IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Quadrotor Neural Network Adaptive Control: Design and Experimental Validation

Gan Yu, Joel Reis, and Carlos Silvestre, *Senior Member, IEEE*

***Abstract*—This letter presents the design and experimental study of an adaptive nonlinear controller for Unmanned Aerial Vehicles (UAVs) in the presence of unknown time-varying disturbances, and model parametric uncertainty. We employ an adaptive Neural Network (NN), used to approximate the partially unknown system, in tandem with a simple controller designed for trajectory tracking, not of the center of mass, but of a point located along the UAVs vertical body axis. This strategy allows: (i) to avoid the two-subsystems control paradigm generally adopted by conventional UAV controllers; (ii) all control inputs to be defined at once; and (iii) to lump all unknown dynamics from both translational and rotational levels into a single vector term. The weights of the NN are determined online by an adaptive law based on the Lyapunov synthesis method. The tracking and adaptation errors are shown to be uniformly ultimately bounded. Simulation and experimental results, including comparison data, are provided to validate and assess the proposed control solution.**

***Index Terms*—Autonomous agents, robust/adaptive control, artificial neural network control.**

I. Introduction

A. Motivation

A major confluence of advancements in engineering technology and scientific research outcomes has made Unmanned Aerial Vehicles (UAVs) crucial to different branches of society. Thanks to the precise maneuvering capabilities of these vehicles, sophisticated and complex operations are now automated, thus helping to save costs and enhance efficiency. However, since model parametric uncertainty is inevitable and UAVs are often subject to aerodynamic and exogenous disturbances, robust controllers are necessary to ensure correct performance.

A myriad of aerodynamic effects influences the motion of UAVs through the air. Although most of these effects cause only minor perturbations, some, like blade flapping and induced drag (see, e.g., [1], [2]), and ground effect (see, e.g., [3], [4]), are especially detrimental to performance. In this letter, we will resort to Neural Networks (NNs) to estimate these effects.

An often highlighted feature of NNs is their outputting of universal function approximations (cf. [5], [6]). In essence, any continuous bounded function can be approximated arbitrarily closely on a compact set using a NN, given the right set of inputs and gains. For instance, in view of the intrinsic relationship between aerodynamic disturbances acting on the UAV and its system states, these disturbances can be approximated by NNs. Exogenous perturbations, on the other hand, cannot.

Naturally, the computational burden associated with the approximation grows with the number of neurons and hidden layers of the NNs. In many cases, a one-layer NN (i.e., no hidden layer) suffices. Indeed, one-layer NNs with specific activation functions (e.g., sigmoids and radial basis functions) hold the universal approximation property [7]; they are also suitable for the control purposes discussed in this work.

B. Proposed Solution

In this letter, an adaptive NN is employed to approximate the partially unknown system, which features model parametric uncertainty, as well as aerodynamic and external disturbances. Using NNs to compensate for the latter has been addressed in [8], [9]. However, this approach is arguably imprecise since, in theory, NNs are not designed to approximate unknown random functions with no system dependencies. Ultimately, it is the adaptive laws governing the NN weights that implicitly render the networks *sensitive* to external disturbances. This is so because adaptive laws are typically function of the tracking errors. Surprisingly, and often overlooked in the literature, the performance of adaptive NN controllers is very much similar to that of conventional approaches, e.g., Integral Action (IA), also referred to as integral backstepping. Indeed, the theoretical stability guarantees are identical: uniform ultimate boundedness of the system errors. Our goal is to find experimental evidence in favor of NNs that may help to highlight their advantages over conventional approaches. Furthermore, we shall propose a reduced-complexity controller, wherein tracking is performed with respect to a point offset from the center of mass.

The highlights and contributions of our strategy are as follows: (i) the controller requires only up to the 2^{nd}-order time derivative of the reference trajectory, as opposed to the usual 4^{th}-order requirement (see, e.g., [10]); (ii) differentiation of the NN is not required, precluding the need for command filters or tracking differentiators; (iii) both the translational and rotational unknown dynamics are lumped into a single vector term approximated by a single-layer NN; (iv) an adaptive law

Manuscript received: 1 September 2022; Revised 15 December 2022; Accepted 17 February 2023. This paper was recommended for publication by Editor Jaydev P. Desai upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by the Macao Science and Technology, Development Fund under Grant FDCT/0031/2020/AFJ, in part by the University of Macau, Macau, China, under Projects MYRG2020-00188-FST and MYRG2022-00205-FST, and in part by the Fundação para a Ciência e a Tecnologia (FCT) through ISR LARSyS-FCT Project under Grant UIDB/50009/2020. (*Corresponding author: Carlos Silvestre.*)

The authors are with the Department of Electrical and Computer Engineering of the Faculty of Science and Technology, University of Macau, Taipa, Macao {ganyu,joelreis,csilvestre}@um.edu.mo

C. Silvestre is on leave from Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal.

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

for the NN weights is derived, based on the Lyapunov synthesis method, that ensures boundedness of the weights through a projection operator; (v) a rigorous stability proof is given, which shows that the total system error in closed-loop with the control and adaptive laws is uniformly ultimately bounded (UUB); and (vi) extensive simulation and experimental results using a quadrotor UAV are provided.

C. Related Works

Traditional methods used to compensate for unknown dynamics include IA [11], Disturbance Observer Based Control (DOBC) [12], and Active Disturbance Rejection Control (ADRC) [13], to name just a few. For instance, in [14], [15], IAs are derived, stemming from the backstepping technique, that are able to deal with model uncertainty and/or unknown time-varying external disturbances. A robust attitude controller using DOBC is proposed in [16] to maneuver a flapping-wing UAV under external disturbances. The work in [17] reports an ADRC strategy for a quadrotor flying under wind gusts.

These three techniques have in common the fact that the controller design can be separated from the disturbance observer design (in DOBC and ADRC) as well as from the adaptive law design (in IA). Similarly, under model uncertainty and external disturbances, a nonlinear model predictive controller augmented with an $\mathcal{L}_1$ adaptive controller is proposed in [18] in order to perform highly aggressive racing trajectories.

Applications with adaptive NNs are also aplenty, having been around for more than two decades, pioneered by the works of [19] and [20]. Recently, they have gained traction, with a deluge of works on the topic of automata [21]–[34].

NNs can learn online the complete dynamics of a UAV, including aerodynamic disturbances [35]. For example, the work in [8] employs two radial basis function NNs (RBFNNs) to approximate and compensate for the unmodeled translational and rotational dynamics of a helicopter. RBFNNs are also adopted in [36], [37] in the context of quadrotor UAVs subject to unknown dynamics. However, extra care must be taken with NNs that are used inside feedback control loops: their weights must remain bounded. This matter is neglected in [8], [36], [37], whose findings are limited to a theoretical scope.

By using multi-layer NNs, the geometric controller developed in [10] is capable of compensating for unknown wind disturbances. Tracking errors are shown to be UUB and the solution is validated with experiments. However, in addition to the inherent complexity of multi-layer NNs, the controller requires up to the 4^{th}-order time derivative of the reference trajectory, and no clue is given as to obtain the derivatives of the NN. High differentiability class orders are prone to rendering control actuations too sensitive, even to slight variations in the system. This trait is also manifested in [8], [36], [37].

II. Framework

A. Notation

Throughout the article, a bold symbol stands for a multidimensional variable. The transpose operator is denoted by $(\bullet)^{\mathsf{T}}$, and the trace function by $\mathrm{tr}(\bullet)$. The set of unit vectors on $\mathbb{R}^3$ is denoted by $\mathsf{S}(2)$. The special orthogonal group of order three is denoted by $\mathsf{SO}(3) := \{\mathbf{X} \in \mathbb{R}^{3\times 3} : \mathbf{X}\mathbf{X}^{\mathsf{T}} = \mathbf{X}^{\mathsf{T}}\mathbf{X} = \mathbf{I},\ \det(\mathbf{X}) = 1\}$. The skew-symmetric matrix representation of $\mathbf{a} \in \mathbb{R}^3$ is denoted by $\mathbf{S}(\mathbf{a})$; given $\mathbf{b} \in \mathbb{R}^3$, one has (the cross-product) $\mathbf{a} \times \mathbf{b} = \mathbf{S}(\mathbf{a})\mathbf{b}$. Given $\mathbf{x} \in \mathbb{R}^n$, with $\mathbf{x} = [x_1, x_2, \ldots, x_n]^{\mathsf{T}}$, its Euclidean norm is denoted by $\|\mathbf{x}\| = \sqrt{\mathbf{x}^{\mathsf{T}}\mathbf{x}}$. The operator $\mathrm{diag}\,(\mathbf{x})$ returns a matrix with $\mathbf{x}$ along the main diagonal. In order to simplify expressions, whenever the argument of a scalar operator is a vector, the operator is applied element-wise to each entry of its vector argument, e.g., $\tanh(\mathbf{x}) := [\tanh(x_1), \tanh(x_2), \ldots, \tanh(x_n)]^{\mathsf{T}}$. We define the smooth operator $\mathbf{T}(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ as $\mathbf{T}(\mathbf{x}) := \mathrm{diag}\,(\tanh(\mathbf{x}/\epsilon))$, where $\epsilon > 0$, with $\|\mathbf{T}(\mathbf{x})\| \leq 1$. The symbol $\odot$ is used to denote the Hadamard (element-wise) product, i.e., $\mathbf{a} \odot \mathbf{b} = \mathrm{diag}\,(\mathbf{a})\,\mathbf{b}$, and the component-wise division of the same dimension, i.e., the Hadamard division, is represented by the symbol $\oslash$, such that $\mathbf{a} \oslash \mathbf{b} = \mathrm{diag}\,(\mathbf{b})^{-1}\,\mathbf{a}$. The symbol $\mathbf{0}$ denotes a matrix/vector of zeros, the symbol $\mathbf{1}$ denotes a matrix/vector of ones, and the symbol $\mathbf{I}$ denotes an identity matrix, all of appropriate dimensions. The operators $\lambda_{\max}(\bullet)$ and $\lambda_{\min}(\bullet)$ return the maximum and minimum eigenvalues of the argument matrix, respectively. For convenience, $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \in \mathsf{S}(2)$ are unit vectors that form a standard orthonormal basis of $\mathbb{R}^3$, i.e., $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3] = \mathbf{I}$.

