e_i) + \beta_i(r_i) \tag{56}$$

where

$$\beta_i(r_i) = \tanh(b_2 r_i)\bar{\epsilon}_i - k_{ri} |\tanh(b_2 r_i)| \le 0,$$

since $k_{ri} \geq |\bar{\epsilon}_i|$.

Now, two cases are analyzed, when $r_i = 0$ and $r_i \neq 0$. Firstly, analyzing the case $r_i = 0$, from the equation (56) it is easy to see that all the terms with $\tanh(b_2 r_i)$ are equal to zero, which leads to

$$\dot{V}_i = -\alpha_i e_i^2 \leq 0,$$

this result guarantees the boundedness of the trajectory tracking error when $r_i = 0$.

Secondly, when $r_i \neq 0$, $\dot{V}$ can be rewritten as

$$\begin{aligned} \dot{V} \quad = \quad & \sum_{i=1}^{n} -\left(b_1 - \frac{\alpha_i r_i}{\tanh(b_2 r_i)}\right)\tanh(b_2 r_i)^2 - \left(\alpha_i^2 - \frac{r_i}{\tanh(b_2 r_i)}\right) e_i \tanh(b_2 r_i) \\ & -\alpha_i e_i^2 + \beta_i(r_i). \end{aligned} \tag{57}$$

Thus, defining the vector $\boldsymbol{E}_i = [e_i \;\; \tanh(b_2 r_i)]^T$ the equation (57) can be expressed as

$$\dot{V} = \sum_{i=1}^{n} -\boldsymbol{E}_i^T Q_i(r_i)\boldsymbol{E}_i - \frac{\alpha_i}{2} e_i^2 + \beta_i(r_i), \tag{58}$$

where $Q_i(r_i)$ is given by

$$Q_i(r_i) = \begin{bmatrix} \frac{\alpha_i}{2} & \frac{\alpha_i^2}{2} - \frac{r_i}{2\tanh(b_2 r_i)} \\ \frac{\alpha_i^2}{2} - \frac{r_i}{2\tanh(b_2 r_i)} & b_1 - \frac{\alpha_i r_i}{\tanh(b_2 r_i)} \end{bmatrix},$$

where the term $-\alpha_i e_i^2$ has been conveniently split up to include one half in the first term of (58).

To guarantee the convergence of the proposed control scheme, it is necessary to prove that the matrix $Q_i(r_i)$ is positive definite. By using Sylvester's criterion it is possible to find the conditions for $Q_i(r_i)$ to be a positive definite matrix. This criterion leads to the following conditions:

$$\alpha_i \quad > \quad 0, \tag{59}$$

$$\frac{\alpha_i}{2}\left(b_1 - \frac{\alpha_i r_i}{\tanh(b_2 r_i)}\right) - \left(\frac{\alpha_i^2}{2} - \frac{r_i}{2\tanh(b_2 r_i)}\right)^2 \quad > \quad 0. \tag{60}$$

The condition (59) is trivially satisfied. By expanding and rearranging the expression (60), we have

$$2\alpha_i b_1 - \alpha_i^4 > \left[\frac{r_i}{\tanh(b_2 r_i)}\right]^2. \tag{61}$$

By using the fact that

$$\left|\frac{r_i}{\tanh(b_2 r_i)}\right| \leq |r_i| + \frac{1}{b_2},$$

it becomes clear that the expression (61) is satisfied if the inequality

$$\alpha_i(2b_1 - \alpha_i^3) > \left[|r_i| + \frac{1}{b_2}\right]^2 \tag{62}$$

is achieved.

The inequality (62), and in consequence (59), is satisfied for $|r_i| < r_{i\,\max}$ for some $r_{i\,\max} > 0$. In fact, the sufficient condition for the existence of some $r_{i\,\max} > 0$ is given in (51), which is always accomplished with b_1 large enough and is a sufficient condition for $\dot{V}$ to be a locally negative definite function.

Considering the definition of V in (52) we can write

$$V \geq \sum_{i=1}^{n} \frac{1}{b_2} \ln(\cosh(b_2 r_i)).$$

By using the following property [64]

$$\frac{1}{b_2} \ln(\cosh(b_2 r_i)) \geq |r_i| - \frac{1}{b_2} \ln(2),$$

and clearing for r_i, we obtain

$$V_{ri} + \frac{1}{b_2} \ln(2) > |r_i|, \tag{63}$$

where $V_{ri} = \frac{1}{b_2} \ln(\cosh(b_2 r_i))$.

Then, replacing the equation (63) into the inequality (62) the following expression is obtained

$$\alpha_i(2b_1 - \alpha_i^3) > \left(V_{ri} + \frac{\ln(2) + 1}{b_2}\right)^2,$$

which is satisfied for "small" r_i as we stated earlier. Therefore, since $\beta_i(r_i) \leq 0$ for all r_i, $\dot{V}$ can be upper-bounded as

$$\dot{V} \leq \sum_{i=1}^{n} -\frac{\alpha_i}{2} e_i^2 \quad \text{for all} \quad |r_i| < r_{i\,\max},\ i = 1, 2, ..., 4, \tag{64}$$

leading to the conclusion that $\dot{V}(t) \leq 0\ \forall\ t \in [0, \infty)$ for initial conditions $r_i(0)$, $e_i(0)$, $\tilde{\Phi}_i(0)$, $\tilde{\boldsymbol{c}}(0)$, and $\tilde{\boldsymbol{\rho}}(0)$ starting at some compact set.

In agreement with the equations (52), (59), and (64), and invoking Barbalat's lemma [65], these results guarantee the boundedness and convergence to zero of the control error signal $e_i(t)$ during the closed-loop operation of the proposed control scheme. In other words, the trajectory tracking control aim is satisfied with the controller (30), (48), (49), and (50).

