Neural Network Based Model Reference Adaptive Attitude Control for a Micro Unmanned Air Vehicle

Salvatore Rosario Bassolillo, Gennaro Raspaolo, Luciano Blasi, Egidio D'Amato, Immacolata Notaro

***Abstract*—During recent decades, Unmanned Aerial Vehicles (UAVs) has increased their success in several areas of applications thanks to their versatility. Furthermore, the growing availability of low-cost electronics has pushed their use in civil consumer applications. Attitude control is one of the needed tasks to effectively carry out missions, whose accuracy and adaptability to high payload imbalances or atmospheric disturbances are fundamental requirements. This paper focuses on the design of a flight control scheme based on Model Reference Adaptive Control with Neural Networks. The effectiveness of the proposed controller is assessed through experimental tests carried out on a Crazyflie 2.1 quadrotor.**

I. Introduction

The use of Unmanned Aerial Vehicles (UAVs) has exponentially grown in the last decades in both civil and military applications. In particular, multirotor drones offer advantages when used for missions in hazardous environments [1] if compared to fixed-wing aircraft, thanks to their characteristics, such as the ability to take-off and land vertically, fly in confined spaces and hover over a specified area. In civil applications, micro and mini UAVs are adopted for tasks such as aerial photography, mapping, weather forecasting, Search and Rescue (SAR) applications in post disaster scenarios [2]–[8].

However, this kind of aircraft needs a reliable and effective attitude controller to successfully carry out a mission, with the ability to ensure stable flight even under operational conditions different from the nominal ones, such as in the presence of atmospheric disturbances. Because of their ease of implementation, the most conventional control methods are based on Proportional-Integral-Derivative (PID) controllers [9], [10], particularly suitable in quasi steady-conditions. This kind of controller offers an intuitive interface to be calibrated and for this reason it is still the most used approach for commercial drones. Over the years, researchers found several alternative to improve performance and reliability of such systems, using optimal control [11]–[15], H_{inf} [16]–[18], Nonlinear Dynamic Inversion [19].

However, in the presence of unknown disturbance, such control systems can fail [20]–[22]. To overcome these difficulties, several adaptive strategies have been proposed by researcher, such as Adaptive Control [23], NN-based Nonlinear Decoupling [24], Model Reference Adaptive Control [25] and Neural Networks (NN) based controllers. [26]. NN-MRAC [27], [28] is particularly interesting in fast dynamics applications, thanks to its ability to automatically adjust the parameters of the control model in real-time when the plant parameters are unknown or suddenly change during operations [29]. This adaptive control technique has been adopted by several authors over the years. In [30], the authors proposed an intelligent Model Reference Adaptive Control (MRAC) based on a neural network for robust tracking control of quadrotor UAV under external disturbances and parameter variations. The authors in [31] proposed a control architecture based on MRAC enabling trajectory tracking for quadcopters, despite uncertainties about the inertial properties of the aircraft and the presence of unknown and unsteady payloads. To enhance adaptability to changing conditions, ensuring efficient and reliable control, the incorporation of Neural Networks for attitude control purposes was also introduced in [32]. The adaptability characteristics of the MRAC controllers were also used in [33], where the authors developed a controller for a quadcopter capable of autonomously adapting control parameters to various operational scenarios different from the reference model, without any external intervention from the pilot. To facilitate the adaptation to dynamic environments, a quadrotor position controller leveraging Neural Networks trained via Reinforcement Learning was proposed in [26].

The main aim of this paper is to design an attitude control system based on a Model Reference Adaptive Control (MRAC) integrated with Neural Networks, to enhance flight control precision and adaptability to unknown external disturbance. To evaluate the performance of the control algorithm, experimental results are presented, comparing two different NN architectures in the presence of a weight unbalance over a commercial quadrotor. In particular, the main contributions of the proposed work are twofold:

- a performance comparison of MRAC with several activation functions;
- design and implementation of experimental tests on a commercial micro quadrotor, such as Crazyflie 2.1+.