B. Neural Network Formulation

Let $\boldsymbol{\nu} \in \mathbb{R}^p$ be a p-dimensional vector, such that there exists a function $\boldsymbol{\phi}(\boldsymbol{\nu}) : \mathbb{R}^p \to \mathbb{R}^3$ that will be approximated through the following linear regression:

$$\boldsymbol{\phi}(\boldsymbol{\nu}) := \boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) + \boldsymbol{\varepsilon}(\boldsymbol{\nu}) \in \mathbb{R}^3, \tag{1}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{n\times 3}$ is an unknown constant parameter matrix that represents the weights of a NN with n neurons given by $\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu})$, where $\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) \in \mathbb{R}^n$ is a vector of known activation functions, and where $\boldsymbol{\varepsilon}(\boldsymbol{\nu}) \in \mathbb{R}^3$ is the NN reconstruction error. The term $\boldsymbol{\nu}$ corresponds to the input of the NN, whose input/output response can be adjusted by tweaking $\boldsymbol{\theta}$, with $\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) := [w_1(\boldsymbol{\nu}), w_2(\boldsymbol{\nu}), \ldots, w_n(\boldsymbol{\nu})]^{\mathsf{T}}$. In this letter, we employ the Gaussian activation function, i.e., $w_i(\boldsymbol{\nu}) := \exp\{-\|\boldsymbol{\nu} - \boldsymbol{\mu}_i\|^2/\sigma^2\} \in \mathbb{R}$, for $i = 1, 2, \ldots, n$, with $\sigma \in \mathbb{R}$ being the width of each Gaussian function, and $\boldsymbol{\mu}_i \in \mathbb{R}^p$ representing the center of the respective field.

Since $\boldsymbol{\theta}$ is unknown, we take $\widehat{\boldsymbol{\theta}} \in \mathbb{R}^{n\times 3}$ as an estimate of $\boldsymbol{\theta}$, and then proceed to define the estimation error as

$$\widetilde{\boldsymbol{\theta}} := \boldsymbol{\theta} - \widehat{\boldsymbol{\theta}} \in \mathbb{R}^{n\times 3}. \tag{2}$$

Assumption 1. *Each dimension of $\boldsymbol{\varepsilon}$ in (1) is bounded by an unknown constant, i.e., $|\mathbf{e}_i^{\mathsf{T}}\boldsymbol{\varepsilon}| \leq \psi_i$, for $i \in \{1, 2, 3\}$.*

Next, let $\boldsymbol{\psi} := [\psi_1, \psi_2, \psi_3]^{\mathsf{T}} \in \mathbb{R}^3$, and further consider $\widehat{\boldsymbol{\psi}} \in \mathbb{R}^3$ to be an estimate of $\boldsymbol{\psi}$. Then, similar to (2), we define

$$\widetilde{\boldsymbol{\psi}} := \boldsymbol{\psi} - \widehat{\boldsymbol{\psi}} \in \mathbb{R}^3. \tag{3}$$

III. Quadrotor Model

A. Rigid body kinematics and dynamics

The classic quadrotor model consists of a rigid body dynamics subjected to gravity and to non-conservative forces and

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

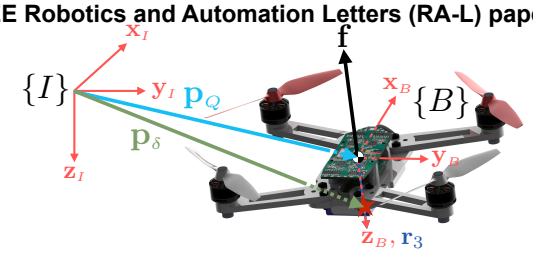

Figure 1. Quadrotor, reference frames, thrust force vector $\mathbf{f}$ drawn along the direction opposite to $\mathbf{r}_3$, and position vectors of both the quadrotor ($\mathbf{p}_Q$) and the *virtual* point ($\mathbf{p}_\delta$), the latter marked with a star.

torques. In order to express the attitude associated with our model, we consider an inertial frame, denoted by $\{I\}$, and a body-fixed frame, denoted by $\{B\}$, whose origin is coincident with the center of mass of the vehicle, see Fig. 1. Throughout the rest of the article, all vectors are expressed in the inertial coordinate frame $\{I\}$ unless otherwise stated.

The kinematic equations describing the translational and rotational motions of the quadrotor are given by

$$\text{Kinematics}: \begin{cases} \dot{\mathbf{p}}_Q = \mathbf{v}_Q, & \text{(4a)} \\ \dot{\mathbf{R}} = \mathbf{R}\mathbf{S}(\boldsymbol{\Omega}), & \text{(4b)} \end{cases}$$

where $\mathbf{p}_Q \in \mathbb{R}^3$ and $\mathbf{v}_Q \in \mathbb{R}^3$ are the position and linear velocity, respectively, of the vehicle, $\boldsymbol{\Omega} \in \mathbb{R}^3$ denotes the vehicle's angular velocity expressed in $\{B\}$, and $\mathbf{R} \in \mathrm{SO}(3)$ represents the rotation matrix from frame $\{B\}$ to frame $\{I\}$.

Assuming all four rotors are identical, we let $T_i \in \mathbb{R}$ denote the thrust generated by the i-th motor, for $i \in \{1,2,3,4\}$. In turn, we denote the net total thrust by $T \in \mathbb{R}$, with $T := \sum_{i=1}^{4} T_i$. Accordingly, the quadrotor thrust force, henceforward denoted by $\mathbf{f} \in \mathbb{R}^3$, is a vector always aligned with the body's z-axis, meaning that $\mathbf{f} := -T\mathbf{R}\mathbf{e}_3$, with $\mathbf{e}_3$ representing the unit vector along the z-axis of $\{B\}$.

Hence, we can write the dynamics of the system as

$$\text{Dynamics}: \begin{cases} m\dot{\mathbf{v}}_Q = \mathbf{f} + \mathbf{f}_d + mg\mathbf{e}_3, & \text{(5a)} \\ \mathbf{J}\dot{\boldsymbol{\Omega}} = -\mathbf{S}(\boldsymbol{\Omega})\mathbf{J}\boldsymbol{\Omega} + \boldsymbol{\tau} + \boldsymbol{\tau}_d, & \text{(5b)} \end{cases}$$

where $m > 0$ and $\mathbf{J} := \mathrm{diag}([j_1, j_2, j_3]) \in \mathbb{R}^{3\times3}$ are the mass and moment of inertia, respectively, of the vehicle, $g > 0$ is the gravitational acceleration, and $\mathbf{f}_d \in \mathbb{R}^3$ and $\boldsymbol{\tau}_d \in \mathbb{R}^3$ stand for aerodynamic and exogenous disturbance forces and torques, respectively, which are assumed bounded. The control inputs to the quadrotor system described in (4) and (5) are the thrust $T \in \mathbb{R}$ and the torque $\boldsymbol{\tau} \in \mathbb{R}^3$.

IV. Trajectory Tracking Controller

In this section, we design an adaptive NN control scheme for the UAV system (4)-(5). A flowchart depicting the overall control process is displayed in Fig. 2.

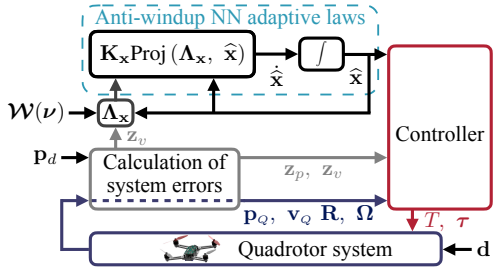

Figure 2. Flowchart of the proposed trajectory tracking control scheme. The symbol $\mathbf{x}$ stands for either the weights $\boldsymbol{\theta}$ or the bounds $\mathcal{V}$.