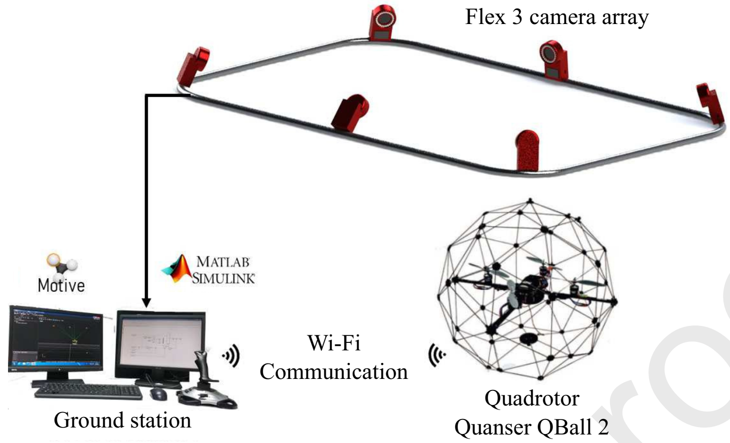


Figure 3: Experimental set up with the motion capture system Optitrack and the QBall 2 quadrotor.

4. Experimental results

In order to validate the proposed control scheme, experimental tests are performed. The experimental validation consisted of implementing the proposed control scheme in the QBall 2 quadrotor and compare it with three different control schemes. The control schemes selected for the comparison are the embedded controller plus a PD scheme, an adaptive model regressor-based control scheme, and an adaptive neural network algorithm. The comparisons were designed to perform the trajectory tracking task for two different trajectories. The selected trajectories are a lemniscate path and a circular path.

4.1. Experimental platform

The real-time experiments are carried out in the QBall 2 quadrotor integrated with the motion capture system Optitrack as shown in Figure 3. The QBall 2 quadrotor is a Quanser experimental platform useful to test different control schemes. The controllers are developed in MATLAB-Simulink, and Quarc software is used to compile and upload the Simulink model to the UAV on-board computer. The position and yaw angle of the quadrotor are sensed by the motion capture system Optitrack using an array of 6 synchronized Flex 3 cameras. The roll and pitch angles and the angular velocities are obtained using the inertial sensors: 3-axis accelerometer and 3-axis gyroscope. The control inputs for the QBall 2 quadrotor are the PWM signals associated with the thrust of each rotor. The sampling rate is 500 [Hz] for both the inertial measurement unit of the quadrotor and the on-board computer where the controller is executed. For the motion capture system, the sampling rate is 30 [Hz].

The embedded controller (3)-(4) and (7)-(10) was implemented in our experimental system by using the following parameters and gains:

$$\begin{aligned} m &= 1.79 \text{ [kg]}, & g &= 9.81 \text{ [m/s}^2\text{]}, \\ \omega_\phi &= 13.944, & \omega_\theta &= 13.944, \\ \xi_\phi &= 1.593, & \xi_\theta &= 1.593, \\ \tau_{\dot{z}} &= 0.728. \end{aligned} \tag{65}$$

In addition, the proposed adaptive GRNN external controller in equations (30), (48), (49), and (50) was implemented with the gains given by

$$\begin{aligned} &b_1 = 1.5, & &b_2 = 1.0, \quad b_3 = 15, \quad K_u = \operatorname{diag}\{15.812, 15.812, 10.6, 2.38\}, \\ &b_4 = 0.1, & &b_5 = 0.1, \quad m = 5, \quad K_r = \operatorname{diag}\{0.001, 0.001, 0.001, 0.001\}, \\ &\alpha = \operatorname{diag}\{3, 2, 5, 2\}. \end{aligned} \tag{66}$$

Random values into the set $[-1,\ 1]$ were selected to initialize the matrix $\hat{\Phi}$ and the vector $\hat{\boldsymbol{c}}$. Unitary values were used to initialize the components of the vector $\hat{\boldsymbol{\rho}}$. Owing to the discontinuous term on our controller the chattering **R3C2** phenomenon is present on the experimental tests. This is an undesired effect that can be attenuated by means of small gains on the discontinuous term. More specifically, to reduce the chattering phenomenon in our controller the values of the matrix K_r that corresponds to the gains of the discontinuous term were selected considerably small in comparison with the other gains. In consequence, the neural network handles the disturbance rejection and deals with parametric uncertainties.

4.2. Control schemes implemented for comparison

The performance of the proposed control scheme is compared with respect to other control schemes. Specifically, the embedded controller (3)-(4) together with a PD outer loop control, the adaptive model regressor-based controller reported in [14] and the adaptive neural network scheme given in [66] are implemented for the comparison.

Embedded controller plus outer PD control loop

The embedded controller (3)-(4) and (7)-(10) was implemented along with a PD outer controller given by

$$\boldsymbol{u} = (RK_u)^{-1}(\ddot{\boldsymbol{x}}_d^w - K_p\boldsymbol{e} - K_d\dot{\boldsymbol{e}}), \tag{67}$$

where $\boldsymbol{e}$ is the control error signal in (24) and K_p, K_d $\in \mathbb{R}^{4\times4}$ are diagonal positive definite matrices. In this experimental case study, the embedded controller in equations (3)-(4) and (7)-(10) was implemented with the gains (65) and the external PD controller used the gains

$$\begin{aligned} K_p &= \operatorname{diag}\{4.5, 3, 7.5, 3\}, \\ K_d &= \operatorname{diag}\{1.5, 1.5.1.5\}. \end{aligned} \tag{68}$$

The controller expressed by (3)-(4), (7)-(10), and (67) will be denoted as ECPD.