The paper is organized as follows: Section II introduces the adopted aircraft model. Section III describes the proposed flight control algorithm. Finally, Section IV presents the pre-

Salvatore Rosario Bassolillo and Egidio D'Amato are with the Department of Science and Technology, Universitá degli Studi di Napoli "Parthenope", Napoli, Italy; e-mail: salvatorerosario.bassolillo@collaboratore.uniparthenope.it (SRB), egidio.damato@uniparthenope.it (ED)

Luciano Blasi, Immacolata Notaro and Gennaro Raspaolo are with the Department of Engineering, Universitá degli Studi della Campania "L. Vanvitelli", 81031, Aversa (CE), Italy; e-mail:luciano.blasi@unicampania.it (LB), immacolata.notaro@unicampania.it (IN), gennaro.raspaolo@unicampania.it (GR)

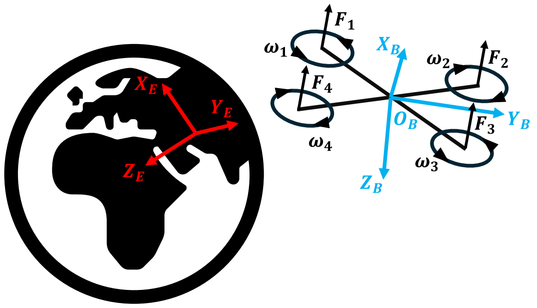

Fig. 1: Definition of NED and Body reference frames.

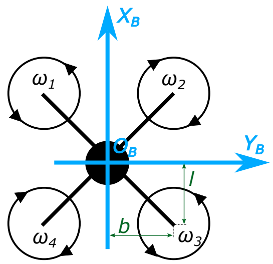

Fig. 2: Quadrotor configuration.

liminary results of an experimental test campaign.

II. Aircraft model

To define the motion of the aircraft and its orientation, two reference frames are considered: an inertial earth-fixed frame, denoted as E, and a body-fixed frame, denoted as B, whose origin O_B is located at the Center of Gravity (CG) of the quadrotor (see Fig. 1).

Before presenting the rigid body equations useful to model the motion of the aircraft, the following assumptions have been considered: (a) the quadrotor structure is symmetrical and rigid, (b) each propeller has a fixed pitch angle, and (c) the center of mass of the quadcopter is coincident with its geometrical center, O_B.

The state of the quadrotor $\mathbf{X} = [\mathbf{V}^T, \mathbf{\Omega}^T, \mathbf{\Theta}^T, \mathbf{z}^T]^T$ is composed of:

- the velocity vector in the body reference frame, $\mathbf{V}$,
- the angular rate in the body frame, $\mathbf{\Omega} = [\omega_x, \omega_y, \omega_z]^T$,
- the attitude in terms of Euler angles, $\mathbf{\Theta} = [\phi, \theta, \psi]^T$,
- the position $\mathbf{z} = [x_E, y_E, z_E]^T$ in the earth-fixed frame.

The governing equations, in the body-fixed frame B, describing the dynamic model of the UAV can subsequently be expressed as:

$$m\left(\dot{\mathbf{V}} + \mathbf{\Omega} \times \mathbf{V}\right) = \mathbf{F}\left(\mathbf{V}, \mathbf{\Omega}, \mathbf{\Theta}, \mathbf{u}\right) \tag{1}$$

$$\mathbf{J}\dot{\mathbf{\Omega}} + \mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega} = \mathbf{T}\left(\mathbf{V}, \mathbf{\Omega}, \mathbf{u}\right) \tag{2}$$

$$\dot{\mathbf{z}} = \mathbf{R}_{BE}^{-1}(\mathbf{\Theta})\mathbf{V} \tag{3}$$

$$\dot{\mathbf{\Theta}} = \mathbf{H}(\mathbf{\Theta})\mathbf{\Omega} \tag{4}$$

where m is the mass of the quadrotor, $\mathbf{J} = \mathrm{diag}(I_{xx}, I_{yy}, I_{zz})$ represents the inertia matrix, assumed diagonal and constant in the Body frame. The vector $\dot{\mathbf{V}}$ denotes the time derivative of the vector $\mathbf{V}$ as observed in the body frame, while $\mathbf{F}$ represents the vector of external forces, which depends both on the motion variables and the vector of propellers rotational speed used for vehicle control, $\mathbf{u}$. The vector $\mathbf{T}$ represents the applied moments with respect to the Center of Gravity (CG), and $\mathbf{R}_{BE}$ denotes the rotation matrix from the earth-fixed frame to the body frame, which depends on the attitude of the UAV.