A. Backstepping Quadrotor Controller

Consider a *virtual* point whose position is given by $\mathbf{p}_\delta := \mathbf{p}_Q + \delta\mathbf{R}\mathbf{e}_3$, for some $\delta \neq 0$. For the sake of illustration, in Fig. 1, the *virtual* point is drawn below the vehicle. We define the position and velocity errors concerning this point as

$$\mathbf{z}_p := \mathbf{p}_\delta - \mathbf{p}_d \in \mathbb{R}^3, \tag{6}$$

where $\mathbf{p}_d \in \mathbb{R}^3$ is a reference curve of class $\mathcal{C}^3$, and

$$\mathbf{z}_v := \dot{\mathbf{z}}_p + k_\Lambda \mathbf{z}_p = \mathbf{v}_Q + \delta\mathbf{R}\mathbf{S}(\boldsymbol{\Omega})\mathbf{e}_3 + k_\Lambda\mathbf{z}_p - \dot{\mathbf{p}}_d \in \mathbb{R}^3. \tag{7}$$

The time derivative of $\mathbf{z}_v$, using (5), follows as $\dot{\mathbf{z}}_v = (1/m)\mathbf{f} - \delta\mathbf{R}\mathbf{S}(\mathbf{e}_3)\mathbf{J}^{-1}\boldsymbol{\tau} + \mathbf{d} - k_\Lambda^2\mathbf{z}_p + g\mathbf{e}_3 - \ddot{\mathbf{p}}_d + \delta\mathbf{R}\mathbf{S}(\mathbf{e}_3)\mathbf{J}^{-1}\mathbf{S}(\boldsymbol{\Omega})\mathbf{J}\boldsymbol{\Omega} + \delta\mathbf{R}\mathbf{S}^2(\boldsymbol{\Omega})\mathbf{e}_3 + k_\Lambda\mathbf{z}_v$, where $\mathbf{d} := (1/m)\mathbf{f}_d - \delta\mathbf{R}\mathbf{S}(\mathbf{e}_3)\mathbf{J}^{-1}\boldsymbol{\tau}_d \in \mathbb{R}^3$. Note from $\dot{\mathbf{z}}_v$ that: (i) T and $\boldsymbol{\tau}$ are already simultaneously available at this stage, which is not the case in conventional methods (see, e.g., [10]); and (ii) both the unknown translational and rotational dynamics can be lumped at once into $\mathbf{d}$.

Since $\mathbf{d}$ is unknown, we will use (1) to obtain an approximation. We can then write $\mathbf{d} = \boldsymbol{\theta}^\mathsf{T}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) + \boldsymbol{\varepsilon}(\boldsymbol{\nu})$[1]. Moreover, according to (2), we can express $\boldsymbol{\theta}^\mathsf{T}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) = \hat{\boldsymbol{\theta}}^\mathsf{T}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) + \tilde{\boldsymbol{\theta}}^\mathsf{T}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu})$. As result, the expression of $\dot{\mathbf{z}}_v$ becomes

$$\begin{aligned} \dot{\mathbf{z}}_v =& (1/m)\mathbf{f} - \delta\mathbf{R}\mathbf{S}(\mathbf{e}_3)\mathbf{J}^{-1}\boldsymbol{\tau} + \hat{\boldsymbol{\theta}}^\mathsf{T}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) + \tilde{\boldsymbol{\theta}}^\mathsf{T}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) \\ &+ \delta\mathbf{R}\mathbf{S}(\mathbf{e}_3)\mathbf{J}^{-1}\mathbf{S}(\boldsymbol{\Omega})\mathbf{J}\boldsymbol{\Omega} + \delta\mathbf{R}\mathbf{S}^2(\boldsymbol{\Omega})\mathbf{e}_3 \\ &+ k_\Lambda\mathbf{z}_v - k_\Lambda^2\mathbf{z}_p + g\mathbf{e}_3 - \ddot{\mathbf{p}}_d + \boldsymbol{\varepsilon}(\boldsymbol{\nu}). \end{aligned} \tag{8}$$

Let us now introduce an auxiliary system parameter matrix defined as $\mathbf{A} := -[\frac{1}{m}\mathbf{e}_3, \delta\mathbf{S}(\mathbf{e}_3)\mathbf{J}^{-1}] \in \mathbb{R}^{3\times4}$, and a control input vector set as

$$\mathbf{u} := \left[T, (\boldsymbol{\tau} + k_\gamma\dot{\gamma}\mathbf{J}\mathbf{e}_3)^\mathsf{T}\right]^\mathsf{T} \in \mathbb{R}^4, \tag{9}$$

where k_γ is a positive gain and $\dot{\gamma}$ is an arbitrary (bounded) scalar function set by the control designer to add an extra term of angular motion about the $\mathbf{e}_3$ axis. It is thus convenient to write the following compact expression:

$$(1/m)\mathbf{f} - \delta\mathbf{R}\mathbf{S}(\mathbf{e}_3)\mathbf{J}^{-1}\boldsymbol{\tau} = \mathbf{R}\mathbf{A}\mathbf{u}. \tag{10}$$

We proceed by gathering the known terms in (8), in addition to new feedback terms $\mathbf{K}_p\mathbf{z}_p + \mathbf{K}_v\mathbf{z}_v$, into $\boldsymbol{\xi} \in \mathbb{R}^3$, given by

$$\begin{aligned} \boldsymbol{\xi} :=& \mathbf{K}_p\mathbf{z}_p + \mathbf{K}_v\mathbf{z}_v + g\mathbf{e}_3 + \delta\mathbf{R}\mathbf{S}^2(\boldsymbol{\Omega})\mathbf{e}_3 + \hat{\boldsymbol{\theta}}^\mathsf{T}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) \\ &+ \delta\mathbf{R}\mathbf{S}(\mathbf{e}_3)\mathbf{J}^{-1}\mathbf{S}(\boldsymbol{\Omega})\mathbf{J}\boldsymbol{\Omega} + k_\Lambda\mathbf{z}_v - k_\Lambda^2\mathbf{z}_p - \ddot{\mathbf{p}}_d \in \mathbb{R}^3, \end{aligned} \tag{11}$$

where $\mathbf{K}_p, \mathbf{K}_v \in \mathbb{R}^{3\times3}$ are positive definite gain matrices. At this stage, we are ready to set the control input as

$$\begin{aligned} \mathbf{u} :=& -\mathbf{A}^\mathsf{T}\left(\mathbf{A}\mathbf{A}^\mathsf{T}\right)^{-1}\mathbf{R}^\mathsf{T}\Big[\boldsymbol{\xi} + \mathbf{R}\mathbf{K}_d\mathbf{S}(\mathbf{e}_3)\,\mathrm{sat}(\boldsymbol{\Omega}) \\ &+ \mathbf{T}(\beta\mathbf{z}_p + \mathbf{z}_v)\,\hat{\psi}\Big] \in \mathbb{R}^4, \end{aligned} \tag{12}$$

where $\beta > 0$ is a control gain, $\mathbf{K}_d \in \mathbb{R}^{3\times3}$ is a positive definite gain matrix, and where the term $\mathbf{R}\mathbf{K}_d\mathbf{S}(\mathbf{e}_3)\,\mathrm{sat}(\boldsymbol{\Omega})$ is included to induce a damping effect on $\boldsymbol{\Omega}$, with $\mathrm{sat}(\bullet)$ denoting an operator that bounds its input to predefined upper and lower saturation values. Note that $(\mathbf{A}\mathbf{A}^\mathsf{T})^{-1} = \mathrm{diag}([j_2^2/\delta^2, j_1^2/\delta^2, m^2])$, which is always well-defined on account of $\delta \neq 0$.

[1]The choice of input $\boldsymbol{\nu}$ is detailed in Section V-C, after the specifics of $\mathbf{d}$ have been established.

From (9) and (12), the control laws are updated as

$$T := [1, \mathbf{0}]\mathbf{u}, \tag{13}$$

$$\boldsymbol{\tau} := [\mathbf{0}, \mathbf{I}]\mathbf{u} - k_\gamma \dot{\gamma} \mathbf{J}\mathbf{e}_3. \tag{14}$$

B. Neural Network Adaptive Laws

Let each column of $\boldsymbol{\theta}$ be denoted by $\boldsymbol{\theta}_i \in \mathbb{R}^n$, for $i = 1, 2, 3$, and let us do the same for $\widehat{\boldsymbol{\theta}}$ and $\widetilde{\boldsymbol{\theta}}$. Then, within the control scheme developed in Section IV-A, the adaptive laws governing $\widehat{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{\psi}}$ shall be set as smooth projections. We begin with $\dot{\widehat{\boldsymbol{\theta}}} := \mathbf{K}_\theta \left[\dot{\widehat{\boldsymbol{\theta}}}_1, \dot{\widehat{\boldsymbol{\theta}}}_2, \dot{\widehat{\boldsymbol{\theta}}}_3\right] \in \mathbb{R}^{n\times 3}$, where $\mathbf{K}_\theta \in \mathbb{R}^{n\times n}$ is a positive definite gain matrix, and where

$$\dot{\widehat{\boldsymbol{\theta}}}_i := \text{Proj}\left(\boldsymbol{\Lambda}_{\theta i}, \widehat{\boldsymbol{\theta}}_i\right) \in \mathbb{R}^n, \quad \text{for } i = 1, 2, 3, \tag{15}$$

with $\boldsymbol{\Lambda}_{\theta i} \in \mathbb{R}^n$ representing the i-th column of

$$\boldsymbol{\Lambda}_\theta := \boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) \left(\beta \mathbf{z}_p + \mathbf{z}_v\right)^\mathsf{T} - k_1 \widehat{\boldsymbol{\theta}} \in \mathbb{R}^{n\times 3}, \tag{16}$$

for some $k_1 > 0$. In turn, the adaptive law for $\boldsymbol{\psi}$ is defined as

$$\dot{\widehat{\boldsymbol{\psi}}} := \mathbf{K}_\psi \text{Proj}\left(\boldsymbol{\Lambda}_\psi, \widehat{\boldsymbol{\psi}}\right) \in \mathbb{R}^3, \tag{17}$$

where $\mathbf{K}_\psi \in \mathbb{R}^{3\times 3}$ is a positive definite gain matrix, and where

$$\boldsymbol{\Lambda}_\psi := \mathbf{T}(\beta \mathbf{z}_p + \mathbf{z}_v) \left(\beta \mathbf{z}_p + \mathbf{z}_v\right) - k_2 \widehat{\boldsymbol{\psi}} \in \mathbb{R}^3, \tag{18}$$

for some $k_2 > 0$. In (16) and (18), k_1 and k_2 are constant design parameters associated with leakage terms.