Adaptive model regressor control scheme

The adaptive controller implemented for the experimental validation was presented in [14], and consists of an outer control loop of position and an inner control loop of attitude. This control scheme is designed considering that the control inputs are the thrust F and the torque $\boldsymbol{\tau}$. The position controller is given by

$$\begin{aligned} F &= \frac{f_z}{\cos(\phi)\cos(\theta)}, \\ \theta_d &= \tan^{-1}\left(\frac{1}{f_z}\left[f_y\sin(\psi_d) + f_x\cos(\psi_d)\right]\right), \\ \phi_d &= \tan^{-1}\left(\frac{\cos(\theta_d)}{f_z}\left[f_x\sin(\psi_d) - f_y\cos(\psi_d)\right]\right), \end{aligned} \tag{69}$$

where $F \in \mathbb{R}$ is the total thrust, θ_d and ϕ_d are the pitch and roll desired angles respectively. The vector $\boldsymbol{f} = [f_x \ f_y \ f_z]^T \in \mathbb{R}^3$ is defined as

$$\boldsymbol{f} = \hat{m}\ddot{\boldsymbol{p}}_d + \hat{m}g\boldsymbol{e}_z + K_p\tilde{\boldsymbol{p}} + K_d\dot{\tilde{\boldsymbol{p}}},$$

with the position error defined as $\tilde{\boldsymbol{p}}(t) = \boldsymbol{p}_d(t) - \boldsymbol{p}(t)$, $\hat{m}$ as a dynamic estimation of the quadrotor mass, $\boldsymbol{e}_z = [0 \ 0 \ 1]^T$ is a unitary vector along the z axis in the inertial reference frame, and $K_p \in \mathbb{R}^{3\times3}$ and $K_d \in \mathbb{R}^{3\times3}$ are positive definite diagonal matrices. The adaptation law for the quadrotor mass is given by

$$\dot{\hat{m}} = \gamma_p\boldsymbol{Y}_p(\ddot{\boldsymbol{p}}_d)^T\dot{\tilde{\boldsymbol{p}}} + \gamma_p\epsilon\boldsymbol{Y}_p(\ddot{\boldsymbol{p}}_d)^T\tilde{\boldsymbol{p}},$$

where γ_p and ϵ are positive constants, $\boldsymbol{Y}_p(\ddot{\boldsymbol{p}}_d) \in \mathbb{R}^{3\times1}$ is the position regression matrix defined as $\boldsymbol{Y}_p(\ddot{\boldsymbol{p}}_d) = \ddot{\boldsymbol{p}}_d + g\boldsymbol{e}_z$. The attitude controller is given by

$$\boldsymbol{\tau} = Y_\eta(\boldsymbol{\eta}, \boldsymbol{\omega}, \boldsymbol{\omega}_r, \dot{\boldsymbol{\omega}}_r)\hat{\boldsymbol{\chi}}_\eta + K_s\boldsymbol{s} \tag{70}$$

where $\hat{\boldsymbol{\chi}}_\eta \in \mathbb{R}^6$ is the estimated parameter vector and $K_s \in \mathbb{R}^{3\times3}$ is a positive definite diagonal matrix, $\boldsymbol{s} = \dot{\tilde{\boldsymbol{\eta}}} + \Lambda\tilde{\boldsymbol{\eta}}$ is the filtered attitude error with the attitude error defined as $\tilde{\boldsymbol{\eta}}(t) = \boldsymbol{\eta}_d(t) - \boldsymbol{\eta}(t)$ and $\Lambda \in \mathbb{R}^{3\times3}$ is a positive definite diagonal matrix. A detailed description of this controller and its implementation can be consulted in [14]. The dynamic adaptive controller in (69) and (70) was implemented experimentally by using the gains

$$\begin{aligned} K_p &= \operatorname{diag}\{7.0, 7.0, 6.5\}, \\ K_d &= \operatorname{diag}\{2.5, 2.5, 4\}, \\ K_s &= \operatorname{diag}\{0.4, 0.4, 1.0\}, \\ \Lambda &= \operatorname{diag}\{4.38, 4.38, 1.5\}, \\ \gamma_p &= 0.014, \\ \epsilon &= 1.39, \\ \Gamma_\eta &= 1.5\times10^{-3}\operatorname{diag}\{1, 1, 1, 1, 1, 1\}. \end{aligned} \tag{71}$$

Hereafter, the controller (69)-(70) will be denoted as AMRC.

Adaptive neural network controller

The controller given in [66] is formed by an integral sliding mode control loop for the attitude and an adaptive neural network-based control loop for the position. The vector $\boldsymbol{u}_A = [\bar{u}_1 F,\, \bar{u}_2 F,\, F]^T$ is the control input for the position subsystem which is defined as

$$\boldsymbol{u}_A = M(\boldsymbol{\eta})^{-1}(g\boldsymbol{e}_3 + \hat{\boldsymbol{f}}(\boldsymbol{X}_{\text{in}}) + k_v\boldsymbol{\gamma} + \ddot{\boldsymbol{\xi}}_r), \tag{72}$$

where $F \in \mathbb{R}$ is the total thrust, $\bar{u}_1 = \cos(\psi)\sin(\theta)\cos(\phi) + \sin(\psi)\sin(\phi)$ and $\bar{u}_2 = \sin(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi)$ are auxiliary control inputs. It is noteworthy to mention that $\ddot{\boldsymbol{\xi}}_d = [\ddot{x}_d,\, \ddot{y}_d,\, \ddot{z}_d]^T$ is an acceleration pre-compensation term added to improve the performance of the controller in the experimental tests. The matrix $M(\boldsymbol{\eta}) = \operatorname{diag}\{1/m,\ 1/m,\ \cos(\phi)\cos(\theta)/m\} \ \in \mathbb{R}^{3\times 3}$, m is the quadrotor mass, g is the gravitational acceleration constant, $\boldsymbol{e}_3 = [0\ 0\ 1]^T$ is a unitary vector along the vertical axis, $\boldsymbol{\gamma} = \dot{\tilde{\boldsymbol{\xi}}} + \Lambda\tilde{\boldsymbol{\xi}}$ is an auxiliary state vector, with $\Lambda = \Lambda^T > 0$, and $\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi}_d - \boldsymbol{\xi}$ as the position error, being $\boldsymbol{\xi}_d$ the position desired signal, and k_v a strictly positive constant. The vector $\hat{\boldsymbol{f}}(\boldsymbol{X}_{\text{in}}) = \hat{W}^T\boldsymbol{P}(\boldsymbol{X}_{\text{in}})$ is the output of a radial basis function neural network, $\boldsymbol{X}_{\text{in}}$ is the input vector of the neural network with the 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 and σ_i are the center and width of the Gaussian function, respectively. The matrix $\hat{W}$ is the estimated weight matrix obtained from