The matrix $\boldsymbol{H}(\mathbf{\Theta})$ is the transformation matrix for angular velocities, defined as follows:

$$\boldsymbol{H}(\mathbf{\Theta}) = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \tag{5}$$

The external forces and moments in (1) and (2) can be decomposed as follows:

$$\mathbf{F}\left(\mathbf{V}, \mathbf{\Omega}, \mathbf{\Theta}, \mathbf{u}\right) = \mathbf{F_g}(\mathbf{\Theta}) + \mathbf{F_p}(\mathbf{V}, \mathbf{\Omega}, \mathbf{u}) + \mathbf{F_a}(\mathbf{V}, \mathbf{\Omega}) \tag{6}$$

$$\mathbf{T}\left(\mathbf{V}, \mathbf{\Omega}, \mathbf{u}\right) = \mathbf{T_p}(\mathbf{V}, \mathbf{\Omega}, \mathbf{u}) + \mathbf{T_a}(\mathbf{V}, \mathbf{\Omega}) \tag{7}$$

where $\mathbf{F_g} = \mathbf{R}_{BE}[0, 0, mg]^T$ represents the gravitational force, $\mathbf{F_p}$ and $\mathbf{T_p}$ are the propulsive forces and moments, whereas $\mathbf{F_a}$ and $\mathbf{T_a}$ consist of the aerodynamic components of forces and moments, respectively.

In practical implementation, PWM signals are involved to control rotational speeds. Experimental observations revealed a proportional relationship between the rotor thrust/torque and the control signal, through a coefficient k_f/k_t.

$$\begin{matrix} F_i = k_f u_i \\ T_i = k_t u_i \end{matrix} \qquad \forall i = 1, 2, 3, 4 \tag{8}$$

The body-frame components of the propulsive force and torque, mainly dependent on the propeller rotational speed of the quadrotor and denoted as $\mathbf{F_p} = [0, 0, F_{pz}]^T$ and $\mathbf{T_p} = [L_p, M_p, N_p]^T$, can be modeled as a combination of the rotors thrust, F_i, and torque, T_i with $i = 1, 2, 3, 4$.

$$\begin{bmatrix} F_{pz} \\ L_p \\ M_p \\ N_p \end{bmatrix} = \begin{bmatrix} k_f & k_f & k_f & k_f \\ bk_f & -bk_f & -bk_f & bk_f \\ lk_f & lk_f & -lk_f & -lk_f \\ k_t & -k_t & k_t & -k_t \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{9}$$

The parameters b and l denote the distance of the rotors center from CG along the y and x axes, respectively (see Fig. 2).

III. Flight Control Algorithm

The proposed Flight Control System (FCS), whose schematic representation is depicted in Fig. 3, is composed of three main components:

- A Control Allocator (CA), responsible for distributing the control effort, in terms of the requested forces and moments, among the available actuators.

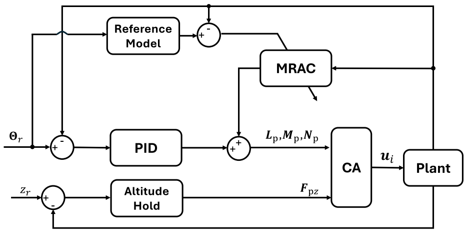

Fig. 3: General scheme of the Flight Control System.

- An inner loop for the attitude control of the vehicle, consisting of a PID and an MRAC-based compensator, able to provide virtual moments through the CA and characterized by fast dynamics.
- An external loop, characterized by slow dynamics, for the altitude control, able to generate virtual force commands to the Control Allocator.

A. Control Allocator

The Control Allocator, designed by manipulating (9), is responsible to translate virtual forces and moments into normalized input commands ready to be translated into PWM signals, as expressed in (10).

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{4k_f} & \frac{1}{4bk_f} & \frac{1}{4lk_f} & \frac{1}{4k_t} \\ \frac{1}{4k_f} & -\frac{1}{4bk_f} & \frac{1}{4lk_f} & -\frac{1}{4k_t} \\ \frac{1}{4k_f} & -\frac{1}{4bk_f} & -\frac{1}{4lk_f} & \frac{1}{4k_t} \\ \frac{1}{4k_f} & \frac{1}{4bk_f} & -\frac{1}{4lk_f} & -\frac{1}{4k_t} \end{bmatrix} \begin{bmatrix} F_{pz} \\ L_p \\ M_p \\ N_p \end{bmatrix} \tag{10}$$

B. Inner Loop Control

The inner loop control allows stabilizing the attitude of the quadrotor through the combination of a baseline PID model and an adaptive controller based on MRAC to compensate for any nonlinearities.

PID controllers are widely used in aerospace applications for their simplicity and robustness, especially for attitude control problems of quadcopter [34], [35].

Let us consider the linear system defined in (11), obtained through the linearization of the equations describing the dynamic model of the UAV, (1) to (4).

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\zeta(t) \tag{11}$$

The vector $\mathbf{x} = [\mathbf{\Omega}^T, \mathbf{\Theta}^T]^T \in \mathbb{R}^m$ is a reduced state vector for attitude control purposes, $\zeta = [L_p, M_p, N_p]^T \in \mathbb{R}^n$ is the control input, $\mathbf{A} \in \mathbb{R}^{mxm}$ and $\mathbf{B} \in \mathbb{R}^{mxn}$ are unknown constant matrices.