The smooth projector operator Proj$(\bullet, \bullet)$ employed in (15) and (17) aims to tackle the windup phenomenon. In this work, we adopt the celebrated projector developed in [38], which, for $\widehat{\mathbf{x}}, \mathbf{y} \in \mathbb{R}^m$, is defined as $\text{Proj}\left(\mathbf{y}, \widehat{\mathbf{x}}\right) := \mathbf{y} - \frac{\eta_1 \eta_2}{2(\varepsilon_1^2 + 2\varepsilon_1 B) 4 B^2} \widehat{\mathbf{x}} \in \mathbb{R}^m$, where $\eta_1 = (\widehat{\mathbf{x}}^\mathsf{T}\widehat{\mathbf{x}} - B^2)^4$ if $\widehat{\mathbf{x}}^\mathsf{T}\widehat{\mathbf{x}} > B^2$, otherwise $\eta_1 = 0$, and where $\eta_2 = \widehat{\mathbf{x}}^\mathsf{T}\mathbf{y} + \sqrt{(\widehat{\mathbf{x}}^\mathsf{T}\mathbf{y})^2 + \varepsilon_2^2}$, with arbitrary parameters $\varepsilon_1, \varepsilon_2 > 0$, and B a predefined bound. Given $\mathbf{x}$ to be the actual value associated with estimate $\widehat{\mathbf{x}}$, then (cf. [38])

$$\|\widehat{\mathbf{x}}\| \leq B + \varepsilon_1 \text{ and } (\mathbf{x} - \widehat{\mathbf{x}})^\mathsf{T} \text{Proj}(\mathbf{y}, \widehat{\mathbf{x}}) \geq (\mathbf{x} - \widehat{\mathbf{x}})^\mathsf{T}\mathbf{y}. \tag{19}$$

C. Stability Analysis

Theorem 1. *Consider the quadrotor system (4)-(5) in closed-loop with the control laws given in (13) and (14), as well as with the NN adaptive laws established in (15) and (17). The closed-loop error system ($\mathbf{z}_p$, $\mathbf{z}_v$, $\widetilde{\boldsymbol{\theta}}$ and $\widetilde{\boldsymbol{\psi}}$) is UUB.*

Proof. Recalling (6) and (7), a first Lyapunov function candidate is devised as $V_1 := 1/2 \mathbf{z}_p^\mathsf{T} \mathbf{K}_p \mathbf{z}_p + 1/2 \mathbf{z}_v^\mathsf{T}\mathbf{z}_v + \beta \mathbf{z}_p^\mathsf{T}\mathbf{z}_v$, which is positive for $\beta^2 \mathbf{I} < \mathbf{K}_p$[2]. Its derivative, using (7), gives $\dot{V}_1 = \mathbf{z}_p^\mathsf{T}\mathbf{K}_p \left(\mathbf{z}_v - k_\Lambda \mathbf{z}_p\right) + \beta(\mathbf{z}_v - k_\Lambda \mathbf{z}_p)^\mathsf{T}\mathbf{z}_v + (\beta \mathbf{z}_p + \mathbf{z}_v)^\mathsf{T}\dot{\mathbf{z}}_v$. Notice from (8), (10) and (11), that $\dot{\mathbf{z}}_v$ can be written as $\dot{\mathbf{z}}_v = \mathbf{R}\mathbf{A}\mathbf{u} + \boldsymbol{\xi} + \widetilde{\boldsymbol{\theta}}^\mathsf{T}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) + \boldsymbol{\varepsilon}(\boldsymbol{\nu}) - \mathbf{K}_p \mathbf{z}_p - \mathbf{K}_v \mathbf{z}_v$. Therefore, and according to (12), we may rewrite the derivative of V_1 as

$$\dot{V}_1 = -W_1 - \frac{\partial V_1}{\partial \mathbf{z}_v} \mathbf{R}\mathbf{K}_d \mathbf{S}(\mathbf{e}_3) \,\text{sat}(\boldsymbol{\Omega}) - \frac{\partial V_1}{\partial \mathbf{z}_v} \mathbf{T}\left(\frac{\partial V_1}{\partial \mathbf{z}_v}\right) \widehat{\boldsymbol{\psi}} + \frac{\partial V_1}{\partial \mathbf{z}_v} \left[\widetilde{\boldsymbol{\theta}}^\mathsf{T}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) + \boldsymbol{\varepsilon}(\boldsymbol{\nu})\right], \tag{20}$$

[2] $\beta < \sqrt{\lambda_{\min}(\mathbf{K}_p)}$ is a sufficient but not necessary condition.

where, for convenience, we have $\partial V_1 / \partial \mathbf{z}_v = (\beta \mathbf{z}_p + \mathbf{z}_v)^\mathsf{T} \in \mathbb{R}^{1\times 3}$, and where $W_1 := (k_\Lambda + \beta)\mathbf{z}_p^\mathsf{T}\mathbf{K}_p\mathbf{z}_p + \beta \mathbf{z}_p^\mathsf{T}(k_\Lambda \mathbf{I} + \mathbf{K}_v)\mathbf{z}_v + \mathbf{z}_v^\mathsf{T}(\mathbf{K}_v - \beta \mathbf{I})\mathbf{z}_v \in \mathbb{R}$ is positive granted that $(k_\Lambda + \beta)\mathbf{K}_p(\mathbf{K}_v - \beta \mathbf{I}) - (\beta/2)^2(k_\Lambda \mathbf{I} + \mathbf{K}_v)^2$ is a positive definite matrix. The final Lyapunov function, which incorporates the NN estimation errors, i.e., $\widetilde{\boldsymbol{\theta}}$ in (2) and $\widetilde{\boldsymbol{\psi}}$ in (3), is designed as $V := V_1 + 1/2 \,\text{tr}(\widetilde{\boldsymbol{\theta}}^\mathsf{T}\mathbf{K}_\theta^{-1}\widetilde{\boldsymbol{\theta}}) + 1/2 \widetilde{\boldsymbol{\psi}}^\mathsf{T}\mathbf{K}_\psi^{-1}\widetilde{\boldsymbol{\psi}}$. Its time derivative, using (2), (3) and (20), as well as employing the fact that $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ are constants, can be computed as

$$\dot{V} = -W_1 - \frac{\partial V_1}{\partial \mathbf{z}_v} \mathbf{R}\mathbf{K}_d \mathbf{S}(\mathbf{e}_3) \,\text{sat}(\boldsymbol{\Omega}) - \frac{\partial V_1}{\partial \mathbf{z}_v} \mathbf{T}\left(\frac{\partial V_1}{\partial \mathbf{z}_v}\right) \widehat{\boldsymbol{\psi}} + \frac{\partial V_1}{\partial \mathbf{z}_v} \left[\widetilde{\boldsymbol{\theta}}^\mathsf{T}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) + \boldsymbol{\varepsilon}(\boldsymbol{\nu})\right] - \text{tr}\left(\widetilde{\boldsymbol{\theta}}^\mathsf{T}\mathbf{K}_\theta^{-1}\dot{\widehat{\boldsymbol{\theta}}}\right) - \widetilde{\boldsymbol{\psi}}^\mathsf{T}\mathbf{K}_\psi^{-1}\dot{\widehat{\boldsymbol{\psi}}}. \tag{21}$$

Simplifying (21) sheds some light on the beneficial role played by the proposed adaptive laws. Recall that the trace of the outer product is equivalent to the inner product. Therefore, $\frac{\partial V_1}{\partial \mathbf{z}_v}\widetilde{\boldsymbol{\theta}}^\mathsf{T}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) = \text{tr}(\widetilde{\boldsymbol{\theta}}^\mathsf{T}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu})\frac{\partial V_1}{\partial \mathbf{z}_v})$, from where it follows that

$$\frac{\partial V_1}{\partial \mathbf{z}_v}\widetilde{\boldsymbol{\theta}}^\mathsf{T}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu}) - \text{tr}\left(\widetilde{\boldsymbol{\theta}}^\mathsf{T}\mathbf{K}_\theta^{-1}\dot{\widehat{\boldsymbol{\theta}}}\right) = \text{tr}\left(\widetilde{\boldsymbol{\theta}}^\mathsf{T}\left[\boldsymbol{\mathcal{W}}(\boldsymbol{\nu})\frac{\partial V_1}{\partial \mathbf{z}_v} - \mathbf{K}_\theta^{-1}\dot{\widehat{\boldsymbol{\theta}}}\right]\right). \tag{22}$$