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

where A is a symmetric positive definite gain matrix. The attitude control loop is given by

$$\boldsymbol{\tau} = M(\boldsymbol{\eta})\dot{\boldsymbol{v}}_\eta + C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \rho_\eta \operatorname{sign}(\boldsymbol{s}_\eta) + k_\eta M\tilde{\boldsymbol{v}}, \tag{73}$$

where ρ_η and k_η are positive constants, $C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$ is the Coriolis matrix, $\tilde{\boldsymbol{v}} = \boldsymbol{v}_\eta - \dot{\boldsymbol{\eta}}$ is the angular velocity error, $\boldsymbol{v}_\eta = \dot{\boldsymbol{\eta}}_d + k_w\tilde{\boldsymbol{\eta}} + \rho_w \operatorname{sign}(\boldsymbol{s}_w)$, the attitude error is defined as $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}_d - \boldsymbol{\eta}$, with $\boldsymbol{\eta}$ as the attitude of the quadrotor, and $\boldsymbol{\eta}_d$ the attitude desired signal, ρ_w and k_w are positive constants. Finally, the sliding surfaces are defined as $\boldsymbol{s}_w = \tilde{\boldsymbol{\eta}} + k_w\int_0^t \tilde{\boldsymbol{\eta}}$ and $\boldsymbol{s}_\eta = \tilde{\boldsymbol{v}} + k_\eta\int_0^t \tilde{\boldsymbol{v}}$. A detailed description of this control scheme can be consulted in [66]. The adaptive neural network controller was experimentally implemented using the following gains

$$\begin{array}{ll}
\boldsymbol{c} = [-1.5, -1, -0.5, 0, 0.5, 1, 1.5]^T, & k_w = 4.5, \\
\boldsymbol{\sigma} = [5, 5, 5, 5, 5, 5, 5]^T, & \rho_w = 1 \times 10^{-6}, \\
\Lambda = \operatorname{diag}\{2.55, 2.55, 3.55\}, & k_\eta = 15, \\
A = 0.15, & \rho_\eta = 1 \times 10^{-4}, \\
k_v = 1.257. &
\end{array} \tag{74}$$

The controller(72)-(73) will be denoted for referencing as ANNC.

4.3. Experimental validation

Two different experiments are carried out in order to validate the proposed controller. The first experiment consists of tracking a lemniscate path, and the second one consists of tracking a circular path.

The gains of the proposed control scheme in (3)-(4), (7)-(10) and (30), the ECPD scheme in (3)-(4), (7)-(10) and (67), the AMRC scheme in (69)-(70), and the ANNC algorithm in (72)-(73) were selected by a trial and error procedure resulting in the values given in (65), (66), (68), (71), and (74).

4.3.1. Experiment 1: Lemniscate path

The lemniscate path is described by the following desired signals

$$\begin{aligned} x_d(t) &= 0.5\sin\left(\frac{2\pi}{4}t\right) \text{ [m]}, \\ y_d(t) &= \cos\left(\frac{2\pi}{8}t\right) \text{ [m]}, \\ z_d(t) &= 1.0 \text{ [m]}, \qquad (75) \\ \psi_d(t) &= 0.0 \text{ [°]}. \qquad (76) \end{aligned}$$

As can be seen in Figure 4, all the control schemes fulfill the assigned task. However, the proposed controller remains closer to the reference than the other controllers that were tested.

The obtained signals of position error $\tilde{x}(t)$, $\tilde{y}(t)$, $\tilde{z}(t)$, and yaw angle error $\tilde{\psi}(t)$ of the quadrotor during the trajectory tracking task are depicted in Figure 5. Note that the proposed controller presents smaller error signals during all the experiment. The control actions correspond to the total thrust $F(t)$ provided by the rotors and the torques around each rotation axis $\boldsymbol{\tau}(t) = [\tau_\phi(t),\ \tau_\theta(t),\ \tau_\psi(t)]^T$. The control action signals $F(t)$ and $\boldsymbol{\tau}(t)$ produced during the experiment are shown in Figure 6. The control actions for all the control schemes are similar. Nevertheless, the control actions $\tau_\phi(t)$ and $\tau_\theta(t)$ provided by the proposed controller are smaller in comparison to that produced by the other controllers and the amplitude of its oscillations is smaller too.

In order to obtain a quantitative comparison index of the controllers performance, the root mean square (RMS) value of the tracking errors and the control signals for each controller are computed. The tracking error signals are calculated with the following expression

$$\tilde{\gamma}_\iota = \gamma_\iota - \gamma_d,$$

where γ represents the signals x, y, z, and ψ, the sub-index d denotes the desired signal, ι indicates the control scheme implemented to obtain that signal, being denoted as "Proposed" for the controller in (3)-(4), (7)-(10), (30), and (48)-(50), ECPD, AMRC and ANNC, which were previously described.

The RMS values of the tracking errors are presented in Table 1. The time interval to compute the RMS values was established in 10 [s] $\leq t \leq$ 35 [s] when all the signals have reached their steady states. The lowest values are

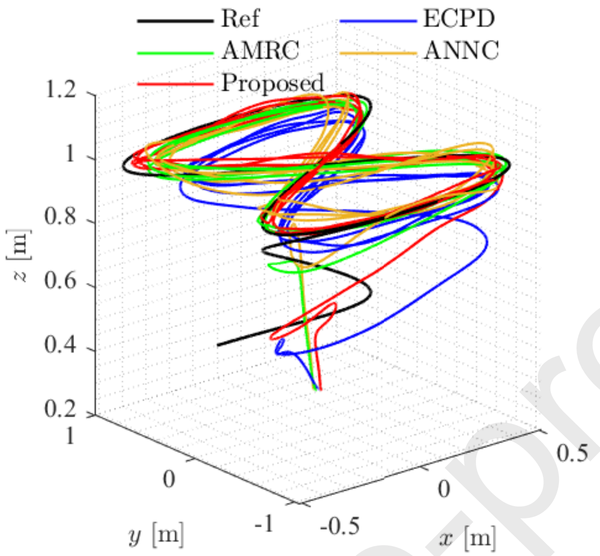
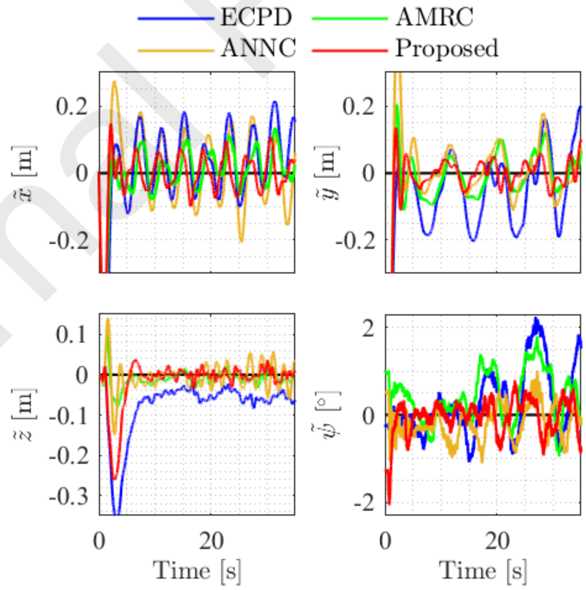