The Altitude Hold block (see Fig. 3), based on a PID controller, is responsible to maintain the desired altitude. The PID controller for attitude stabilization is able to provide the requested control signals in terms of requested moments, L_p, M_p, and N_p, to the Control Allocator, providing the appropriate input commands to the quadcopter (see (10)).

The presence of nonlinearities due to weight unbalance or engine failures, can lead to degraded controller performance. To address this issue, the MRAC is used to dynamically compensate the PID control signal, ensuring that the system is able to follow a reference model subjected to the same input conditions.

To account for system nonlinearities, let us introduce an uncertainty function, $\mathbf{f_T}(\mathbf{X})$, into the model described in (11). This way, the set of equations describing the state dynamics is modified in accordance to (12).

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{\Lambda}\left(\zeta(t) + \mathbf{f_T}(\mathbf{X})\right) \tag{12}$$

The diagonal matrix $\mathbf{\Lambda}$ is designed to take into account potential uncertainties in the effectiveness of the control system, whose diagonal elements are strictly positive.

Since the model shown in (11) is a linear approximation of a nonlinear system, valid in a small region around a particular flight condition, the non-linear function $\mathbf{f_T}(\mathbf{X})$ allows to take into account the uncertainties of the system and disturbances that may arise from flight conditions different from those considered during linearization. In this regard, $\mathbf{f_T}(\mathbf{X})$ can be defined according to (13).

$$\mathbf{f_T}(\mathbf{X}) = \mathbf{T_a}(\mathbf{V}, \mathbf{\Omega}) + \mathbf{T_p}(\mathbf{V}, \mathbf{\Omega}) + \mathbf{T_m}(\mathbf{V}, \mathbf{\Omega}) \tag{13}$$

The aerodynamic components, $\mathbf{T_a}$, account for uncertainties arising from an incomplete aerodynamic model under flight conditions different from the hovering phase. Similarly, the propulsive components, $\mathbf{T_p}$, address uncertainties related to actuator characterization, as well as disturbances related to a loss of actuator performance. Finally, $\mathbf{T_m}$ takes into account uncertainties about inertial and miscellaneous effects not related to aerodynamic and propulsive components.

In order to ensure a desired behaviour, the adaptive control law must guide the system defined in (12) to track the state $\mathbf{x}_r$ of a given reference plant model, as defined in (14), even in the presence of disturbances and uncertainties.

$$\dot{\mathbf{x}}_r(t) = \mathbf{A}_r\mathbf{x}_r(t) + \mathbf{B}_r\mathbf{r}(t) \tag{14}$$

The matrix $\mathbf{A}_r \in \mathbb{R}^{mxm}$ is a stable matrix, $\mathbf{B}_r \in \mathbb{R}^{mxn}$, and $\mathbf{r}(t) \in \mathbb{R}^n$ is the reference input vector.

Considering a neural network with two layers, a single layer of N hidden neurons and a linear output layer with 1 neuron (see Fig. 4), the MRAC controller computes the control input according to (15).

$$\mathbf{u}_{MRAC}(t) = -\mathbf{w}(t)^T\Phi(\mathbf{x}) \tag{15}$$

$\Phi(\mathbf{x}) = [\Phi_1(\mathbf{x}), \Phi_2(\mathbf{x}), ..., \Phi_N(\mathbf{x}), 1]^T$ represents the vector of activation functions plus a unit term used to add a bias, whereas $\mathbf{w}(t) \in \mathbb{R}^{(N+1)xn}$ is the hidden layer weights matrix [36]. The rectangular matrix $\mathbf{w}(t)$ is defined as:

$$\mathbf{w}(t) = \begin{bmatrix} w_{1,1}(t) & \dots & w_{1,n}(t) \\ \dots & \dots & \dots \\ w_{N+1,1}(t) & \dots & w_{N+1,n}(t) \end{bmatrix} \tag{16}$$

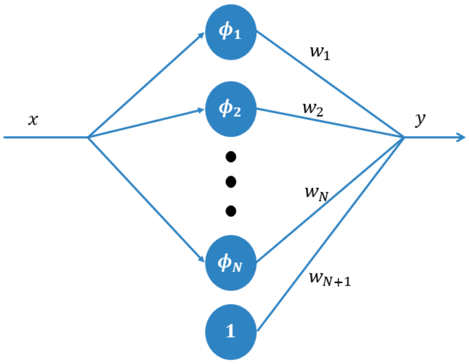

Fig. 4: Graphical representation of the adopted neural network.