Lemma 1 (Lemma 1 in [19])**.** *For any $\eta \in \mathbb{R}$, and for any $\epsilon > 0$, the following inequality holds (with $\kappa = 0.2785$): $0 \leq |\eta| - \eta \tanh(\eta/\epsilon) \leq \kappa\epsilon$.*

According to **Assumption** 1, and based on (3) and Lemma 1, it follows that

$$\frac{\partial V_1}{\partial \mathbf{z}_v}\boldsymbol{\varepsilon}(\boldsymbol{\nu}) - \frac{\partial V_1}{\partial \mathbf{z}_v}\mathbf{T}\left(\frac{\partial V_1}{\partial \mathbf{z}_v}\right)\widehat{\boldsymbol{\psi}} \leq \kappa\epsilon \mathbf{1}^\mathsf{T}\boldsymbol{\psi} + \frac{\partial V_1}{\partial \mathbf{z}_v}\mathbf{T}\left(\frac{\partial V_1}{\partial \mathbf{z}_v}\right)\widetilde{\boldsymbol{\psi}}. \tag{23}$$

Consequently, using (22) and (23), $\dot{V}$ in (21) becomes

$$\dot{V} = -W_1 + \kappa\epsilon \mathbf{1}^\mathsf{T}\boldsymbol{\psi} + \text{tr}\left(\widetilde{\boldsymbol{\theta}}^\mathsf{T}\left\{\boldsymbol{\mathcal{W}}(\boldsymbol{\nu})\frac{\partial V_1}{\partial \mathbf{z}_v} - \mathbf{K}_\theta^{-1}\dot{\widehat{\boldsymbol{\theta}}}\right\}\right) - \frac{\partial V_1}{\partial \mathbf{z}_v}\mathbf{R}\mathbf{K}_d\mathbf{S}(\mathbf{e}_3)\,\text{sat}(\boldsymbol{\Omega}) - \widetilde{\boldsymbol{\psi}}^\mathsf{T}\left[\mathbf{K}_\psi^{-1}\dot{\widehat{\boldsymbol{\psi}}} - \mathbf{T}\left(\frac{\partial V_1}{\partial \mathbf{z}_v}\right)\left(\frac{\partial V_1}{\partial \mathbf{z}_v}\right)^\mathsf{T}\right]. \tag{24}$$

Next, we use (15), (17) and (19), in addition to the linear mapping property of the trace operator, to simplify (24) as

$$\dot{V} \leq -W_1 - \frac{\partial V_1}{\partial \mathbf{z}_v}\mathbf{R}\mathbf{K}_d\mathbf{S}(\mathbf{e}_3)\boldsymbol{\Omega} + \kappa\epsilon\mathbf{1}^\mathsf{T}\boldsymbol{\psi} + k_1 \,\text{tr}\left(\widetilde{\boldsymbol{\theta}}^\mathsf{T}\widehat{\boldsymbol{\theta}}\right) + k_2\widetilde{\boldsymbol{\psi}}^\mathsf{T}\widehat{\boldsymbol{\psi}}. \tag{25}$$

A useful bound concerning (3) can be written as

$$\widetilde{\boldsymbol{\psi}}^\mathsf{T}\widehat{\boldsymbol{\psi}} = \frac{1}{2}\|\boldsymbol{\psi}\|^2 - \frac{1}{2}\|\widehat{\boldsymbol{\psi}}\|^2 - \frac{1}{2}\|\widetilde{\boldsymbol{\psi}}\|^2 \leq \frac{1}{2}\|\boldsymbol{\psi}\|^2 - \frac{1}{2}\|\widetilde{\boldsymbol{\psi}}\|^2. \tag{26}$$

Note that (26) establishes a well known result associated with the sum of vectors. Naturally, it also applies to (2). Another property of the trace operator states that

$$\text{tr}\left(\boldsymbol{\theta}^\mathsf{T}\boldsymbol{\theta}\right) = \text{tr}\left(\text{vec}(\boldsymbol{\theta})^\mathsf{T}\,\text{vec}(\boldsymbol{\theta})\right) = \sum_{i=1}^{3}\boldsymbol{\theta}_i^\mathsf{T}\boldsymbol{\theta}_i, \tag{27}$$

where $\text{vec}(\boldsymbol{\theta}) := [\boldsymbol{\theta}_1^\mathsf{T}, \boldsymbol{\theta}_2^\mathsf{T}, \boldsymbol{\theta}_3^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{3n}$ denotes the (column) vectorization of $\boldsymbol{\theta}$. It follows then, using (26) and (27), that,

$$\text{tr}\left(\widetilde{\boldsymbol{\theta}}^\mathsf{T}\widehat{\boldsymbol{\theta}}\right) = \sum_{i=1}^{3}\widetilde{\boldsymbol{\theta}}_i^\mathsf{T}\widehat{\boldsymbol{\theta}}_i \leq \sum_{i=1}^{3}\left(\frac{1}{2}\|\boldsymbol{\theta}_i\|^2 - \frac{1}{2}\|\widetilde{\boldsymbol{\theta}}_i\|^2\right). \tag{28}$$

In addition, Youngs inequality leads to:

$$-\frac{\partial V_1}{\partial \mathbf{z}_v}\mathbf{R}\mathbf{K}_d\mathbf{S}(\mathbf{e}_3)\,\mathrm{sat}(\mathbf{\Omega}) \leq \frac{1}{2}\|\mathbf{z}_p\|^2 + \frac{1}{2}\|\mathbf{z}_v\|^2 + \frac{\beta^2+1}{2}\lambda^2_{\max}(\mathbf{K}_d)\Omega^2_{\max}, \tag{29}$$

where $\Omega_{\max} > 0$ denotes the maximum value of $|\mathrm{sat}(\mathbf{\Omega})|$. According to W_1 in (20), there exists a $w_1 := \lambda_{\min}([(k_\Lambda + \beta)\mathbf{K}_p, \beta/2(k_\Lambda\mathbf{I}+\mathbf{K}_v); \beta/2(k_\Lambda\mathbf{I}+\mathbf{K}_v), \mathbf{K}_v - \beta\mathbf{I}])$, with $w_1 > 1/2$, such that $W_1 \geq w_1(\|\mathbf{z}_p\|^2+\|\mathbf{z}_v\|^2)$. By using (26), (28) and (29), $\dot{V}$ in (25) satisfies the following bound: $\dot{V} \leq -W_2+\mu$, where $W_2 := (w_1-1/2)(\|\mathbf{z}_p\|^2+\|\mathbf{z}_v\|^2)+k_1/2\sum_{i=1}^3\|\widetilde{\boldsymbol{\theta}}_i\|^2+k_2/2\|\widetilde{\boldsymbol{\psi}}\|^2 \in \mathbb{R}$ is also a positive quantity, and where $\mu := k_1/2\sum_{i=1}^3\|\boldsymbol{\theta}_i\|^2 + k_2/2\|\boldsymbol{\psi}\|^2 + \kappa\epsilon\mathbf{1}^\mathsf{T}\boldsymbol{\psi} + (\beta^2+1)\lambda^2_{\max}(\mathbf{K}_d)\Omega^2_{\max}/2 \in \mathbb{R}$ is also a bounded positive constant. Furthermore, let us define $\mathbf{z} := [\mathbf{z}_p^\mathsf{T}, \mathbf{z}_v^\mathsf{T}, \widetilde{\boldsymbol{\theta}}_1^\mathsf{T}, \widetilde{\boldsymbol{\theta}}_2^\mathsf{T}, \widetilde{\boldsymbol{\theta}}_3^\mathsf{T}, \widetilde{\boldsymbol{\psi}}^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{9+3n}$. Hence, an updated bound for $\dot{V}$ satisfies

$$\dot{V} \leq -\alpha\|\mathbf{z}\|^2 + \bar{\mu}, \tag{30}$$

where $\alpha := \min(w_1 - 1/2, k_1/2, k_2/2) \in \mathbb{R}$ and where $\bar{\mu}$ is a positive bounded constant such that $\|\mu(t)\| \leq \bar{\mu}$ for all t. The result in (30) implies that the error vector $\mathbf{z}$ is UUB, more specifically, there exists a $t^* > 0$, such that $\|\mathbf{z}(t)\| \leq \sqrt{\bar{\mu}/\alpha}$, for $t \geq t^*$. Thus all errors contained in $\mathbf{z}$ are UUB as well. □

V. Simulation and Experimental Results

A. Blade flapping, induced drag, and ground effect

To model the blade flapping and induced drags, we start by considering that the linear velocity of the i-th rotor hub, expressed in coordinate frame $\{B\}$ is given by $\mathbf{v}_{r_i} = \mathbf{R}^\mathsf{T}(\mathbf{v}_Q - \mathbf{v}_w) + \mathbf{\Omega} \times \mathbf{d}_i$, where, for $i = \{1,2,3,4\}$, $\mathbf{v}_w \in \mathbb{R}^3$ denotes the wind velocity and $\mathbf{d}_i \in \mathbb{R}^3$ is the displacement, expressed in $\{B\}$, of the i-th rotor w.r.t. the vehicle's center of mass. Then, the drag forces caused by the blade flapping effect and the induced drag can be respectively modeled by (cf. [1], [39]): $\mathbf{f}_{bf} := -c_t\mathbf{R}\sum_{i=1}^4\varpi_i^2[c_a, -c_b, 0; c_b, c_a, 0; 0, 0, 0]\mathbf{v}_{r_i}$, and $\mathbf{f}_{id} := -c_t\mathbf{R}\sum_{i=1}^4\varpi_i^2\,\mathrm{diag}\left([c_{dx}, c_{dy}, 0]\right)\mathbf{v}_{r_i}$, where $c_t > 0$ is a constant shared by the four rotors; $c_a, c_b > 0$ are blade flapping coefficients, $c_{dx}, c_{dy} > 0$ are induced drag coefficients, and $\varpi_i \in \mathbb{R}$ is the angular speed of the i-th motor.