Figure 4: **Experiment 1:** Path $(x(t), y(t), z(t))$ drawn by the quadrotor for the specification of the desired trajectory in (75) and (76) when implementing the ECPD scheme, the AMRC controller, the ANNC algorithm, and the Proposed scheme.



R3C3

Figure 5: **Experiment 1:** Time evolution of position error $\tilde{x}(t)$, $\tilde{y}(t)$, $\tilde{z}(t)$, and yaw angle error $\tilde{\psi}(t)$ for the specification of the desired trajectory in (75) and (76) when implementing the ECPD scheme, the AMRC controller, the ANNC algorithm, and the Proposed scheme.

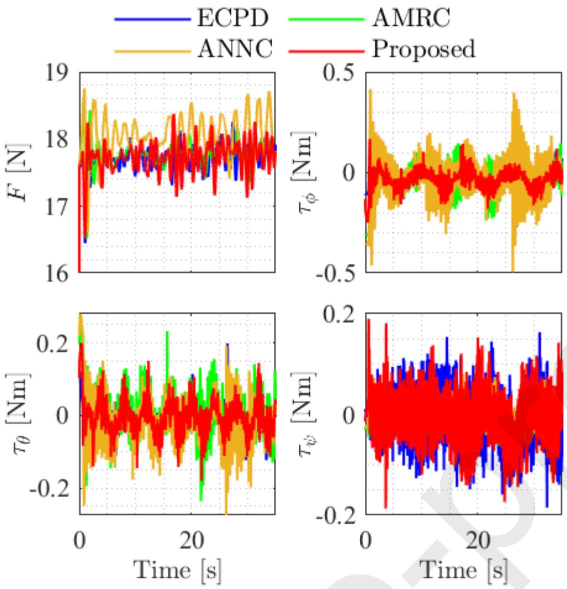


Figure 6: **Experiment 1:** Control actions provided by the ECPD scheme, the AMRC controller, the ANNC algorithm, and the Proposed scheme.

Table 1: **Experiment 1:** RMS values of position error $\tilde{x}(t)$, $\tilde{y}(t)$, $\tilde{z}(t)$, and yaw angle error $\tilde{\psi}(t)$ in the time interval 10 [s] $\leq t \leq$ 35 [s].

Error	Units	ECPD	AMRC	($\mathrm{P_{imp}}$%)	ANNC	($\mathrm{P_{imp}}$%)	Proposed	($\mathrm{P_{imp}}$%)
$\tilde{x}$	[m]	0.0981	0.0529	46.12	0.0952	2.97	**0.0480**	51.12
$\tilde{y}$	[m]	0.1104	0.0525	52.47	0.0637	42.32	**0.0348**	68.47
$\tilde{z}$	[m]	0.0525	0.0137	73.86	0.0256	51.19	**0.0125**	76.17
$\tilde{\psi}$	[°]	0.9445	0.7952	15.81	0.4566	51.66	**0.3427**	63.72

in bold font and indicate better performance for the trajectory tracking task. In addition, the relative percentage of improvement $\mathrm{P_{imp}}$% with respect to the ECPD controller was also computed aiming to provide a better understanding of the enhance obtained with the AMRC, ANNC and proposed controller, being computed as

$$\mathrm{P_{imp}}\%(\tilde{\gamma}_\varsigma) = \frac{\mathrm{RMS(ECPD)} - \mathrm{RMS}(\varsigma)}{\mathrm{RMS(ECPD)}} \times 100\%,$$

where ς represents either the proposed, the AMRC or the ANNC scheme implemented to obtain that error signal.

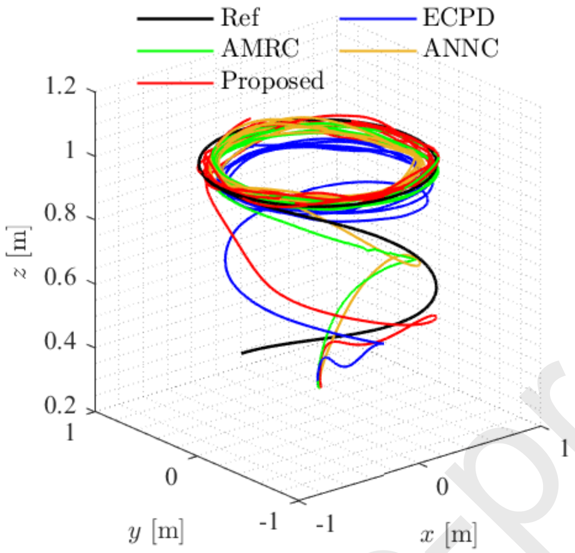


Figure 7: **Experiment 2:** Path $(x(t), y(t), z(t))$ drawn by the quadrotor for the specification of the desired trajectory in (77) and (78) when implementing the ECPD scheme, the AMRC controller, the ANNC algorithm, and the Proposed scheme.