TABLE I: Quadcopter main characteristics.

	Value
m $[kg]$	0.031
l $[m]$	0.098
b $[m]$	0.098
$\mathbf{J}$ $[kg * m^2]$	$diag(2.4 * 10^{-5}, 2.4 * 10^{-5}, 3.2 * 10^{-5})$

TABLE II: MRAC controller parameters.

	Value
N	10
Γ	$6 \cdot 10^{-4}$
$\mathbf{w}(0)$	$zeros(N+1, n)$
α	$[-\frac{\pi}{6} : N : \frac{\pi}{6}]$
σ	0.05

The MRAC controller updates $\mathbf{w}(t)$ using the tracking error $\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{x}_r(t)$ between the vehicle and the reference model as follows.

$$\dot{\mathbf{w}}(t) = \Gamma \Phi \mathbf{e}^T \mathbb{P} \mathbf{B} - \sigma \mathbf{w} \tag{17}$$

The parameter Γ represents the adaptation rate of the weights, $\mathbb{P}$ is the solution of the Lyapunov equation [36], and σ ensures that the weights remain bounded without the need for persistent excitation [32], [37].

The total amount of attitude control signal provided to the plant is a combination of PID and MRAC outputs as defined in (18).

$$\mathbf{u}_{\Theta}(t) = \mathbf{u}_{PID}(t) + \mathbf{u}_{MRAC}(t) \tag{18}$$

C. Outer Loop Control

The outer loop control block, based on a PID controller [34], provides the appropriate control effort, in terms of force command, F_{pz}, to track a reference altitude, z_r.

IV. Experimental Simulation Results

To validate the proposed control architecture, preliminary experimental results were obtained considering a Crazyflie 2.1 quadrotor, whose main simulation parameters, presented in [38], are summarized in Table I. PID parameters were optimized in simulation, by using a digital twin of the quadrotor. In particular, to test the attitude control system, the quadrotor was constrained to a custom-built tilting platform, specifically designed to allow only attitude maneuvers, as shown in Fig. 5.

Table II shows the operating parameters for the proposed MRAC controller. A comparison between two neural architectures were conducted, specifically varying the activation function. The first uses a shifted ReLU (DReLU), a slightly modified version of the ReLU-type, which allows for a greater flexibility with respect to the ReLU-type in the distribution of functions within the disturbance domain. The second employs a RBF-type (Radial Basis Function), a computationally heavier activation function in terms of computational burden, but able to better approximate the non-linearity of the disturbance. As a reference, the results are compared with a conventional PID-based control strategy.

DReLU-type [39] and RBF activation functions can be expressed as follows:

$$DReLU(x) = \max(\alpha, x) \tag{19}$$

$$RBF(x) = e^{-(x+\alpha)^2} \tag{20}$$

In both cases, the parameter α is used to cover the disturbance domain.

To simulate the effect of an unknown disturbance on the model, we introduced an external force inducing an unbalancing moment by mounting a weight of $2 \cdot 10^{-3} kg$ below motor M1, as shown in Fig. 5.

The performance of the proposed control architecture is evaluated considering a sequence of distinct roll and pitch maneuvers around the body axis. Specifically, the maneuvers are imposed by applying an initial sequence of doublets with increasing amplitude, ranging from $5deg$ to $15deg$, with a duration of $3s$. After a training interval of $60s$, two additional maneuvers are performed, with a peak amplitude of $15deg$ and $10deg$ and a duration of $8s$ and $5s$, respectively.

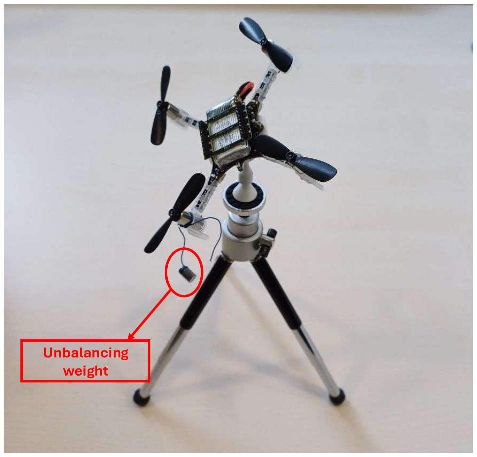

Fig. 5: Experimental setup used to test the control algorithms.