The ground effect, i.e., the force induced when the vehicle is below a threshold distance z_0 from the floor, is given by (cf. [40]): $\mathbf{f}_g := -[c_g/(4p_z^2) - c_g/(z_0 - p_z)^2]\mathbf{e}_3$, if $p_z < z_0$, otherwise $\mathbf{f}_g := \mathbf{0}$, where $c_g > 0$ is the ground effect coefficient, and p_z is the distance between the vehicle and the floor.

In summary, the disturbance forces $\mathbf{f}_{bf}$, $\mathbf{f}_{id}$ and $\mathbf{f}_g$ may be lumped together into a *secondary* aerodynamic force $\mathbf{f}_{\text{aero}}(p_z, \mathbf{v}_Q, \mathbf{R}, \mathbf{\Omega}) := \mathbf{f}_{bf} + \mathbf{f}_{id} + \mathbf{f}_g$. In similar fashion, the aerodynamic torque acting on the vehicle due to $\mathbf{f}_{bf}$ and $\mathbf{f}_{id}$ can be expressed as $\boldsymbol{\tau}_{\text{aero}}(\mathbf{v}_Q, \mathbf{R}, \mathbf{\Omega}) := \sum_{i=1}^4\mathbf{S}(\mathbf{d}_i)(\mathbf{f}_{bf_i} + \mathbf{f}_{id_i})$.

It is important to remark that the aerodynamic disturbances perturbing the quadrotor are function of the system states.

B. Wind Disturbance

Suppose there is a wind source characterized by a pseudo-random Brownian motion, whose velocity is governed by the following differential equation $\dot{\mathbf{v}}_w := [-\mathbf{v}_w + [80, 75, 24] \odot \mathbf{w_d}] \oslash [1.65, 1.5, 1.9]$ m/s, where $\mathbf{w_d} \in \mathbb{R}^3$ is a vector of zero-mean white Gaussian noise sequences generated from different seeds, with noise power set to 10^{-3}. This wind will generate a force $\mathbf{f}_w$ with each of its components being calculated according to the classic drag equation as (area of air impact m^2) $\times$ (air density $\mathrm{kg/m}^3$) $\times$ (wind speed $\mathrm{m/s}$)^2. We set $\mathbf{f}_w = [0.0035v_{w_x}^2, 0.0035v_{w_y}^2, 0.0246v_{w_z}^2]$, where v_{wi}, for $i = \{x, y, z\}$ are the components of $\mathbf{w_d}$.

C. Simulation Results

Table I
PARAMETERS AND GAINS USED IN THE SIMULATIONS AND EXPERIMENTS.

	Symbol	Value
Initial conditions	$\mathbf{p}_Q(0), \mathbf{v}_Q(0)$	$[1, 1, -0.5]^\mathsf{T}$ m, $\mathbf{0}$ m/s
	$\mathbf{R}(0), \mathbf{\Omega}(0)$	$\mathbf{I}, \mathbf{0}$ rad/s
Model parameters	m, g, δ	0.247 kg, 9.81 m/s^2, –0.005
	$\mathbf{J}$	diag ([7, 7, 13]) 10^{-4}
Control gains	$\mathbf{K}_p, k_\Lambda$	diag ([3.8, 3.8, 3.8]) , 0.11
	$\mathbf{K}_v, \beta$	diag ([2.5, 2.5, 2.5]) , 0.5
	$\mathbf{K}_d$	diag ([0.68, 0.7, 1.2])
	$\mathbf{K}_\theta, \mathbf{K}_\psi, k_1, k_2$	$0.15\mathbf{I}, 0.01\mathbf{I}, 0.01, 0.01$
NN parameters	n, σ	1024, 7.07

Consider a reference Clelia curve described by

$$\mathbf{p}_d(\gamma(t)) := \begin{bmatrix} 1.3\cos(\varrho(t))\cos(\varrho(t)/4) \\ 1.3\cos(\varrho(t))\sin(\varrho(t)/4) \\ 1.3\sin(\varrho(t)) - 2.2 \end{bmatrix} \text{ (m)}, \tag{31}$$

where, in order to ensure constant speed along the trajectory, i.e., $\|\dot{\mathbf{p}}_d(t)\| = 1$ m/s for all $t \geq 0$, $\varrho(t) \in \mathbb{R}$ is a parameter set as $\varrho(t) := \int_0^t \|\partial\mathbf{p}_d(\varrho(\tau))/\partial\varrho(\tau)\|^{-1}\,d\tau$. The results in this section are obtained under the effect of the aerodynamic disturbances and the wind force, i.e., according to (5), $\mathbf{f}_d := \mathbf{f}_{aero} + \mathbf{f}_w$ and $\boldsymbol{\tau}_d := \boldsymbol{\tau}_{aero}$. Initial conditions, physical parameters and control gains are listed in Table I.

Based on Section V-A, we choose the input to the NN as $\boldsymbol{\nu} := [\mathbf{R}^\mathsf{T}\mathbf{z}_p, \mathbf{R}^\mathsf{T}\mathbf{z}_v, p_z, \mathbf{v}_Q, \mathbf{\Omega}] \in \mathbb{R}^{13}$. The centers of the activation functions are laid out as $\boldsymbol{\mu} := [h_p\mathbf{1}, h_v\mathbf{1}, 2.5h_p, 2.5h_v\mathbf{1}, 10\mathbf{1}]^\mathsf{T} \otimes \mathbf{\Delta} \in \mathbb{R}^{13\times512}$, where $h_p, h_v > 0$ are tuning parameters, $\mathbf{\Delta} := [2/n(-n/2, \ldots, -1, 0, 1, \ldots, n/2-1) \in \mathbb{R}^{1\times512}$ is an auxiliary sequence, and where $\otimes$ denotes the Kronecker product. We resort to the Monte Carlo method to obtain suitable h_p and h_v. Each run of the algorithm consists in having the vehicle tracking the reference (31) while under the influence of randomly generated wind disturbance sequences. The resulting analysis is displayed in Fig. 3, where the steady-state mean position error, computed for $t \geq 30$ s, is color mapped for different combinations of h_p (horizontal axis) and h_v (vertical axis). Statistically, smaller gains lead to better performances, which corresponds to a less discriminatory NN.

Upon data inspection, we concluded that the best result was obtained for $h_p = 2.5$, and $h_v = 1.5$, with a steady-state mean position error of 4.3 cm. The plot in Fig. 4 corroborates this result by showing the NN output, given by $\widehat{\mathbf{d}} := \widehat{\boldsymbol{\theta}}^\mathsf{T}\boldsymbol{\mathcal{W}}$, compared to the actual values of the lumped disturbances.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

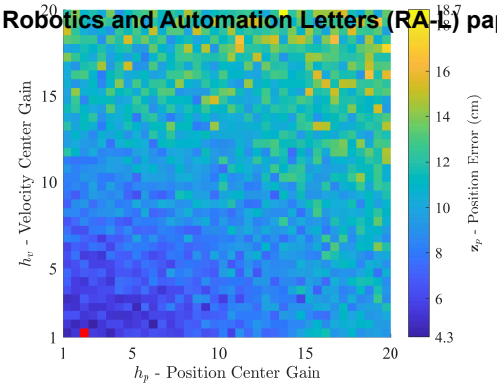

Figure 3. Steady-state mean position error, for combinations of h_p and h_v. Red square marks infeasible/unrealistic simulation.

Notice the good tracking performance, which attests to the strong estimation capabilities of the implemented NN.

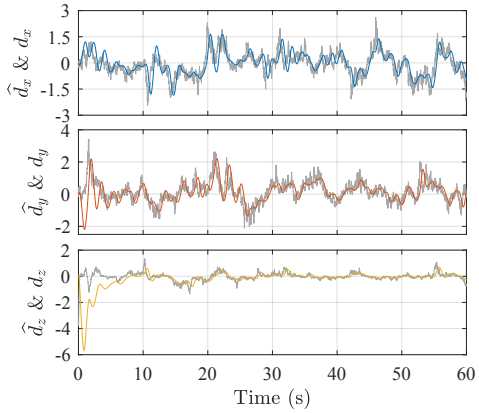

Figure 4. Component-wise comparison between the actual lumped disturbance $\mathbf{d}$ and the NN output $\widehat{\mathbf{d}}$. The observed adaptation delay is caused by the random/unpredictable nature of the time-varying disturbances, and naturally by the finite bandwidth of the NN.