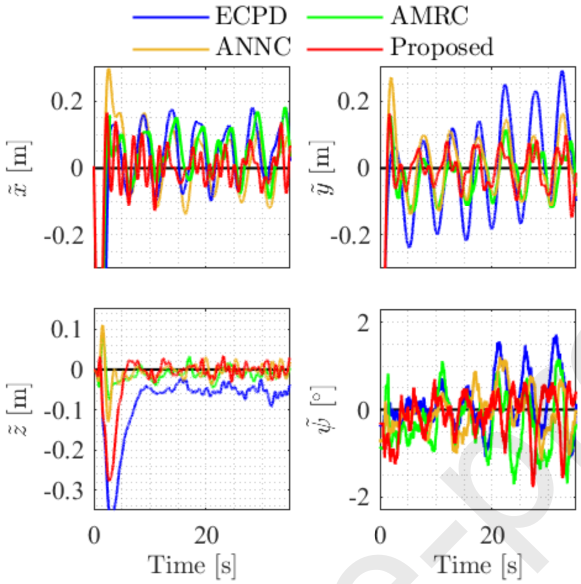
4.3.2. Experiment 2: Circular path

The circular path is described by the following desired signals

$$\begin{aligned} x_d &= 0.75\sin\left(\frac{2\pi}{5}t\right) \text{ [m]}, \\ y_d &= 0.75\cos\left(\frac{2\pi}{5}t\right) \text{ [m]}, \\ z_d &= 1.0 \text{ [m]}, && (77) \\ \psi_d &= 0.0 \text{ [°]}. && (78) \end{aligned}$$

In Figure 7, a tridimensional view of the quadrotor path is depicted. The error signals of the position and yaw angle of the quadrotor during the circular path tracking are depicted in Figure 8. The control action signals are shown in Figure 9.

The RMS values of the tracking errors for the circular path tracking task are presented in Table 2. The time interval is established in 10 [s] $\leq t \leq$ 35 [s] as in Experiment 1. The lowest values are in bold font to identify which control scheme provides better performance. The results are accompanied by their respective percentage of improvement. The proposed control scheme presents the best tracking accuracy, which confirms the advantage of the GRNN.



R3C3

Figure 8: **Experiment 2:** Time evolution of position error $\tilde{x}(t)$, $\tilde{y}(t)$, $\tilde{z}(t)$, and yaw angle error $\tilde{\psi}(t)$ for the specification of the desired trajectory in (77) and (78) when implementing the ECPD scheme, the AMRC controller, the ANNC algorithm, and the Proposed scheme.

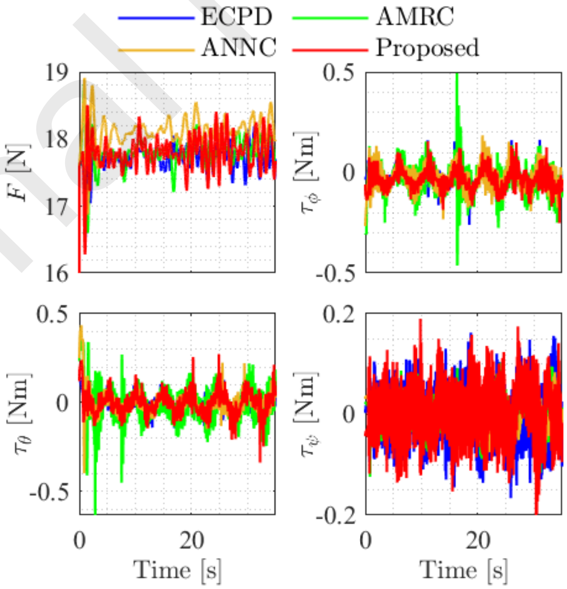


Figure 9: **Experiment 2:** Control actions provided by the ECPD scheme, the AMRC controller, the ANNC algorithm, and the Proposed scheme.

Table 2: **Experiment 2:** RMS values of position error $\tilde{x}(t)$, $\tilde{y}(t)$, $\tilde{z}(t)$, and yaw angle error $\tilde{\psi}(t)$ in the time interval 10 [s] $\leq t \leq$ 35 [s].

Error	Units	ECPD	AMRC	($\mathrm{P_{imp}}$%)	ANNC	($\mathrm{P_{imp}}$%)	Proposed	($\mathrm{P_{imp}}$%)
$\tilde{x}$	[m]	0.0949	0.0812	14.41	0.0730	23.05	**0.0446**	53.01
$\tilde{y}$	[m]	0.1416	0.0615	56.57	0.0777	45.14	**0.0534**	62.30
$\tilde{z}$	[m]	0.0536	0.0174	67.58	0.0141	73.67	**0.0140**	73.86
$\tilde{\psi}$	[°]	0.7143	0.7590	-6.26	0.5362	24.93	**0.4857**	32.01

5. Conclusions

This paper explored the modeling and control of UAVs assuming the presence of an embedded controller. An external control loop consisting of a robust online learning GRNN was introduced. An analysis of the trajectories of the closed-loop system was presented. The proposed control scheme was successfully implemented in a QBall 2 quadrotor. Experimental comparisons of the proposed controller with respect to the embedded controller plus an outer PD control loop, an adaptive model regressor control, and an adaptive neural network algorithm were carried out. The obtained results by using the proposed controller showed smaller tracking error values of position and yaw angle than the obtained with the other three control schemes. The relative percentages of improvement proved the advantages of using the proposed control scheme.

References

[1] K. Dalamagkidis, K. P. Valavanis, L. A. Piegl, Aviation history and unmanned flight, in: On integrating unmanned aircraft systems into the national airspace system, Springer, Dordrecht, Netherlands, 2012, pp. 11–42.

[2] L. V. Santana, A. S. Brandao, M. Sarcinelli-Filho, R. Carelli, A trajectory tracking and 3D positioning controller for the AR.Drone quadrotor, in: Proc. Int. Conf. on Unmanned Aircraft Systems, 2014, pp. 756–767.

[3] R. W. Beard, T. W. McLain, Small unmanned aircraft: Theory and practice, Princeton university press, Princeton, NJ, USA, 2012.

[4] R. X. Guo, J. K. Dong, Y. Zhu, Disturbance rejection and asymptotically stabilizing control for a quadrotor UAV, Journal of Control Engineering and Applied Informatics 17 (4) (2015) 33–41.