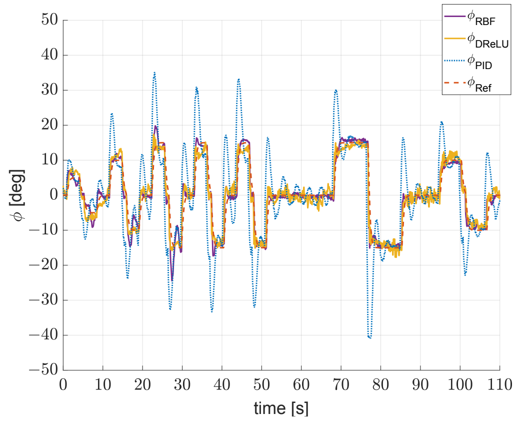

Fig. 6: Experimental results on ϕ channel, with and without MRAC controller. ϕ_{RBF} represents the response of the model with MRAC controller and RBF activation function. ϕ_{DReLU} is the response of the model with MRAC controller and DReLU activation function. ϕ_{PID} is the response of the system without the MRAC controller. ϕ_{Ref} indicates the desired response used as a reference model for the MRAC.

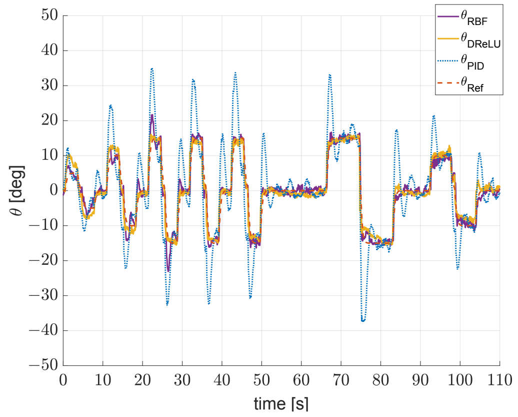

Fig. 7: Experimental results on θ channel, with and without MRAC controller. θ_{RBF} represents the response of the model with MRAC controller and RBF activation function. θ_{DReLU} is the response of the model with MRAC controller and DReLU activation function. θ_{PID} is the response of the system without the MRAC controller. θ_{Ref} indicates the desired response used as a reference model for the MRAC.

As depicted in Fig. 6, the use of a PID controller suffers of a significant bias around the roll axis due to the unbalancing moment, whereas the MRAC-based architectures, both with DReLU and RBF activation functions, are able to effectively compensate for this bias, providing satisfactory results, with limited overshoot in tracking error. A similar behavior can be observed during the pitch maneuver (see Fig. 7), where the MRAC controller is able to effectively mitigate the external disturbance, outperforming the performance of the PID controller.

It is worth noting that the use of DReLU activation function does not impact the performance of the MRAC controller in terms of attitude error, providing a good trade-off between the performance and the computational cost, given the linear nature of DReLU, less demanding compared to RBFs [40].

TABLE III: Performance comparisons of the proposed DReLU-based MRAC control architecture in terms of the average error, μ, and standard deviation, σ, for roll (μ^{ϕ} and σ^{ϕ}) and pitch (μ^{θ} and σ^{θ}) attitude control.

	μ^{ϕ} $[deg]$	σ^{ϕ} $[deg]$	μ^{θ} $[deg]$	σ^{θ} $[deg]$
DReLU	1.81	2.54	1.54	2.18
RBF	1.57	2.54	1.33	2.01
PID	7.03	10.79	5.70	8.20

To further highlight the good performance of the DReLU-based MRAC controller, two performance indices were introduced. Denote with η the generic component of the vector $\mathbf{\Theta}$, let us define the error between the response of the reference model, η_{Ref}, and the response of the system as in (21).

$$e^{\eta}(t) = \|\eta_{Ref}(t) - \eta(t)\| \tag{21}$$

The average error μ^{η} in a time window composed by n_t time steps is expressed in (22).

$$\mu^{\eta} = \frac{1}{n_t}\sum_{t=1}^{n_t} e^{\eta}(t) \tag{22}$$

Then, it is possible to define the standard deviation σ^{η} with respect to the average error as in (23).

$$\sigma^{\eta} = \sqrt{\frac{1}{(n_t)}\sum_{t=1}^{n_t}\left(e^{\eta}(t) - \mu^{\eta}\right)^2} \tag{23}$$

Table III shows a tabular comparison in terms of standard deviation and mean absolute error.

V. Conclusions

In this paper, the design and testing of a flight control scheme based on PID and MRAC with Neural Networks is presented. The effectiveness of the proposed control algorithm has been validated through experimental simulations on a Crazyflie 2.1 quadcopter, performing a series of maneuvers around roll and pitch axis with the presence of an external unbalancing weight mounted on one arm of the drone. Results have demonstrated that the proposed MRAC-based control scheme outperforms traditional control architectures based on PID control, ensuring limited tracking errors and avoiding the excessive overshoot shown in the response to the input commands without the use of the MRAC. As future work different neural networks architecture will be tested and a sensitivity analysis will be carried out to check how the parameters of the neural network influence the performance.