D. Experimental Setup

A rapid prototyping and testing architecture by means of a MATLAB/Simulink environment was used to seamlessly integrate the sensors, the control algorithm, and the communication with a vehicle whose frame was 3D printed in-house using a tough polylactic acid material. The power distribution board, main flight control board, and motors are identical to the ones equipping the radio-controlled Blade 200 QX quadrotor.

Due to the lack of payload for onboard sensors, the states of the quadrotor were measured through external sensors. The indoors drone-testing arena in the SCORE Laboratory of the University of Macau is an $11 \times 5 \times 7$ m^3 volume featuring a VICON motion capture system comprising a total of 30 Vantage cameras. This high-performance system operates with sub-millimeter accuracy at 100 Hz, which allows for an accurate tracking of the reflective markers attached to the vehicle, resulting in position and attitude measurements, i.e., $\mathbf{p}_Q$ and $\mathbf{R}$. The linear and angular velocities, i.e. $\mathbf{v}_Q$ and $\boldsymbol{\Omega}$, are obtained from the first difference of position and attitude time series. Command signals are sent to the quadrotor through RF transmitter at 45 Hz. This rate is much faster than the system dynamics and does not alter significantly the control feedback.

The physical constants, control gains, and NN parameters employed in the experiments are the same as in Section V-C. Note that all values mentioned above remained unchanged across the different experiments reported in this section, further attesting to the robustness of the proposed strategy.

E. Trajectory Tracking

Our first batch of experiments focused on trajectory tracking. For comparison purposes, we performed, in identical conditions, three tests where: (i) the NN was kept off; (ii) an IA approach adapted from [15] was used; and (iii) the geometric multiple-layer NN controller [10] was used. The information fed to the controller about the UAV mass had a 20% discrepancy, (0.1976 kg instead of the actual 0.247 kg). This discrepancy, according to (5a), produces a time-varying model uncertainty with repercussions across the whole system.

The comparison data concerning the tracking of (31) are plotted in Fig. 5. From the leftmost plots, we see that our controller with the NN off performs worst: the noticeable vertical offset is attributed to the lack of compensation for disturbances, particularly unknown mass. This offset notwithstanding, even when accounting just for the motion on the $x-y$ plane, the trajectory tracking accuracy of the controller with NN off was calculated at 30.5 cm in terms of steady state mean magnitude of the position error, whereas the proposed controller with NN attained a 4.8 cm accuracy under the exact same conditions. Mass discrepancies aside, another reason why the controller with NN off performs so poorly is that our theoretical dynamic model deviates considerably from the actual model of a 3D printed vehicle. These pieces of evidence underline the importance of always including adaptation in controllers.

Starting with large initial position error (2.086 m) and velocity error (1 m/s), the maneuver with NN on enters steady-state roughly at the 10 s mark. The steady-state mean position error is 5 cm (better than the 7.3 cm attained by the IA controller [15], and the 5.3 cm attained by the geometric NN controller [10]. The corresponding standard deviations are 1.7 cm, 2.6 cm and 1.8 cm, respectively. Most importantly, both T and $\boldsymbol{\tau}$ control laws are well behaved for the entirety of the maneuver, and the vehicle's angular velocity $\boldsymbol{\Omega}$ remains low in steady-state. The robustness of the proposed controller, in particular its action against the mass discrepancy and unmodeled dynamics is highlighted in the evolution of the NN output $\widehat{\mathbf{d}}$ (see bottom right plot in Fig. 5).

F. Robustness to Wind Disturbances

A hovering maneuver in the presence of external wind disturbances was carried out to further attest to the robustness of the proposed NN controller. The controller in [10] was tested for comparison. Wind was generated by a mechanical office desk fan. Using an anemometer, the mean wind speed was measured at roughly 5.79 m/s within a 0.5 m radius from the source. The results are showcased in Fig. 6.

In this maneuver, a period of nominal hovering ($t = [0, 20]$ s) with the fan turned off was followed by a period of strong wind disturbances ($t = [20, 50]$ s). The vehicle was commanded to hover at a constant distance of 0.5 m from the fan. During the nominal hovering, the steady-state mean norm position errors are 5.3 cm (proposed) and 6.4 cm [10]. Even

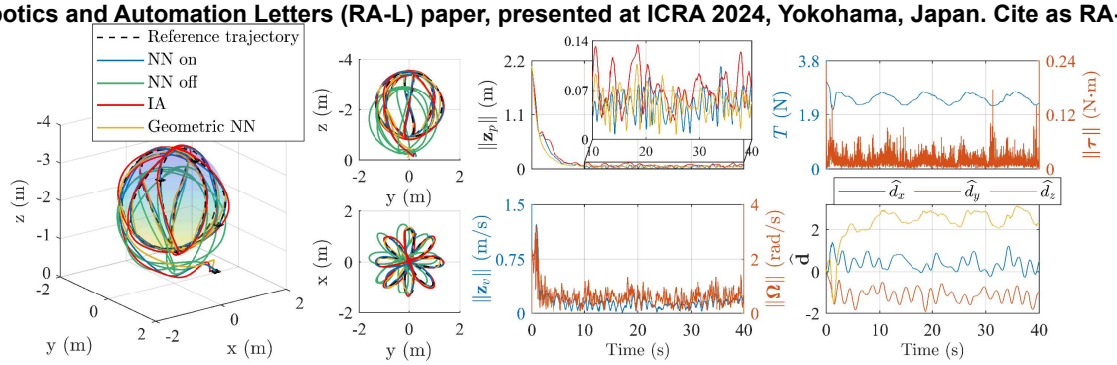

Figure 5. Left: superposition of reference trajectory (31) and actual 3D trajectories, along with two plane perspectives. Quadrotor is plotted every 10 s, with realistic depictions of position and attitude. On the center (from top to bottom), time evolution of: norm of the position error $\mathbf{z}_p$; norms of the velocity error $\mathbf{z}_v$ and angular velocity $\boldsymbol{\Omega}$. Right (from top to bottom): time evolution of input commands T and $\|\boldsymbol{\tau}\|$, and component-wise evolution of the NN output $\widehat{\mathbf{d}}$.

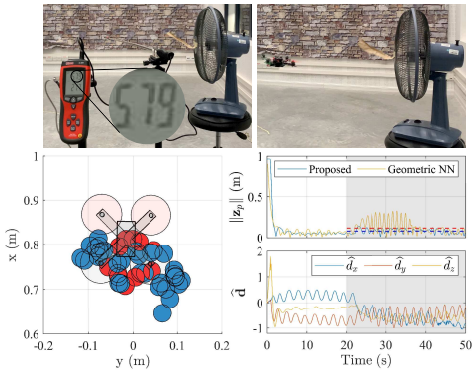

Figure 6. Top: anemometer reading and mechanical fan setup. Bottom left: top-view sketch of the quadrotor; red circles mark the positions in $t = [10, 19]$ s (steady-state maneuver with **fan off**), blue circles mark the positions in $t = [20, 50]$ s (steady-state with **fan on**). Bottom right: plots with the time evolution of the norms of the position errors, and the NN output $\widehat{\mathbf{d}}$; gray shaded areas highlight the period when the fan was blowing air.

in the presence of strong wind, the vehicle commanded by the proposed solution remained rather still; this robustness is substantiated by an 8.1 cm steady-state mean norm position error. The NN proposed in [10] displays a slower convergence, but ultimately achieves a similar performance, with a 11.7 cm steady-state mean norm position error. The strength of the wind disturbance can be inferred from the evolution of $\widehat{\mathbf{d}}$, which features an abrupt change along the x-axis at $t = 20$ s.

G. Robustness to Ground Effect

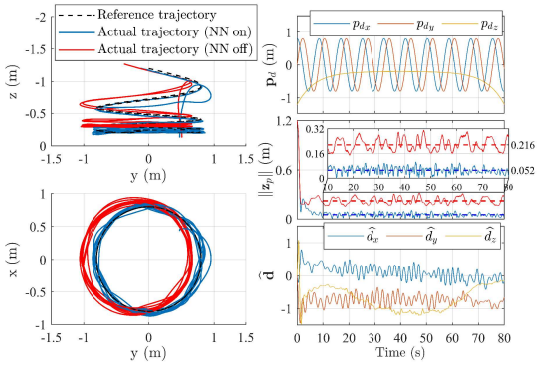

Figure 7. Left: plane perspectives of superposition of reference trajectory and actual trajectories. Right (top to bottom): evolution of reference trajectory ($\mathbf{p}_d$); norm of position error ($\|\mathbf{z}_p\|$); NN output ($\widehat{\mathbf{d}}$).

Our final batch of tests aimed at evaluating the performance of the proposed controller under the influence of ground effect. To this end, a trajectory with reference altitude approaching and moving away from the ground was implemented (see Fig. 7). To contrast performances, two runs were carried out: with NN on and with NN off. In the latter, the altitude of the actual trajectory is noticeably higher than the reference, which is caused by lift induced by ground effect. On the other hand, the maneuver with the NN on retained a high accuracy in tracking (5.3 cm mean steady-state position error). The compensation for ground effect is evident from the z component of $\widehat{\mathbf{d}}$ in Fig. 7 (bottom right subplot).