[5] R. Sanz, P. Garcia, Q. C. Zhong, P. Albertos, Predictor-based control of a class of time-delay systems and its application to quadrotors, IEEE Transactions on Industrial Electronics 64 (1) (2016) 459–469.

[6] K. Wu, Z. Zhang, C. Sun, Disturbance-observer-based output feedback control of non-linear cascaded systems with external disturbance, Institution of

Engineering and Technology Control Theory & Applications 12 (6) (2017) 738–744.

[7] A. Castillo, R. Sanz, P. Garcia, W. Qiu, H. Wang, C. Xu, Disturbance observer-based quadrotor attitude tracking control for aggressive maneuvers, Control Engineering Practice 82 (2019) 14–23.

[8] D. Cabecinhas, R. Cunha, C. Silvestre, A nonlinear quadrotor trajectory tracking controller with disturbance rejection, Control Engineering Practice 26 (2014) 1–10.

[9] J. B. Aldrich, Attitude control with analytic disturbance-rejection guarantee, Journal of Guidance, Control, and Dynamics 37 (6) (2014) 1791–1807.

[10] Y. Huang, W. Xue, Active disturbance rejection control: methodology and theoretical analysis, ISA Transactions 53 (4) (2014) 963–976.

[11] J. Jia, Z. Jiao, D. Sun, Y. Shang, Aircraft anti-skid braking active disturbance rejection control based on optimal slip ratio, in: Proc. Int. Conf. on Aircraft Utility Systems, 2018, pp. 1–6.

[12] B. Zhao, B. Xian, Y. Zhang, X. Zhang, Nonlinear robust adaptive tracking control of a quadrotor UAV via immersion and invariance methodology, IEEE Transactions on Industrial Electronics 62 (5) (2014) 2891–2902.

[13] O. Mofid, S. Mobayen, Adaptive sliding mode control for finite-time stability of quad-rotor UAVs with parametric uncertainties, ISA Transactions 72 (2018) 1–14.

[14] R. Pérez-Alcocer, J. Moreno-Valenzuela, Adaptive control for quadrotor trajectory tracking with accurate parametrization, IEEE Access 7 (2019) 53236–53247.

[15] P. J. Rossomando, The achievement of spacecraft autonomy through the thematic application of multiple cooperating intelligent agents, Telematics and Informatics 9 (3-4) (1992) 205–219.

[16] E. Mehiel, M. Balas, Optimization of direct adaptive disturbance rejection systems, in: American Institute of Aeronautics and Astronautics Guidance, Navigation, and Control Conf. and Exhibit, 2003, pp. 1–8.

[17] O. Naghash-Almasi, M. H. Khooban, PI adaptive LS-SVR control scheme with disturbance rejection for a class of uncertain nonlinear systems, Engineering Applications of Artificial Intelligence 52 (2016) 135–144.

[18] 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.

[19] L. Liu, Y.-J. Liu, A. Chen, S. Tong, C. P. Chen, Integral barrier Lyapunov function-based adaptive control for switched nonlinear systems, Science China Information Sciences 63 (3) (2020) 1–14.

[20] L. Liu, Y.-J. Liu, D. Li, S. Tong, Z. Wang, Barrier Lyapunov function-based adaptive fuzzy FTC for switched systems and its applications to resistance-inductance-capacitance circuit system, IEEE Transactions on Cybernetics 50 (8) (2019) 3491 – 3502.

[21] Y. Pan, P. Du, H. Xue, H.-K. Lam, Singularity-free fixed-time fuzzy control for robotic systems with user-defined performance, IEEE Transactions on Fuzzy Systems (2020).

[22] M. C. Santos, L. V. Santana, M. M. Martins, A. S. Brandão, M. Sarcinelli-Filho, Estimating and controlling UAV position using RGB-D/IMU data fusion with decentralized information/Kalman filter, in: Proc. 2015 IEEE Int. Conf. on Industrial Technology, pp. 232–239.

[23] L. Tang, D. Ma, J. Zhao, Adaptive neural control for switched non-linear systems with multiple tracking error constraints, IET Signal Processing 13 (3) (2018) 330–337.

[24] J. Ni, Z. Wu, L. Liu, C. Liu, Fixed-time adaptive neural network control for nonstrict-feedback nonlinear systems with deadzone and output constraint, ISA Transactions 97 (2020) 458–473.

[25] J. Fei, H. Wang, Recurrent neural network fractional-order sliding mode control of dynamic systems, Journal of the Franklin Institute 357 (2020) 4574–4591.

[26] J. Dunfied, M. Tarbouchi, G. Labonte, Neural network based control of a four rotor helicopter, in: Proc. 2004 IEEE Int. Conf. on Industrial Technology, Vol. 3, pp. 1543–1548.

[27] 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.

[28] T. Dierks, S. Jagannathan, Neural network control of quadrotor UAV formations, in: 2009 American Control Conf., 2009, pp. 2990–2996.

[29] D. Nodland, H. Zargarzadeh, S. Jagannathan, Neural network-based optimal adaptive output feedback control of a helicopter UAV, IEEE Transactions on Neural Networks and Learning Systems 24 (7) (2013) 1061–1073.

[30] M. Shirzadeh, A. Amirkhani, A. Jalali, M. R. Mosavi, An indirect adaptive neural control of a visual-based quadrotor robot for pursuing a moving target, ISA Transactions 59 (2015) 290–302.

[31] F. Jiang, F. Pourpanah, Q. Hao, Design, implementation, and evaluation of a neural-network-based quadcopter UAV system, IEEE Transactions on Industrial Electronics 67 (3) (2019) 2076–2085.

[32] W. Shen, Z. Li, Backstepping sliding mode RBF network adaptive control for quadrotor UAV, in: Proc. Chinese Automation Congress, 2019, pp. 4086–4091.

[33] C. Luo, J. Song, C. Lv, Robust adaptive control design based on RBFNN with an unmanned quadrotor for transporting tasks, in: Proc. 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conf., 2019, pp. 614–618.

[34] 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.

[35] D. F. Specht, A general regression neural network, IEEE Transactions on Neural Networks 2 (6) (1991) 568–576.