References

[1] A. Jaimes, S. Kota, and J. Gomez, "An approach to surveillance an area using swarm of fixed wing and quad-rotor unmanned aerial vehicles uav (s)," in *2008 IEEE International Conference on System of Systems Engineering*, pp. 1–6, IEEE, 2008.

[2] J. Guo, X. Liu, L. Bi, H. Liu, and H. Lou, "Un-yolov5s: A uav-based aerial photography detection algorithm," *Sensors*, vol. 23, no. 13, p. 5907, 2023.

[3] F. Nex and F. Remondino, "Uav for 3d mapping applications: a review," *Applied geomatics*, vol. 6, pp. 1–15, 2014.

[4] A. Thibbotuwawa, G. Bocewicz, P. Nielsen, and B. Zbigniew, "Planning deliveries with uav routing under weather forecast and energy consumption constraints," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 820–825, 2019.

[5] A. Ferraro and V. Scordamaglia, "Coordinated trajectory planning for ground-based multi-robot systems in exploration and surveillance missions," in *2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense)*, pp. 131–136, IEEE, 2023.

[6] A. Ferraro, V. Nardi, E. D'Amato, I. Notaro, and V. Scordamaglia, "Multi-drone systems for search and rescue operations: problems, technical solutions and open issues," in *2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense)*, pp. 159–164, IEEE, 2023.

[7] V. Scordamaglia, A. Ferraro, F. Tedesco, and G. Franzè, "Set-theoretic approach for autonomous tracked vehicles involved in post-disaster first relief operations," in *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 2006–2011, IEEE, 2024.

[8] A. Ferraro, C. De Capua, and V. Scordamaglia, "Integrated trajectory planning and fault detection for a skid-steered mobile robot deployed in post-disaster scenarios," in *2024 IEEE International Workshop on Technologies for Defense and Security (TechDefense)*, pp. 413–418, IEEE, 2024.

[9] H. Lim, J. Park, D. Lee, and H. J. Kim, "Build your own quadrotor: Open-source projects on unmanned aerial vehicles," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 33–45, 2012.

[10] Q. Jiao, J. Liu, Y. Zhang, and W. Lian, "Analysis and design the controller for quadrotors based on pid control method," in *2018 33rd youth academic annual conference of Chinese association of automation (YAC)*, pp. 88–92, IEEE, 2018.

[11] Z. Shulong, A. Honglei, Z. Daibing, and S. Lincheng, "A new feedback linearization lqr control for attitude of quadrotor," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 1593–1597, IEEE, 2014.

[12] S. Khatoon, D. Gupta, and L. Das, "Pid & lqr control for a quadrotor: Modeling and simulation," in *2014 international conference on advances in computing, communications and informatics (ICACCI)*, pp. 796–802, IEEE, 2014.

[13] G. Franze, M. Mattei, L. Ollìo, and V. Scordamaglia, "A norm-bounded robust mpc strategy with partial state measurements," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 7449–7454, 2014.

[14] G. Franze, M. Mattei, and V. Scordamaglia, "Embedding norm-bounded model predictive control allocation strategy for the high altitude performance demonstrator (hapd) aircraft," in *52nd IEEE Conference on Decision and Control*, pp. 6409–6414, IEEE, 2013.

[15] G. Franze, M. Mattei, L. Ollio, and V. Scordamaglia, "A receding horizon control scheme with partial state measurements: Control augmentation of a flexible uav," in *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*, pp. 158–163, IEEE, 2013.

[16] A. Jafar, S. Fasih-UR-Rehman, S. Fazal-UR-Rehman, N. Ahmed, and M. U. Shehzad, "A robust h control for unmanned aerial vehicle against atmospheric turbulence," in *2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*, pp. 1–6, IEEE, 2016.

[17] M. Babar, S. Ali, M. Shah, R. Samar, A. Bhatti, and W. Afzal, "Robust control of uavs using h control paradigm," in *2013 IEEE 9th International Conference on Emerging Technologies (ICET)*, pp. 1–5, IEEE, 2013.

[18] V. Scordamaglia, A. Sollazzo, and M. Mattei, "Fixed structure flight control design of an over-actuated aircraft in the presence of actuators with different dynamic performance," *IFAC Proceedings Volumes*, vol. 45, no. 13, pp. 127–132, 2012.