VI. Discussion

Despite many contributions on the topic of adaptive NN control, the frequent lack of experimental results, and the omission of details surrounding the NN implementations create a certain degree of opaqueness that hinders any chance of objective analyses. One key objective of this letter is to lift the veil on some of the idiosyncrasies of NNs in the context of adaptive control, where many times the role played by NNs is not discussed, or properly contrasted with, e.g., IA strategies.

In theory, based on the universal approximation theorem, the NN input should abide by the nature of the functions which the NN wants to approximate. This motivated our choice of input, which includes system states related to the aerodynamic disturbances, in addition to tracking errors, which can yield improved tracking precision under time-varying disturbances.

The fact that these adaptive NNs can approximate unknown time-varying disturbances is also worth discussing. Simulation and experimental analyses confirm this ability, but little has been discussed about the underlying mathematical agent responsible for it. Adaptive NNs trace back to the works of [19] and [20]. They are structurally akin to conventional backstepping IAs. Indeed, when the NN input $\boldsymbol{\nu}$ is set to be identical to $\boldsymbol{\mu}_i$, the term $\widehat{\boldsymbol{\theta}}^{\mathsf{T}}\boldsymbol{\mathcal{W}}(\boldsymbol{\nu})$ becomes $\widehat{\boldsymbol{\theta}}^{\mathsf{T}}\mathbf{1}$. In other words, the NN degenerates into an IA whose adaptive law is function of the tracking errors. This explains why adaptive NNs can approximate time-varying disturbances while offering UUB guarantees. Nevertheless, adaptive NNs are still an upgraded iteration of IAs. The centers and width of the activation functions act as extra tuning parameters that enable a finer adjustment of performance. Adjusting these gains manually is cumbersome, which justifies a Monte Carlo statistical analysis like the one reported in this letter.

VII. Conclusion

This study set out to solve the trajectory tracking problem of UAVs in the presence of unknown dynamics, including unknown time-varying disturbances, and model parametric uncertainty. A control framework was first designed that allows to compute all control inputs at once while lumping all unknown dynamics into a single term. A NN was then employed to deal with the lumped unknown system dynamics. The resulting errors stemming from the closed-loop system were shown to be UUB. Simulation and experimental results were presented to demonstrate the performance of our approach. Experimental comparisons with state-of-the-art controllers further confirmed the good performance attained by the proposed solution.

References

[1] R. Mahony, V. Kumar, and P. Corke, "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[2] J. Svacha, J. Paulos, G. Loianno, and V. Kumar, "Imu-based inertia estimation for a quadrotor using newton-euler dynamics," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3861–3867, 2020.

[3] E. Davis and P. E. I. Pounds, "Passive Position Control of a Quadrotor With Ground Effect Interaction," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 539–545, 2016.

[4] X. Kan, J. Thomas, H. Teng, H. G. Tanner, V. Kumar, and K. Karydis, "Analysis of ground effect for small-scale uavs in forward flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3860–3867, 2019.

[5] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, pp. 303–314, 1989.

[6] R. DeVore, B. Hanin, and G. Petrova, "Neural network approximation," *Acta Numerica*, vol. 30, pp. 327–444, 2021.

[7] F. L. Lewis, S. Jagannatha, and A. Yesildirak, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. CRC press, 1998.

[8] Y. Zou and Z. Zheng, "A Robust Adaptive RBFNN Augmenting Backstepping Control Approach for a Model-Scaled Helicopter," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2344–2352, 2015.

[9] J. Gao, A. A. Proctor, Y. Shi, and C. Bradley, "Hierarchical Model Predictive Image-Based Visual Servoing of Underwater Vehicles With Adaptive Neural Network Dynamic Control," *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2323–2334, 2016.

[10] M. Bisheban and T. Lee, "Geometric Adaptive Control With Neural Networks for a Quadrotor in Wind Fields," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1533–1548, 2021.

[11] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 153–158, 2007.

[12] W.-H. Chen, J. Yang, L. Guo, and S. Li, "Disturbance-Observer-Based Control and Related MethodsAn Overview," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1083–1095, 2016.

[13] J. Han, "From pid to active disturbance rejection control," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, pp. 900–906, 2009.

[14] P. Pereira, J. Cortés, and D. Dimarogonas, "Aerial Slung-Load Position Tracking Under Unknown Wind Forces," *IEEE Transactions on Automatic Control*, vol. 66, no. 9, pp. 3952–3968, 2021.

[15] J. Reis, G. Yu, D. Cabecinhas, and C. Silvestre, "High-performance quadrotor slung load transportation with damped oscillations," *International Journal of Robust and Nonlinear Control*, 2022.

[16] J. Lee, S. Ryu, and H. J. Kim, "Stable Flight of a Flapping-Wing Micro Air Vehicle Under Wind Disturbance," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5685–5692, 2020.

[17] H. Yang, L. Cheng, Y. Xia, and Y. Yuan, "Active Disturbance Rejection Attitude Control for a Dual Closed-Loop Quadrotor Under Gust Wind," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1400–1405, 2018.

[18] D. Hanover, P. Foehn, S. Sun, E. Kaufmann, and D. Scaramuzza, "Performance, Precision, and Payloads: Adaptive Nonlinear MPC for Quadrotors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 690–697, 2022.

[19] M. M. Polycarpou, "Stable adaptive neural control scheme for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 447–451, Mar. 1996.

[20] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 388–399, 1996.

[21] T. Zhang, S. Ge, and C. Hang, "Adaptive neural network control for strict-feedback nonlinear systems using backstepping design," *Automatica*, vol. 36, no. 12, pp. 1835–1846, 2000.

[22] T. Wang, H. Gao, and J. Qiu, "A Combined Adaptive Neural Network and Nonlinear Model Predictive Control for Multirate Networked Industrial Process Control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 416–425, 2016.

[23] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive neural network control of auvs with control input nonlinearities using reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 6, pp. 1019–1029, 2017.

[24] X. Yu, B. Li, W. He, Y. Feng, L. Cheng, and C. Silvestre, "Adaptive-Constrained Impedance Control for Human-Robot Co-Transportation," *IEEE Transactions on Cybernetics*, pp. 1–13, 2021.

[25] A. Melingui, O. Lakhal, B. Daachi, J. Mbede, and R. Merzouki, "Adaptive Neural Network Control of a Compact Bionic Handling Arm," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 2862–2875, 2015.

[26] C.-C. Tsai, H.-C. Huang, and S.-C. Lin, "Adaptive Neural Network Control of a Self-Balancing Two-Wheeled Scooter," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1420–1428, 2010.

[27] Y.-J. Liu, Q. Zeng, S. Tong, C. L. P. Chen, and L. Liu, "Adaptive Neural Network Control for Active Suspension Systems With Time-Varying Vertical Displacement and Speed Constraints," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9458–9466, 2019.

[28] T. Sun, H. Pei, Y. Pan, and C. Zhang, "Robust adaptive neural network control for environmental boundary tracking by mobile robots," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 2, pp. 123–136, 2013.

[29] W. Sun, N. Akashi, Y. Kuniyoshi, and K. Nakajima, "Physics-Informed Recurrent Neural Networks for Soft Pneumatic Actuators," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6862–6869, 2022.

[30] S.-W. Kim, B. Cho, S. Shin, J.-H. Oh, J. Hwangbo, and H.-W. Park, "Force Control of a Hydraulic Actuator With a Neural Network Inverse Model," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2814–2821, 2021.

[31] F. Cursi, W. Bai, W. Li, E. M. Yeatman, and P. Kormushev, "Augmented Neural Network for Full Robot Kinematic Modelling in SE(3)," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7140–7147, 2022.

[32] M. Vrba and M. Saska, "Marker-less micro aerial vehicle detection and localization using convolutional neural networks," *IEEE Robotics and Automation Letters*.

[33] T. Gao, Y.-J. Liu, L. Liu, and D. Li, "Adaptive neural network-based control for a class of nonlinear pure-feedback systems with time-varying full state constraints," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 5, pp. 923–933, 2018.

[34] H. E. Psillakis, "Further Results on the Use of Nussbaum Gains in Adaptive Neural Network Control," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2841–2846, 2010.

[35] T. Dierks and S. Jagannathan, "Output Feedback Control of a Quadrotor UAV Using Neural Networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 50–66, 2010.

[36] Q. Xu, Z. Wang, and Z. Zhen, "Adaptive neural network finite time control for quadrotor UAV with unknown input saturation," *Nonlinear Dynamics*, vol. 98, pp. 1973–1998, 2019.

[37] Y. Ouyang, L. Xue, L. Dong, and C. Sun, "Neural Network-Based Finite-Time Distributed Formation-Containment Control of Two-Layer Quadrotor UAVs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 8, pp. 4836–4848, 2022.

[38] Z. Cai, M. S. de Queiroz, and D. M. Dawson, "A sufficiently smooth projection operator," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 135–139, 2006.

[39] S. Omari, M.-D. Hua, G. Ducard, and T. Hamel, "Nonlinear control of VTOL UAVs incorporating flapping dynamics," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2419–2425, 2013.

[40] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of Control, Automation and Systems*, vol. 7, pp. 419–428, 2009.