[36] T. L. Seng, M. Khalid, R. Yusof, Adaptive GRNN for the modelling of dynamic plants, in: Proc. IEEE Internatinal Symp. on Intelligent Control, 2002, pp. 217–222.

[37] Ö. Polat, T. Yıldırım, Genetic optimization of GRNN for pattern recognition without feature extraction, Expert Systems with Applications 34 (4) (2008) 2444–2448.

[38] H.-Z. Li, S. Guo, C.-J. Li, J.-Q. Sun, A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm, Knowledge-Based Systems 37 (2013) 378–387.

[39] A. J. Al-Mahasneh, S. G. Anavatti, M. A. Garratt, Altitude identification and intelligent control of a flapping wing micro aerial vehicle using modified generalized regression neural networks, in: Proc. IEEE Symp. Series on Computational Intelligence, 2017, pp. 1–6.

[40] X.-h. Xie, L. Xu, L. Zhou, Y. Tan, GRNN model for fault diagnosis of unmanned helicopter rotor's unbalance, in: Proc. 5th Int. Conf. on Electrical Engineering and Automatic Control, 2016, pp. 539–547.

[41] M. C. P. Santos, C. D. Rosales, M. Sarcinelli-Filho, R. Carelli, A novel null-space-based UAV trajectory tracking controller with collision avoidance, IEEE/ASME Transactions on Mechatronics 22 (6) (2017) 2543–2553.

[42] M. C. P. Santos, C. D. Rosales, J. A. Sarapura, M. Sarcinelli-Filho, R. Carelli, An adaptive dynamic controller for quadrotor to perform trajectory tracking tasks, Journal of Intelligent & Robotic Systems 93 (1-2) (2019) 5–16.

[43] F. Kendoul, Z. Yu, K. Nonami, Guidance and nonlinear control system for autonomous flight of minirotorcraft unmanned aerial vehicles, Journal of Field Robotics 27 (3) (2010) 311–334.

[44] 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.

[45] H. Bouadi, M. Tadjine, Nonlinear observer design and sliding mode control of four rotors helicopter, Int. Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering 1 (7) (2007) 354–359.

[46] H. Bouadi, M. Bouchoucha, M. Tadjine, Sliding mode control based on backstepping approach for an UAV type-quadrotor, World Academy of Science, Engineering and Technology 26 (5) (2007) 22–27.

[47] T. Luukkonen, Modelling and control of quadcopter (Aalto University, Espoo, Finland, Int. Rep. 22, 2011).

[48] M. Bangura, R. Mahony, Nonlinear dynamic modeling for high performance control of a quadrotor, in: Proc. Australasian Conf. on Robotics and Automation, 2012, pp. 115–124.

[49] K. T, V. Vonásek, D. Fišer, J. Faigl, AR-Drone as a robotic platform for research and education, in: Proc. Int. Conf. on Research and Education in Robotics, 2011, pp. 1–15.

[50] J. Engel, J. Sturm, D. Cremers, Camera-based navigation of a low-cost quadrocopter, in: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2012, pp. 2815–2821.

[51] A. Hernandez, C. Copot, R. De Keyser, T. Vlas, I. Nascu, Identification and path following control of an AR.Drone quadrotor, in: Proc. 17th Int. Conf. on System Theory, Control and Computing, 2013, pp. 583–588.

[52] P. Falcón, A. Barreiro, M. D. Cacho, Modeling of Parrot Ardrone and passivity-based reset control, in: Proc. 9th Asian Control Conf., 2013, pp. 1–6.

[53] L. V. Santana, A. S. Brandão, M. Sarcinelli-Filho, Navigation and cooperative control using the AR.Drone quadrotor, Journal of Intelligent & Robotic Systems 84 (1-4) (2016) 327–350.

[54] 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.

[55] C. D. Rosales, S. R. Tosetti, C. M. Soria, F. G. Rossomando, Neural adaptive PID control of a quadrotor using EFK, IEEE Latin America Transactions 16 (11) (2018) 2722–2730.

[56] C. Rosales, C. M. Soria, F. G. Rossomando, Identification and adaptive PID control of a hexacopter UAV based on neural networks, Int. Journal of Adaptive Control and Signal Processing 33 (1) (2019) 74–91.

[57] 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.

[58] C. Rosales, J. Gimenez, F. Rossomando, C. Soria, M. Sarcinelli-Filho, R. Carelli, UAVs formation control with dynamic compensation using neuro adaptive SMC, in: Int. Conf. on Unmanned Aircraft Systems, 2019, pp. 93–99.

[59] F. Rossomando, C. Rosales, J. Gimenez, L. Salinas, C. Soria, M. Sarcinelli-Filho, R. Carelli, Aerial load transportation with multiple quadrotors based on a kinematic controller and a neural SMC dynamic compensation, Journal of Intelligent & Robotic Systems (2020).

[60] F. Capraro, F. G. Rossomando, C. Soria, G. Scaglia, Cascade sliding control for trajectory tracking of a nonholonomic mobile robot with adaptive neural compensator, Mathematical Problems in Engineering 2017 (2017) 1–13.

[61] T. Chen, H. Chen, Approximations of continuous functionals by neural networks with application to dynamic systems, IEEE Transactions on Neural Networks 4 (6) (1993) 910–918.

[62] F. L. Lewis, J. Campos, R. Selmic, Neuro-fuzzy control of industrial systems with actuator nonlinearities, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.

[63] Z. Liu, C. P. Chen, Broad learning system: Structural extensions on single-layer and multi-layer neural networks, in: Proc. Int. Conf. on Security, Pattern Analysis, and Cybernetics, 2017, pp. 136–141.

[64] M. G. Feemster, Y. Fang, D. M. Dawson, Disturbance rejection for a magnetic levitation system, IEEE/ASME Transactions on Mechatronics 11 (6) (2006) 709–717.

[65] R. Vepa, Nonlinear Control of Robots and Unmanned Aerial Vehicles: An Integrated Approach, CRC Press, Boca Raton, FL, USA, 2016.

[66] S. Li, Y. Wang, J. Tan, Y. Zheng, Adaptive RBFNNs/integral sliding mode control for a quadrotor aircraft, Neurocomputing 216 (2016) 126–134.