[19] H. Das, K. Sridhar, and R. Padhi, "Bio-inspired landing of quadrotor using improved state estimation," *IFAC-PapersOnLine*, vol. 51, no. 1, pp. 462–467, 2018.

[20] Y. Jiang, Q. Hu, and G. Ma, "Adaptive backstepping fault-tolerant control for flexible spacecraft with unknown bounded disturbances and actuator failures," *ISA transactions*, vol. 49, no. 1, pp. 57–69, 2010.

[21] A. Ferraro and G. Martino, "Time-delay approach to coordinate movements of mobile robots subject to uncertainties and external disturbances," in *2024 IEEE International Workshop on Technologies for Defense and Security (TechDefense)*, pp. 462–467, IEEE, 2024.

[22] V. Scordamaglia and A. Ferraro, "A set-based method for planning coordinated trajectories for skid-steered robotic units subject to constraints, uncertainties and external disturbances," *IEEE Access*, 2025.

[23] M. C. P. Santos, C. D. Rosales, J. A. Sarapura, M. Sarcinelli-Filho, and R. Carelli, "An adaptive dynamic controller for quadrotor to perform trajectory tracking tasks," *Journal of Intelligent & Robotic Systems*, vol. 93, pp. 5–16, 2019.

[24] Y. Fu and T. Chai, "Neural-network-based nonlinear adaptive dynamical decoupling control," *IEEE transactions on neural networks*, vol. 18, no. 3, pp. 921–925, 2007.

[25] B. Whitehead and S. Bieniawski, "Model reference adaptive control of a quadrotor uav," in *AIAA Guidance, Navigation, and Control Conference*, p. 8148, 2010.

[26] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2096–2103, 2017.

[27] N. T. Nguyen and N. T. Nguyen, *Model-reference adaptive control*. Springer, 2018.

[28] R. B. Anderson, J. A. Marshall, and A. L'Afflitto, "Novel model reference adaptive control laws for improved transient dynamics and guaranteed saturation constraints," *Journal of the Franklin Institute*, vol. 358, no. 12, pp. 6281–6308, 2021.

[29] D. Lara, G. Romero, R. Ibarra, A. Sánchez, and C. Pegard, "Onboard system for flight control of a small uav," in *World Automation Congress 2012*, pp. 1–6, IEEE, 2012.

[30] M. M. Madebo, C. M. Abdissa, L. N. Lemma, and D. S. Negash, "Robust tracking control for quadrotor uav with external disturbances and uncertainties using neural network based mrac," *IEEE Access*, 2024.

[31] R. B. Anderson, J. A. Marshall, and A. L'Afflitto, "Constrained robust model reference adaptive control of a tilt-rotor quadcopter pulling an unmodeled cart," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 39–54, 2020.

[32] E. D'Amato, G. Tartaglione, L. Blasi, M. Mattei, *et al.*, "Nonlinear dynamic inversion with neural networks for the flight control of a low cost tilt rotor uav," in *30th Congress of the International Council of the Aeronautical Sciences, ICAS 2016*, International Council of the Aeronautical Sciences, 2016.

[33] E. A. Niit and W. J. Smit, "Integration of model reference adaptive control (mrac) with px4 firmware for quadcopters," in *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, pp. 1–6, IEEE, 2017.

[34] M. Santos, V. Pereira, A. Ribeiro, M. Silva, M. do Carmo, V. Vidal, L. Honório, A. Cerqueira, and E. Oliveira, "Simulation and comparison between a linear and nonlinear technique applied to altitude control in quadcopters," in *2017 18th International Carpathian Control Conference (ICCC)*, pp. 234–239, IEEE, 2017.

[35] I. Lopez-Sanchez and J. Moreno-Valenzuela, "Pid control of quadrotor uavs: A survey," *Annual Reviews in Control*, vol. 56, p. 100900, 2023.

[36] P. A. Ioannou and J. Sun, *Robust adaptive control*, vol. 1. PTR Prentice-Hall Upper Saddle River, NJ, 1996.

[37] K. Tsakalis and P. Ioannou, "Adaptive control of linear time-varying plants," *Automatica*, vol. 23, no. 4, pp. 459–468, 1987.

[38] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski, "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering," in *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 37–42, IEEE, 2017.

[39] D. Macêdo, C. Zanchettin, A. L. Oliveira, and T. Ludermir, "Enhancing batch normalized convolutional networks using displaced rectifier linear units: A systematic comparative study," *Expert Systems with Applications*, vol. 124, pp. 271–281, 2019.

[40] A. Wuraola, N. Patel, and S. K. Nguang, "Efficient activation functions for embedded inference engines," *Neurocomputing*, vol. 442, pp. 73–88, 2021.